PASS SQLSATURDAY

SQL BAHIA

Databases on AWS:

The Right Tool for the Right Job

Bruno Lopes, Technical Trainer, AWS

PASS SQLSATURDAY

Who am I?

Bruno Lopes

Technical Trainer | Amazon Web Services

lopbruno@amazon.com

/in/**blopesinfo**

/brunokktro/**auladobruno**

Microsoft Most Valuable Professional

MVP Reconnect

PASS SQLSATURDAY

What is your database strategy?

Two fundamental areas of focus

“Lift and shift” existing apps to the cloud

Quickly build new apps in the cloud

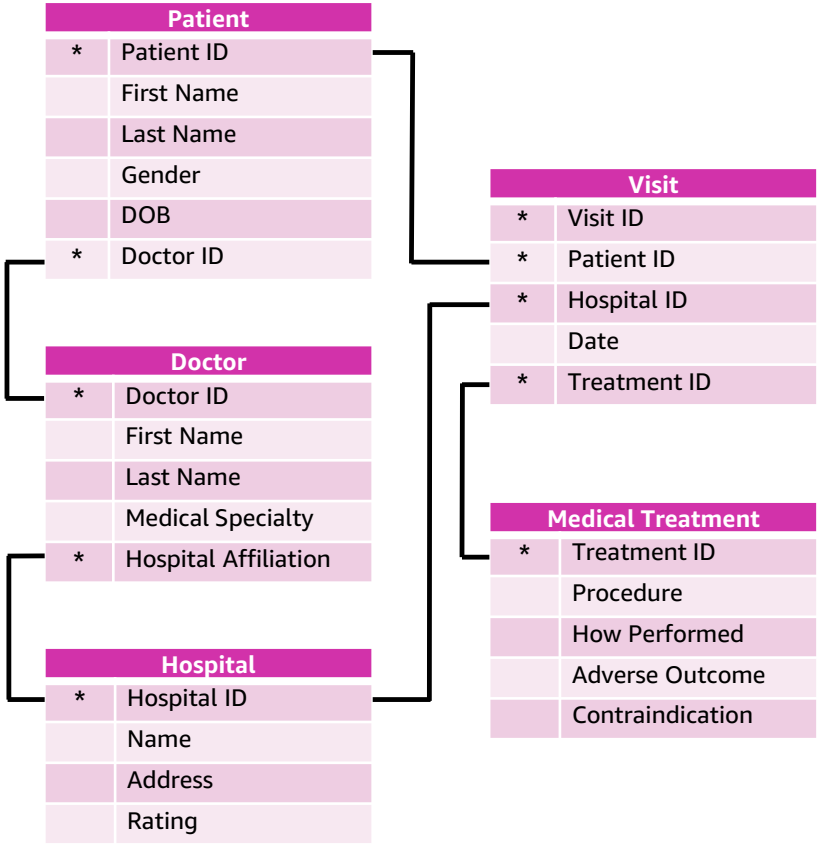
“Lift and shift” existing apps to the cloud

“Lift and shift” existing apps to the cloud

Quickly build new apps in the cloud

Relational data

- Divide data among tables
- Highly structured
- Relationships established via keys enforced by the system
- Data accuracy and consistency



Amazon Relational Database Service (RDS)

Managed relational database service with a choice of six popular database engines

Amazon Aurora | MySQL | PostgreSQL | MariaDB | Microsoft SQL Server | ORACLE

Easy to administer

No need for infrastructure provisioning, installing, and maintaining DB software

Available and durable

Automatic Multi-AZ data replication; automated backup, snapshots, failover

Highly scalable

Scale database compute and storage with a few clicks with no app downtime

Fast and secure

SSD storage and guaranteed provisioned I/O; data encryption at rest and in transit

aws

Amazon Aurora

MySQL and PostgreSQL-compatible relational database built for the cloud

Performance and availability of commercial-grade databases at 1/10th the cost

Performance and scalability

5x throughput of standard MySQL and 3x of standard PostgreSQL; scale-out up to 15 read replicas

Availability and durability

Fault-tolerant, self-healing storage; six copies of data across three Availability Zones; continuous backup to Amazon S3

Highly secure

Network isolation, encryption at rest/transit

Fully managed

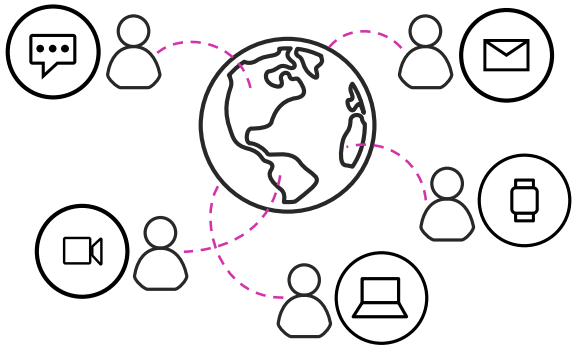
Managed by RDS:
No hardware provisioning, software patching, setup, configuration, or backups

Quickly build new apps in the cloud

"Lift and shift" existing apps to the cloud

Quickly build new apps in the cloud

Modern apps create new requirements



Ride hailing

Media streaming

Social media

Dating

Users:	1 million+
Data volume:	TB–PB–EB
Locality:	Global
Performance:	Milliseconds–microseconds
Request rate:	Millions
Access:	Web, mobile, IoT, devices
Scale:	Up-down, Out-in
Economics:	Pay for what you use
Developer access:	No assembly required

Common data categories and use cases

Relational	Key-value	Document	In-memory	Graph	Time-series	Ledger
Referential integrity, ACID transactions, schema-on-write	High throughput, low-latency reads and writes, endless scale	Store documents and quickly access querying on any attribute	Query by key with microsecond latency	Quickly and easily create and navigate relationships between data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data
Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial

Purpose-built Databases

Relational	Key-value	Wide Column	Document	In-memory	Graph	Time-series	Ledger
RDS	DynamoDB	Managed Cassandra Service with Cassandra Compatibility	DocumentDB with MongDB Compatibility	ElastiCache	Neptune	Timestream	QLDB

