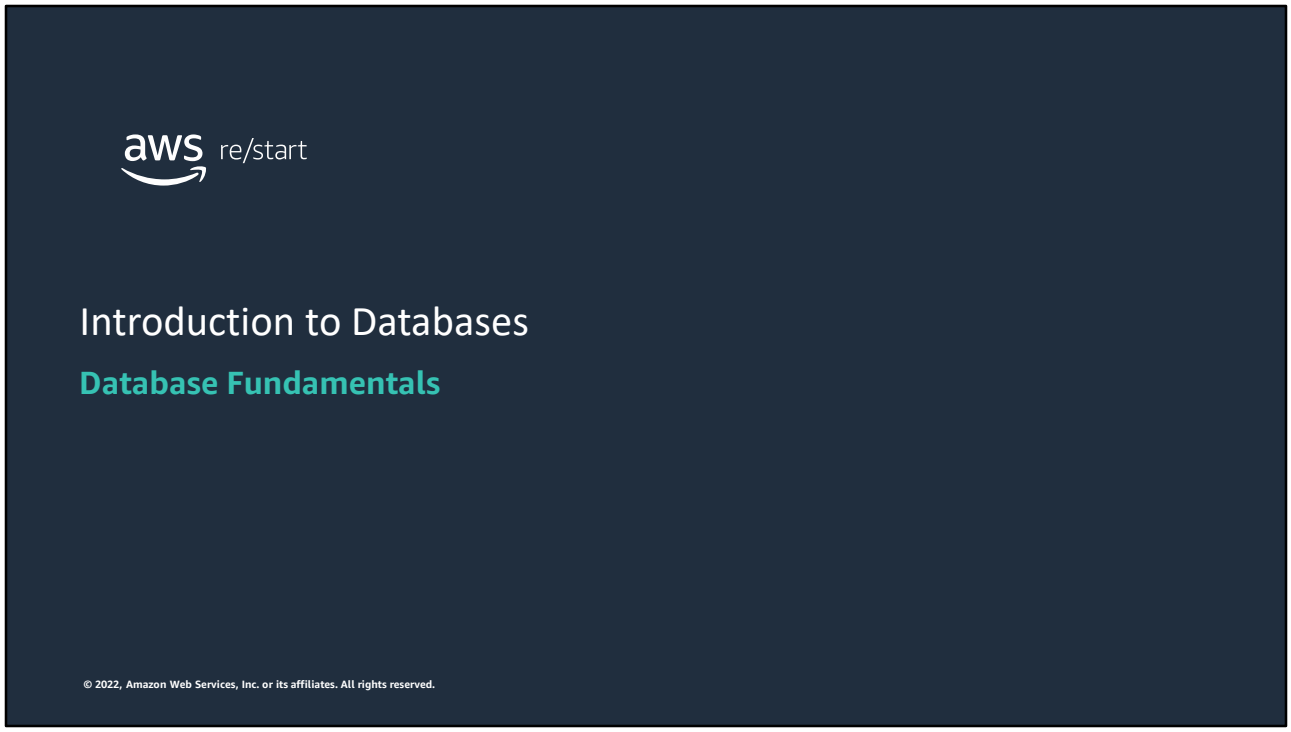


Welcome to Introduction to Databases.

What you will learn

At the core of the lesson

You will learn how to do the following:

- Identify different components of a database.
- Differentiate between relational databases and nonrelational databases.
- List the elements of a well-designed database.
- Describe the purpose and functions of a database management system (DBMS).

Key terms:

- Data
- Database
- Data model
- Relational data model
- Schema
- Transactions
- Relational databases
- Nonrelational databases
- Database management system (DBMS)

aws re/start

In this module, you will learn how to do the following:

- Identify different components of a database.
- Differentiate between relational databases and nonrelational databases.
- List the traits of a well-designed database.
- Describe the purpose and functions of a database management system (DBMS).

Data and databases

What is data?

- Data is raw bits and pieces of information.
  - Images, words, and phone numbers are examples of data.

What is a database?

- A database is a collection of data that is organized into files that are called tables.
  - Tables are a logical way of accessing, managing, and updating data.

