Chapter 5

Objectives: to learn about
- the extended entity relationship (E-ER) model
- How entity clusters are used to represent multiple entities and relationships
- The characteristics of good primary keys and how to select them
- using flexible solutions for special data modeling cases

The Extended Entity Relationship Model

- Result of adding more semantic constructs to original entity relationship (ER) model
- Diagram using this model is called an EER diagram (EERD)
- Combines some of the Object-oriented concepts with Entity Relationship concepts.

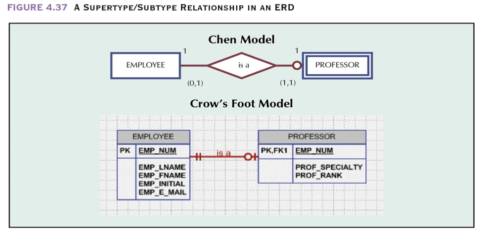
Entity Supertypes and Subtypes

- **Entity supertype**
  - Generic entity type related to one or more entity subtypes
  - Contains common characteristics
- **Entity subtype**
  - Contains unique characteristics of each entity subtype
  - Avoids unnecessary null attributes when not shared by all super entity types.

Supertype/Subtype Relationship in an ERD

- Example of employee (Super type), and Professor (Sub type)
- Note the cardinality for a 1:1 relationship

FIGURE 4.37 A SUPERTYPE/SUBTYPE RELATIONSHIP IN AN ERD



Entity Supertypes and Subtypes

- Example data set with and without sub-type entity, where certain employees contain additional data.

FIGURE 5.1 Nulls created by unique attributes

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_LICENSE	EMP_RATINGS	EMP_MED_TYPE	EMP_HIRE_DATE
100	Kolmycz	Xavier	T				15-Mar-88
101	Lewis	Marcos		ATP	SEL/MEL/Instr/CFII	1	25-Apr-89
102	Vandam	Jean					20-Dec-93
103	Jones	Victoria	R				28-Aug-03
104	Lange	Edith		ATP	SEL/MEL/Instr	1	20-Oct-97
105	Williams	Gabriel	U	COM	SEL/MEL/Instr/CFI	2	08-Nov-97
106	Duzak	Mario		COM	SEL/MEL/Instr	2	05-Jan-04
107	Diante	Venite	L				02-Jul-97
108	Wiesenbach	Joni					18-Nov-95
109	Travis	Brett	T	COM	SEL/MEL/SES/Instr/CFII	1	14-Apr-01
110	Genkazi	Stan					01-Dec-03

FIGURE 5.3 The EMPLOYEE-PILOT supertype-subtype relationship

Table Name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIRE_DATE	EMP_TYPE
100	Kolmycz	Xavier	T	15-Mar-88	
101	Lewis	Marcos		25-Apr-89	P
102	Vandam	Jean		20-Dec-93	A
103	Jones	Victoria	R	28-Aug-03	
104	Lange	Edith		20-Oct-97	P
105	Williams	Gabriel	U	08-Nov-97	P
106	Duzak	Mario		05-Jan-04	P
107	Diante	Venite	L	02-Jul-97	M
108	Wiesenbach	Joni		18-Nov-95	M
109	Travis	Brett	T	14-Apr-01	P
110	Genkazi	Stan		01-Dec-03	A

Table Name: PILOT

EMP_NUM	PIL_LICENSE	PIL_RATINGS	PIL_MED_TYPE
101	ATP	SEL/MEL/Instr/CFII	1
104	ATP	SEL/MEL/Instr	1
105	COM	SEL/MEL/Instr/CFI	2
106	COM	SEL/MEL/Instr	2
109	COM	SEL/MEL/SES/Instr/CFII	1

Specialization Hierarchy

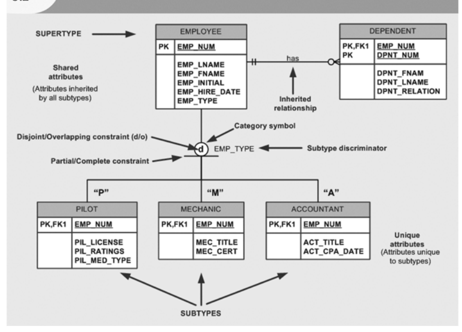
- Depicts arrangement of higher-level entity supertypes and lower-level entity subtypes
- Relationships described in terms of "IS-A" relationships
- Subtype exists only within context of supertype
- Every subtype has only one supertype to which it is directly related
- Can have many levels of supertype/subtype relationships