RDS:
- Aurora: MySQL, PostgreSQL
- Community: MySQL, PostgreSQL, MariaDB
- Commercial: ORACLE, Microsoft SQL Server

ElastiCache: redis, Memcached

Key-value use case

// Status of Hammer57

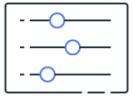
```
GET {
  TableName:"Gamers",
  Key: {
    "GamerTag":"Hammer57",
    "Type":"Status" } }
```

// Return all Hammer57

```
QUERY {
  TableName:"Gamers",
  KeyConditionExpression:"GamerTag = :a",
  ExpressionAttributeValues: {
      ":a":"Hammer57" } }
```

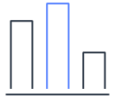
Gamers				
Primary Key		Attributes		
Gamer Tag	Type			
Hammer57	Rank	Level	Points	Tier
		87	4050	Elite
	Status	Health	Progress	
		90	30	
	Weapon	Class	Damage	Range
		Taser	87%	50
FluffyDuffy	Rank	Level	Points	Tier
		5	1072	Trainee
	Status	Health	Progress	
		37	8	

Amazon DynamoDB – Key concepts

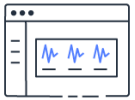


Fully managed NoSQL

Document or key-value



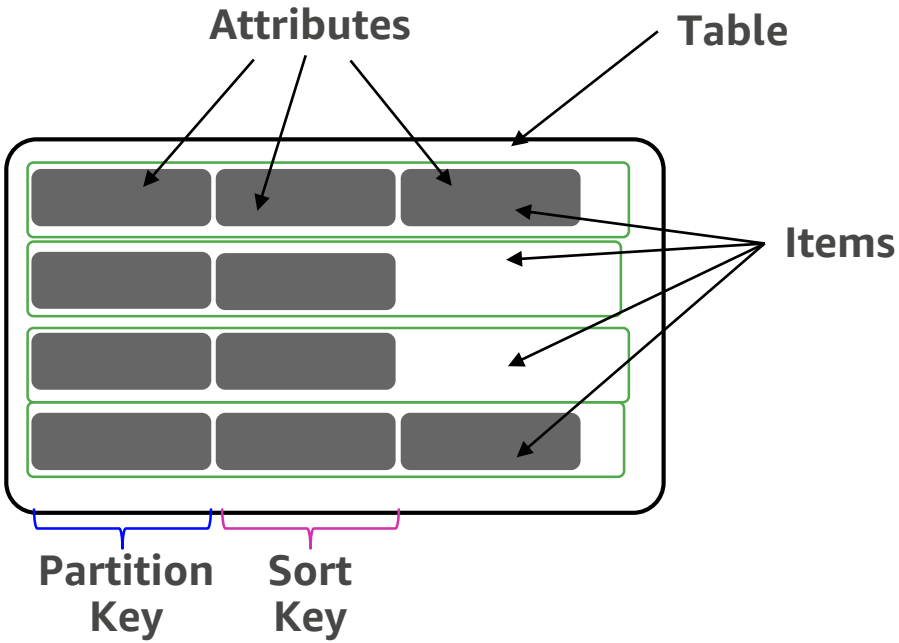
Scales to any workload



Fast and consistent

Access control

Event-driven programming



- Global secondary index
- Local secondary index

aws

Amazon Keyspaces (for Apache Cassandra)

Scalable, highly available, and managed Cassandra-compatible database service

Apache Cassandra-compatible

Use the same application code, licensed drivers, and tools built on Cassandra

No servers to manage

No need to provision, configure, and operate large Cassandra clusters or add and remove nodes manually

Single-digit millisecond performance at scale

Single-digit millisecond performance

Scale tables up and down automatically based on application traffic

Virtually unlimited throughput and storage

Simple migration

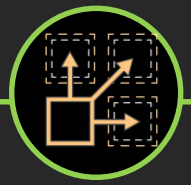
Simple migration to Managed Cassandra Service for Cassandra databases on premises or on EC2

Amazon DocumentDB (with MongoDB compatibility)

Fully managed

Managed by AWS: no hardware provisioning; auto patching, quick setup, secure, and automatic backups

Scalable



Separation of compute and storage enables both layers to scale independently; scale out to 15 read replicas in minutes

MongoDB compatible

Compatible with MongoDB 3.6; use the same SDKs, tools, and applications with Amazon DocumentDB

Cloud-native database architecture

Amazon DocumentDB (with MongoDB compatibility)