Country table

Code	Name	Continent	Region
ARG	Argentina	South America	South America
AUS	Australia	Oceania	Australia and New Zealand

aws re/start

Data is raw bits and pieces of information. The bits of information that make up images, words, and phone numbers that you see on mobile phones or computer monitors are examples of data.

A database is a collection of bits of data that is organized into files, which are called tables. Tables are a logical way of accessing, managing, and updating data.

In a database, data can also appear in other formats, such as figures, graphics, images, and audio-video recordings.

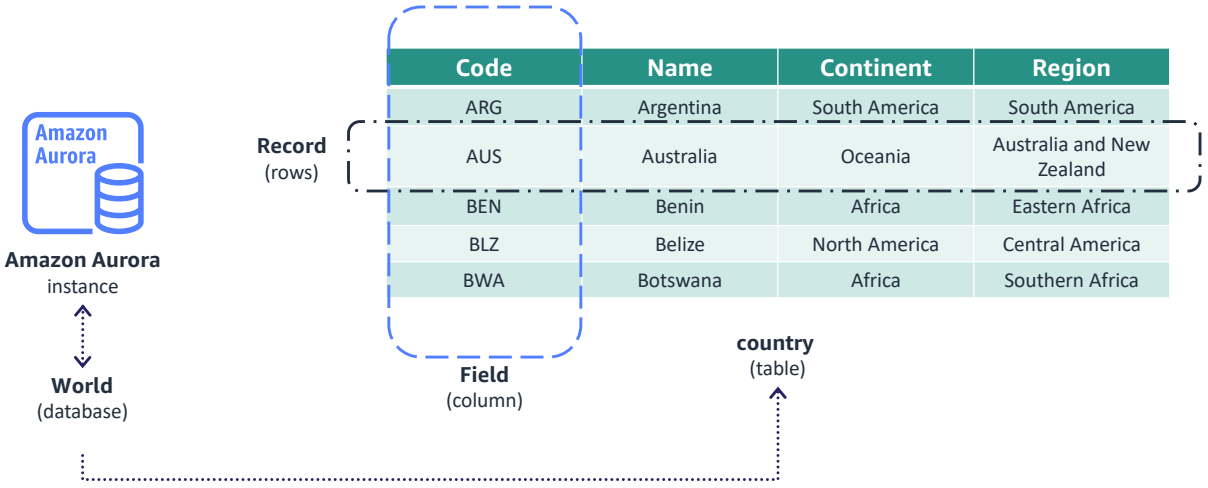
Data models

What is a data model?

- A data model represents the logical structure of the data that is stored in a database.
- Data models are used to determine how data can be stored and organized.
- The following items are examples of data models:
  - Relational
  - Semi-structured
  - Entity relationship
  - Object-based

A data model represents the logical design and structure of a database. Data models are used to determine how data can be stored and organized. Relational, semi-structured, entity relationship, and object-based are all data model types. This course focuses mainly on the relational data model.

Relational model



aws re/start

The relational model is a data model that Dr. Edgar F. Codd, a mathematician at IBM, developed in the late 1960s. Codd developed it to improve handling of large amounts of data. The relational model is based on a mathematical domain that is called relational algebra.

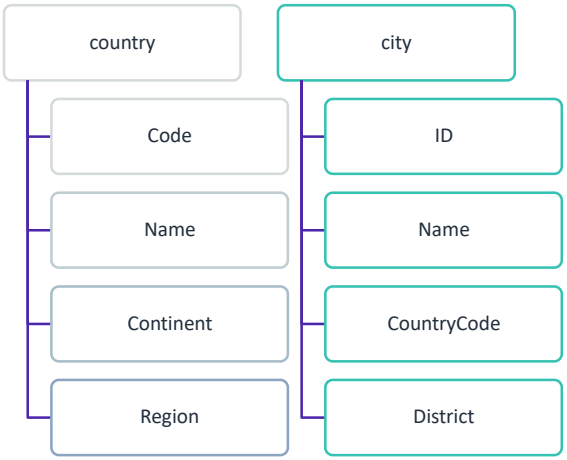
This slide provides an example of a relational model. Looking at the model, start with the database that is named World. Inside the World database is a table. Tables are used to hold information about the objects to be represented in the database. The country table can have multiple records (which are also known as rows) and fields (which are also known as columns). Each row is a set of values for the columns of the table. In this example, each record or row contains information about a country. This table contains the country code, the country name, the country's continent name, and the country's region name.