Inheritance

- Enables entity subtype to inherit attributes and relationships of supertype
- All entity subtypes inherit their primary key attribute from their supertype
- At implementation level, supertype and its subtype(s) maintain a 1:1 relationship
- Entity subtypes inherit all relationships in which supertype entity participates
- Lower-level subtypes inherit all attributes and relationships from all upper-level supertypes

Specialization Hierarchy

FIGURE 5.2 A specialization hierarchy



Subtype Discriminator

- The Subtype Discriminator is an attribute in supertype entity
  - Determines to which entity subtype each supertype occurrence is related
- Default comparison condition for subtype discriminator attribute is equality comparison
- Subtype discriminator may be based on other comparison condition

Disjoint and Overlapping Constraints

- **Disjoint subtypes**
  - Also called **nonoverlapping subtypes**
  - Subtypes that contain unique subset of supertype entity set
  - Single attribute is coded for the type
- **Overlapping subtypes**
  - Subtypes that contain non-unique subsets of supertype entity set
  - Multiple attributes are necessary, each representing a possible type.

Disjoint and Overlapping Constraints

- Implementing Overlapping subtypes
  - The supertype entity can hold multiple discriminators.

TABLE 5.1 Discriminator Attributes with Overlapping Subtypes

DISCRIMINATOR ATTRIBUTES		COMMENT
Professor	Administrator	
"Y"	"N"	The Employee is a member of the Professor subtype.
"N"	"Y"	The Employee is a member of the Administrator subtype.
"Y"	"Y"	The Employee is both a Professor and an Administrator.

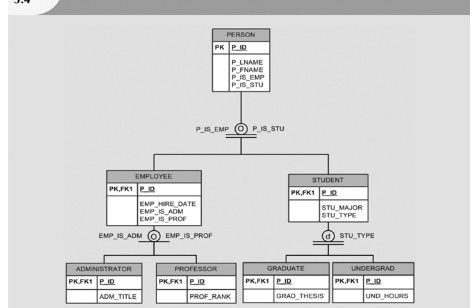
Diagramming Symbols for Disjoint & Overlapping Constraints

TABLE 5.2 Specialization Hierarchy Constraint Scenarios

TYPE	DISJOINT CONSTRAINT	OVERLAPPING CONSTRAINT
Partial	Supertype has optional subtypes. Subtype discriminator can be null. Subtype sets are unique.	Supertype has optional subtypes. Subtype discriminators can be null. Subtype sets are not unique.
Total	Every supertype occurrence is a member of a (at least one) subtype. Subtype discriminator cannot be null. Subtype sets are unique.	Every supertype occurrence is a member of a (at least one) subtype. Subtype discriminators cannot be null. Subtype sets are not unique.

Disjoint and Overlapping Constraints

FIGURE 5.4 Specialization hierarchy with overlapping subtypes



Completeness Constraint

- Specifies whether entity supertype occurrence must be a member of at least one subtype
- **Partial completeness**
  - Some supertype occurrences are not members of any subtype
  - symbolized by a circle over a single line
- **Total completeness**
  - Every supertype occurrence must be member of at least one subtype
  - symbolized by a circle over a double line

Completeness Constraint

- Partial or Total Completeness with Disjoint or Overlapping constraints.

TABLE 6.2 Specialization Hierarchy Constraint Scenarios

TYPE	DISJOINT CONSTRAINT	OVERLAPPING CONSTRAINT
Partial	Supertype has optional subtypes. Subtype discriminator can be null. Subtype sets are unique.	Supertype has optional subtypes. Subtype discriminators can be null. Subtype sets are not unique.
Total	Every supertype instance is a member of a (at least one) subtype. Subtype discriminator cannot be null. Subtype sets are unique.	Every supertype instance is a member of a (at least one) subtype. Subtype discriminators cannot be null. Subtype sets are not unique.