Scale out in minutes

Scale to 15 read replicas, millions of reads

Scale up in minutes

Scale from 16 to 768 GiB or RAM

Autoscaling storage

Storage automatically grow from 10 GB to 64 TB

Load balancing

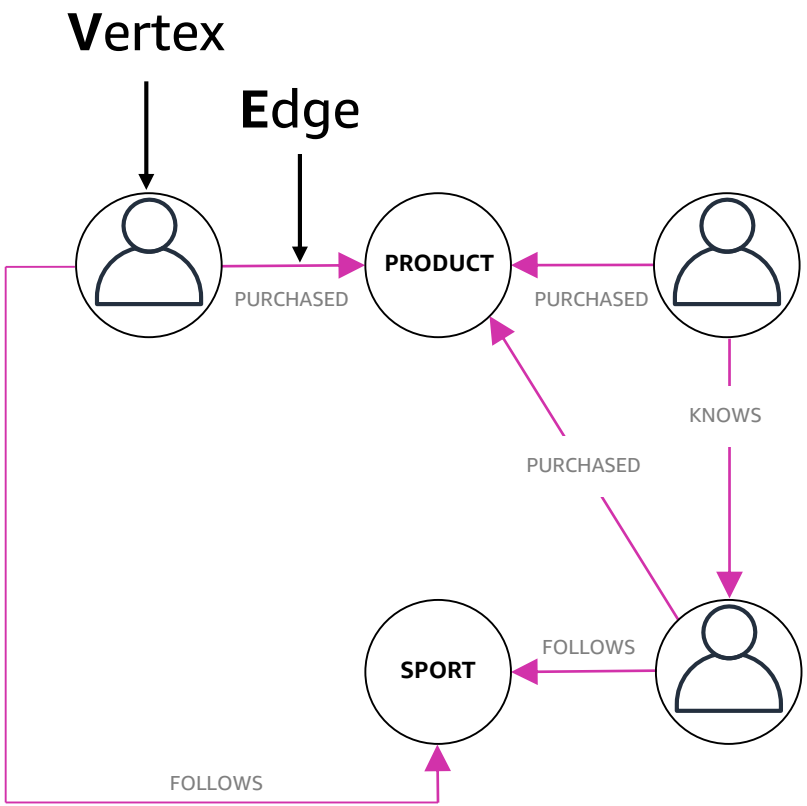
Scale reads across replicas

Iron

"Adopting Amazon DocumentDB is a game-changer . . . with Amazon DocumentDB, we can add or scale instances in minutes, regardless of data size."

Graph data

- Relationships are first-class objects
- Vertices connected by Edges



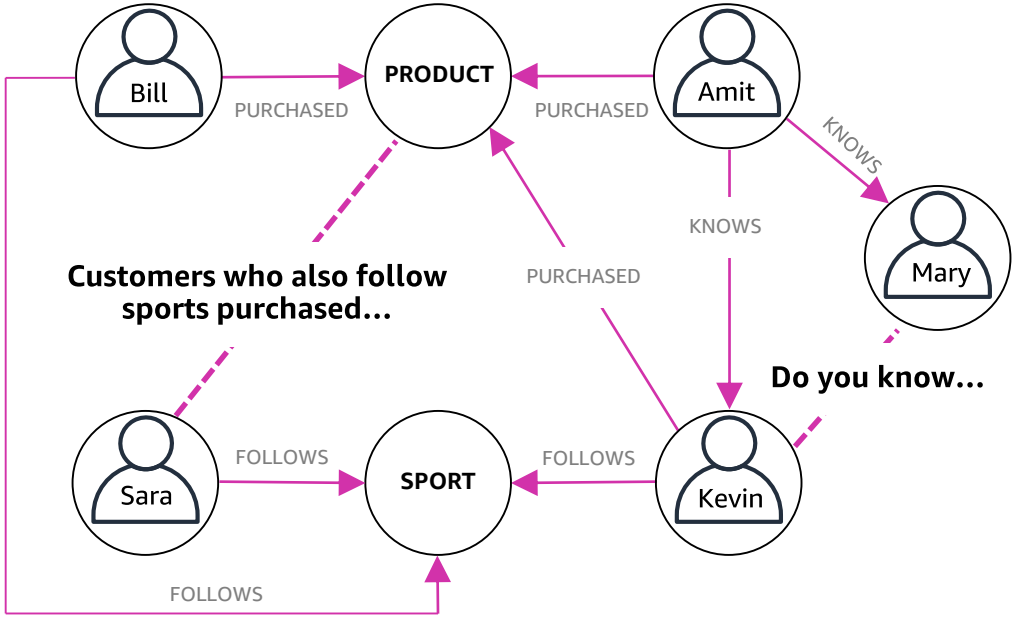
Graph use case

// Product recommendation to a user

```
gremlin> g.V().has(‘name’,’sara’).as(‘customer’).out(‘follows’).in(‘follows’).out(‘purchased’)
where(neq(‘customer’)).dedup().by(‘name’).properties('name')
```

// Identify a friend in common and make a recommendation

```
gremlin> g.V().has('name','mary').as(‘start’).
both('knows').both('knows').
where(neq(‘start’)).
dedup().by('name').properties('name')
```



AMAZON NEPTUNE

Fully managed graph database

FAST

Query billions of relationships with millisecond latency

RELIABLE

6 replicas of your data across 3 AZs with full backup and restore

EASY

Build powerful queries easily with Gremlin and SPARQL

OPEN

Supports Apache TinkerPop & W3C RDF graph models

Airbnb uses different databases based on the purpose

User search history: **Amazon DynamoDB**

- Massive data volume
- Need quick lookups for personalized search

Session state: **Amazon ElastiCache**

- In-memory store for submillisecond site rendering

Relational data: **Amazon RDS**

- Referential integrity
- Primary transactional database

duolingo

300M total users

7B exercises per month

CHALLENGE

Wanted to enable anyone to learn a language for free.

SOLUTION

Purpose-built databases from AWS:

- **DynamoDB**: 31B items tracking which language exercises completed
- **Aurora**: primary transactional database for user data
- **ElastiCache**: instant access to common words and phrases

Result:

More people learning a language on Duolingo than entire US school system

PASS
SQLSATURDAY

Retail demo application

Key-value	Graph	In-memory	Search
High throughput, Low-latency reads and writes, endless scale	Quickly and easily create and navigate relationships between data	Query by key with microsecond latency	Indexing and searching semistructured logs and data
Shopping cart, user profile	Product recommendation	Product leaderboard	Product search
DynamoDB	Amazon Neptune	ElastiCache	Amazon Elasticsearch Service

Demo application:

1. Available today

2. On GitHub: /aws-samples/aws-bookstore-demo-app

3. One click CloudFormation deployment

Amazon Quantum Ledger Database (QLDB)

Fully managed ledger database
Track and verify history of all changes made to your application's data

Immutable

Maintains a sequenced record of all changes to your data, which cannot be deleted or modified; you have the ability to query and analyze the full history

Cryptographically verifiable

Uses cryptography to generate a secure output file of your data's history

Highly scalable

Executes 2–3X as many transactions than ledgers in common blockchain frameworks

Easy to use

Easy to use, letting you use familiar database capabilities like SQL APIs for querying the data

Common customer use cases

Ledgers with centralized control

Healthcare
Verify and track hospital equipment inventory

Government
Track vehicle title history

Manufacturers
Track distribution of a recalled product

HR & Payroll
Track changes to an individual's profile

Amazon Timestream

Fast, scalable, fully managed time-series database

1,000x faster and 1/10th the cost of relational databases

Collect data at the rate of millions of inserts per second (10M/second)

