IBM Developing Applications Using IBM Informix ESQL/C

Duration: 4 Days **Course Code: IX150G**

Overview:

This is the Classroom version of Instructor-Led Online course Developing Applications Using IBM Informix ESQL/C - Instructor Led Online (3X150) and Self-Paced Virtual course Developing Applications Using IBM Informix ESQL/C (SPVC) (2X150).
In this course, you will use IBM Informix ESQL/C tools to write applications that contain embedded SQL commands and queries. You will write applications to perform queries that return single and multiple rows; insert, update, and delete rows; create and use forms to display one or more rows of data; manage cursors, and handle various types of data, including large objects.
This course replaces US Course Developing Applications Using IBM Informix ESQL/C (L1112).

Target Audience:

This intermediate course is designed for application developers.

Objectives:

- Include SQL statements in a C program to add, retrieve, and alter data in an Informix database
- Set up a cursor structure for manipulating a set of rows
- Work with all SQL data types
- Set up and execute dynamically defined SQL statements
- Write effective, efficient SQL-based applications in C

Prerequisites:

You should have:

- IBM Informix Structured Query Language or equivalent knowledge
- ANSI C programming or C programming experience

Content:

Introduction to IBM Informix ESQL/C

- Identify the components of ESQL/C
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure

from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

An Overview of Cursors

- Name the three types of cursors

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility

different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups

appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3

- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7

- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

The Demonstration Database

- The tables in the stores_demo demonstration database

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the

- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Using Scroll Cursors

- Use SQL statements to set up a scrolling cursor

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data

- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were

selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing

fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18

ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored

- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Embedding SQL Statements

- Embed SQL statements in C

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements

BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor

INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database

- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types

- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a

- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate

- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement

database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Using CONNECT TO

- Use the CONNECT TO, DISCONNECT, and SET CONNECTION SQL statements

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16

program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Using Scroll Cursors Effectively

- Use a scrolling cursor to select a primary key

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into

- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information

- library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10

- a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate

- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server

- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic

- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program

- Exercise 19

Compiling an ESQL/C Program

- Incorporate other files into your source code

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Day 2

- Unit 9: Inserting Rows

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL

from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of

- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is

data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors

different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups

stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Using an Update Cursor

- Use an update cursor to lock rows that might be updated

Using a Insert Cursor

- Determine if an INSERT cursor is needed for adding rows to a database

- Install ESQL/C as part of Client SDK
- Set environment variables and execute

Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging

- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network

the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak

- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables

messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

ESQL/C Data Types

- Determine which C data type to use as a host variable

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the

- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Time Data Types

- Declare host variables for SQL DATE, DATETIME and INTERVAL data types

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix

- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1

ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3

ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions

- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data

- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in

- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups

- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16

- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor

- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers

- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Character and String Data Types

- Declare host variables for CHAR, CHAR *, VARCHAR, and LVARCHAR data types

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement

- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Simple Large Objects

- Declare host variables for BYTE and TEXT data types

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows

- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK

- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality

- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections

- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure

- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and

- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Numeric Data Types

- Recognize the storage needs for numeric data

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Dynamic SQL

- Use dynamic SQL and the associated data structures and commands

Dynamic SQL: Constructing INSERT Statements

- Use dynamic SQL to construct insert statements at runtime

Working with the Database Server

- Explain how to control the database server process with these functions:
- sqlexit

string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10

information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large

- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the

- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak

- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility

- selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data

- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory

- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Agenda

Day 1

- Welcome

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Day 3

- Unit 14: An Overview of Cursors

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Inserting Rows

- Code an appropriate embedded SQL statement that inserts a row into a database

The SQL Communications Area

- Explain the need for and the structure of the SQL Communication Area

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6

- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

SQL Exception Testing

- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows

- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error

- Describe the kinds of exceptions that might occur during execution of an SQL statement
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19
- information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17

- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Using GET DIAGNOSTICS

- Describe the preferred error detection methods

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure

- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database

data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C

- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with

- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Singleton Queries and Lookups

- Use the SELECT statement to retrieve a row from your database

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5

- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure

- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure

- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15

- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15

- Unit 18: Using an Insert Cursor

- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the

- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd

primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables

statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program

- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO

ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access

- data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively

- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row

- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Day 4

- Unit 20: Simple Large Objects

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the

- ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix

ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data

- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS

- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows

- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach

- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of

- different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9

- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic

- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server

Further Information:

For More information, or to book your course, please call us on 00 966 92000 9278

training@globalknowledge.com.sa

www.globalknowledge.com.sa

Global Knowledge - KSA, 393 Al-Uroubah Road, Al Worood, Riyadh 3140, Saudi Arabia