Specialization and Generalization

- **Specialization**
  - Identifies more specific entity subtypes from higher-level entity supertype
  - Top-down process
  - Based on grouping unique characteristics and relationships of the subtypes
- **Generalization**
  - Identifies more generic entity supertype from lower-level entity subtypes
  - Bottom-up process
  - Based on grouping common characteristics and relationships of the subtypes

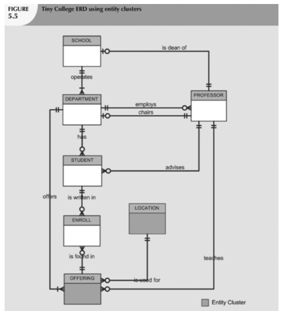
Entity Clustering

- “Virtual” entity type used to represent multiple entities and relationships in ERD
- Considered “virtual” or “abstract” because it is not actually an entity in final ERD
- Temporary entity used to represent multiple entities and relationships
- Eliminate undesirable consequences
  - Avoid display of attributes when entity clusters are used

Entity Clustering

- A technique used to simplify the ERD
  - Useful when the target audience of the ERD is not directly involved with the subsystem represented by the virtual entity.
  - Makes sure that the relationship to the cluster is not forgotten.

FIGURE 5.5 Tiny College ERD using entity clusters



Selecting Primary Keys

- Primary key is the most important characteristic of an entity
  - Single attribute or some combination of attributes
- Primary key's function is to guarantee entity integrity, i.e. the uniqueness of each entity row.
  - It's purpose is to guarantee uniqueness, not to "describe" the entity
- Primary keys and foreign keys work together to implement relationships
- Properly selecting primary key has direct bearing on efficiency and effectiveness

Primary Key Guidelines

TABLE 6.3 Desirable Primary Key Characteristics

PK CHARACTERISTIC	RATIONALE
Unique values	The PK must uniquely identify each entity instance. A primary key must be able to guarantee unique values. It cannot contain nulls.
Nonintelligent	The PK should not have embedded semantic meaning. An attribute with embedded semantic meaning is probably better used as a descriptive characteristic of the entity rather than as an identifier. In other words, a student ID of "650973" would be preferred over "Smith, Martha L." as a primary key identifier.
No change over time	If an attribute has semantic meaning, it may be subject to updates. This is why names do not make good primary keys. If you have "Vickie Smith" as the primary key, what happens when she gets married? If a primary key is subject to change, the foreign key values must be updated, thus adding to the database work load. Furthermore, changing a primary key value means that you are basically changing the identity of an entity.
Preferably single-attribute	A primary key should have the minimum number of attributes possible. Single-attribute primary keys are desirable but not required. Single-attribute primary keys simplify the implementation of foreign keys. Having multiple-attribute primary keys can cause primary keys of related entities to grow through the possible addition of many attributes, thus adding to the database work load and making (application) coding more cumbersome.
Preferably numeric	Unique values can be better managed when they are numeric because the database can use internal routines to implement a "counter-style" attribute that automatically increments values with the addition of each new row. In fact, most database systems include the ability to use special constructs, such as Autonumber in MS Access, to support self-incrementing primary key attributes.
Security complaint	The selected primary key must not be composed of any attribute(s) that might be considered a security risk or violation. For example, using a Social Security number as a PK in an EMPLOYEE table is not a good idea.

Selecting Primary Keys

- **A Natural key** is a real-world identifier used to uniquely identify real-world objects
  - Familiar to end users and forms part of their day-to-day business vocabulary
  - is generally used as the primary key of entity being modeled
- **Composite** keys are useful in two cases:
  - As identifiers of composite(bridge) entities
  - As identifiers of weak entities
- **Surrogate** keys may be generated when a Natural or Composite key is not available.

Use of Composite Primary Keys

- As identifier of Composite Entity Enroll

FIGURE 5.6 The M:N relationship between STUDENT and CLASS

Database name: Ch06_Tinycollege

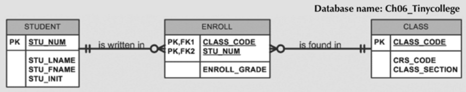


Table name: STUDENT (first four fields)

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT
321452	Bowser	William	C
324257	Smithson	Anne	K
324258	Brewer	Juliette	
324269	Oblonski	Walter	H
324273	Smith	John	D
324274	Katinga	Raphael	P
324291	Robertson	Gerald	T
324299	Smith	John	B

Table name: ENROLL

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT
321452	Bowser	William	C
324257	Smithson	Anne	K
324258	Brewer	Juliette	
324269	Oblonski	Walter	H
324273	Smith	John	D
324274	Katinga	Raphael	P
324291	Robertson	Gerald	T
324299	Smith	John	B

