Database Objects and Hierarchy (Including Schemas)
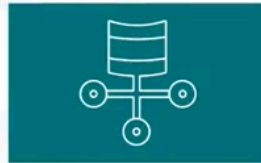
What will you learn

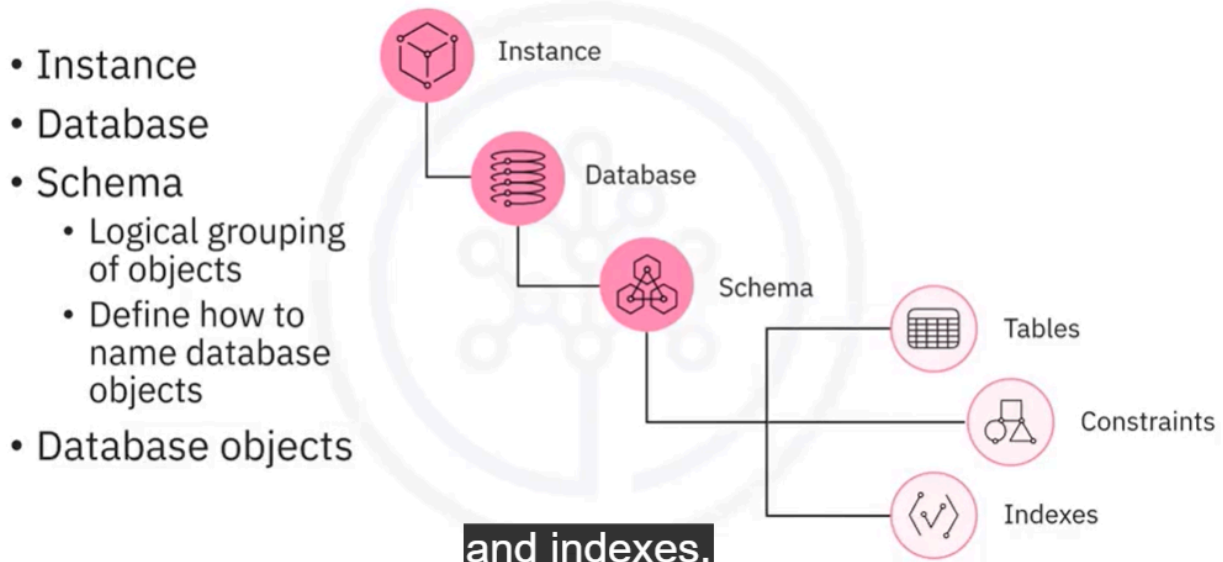|			
Describe the hierarchical structure of database objects	Explain the significance of database instances	Define relational database	Describe schema functionality and usage

Database hierarchy

- RDBMSs contain many objects
- Enables to:
  - Manage security
  - Maintenance
  - Accessibility
- Provides an overview of the RDBMS structure

- Relational database management systems RDBMSs contain many objects that database engineers and database administrators must organize. Storing tables, constraints, indexes, and other database objects in a hierarchical

structure allows database administrators to manage security, maintenance, and accessibility. This hierarchy gives you an overview of how RDBMSs are structured.

Database hierarchy

- Instance
- Database
- Schema
  - Logical grouping of objects
  - Define how to name database objects
- Database objects

Instance

Database

Schema

Tables

Constraints

Indexes

and indexes.

- most RDBMSs begin with an instance, a single way of organizing the database and everything it contains. Many RDBMSs permit more than one database within a single instance. You will generally find at least one schema at some level in the hierarchy. A schema is a logical grouping of objects within a database. Schemas define how database objects are named and prevent ambiguous references. Some RDBMSs consider the schema a parent object of a database, and others consider it a database object. There are database objects within a schema, including tables, constraints, and indexes.

Instances

Instance:
- Logical boundary for databases
- Organize objects
- Set configuration parameters

- Provides unique database server environment
- Allows isolation between databases
- Not all RDBMSs use instances

- An instance is a logical boundary for a database, or set of databases where you organize database objects and set configuration parameters. Every database within an instance is assigned a unique name, has its own set of system catalog tables, which keep track of the objects within the database, and has its own configuration files. You have the option to generate multiple instances on a single physical server, ensuring a distinct database server environment for each instance. The databases and other objects within one instance are isolated from those in any other instance. It is possible to establish multiple instances on a single physical server, thereby creating distinct database server environments for each instance. Not all RDBMSs use the concept of instances, often managing database configuration information in a special database instead. In cloud based RDBMs, the term instance means a specific running copy of a service.
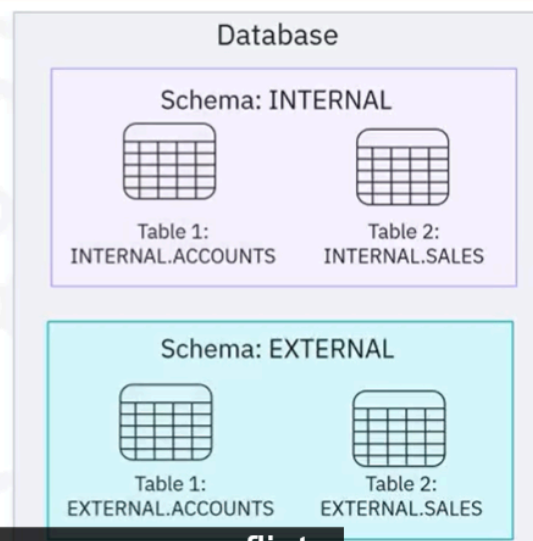
Relational databases