Trillions of daily events

Adaptive query processing engine maintains steady, predictable performance

Time-series analytics

Built-in functions for interpolation, smoothing, and approximation

Serverless

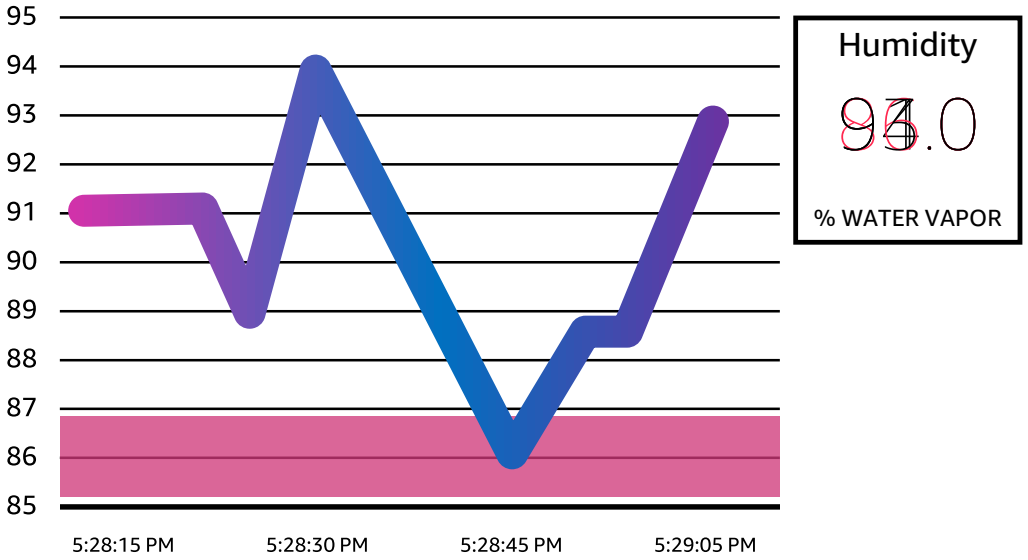
Automated setup, configuration, server provisioning, software patching

Time-series data

What is time-series data?

What is special about a time-series database?

Time-series use case

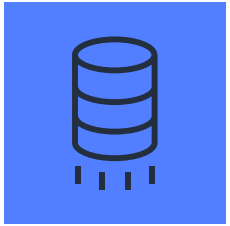


1. Application events
2. IoT Sensor Readings
3. DevOps data

PASS SQLSATURDAY

Database Migration

AWS migration tooling

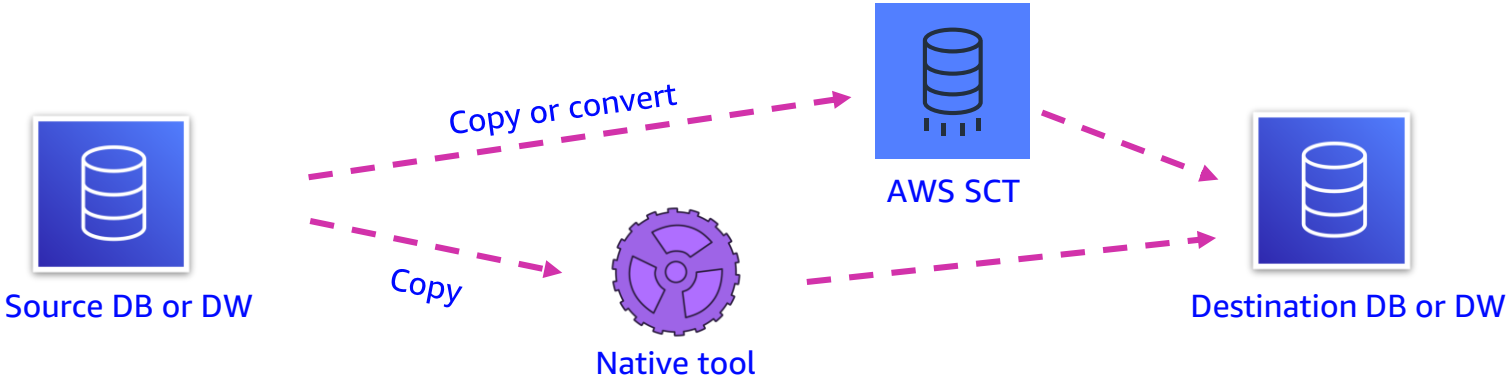


AWS Schema Conversion Tool converts your commercial database and data warehouse schemas to open-source engines or AWS-native services such as Amazon Aurora and Amazon Redshift

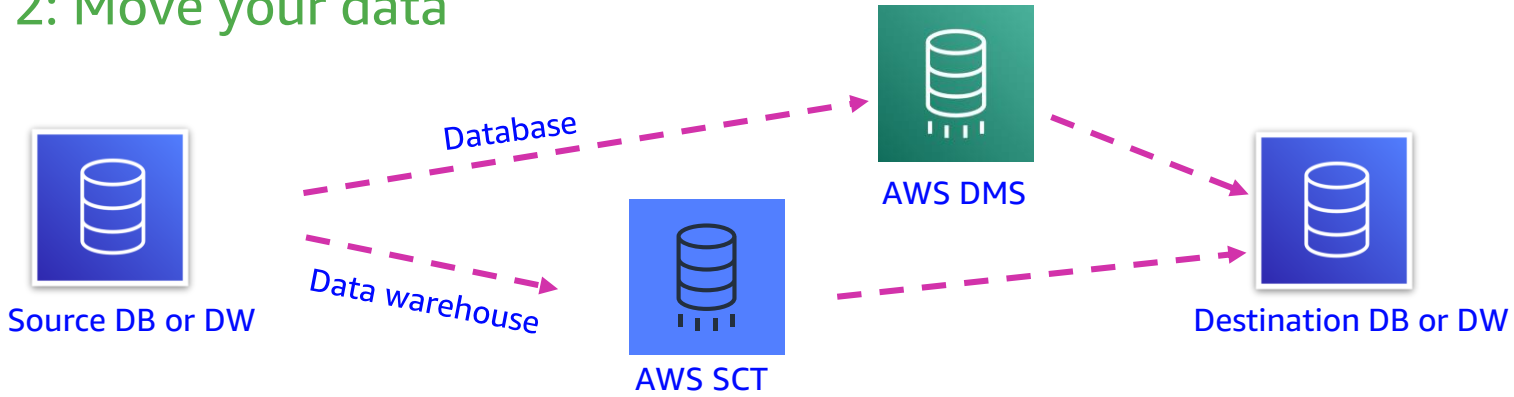
AWS Database Migration Service (AWS DMS) easily and securely migrates and/or replicates your databases and data warehouses to AWS