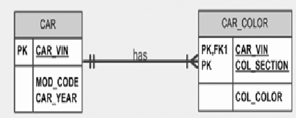
Table name: CLASS (first three fields)

CLASS_CODE	CRS_CODE	CLASS_SECTION
10012	ACCT-211	1
10013	ACCT-211	2
10014	ACCT-211	3
10015	ACCT-212	1
10016	ACCT-212	2
10017	CIS-220	1
10018	CIS-220	2
10019	CIS-220	3
10020	CIS-420	1
10021	QM-261	1
10022	QM-261	2
10023	QM-362	1
10024	QM-362	2
10025	MATH-243	1

Use of Composite Primary Keys

- As identifier of a Weak Entity
- Dependent entity exists only when it is related to parent entity

FIGURE 4.5 A new entity set composed of a multivalued attribute's components



Use of Surrogate Primary Keys

- Especially helpful when there is:
  1. No natural key
  2. Selected candidate key has embedded semantic contents
  3. Selected candidate key is too long or cumbersome
- If you use surrogate key,
  – ensure the candidate key of entity in question performs properly through use of "unique index" and "not null" constraints

Using Surrogate Primary Keys

- Example - Data set for Events
  – Uniqueness requires multiple attributes
  – Composite key is cumbersome and has embedded information.

TABLE 5.4 Data Used to Keep Track of Events

DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2010	11:00AM	2:00PM	Allure	Burton Wedding	60
6/17/2010	11:00AM	2:00PM	Bonanza	Adams Office	12
6/17/2010	3:00PM	5:30PM	Allure	Smith Family	15
6/17/2010	3:30PM	5:30PM	Bonanza	Adams Office	12
6/18/2010	1:00PM	3:00PM	Bonanza	Boy Scouts	33
6/18/2010	11:00AM	2:00PM	Allure	March of Dimes	25
6/18/2010	11:00AM	12:30PM	Bonanza	Smith Family	12

Design Cases: Learning Flexible Database Design

- Data modeling and design requires skills acquired through experience
- Experience acquired through practice
- Four special design cases that highlight:
  - Importance of flexible design
  - Proper identification of primary keys
  - Placement of foreign keys

Design Case #1: Implementing 1:1 Relationships

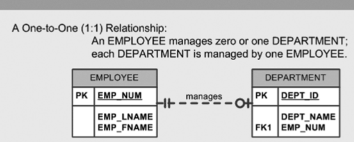
- Foreign keys work with primary keys to properly implement relationships in relational model
- Although conceivable to have put the primary key of each table into the other table as a foreign key, it is unnecessary.
- Put primary key of the "one" side (parent entity) on the "many" side (dependent entity) as foreign key

Design Case #1: Implementing 1:1 Relationships

TABLE 5.5 Selection of Foreign Key in a 1:1 Relationship

CASE	ER RELATIONSHIP CONSTRAINTS	ACTION
I	One side is mandatory and the other side is optional.	Place the PK of the entity on the mandatory side in the entity on the optional side as a FK, and make the FK mandatory.
II	Both sides are optional.	Select the FK that causes the fewest nulls, or place the FK in the entity in which the (relationship) role is played.
III	Both sides are mandatory.	See Case II, or consider revising your model to ensure that the two entities do not belong together in a single entity.

FIGURE 5.7 The 1:1 relationship between DEPARTMENT and EMPLOYEE

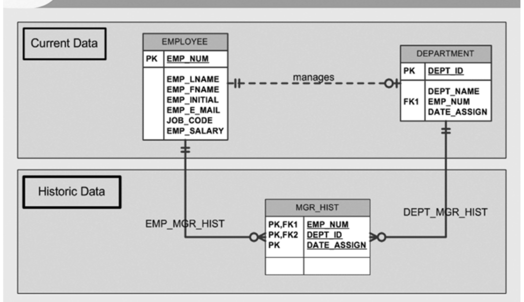


Design Case #2: Maintaining History of Time-Variant Data

- Time-variant data refers to data whose values change over time and for which you must keep a history of data changes
  - Values change over time
  - Must keep a history of data changes
- Keeping history of time-variant data equivalent to having a multivalued attribute in your entity
- Must create new entity in 1:M relationships with original entity
- New entity contains new value, date of change