Databases exist on the backend, or behind most data-driven applications, data-driven websites, and data-driven mobile apps.

Schema

What is a schema?

- A database schema defines the organization of a database.
- It is based on the data model.
- It describes the elements of a database's design:
  - Tables
  - Columns
  - Relationships
  - Constraints



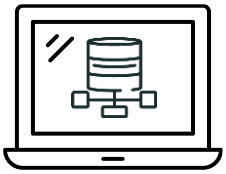
aws re/start

A database schema describes the organization of the data in a database. It contains the definition of tables, columns, constraints, and other elements identified in the database's data model.

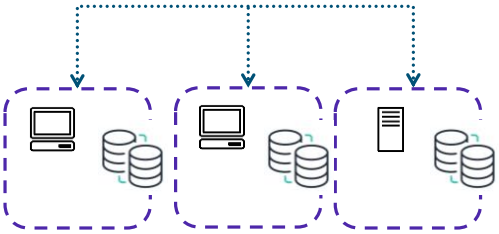
The following are examples of elements defined in a schema:

- Tables represent the objects in the data model. For example, in the World database, country and city are tables.
- Columns represent the attributes of the objects in a table. For, example, an object in the country table has a Code, Name, Continent, and Region attribute.
- Relationships represent the relations that exist between tables. For example, a city is located in a country. Therefore, in the city table, an attribute named CountryCode holds the code for the country where the city is located.
- Constraints represent the rules that affect the types of data or object relationships in the database. For example, a CountryCode value in the city table must exist as a Code value in the country table.

Small scale or distributed



Small scale or local



Distributed across a company's network

Aurora instance in the AWS Cloud

aws re/start

Databases can operate on a small scale. They might run on one computer and support a single user.

Other databases are distributed across multiple systems and locations across a company's network, or they can be cloud-based databases that support millions of users.

Relational databases

You will now learn more information about relational databases.

Relational database basics (1 of 2)

What is a relational database?

- A relational database is a collection of data items that have predefined relationships between them.
- A relational database is often referred to as a structured query language (SQL) database.
- The database requires a fixed definition of the structure of the data.
- The data is stored in tables with rows and columns.

The following information describes relational database design:

- Tables are used to hold information about the objects to be represented in the database.
- The data is spread across multiple tables and stored in rows and columns.
- Data is accessed and queried based on the database schema.

Relational database basics (2 of 2)

Main reasons to use a relational database:

- Natively supports SQL
- Provides data integrity
- Supports transactions

Use cases:

- Ecommerce
- Customer relationship management (CRM): Managing interactions with customers
- Business intelligence (BI) tools: Finance reporting and data analysis

Example relational databases:

- MySQL
- Amazon Aurora
- PostgreSQL
- Microsoft SQL Server
- Oracle

aws re/start

Relational databases are popular and continue to be used because they provide many benefits. For example, they support SQL, a language that is relatively simple to learn and get started with to query the data in a relational database. Also, a relational database provides features to protect the integrity of the data that you store. For example, each column attribute in a table is defined to hold only a certain type of data. If you attempt the enter a value that is of a different data type, the database will not allow it.

Relational database examples

Country table

Code	Name	Continent
ARG	Argentina	South America
AUS	Australia	Oceania
BEN	Benin	Africa
BLZ	Belize	North America
BWA	Botswana	Africa

City table

ID	Name	CountryCode
2	Qandahar	AFG
5	Amsterdam	NLD
65	Abu Dhabi	ARE
246	Belford Roxo	BRA
456	London	GBR

aws re/start

In relational databases, tables are used to hold information about the objects to be represented in the database. Each table is organized in columns and rows. Each column (or field) holds a special type of data, such as numbers, strings, or dates. Each row is a set of values for the columns of the table. Each row in a table can also be marked with a unique identifier, which is called a primary key. Rows among multiple tables can become related by using foreign keys.