Database migration process

Step 1: Convert or copy your schema



Step 2: Move your data



AWS Schema Conversion Tool

The AWS Schema Conversion Tool helps automate database schema and code conversion tasks when migrating from source to target database engines

Features

Create assessment reports for homogeneous/heterogeneous migrations

Convert database schema

Convert data warehouse schema

Convert embedded application code

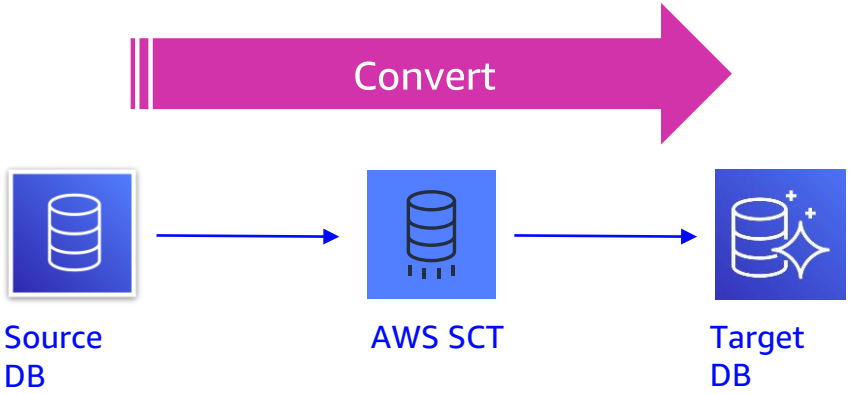
Code browser that highlights places where manual edits are required

Secure connections to your databases with SSL

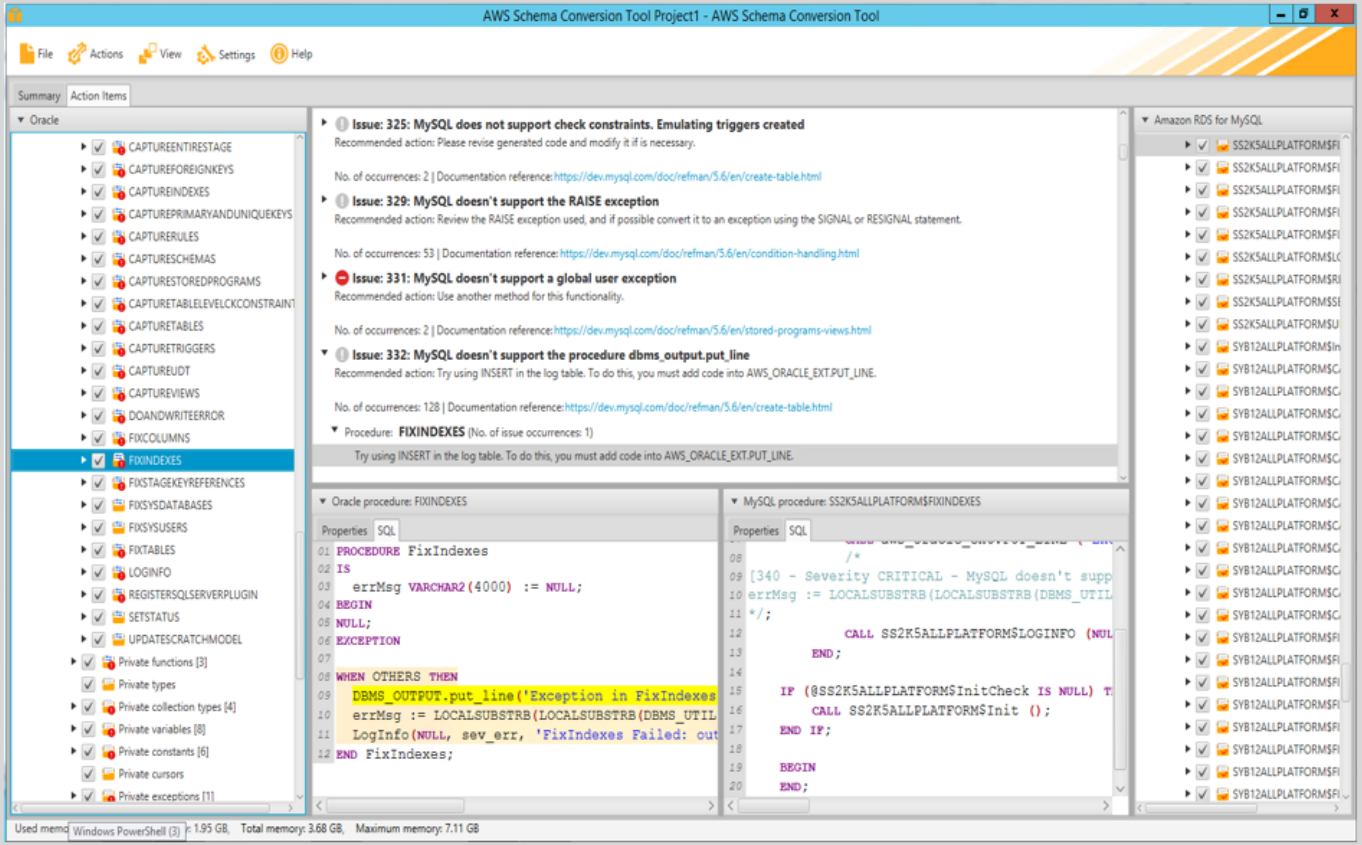
Service substitutions/ETL modernization to AWS Glue