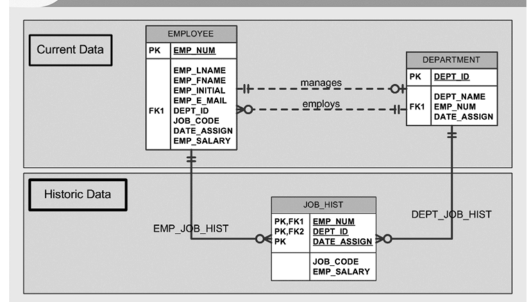
Design Case #2: Maintaining History of Time-Variant Data

FIGURE 5.8 Maintaining manager history



Design Case #2: Maintaining History of Time-Variant Data

FIGURE 5.9 Maintaining job history



Design Case #3: Fan Traps

- Design trap occurs when relationship is improperly or incompletely identified
  - Represented in a way not consistent with the real world
  - Most common design trap is known as fan trap
- Fan trap occurs when one entity is in two 1:M relationships to other entities
  - Produces an association among other entities not expressed in the model

Design Issue #3: Fan Traps

FIGURE 6.10 Incorrect ERD with Fan Trap problem

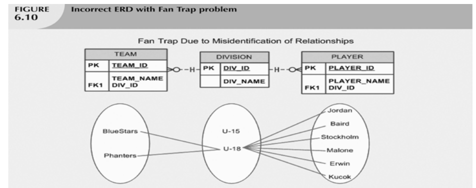
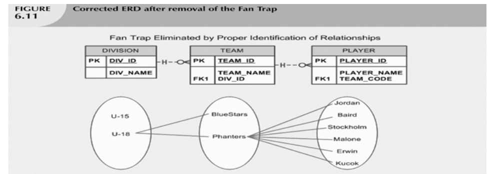


FIGURE 6.11 Corrected ERD after removal of the Fan Trap



Design Case #4: Redundant Relationships

- Redundancy is seldom a good thing in database environment
  - Occurs when there are multiple relationship paths between related entities
- Some designs use redundant relationships to simplify the design, or to account for time-variant data.
- The concern is that redundant relationships remain consistent across model

Design Case #4: Redundant Relationships

- Player relates to Division
- Player relates to Team which relates to Division

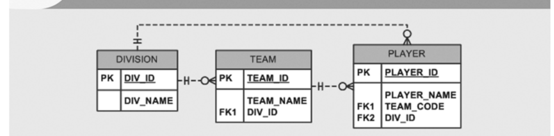


FIGURE 5.12 A redundant relationship

Summary

- Extended entity relationship (EER) model adds semantics to ER model
  - Adds semantics via entity supertypes, subtypes, and clusters
  - Entity supertype is a generic entity type related to one or more entity subtypes
- Specialization hierarchy
  - Depicts arrangement and relationships between entity supertypes and entity subtypes
- Inheritance means an entity subtype inherits attributes and relationships of supertype

Summary

- Subtype discriminator determines which entity subtype the supertype occurrence is related to:
  - Partial or total completeness
  - Specialization vs. generalization
- Entity cluster is "virtual" entity type
  - Represents multiple entities and relationships in ERD
  - Formed by combining multiple interrelated entities and relationships into a single object
- Natural keys are identifiers that exist in real world
  - Sometimes make good primary keys

Summary

- Characteristics of primary keys:
  - Must have unique values
  - Should be nonintelligent
  - Must not change over time
  - Preferably numeric or composed of single attribute
- Composite keys are useful to represent
  - M:N relationships
  - Weak (strong-identifying) entities
- Surrogate primary keys are useful when no suitable natural key makes primary key

Summary

- For 1:1 relationship, put the PK of mandatory entity
  - As FK in optional entity
  - As FK in entity that causes least number of nulls
  - As FK where the role is played
- Time-variant data
  - Data whose values change over time
  - Requires keeping a history of changes
- To maintain history of time-variant data:
  - Create entity containing the new value, date of change, other time-relevant data
  - Entity maintains 1:M relationship with entity for which history maintained

Summary

- Fan trap:
  - One entity in two 1:M relationships to other entities
  - Association among the other entities not expressed in model
- Redundant relationships occur when multiple relationship paths between related entities
  - Main concern is that they remain consistent across the model
- Data modeling checklist provides way to check that the ERD meets minimum requirements (see the front cover, inside page)