The following information provides examples:

- Each column in the table holds a certain type of data and stores the actual value of an attribute. In the country table, under the Name column, the kind of data expected is text or strings. Argentina and Belize are examples of actual values.
- The rows in the table represent a collection of related values of one object or entity. In the city table, the row that has an ID of *2* represents the attributes of the city of Qandahar. The ID value is followed by the values of Qandahar and AFG, which represent the city's name and country code, respectively.

Can you identify how the first table can relate to the second table? The tables can be related by the CountryCode field.

These related fields can link multiple tables and are called primary keys and foreign keys.

Nonrelational databases

You will now learn more information about nonrelational databases.

Nonrelational database basics (1 of 2)

What is a nonrelational database?

- A nonrelational database is a database that does not follow the relational model.
- An nonrelational database does not require a fixed definition of the structure of the data.
- A nonrelational database, which is often referred to as a NoSQL database, is a database that does not use a table structure to store data.

aws re/start

A nonrelational database is often referred to as a NoSQL database. *NoSQL* stands for *Not only SQL* because you can use other languages in addition to SQL to access and query a nonrelational database.

Nonrelational database basics (2 of 2)

Use cases:

- Fraud detection
- Internet of Things (IoT)
- Social networks

Example NoSQL databases:

- Amazon DynamoDB
- MongoDB
- Apache Hbase

Example of data that is stored as a JSON document:

```
[
   {
      "ID":1024,
      "Name":"Mumbai (Bombay)",
      "CountryCode":"IND",
      "District":"Maharashtra",
      "Population":"10500000"
   }
]
```


aws re/start

In a relational database, objects are broken down into multiple tables (customers, orders, and more) by using a fixed schema. With a nonrelational database, objects are stored instead with flexible schemas with no constraints, or as documents in JSON or XML formats.

NoSQL databases use various data models for accessing and managing data. These types of databases are optimized specifically for applications that require large data volumes, low latency, and flexible data models. These requirements are achieved by relaxing some of the restrictions around data consistency that other databases use.

Pros and cons of relational and nonrelational databases

Relational databases (SQL)

Pros:

- Known and reliable technology
- Simple-to-write complex queries
- Well-known SQL language
- Well-supported transactions

Cons:

- Use vertical scaling
- Include a fixed schema

Nonrelational databases (NoSQL)

Pros:

- Flexible schema
- Good fit for storing and fast retrieval of massive amounts of data of different types
- Horizontal scaling
- Good fit for hierarchical data

Cons:

- Are a relatively new technology
- Do not guarantee data integrity
- Are not a good fit for complex queries or transactional applications

aws re/start

This slide shows the pros and cons of using relational (SQL) databases.

DBMS

How do you manage stored data?

A DBMS is software that provides database functionality.

DBMS basics (1 of 2)

What is a DBMS?

- A DBMS is software or database as a service (DBaaS) that provides database functionality.
- It is used mainly for the following:
  - Creating databases
  - Inserting data into a database
  - Storing, retrieving, updating, or deleting data in a database
- The primary benefit of a DBaaS is to avoid the cost of installing and maintaining servers.

The following are two variations of DBMSs:

- Single-user DBMS applications, such as Microsoft Access
- Multiple-user DMBS applications, such as Oracle Database, Microsoft SQL Server, MySQL, and IBM Db2

aws re/start

How do you store and manage data in a database? You use a DBMS.

A DBMS is software or database as a service (DBaaS). It is used mainly for creating databases; inserting data into a database; and storing, retrieving, updating, or deleting data in the database.

If you want to avoid the cost of installing and maintaining the servers that your database uses, use a DBaaS service. A DBaaS manages the infrastructure to support your database so that you can focus on designing, creating, and maintaining the data in your database.

DBMS basics (2 of 2)

Locations

On-premises (data center): Data is stored on computers and networks that the organization owns. These computers and networks are located in the organization's own data center.

In the cloud (virtualized data center): Data is located outside the organization, such as in remote data centers that are owned by cloud providers such as Amazon Web Services (AWS).

aws re/start

With traditional database management, a company uses a DBMS in an on-premises data center. The company must invest in the hardware and software resources needed to support the DBMS.

With DBaaS, the DMBS is in the cloud in a virtualized data center. The company needs to pay for only the database-related cloud resources that they use.