Migrate data to data warehouses using SCT data extractors

Optimize schemas in Amazon Redshift

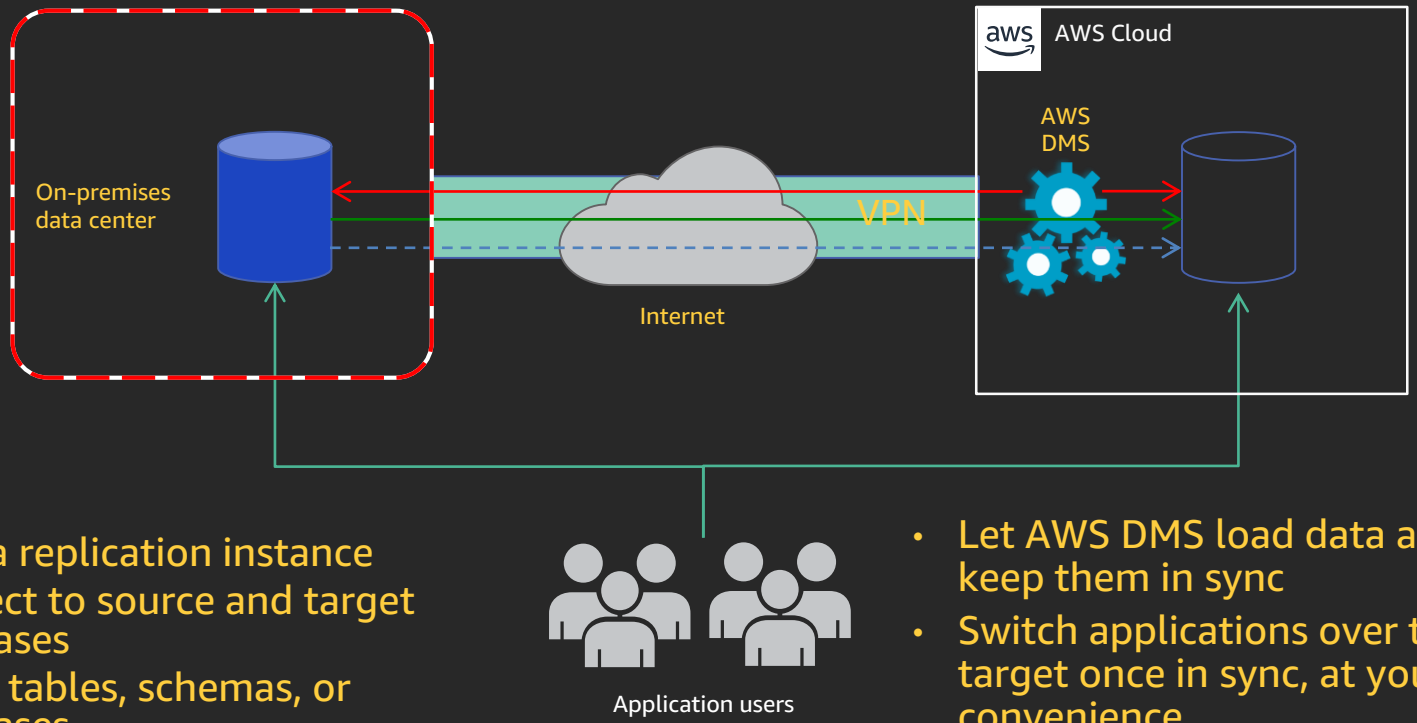


SCT helps with converting tables, views, and code



- Sequences
- User-defined types
- Synonyms
- Packages
- Stored procedures
- Functions
- Triggers
- Schemas
- Tables
- Indexes
- Views
- Sort and distribution keys

The data migration process



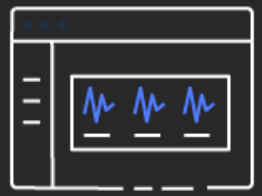
- Start a replication instance
- Connect to source and target databases
- Select tables, schemas, or databases

- Let AWS DMS load data and keep them in sync
- Switch applications over to the target once in sync, at your convenience