- Stores, manages, and provides access to data
- Contains objects like tables, views, indexes
- Uses related tables to reduce data redundancy
- Distributed across multiple systems
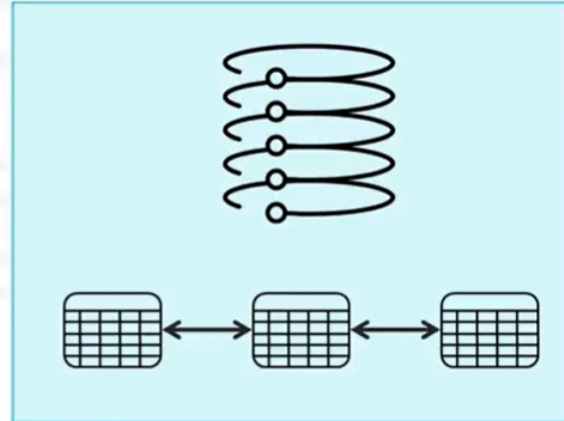
- A relational database comprises objects designed for the storage, management, and retrieval of data. These objects include tables, views, indexes, functions, triggers, and packages. Database objects can be either defined by the system built-in objects, or defined by the user. User defined objects in a relational database database engineers establish relationships between tables to reduce redundant data and improve data integrity. A distributed relational database shares tables and other objects across different but interconnected computer systems.

Schemas

- Organize database objects
- Include various objects
- Assign it to a schema
- Include schema name
- Default schema is the user schema
- Provides a naming context
- Enables you to distinguish objects without encountering namespace conflicts

- A schema is a specialized database object that provides a way to group other database objects logically. A schema has the capability to include tables, views, nicknames, triggers, functions, packages, and various other objects. When you create a database object, you can assign it to a schema. If you want to assign the object to a specific schema, you can explicitly include the schema name. If you don't include the schema name, the object is implicitly assigned to the current schema. In most RDBMSs, the default schema is the user schema for the currently logged on user. A schema also provides a naming context, using the schema name as a name qualifier enables you to distinguish between objects with the same name in different schemas. For example, the schema names internal and external make it easy to distinguish two different sales tables, internal sales in the internal schema, and external sales in the external schema. Thus, schemas enable multiple applications to store data in a single database without encountering namespace conflicts.

Schemas

- Many RDBMSs use a specialized schema
- Tables in a system schema store:
    - Lists of database users
    - Access permissions
    - Information about the indexes
    - Details of any database partitions
    - User-defined data types

- Many RDBMSs use a specialized schema to hold configuration information and metadata about a particular database. For example, tables in a system schema can store lists of database users and their access permissions, information about the indexes on tables, details of any database partitions that exist, and user-defined data types.
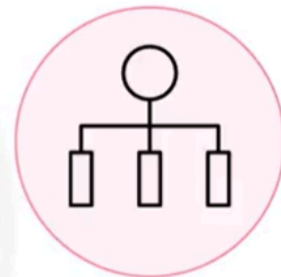
Database partitions

- Data managed across multiple partitions
- Split tables that contain very large quantities of data
- Hold a subset of the data
- Common in data warehousing
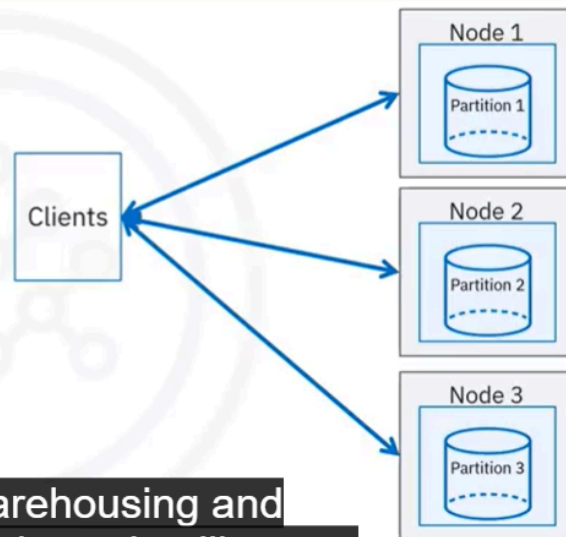
such as data warehousing and data analysis for business intelligence

- A partitioned relational database is one where data is distributed and managed across multiple database partitions within the relational database system. You can partition tables that need to contain large quantities of data into multiple logical partitions, with each partition containing a subset of the overall data. Database partitioning is used in scenarios that involve large volumes of data, such as data warehousing and data analysis for business intelligence.

Database objects

Physical database design consists of defining database objects

Database objects

Common database objects:

- Tables
- Constraints
- Indexes
- Views
- Aliases

Create using Graphical tools, scripting, APIs, SQL

Tables   Constraints   Indexes

- Database objects encompass items present within the database. The process of database design includes defining database objects and their relationships with each other.

- In most RDBMSs, you can create objects such as tables, these are logical structures consisting of rows and columns that store data. Constraints within any business, data is often subject to certain restrictions or rules. For example, it is imperative for an employee number to be distinct or unique. Constraints provide a way to enforce such rules. Indexes, an index is a set of pointers used to improve performance and ensure the uniqueness of the data. Views, a view provides a different way of representing the data in one or more tables. It's important to note that a view is not an actual table, and does not demand permanent storage. Aliases, an alias serves as an alternative name for an object such as a table. It can be used to provide a shorter, simpler name to reference an object. You can create and manage database objects through graphical database management tools, scripting or accessing the database through an API. If you are using SQL to create or manage the object, you will use data definition language statements like create or alter.