DBaaS

A few key points about cloud-based databases

- **Hosted by third-party providers:**
  - These database servers are hosted in third-party data centers and accessed over the internet (the cloud) instead of being hosted on local networks.
- **Reduced cost:**
  - These databases reduce the cost of installing and maintaining servers.
- **Fully managed:**
  - For example, with managed AWS databases, you don't need to manage database management tasks, such as server provisioning, patching, setup, configuration, backups, or recovery.
- **Faster:**
  - With these databases, you can use companies, such as AWS, that offer large amounts of storage and processing power in their data centers.

aws re/start

With DBaaS, database servers are hosted in third-party data centers. They are accessed over the internet (the cloud) instead of being hosted on local networks.

DBaaS examples

Amazon Relational Database Service (Amazon RDS)

Amazon RDS manages common relational database administrative tasks.

Amazon Aurora

A part of Amazon RDS, Aurora is a fully managed relational database engine.

Amazon DynamoDB

DynamoDB is a fully managed NoSQL database service.

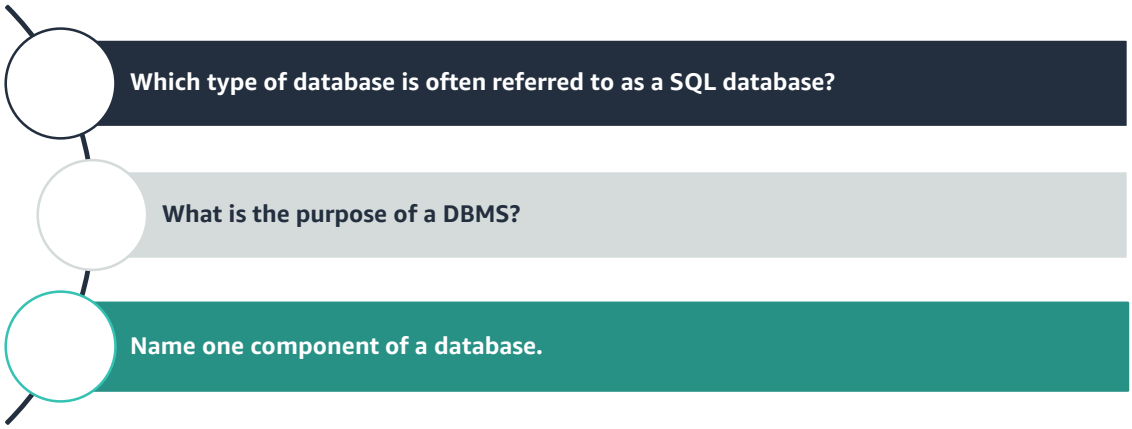
aws re/start

Amazon Relational Database Service (Amazon RDS) is a web service that facilitates setting up, operating, and scaling a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Amazon Aurora is a fully managed relational database engine that is compatible with MySQL and PostgreSQL. Aurora is part of the managed database service Amazon RDS.

Amazon DynamoDB is a fully managed NoSQL database service. You can use DynamoDB to create a database table that stores and retrieves any amount of data and serves any level of request traffic. DynamoDB automatically spreads the table's data and traffic over a sufficient number of servers to handle the request capacity that the customer specifies. It can handle the amount of data that is stored while maintaining consistent and fast performance.

Checkpoint questions



aws re/start

1. Which type of database is often referred to as a SQL database?

   A relational database.

2. What is the purpose of a DBMS?

   A DBMS is software that provides database functionality.

3. Name one component of a database.

   Any one of the following answers is correct: data, data models, schema, and small scale or distributed.

Key takeaways

- A relational database, which is often referred to as a SQL database, provides the ability for data to be spread across multiple tables.
- A nonrelational database, which is often referred to as a NoSQL database, does not use a table structure to store data.
- A DBMS is software that provides database functionality.

aws re/start

This module includes the following key takeaways:

- A relational database, which is often referred to as a SQL database, provides the ability for data to be spread across multiple tables.
- A nonrelational database, which is often referred to as a NoSQL database, does not use a table structure to store data.
- A DBMS is software that provides database functionality.

Thank you

Thank you for completing this module.