AWS DMS product highlights

Secure

Assess



Validate

Snowball integration

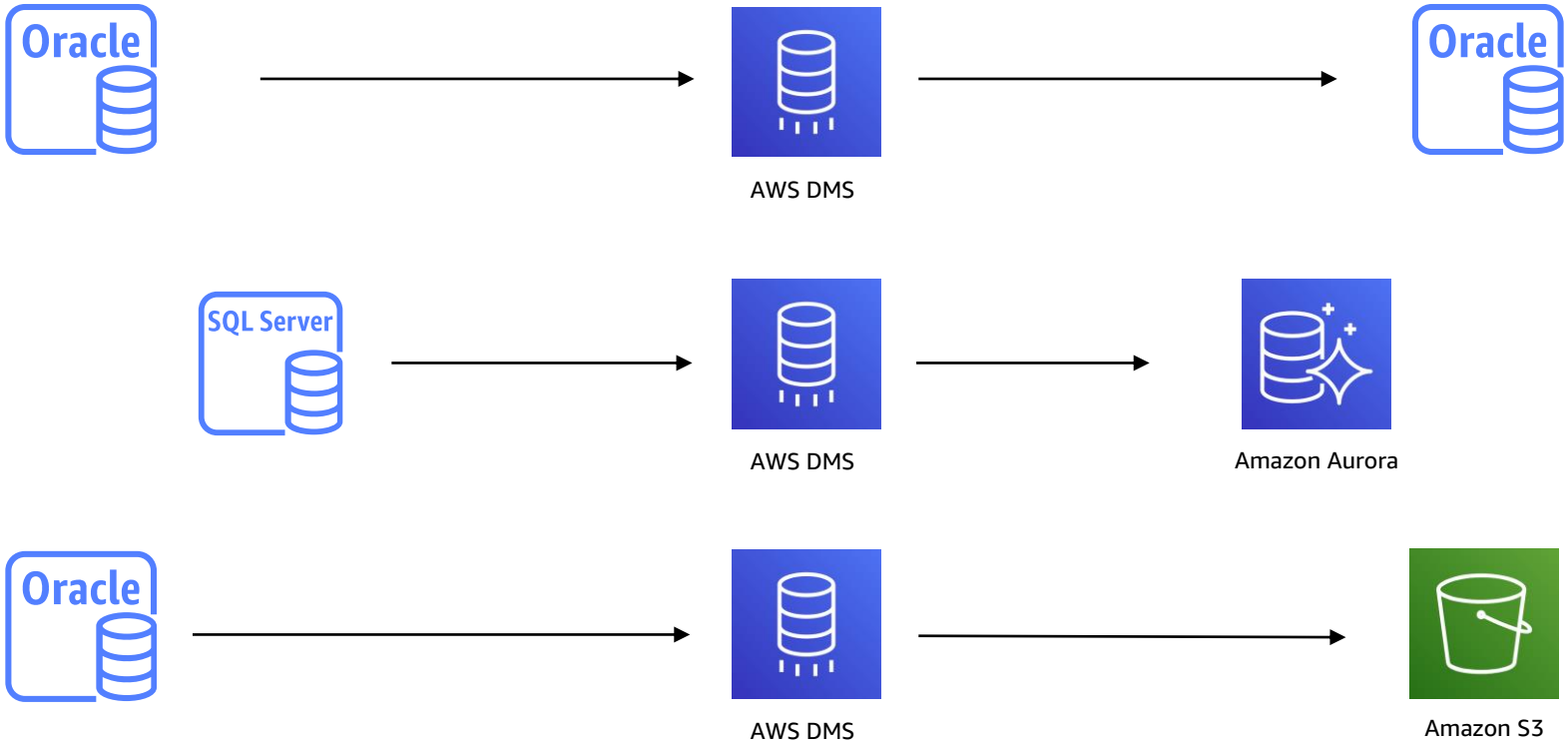
Monitor

Stream data

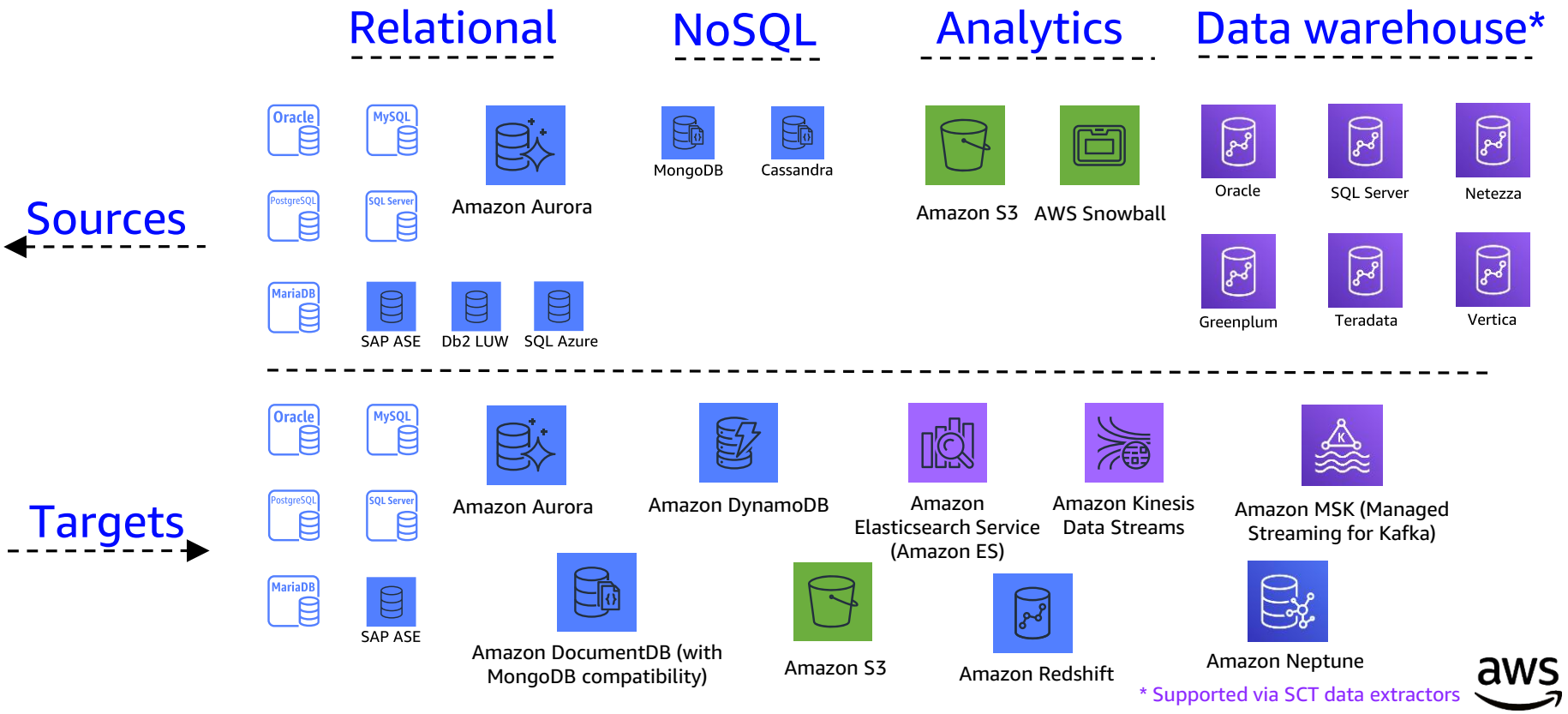
Low cost

Multiple options

Homogenous or heterogeneous



Supported source and targets



Old world to AWS migration playbooks

- **Topic-by-topic** overview of how to migrate databases and data warehouses to AWS services
- Covers all proprietary features and the different database objects
- Migration **best practices**
- Oracle to Aurora PostgreSQL (available)
- SQL Server to Aurora MySQL (available)
- SQL Server to Aurora PostgreSQL (available)

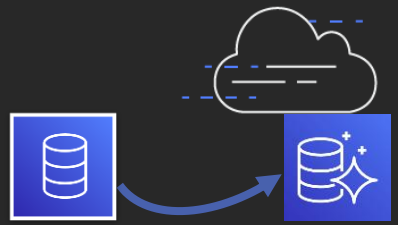
	Oracle Feature	PostgreSQL Feature	Compatibility
Link	Index Organized Tables (IOTs)	PostgreSQL "Cluster" Tables	Yes*
Link	Common Data Types	Common Data Types	Yes
Link	Table Constraints	Table Constraints	Yes
Link	Table Partitioning including: RANGE, LIST, HASH, COMPOSITE, Automatic LIST	Table Partitioning including: RANGE, LIST	Yes*
Link	Exchange & Split Partitions	N/A	None
Link	Temporary Tables	Temporary Tables	Yes*
Link	Unused Columns	ALTER TABLE DROP COLUMN	Yes
Link	Virtual Columns	Views and/or Function as a Column	Yes*
Link	User Defined Types (UDTs)	User Defined Types (UDTs)	Yes
Link	Read Only Tables & Table Partitions	Read Only Roles and/or Triggers	Yes*
Link	Index Types	Index Types	Yes*
Link	B-Tree Indexes	B-Tree Indexes	Yes
Link	Composite Indexes	Multi-Column Indexes	Yes
Link	BITMAP Indexes	BRIN Indexes	Minimal
Link	Function-Based Indexes	Expression Indexes	Yes
Link	Global and Local Partitioned Indexes	Partitioned Indexes	Yes*
Link	Identity Columns	Serial Data Type	Yes*
Link	MVCC (Table & Row Locks)	MVCC (Table & Row Locks)	Yes*
Link	Character Sets	Encoding	Yes*
Link	Transactional Model	Transactional Model	Yes*

Schema + Data + Best practices

AWS SCT + AWS DMS + Playbook

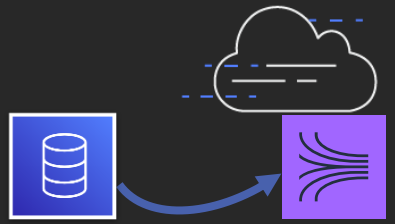
Use cases

Migrate



- Migrate business-critical applications
- Migrate data warehouses to Amazon Redshift
- Upgrade to a minor/major version
- Consolidate shards into Amazon Aurora
- Archive old data to Amazon S3
- Migrate from NoSQL to SQL, SQL to NoSQL, or NoSQL to NoSQL

Replicate



- Create cross-Region read replicas
- Run your analytics in the cloud
- Hydrate your data lakes
- Replicate to streaming platforms

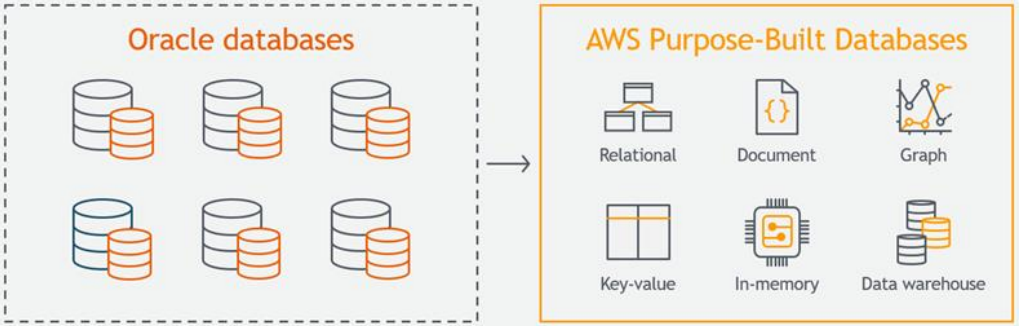
>200,000 databases migrated with DMS

More in 2019 than all of 2016-2018 combined

AUTODESK
SAMSUNG
finra
coursera
docomo
3M
U.S. Department of Veterans Affairs
zendesk
SK planet
DHI
nielsen
Johnson & Johnson
pitney bowes
Amway
ancestry.com
Arizona State University
asurion
dataxu
Cerner
COMCAST
ATLASSIAN
AMGEN
SIEMENS
gogo
Capital One
Expedia
Bristol-Myers Squibb
Liberty Mutual INSURANCE
Pearson
SCHOLASTIC
Agero
Adobe
ticketmaster
FT FINANCIAL TIMES
THOMSON REUTERS
Flipboard
Currencycloud
Globe
shaadi.com
ORION HEALTH
intuit
SendGrid
Meetup
fiverr
wave accounting
APPRISS
SysAid
gumgum
PEGA
halodoc
Smile Brands Inc.
bazaarvoice
hotelbeds
verizon
oddity
yapstone
CUPONATION
entradas.com
Sotheby's
Travelex
SnappCar
TOTANGO

Use Case – amazon.com migration

- We migrated **75 petabytes** of internal data stored
- Nearly **7,500 Oracle databases** to multiple AWS database services
- The migrations were accomplished with little or **no downtime**, and covered 100% of our proprietary systems.



Reduced our database costs by over **60%**, latency of our consumer-facing applications by **40%**, and database admin overhead by **70%**.

PASS
SQLSATURDAY

Thank you

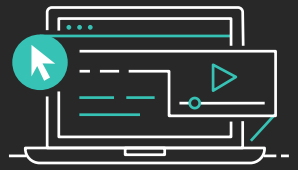
Reference

- Migration Complete – Amazon's Consumer Business Just Turned off its Final Oracle Database
- AWS re:Invent 2018: Databases on AWS: The Right Tool for the Right Job
- AWS re:Invent 2019: Dive deep into AWS SCT and AWS DMS
- Office Hours: Database Deep Dive | S1 E1 – How to Choose the Right Database for the Job
- Database Freedom
- Learning Paths - Databases

PASS

Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS

Validate expertise with the new **AWS Certified Database - Specialty** exam

Visit aws.training

aws training and certification


PASS