**What is Database?** – Place where organized collection of data is stored where data can be stored, manipulated, retrieved from and accessed efficiently. Database comprises of storage, memory, processor and software.

**What is DBMS?** – Database Management System is a collection of programs that enables the users to access, manipulate and retrieve the databases.

**Major types of DBMS:**

1. **Hierarchical DBMS** – Parent-child relationships, Data storage is like tree with nodes and branches. Windows registry is an example.
2. **Network DBMS** – [RDM](https://www.techtarget.com/searchvmware/definition/raw-device-mapping-RDM) server is an example. Many-to-Many relationships. Generally, results in complex DB structures.
3. **Relational DBMS** – The most widely used DBMS. Oracle, MySQL, MsSQL use this type. Does **NOT** support many-to-many relationship. RDBMS have entities such as tables and relations are established between these entities. Type of data that can be stored in a RDBMS is predefined.
4. **Object-Relational DBMS** - Supports storing new datatypes because we can store our won objects in it. PostgreSQL is the most commonly known ORDBMS.

**Oracle DB Features:**

* Security
* Performance
* Scalability
* Powerful Coding (using Oracle SQL/ PLSQL)
* My Oracle support

**Table in DB:**

* The data in a DB are stored in Tables.
* Each cell in a Table contains one datum.
* The union of horizontal cells is called a **record** or a **row**.
* The union of horizontal cells is called a **column**.
* **Rows** – Any type of data (string, int, etc.) can be stored.
* **Columns** – Only one type of data can be stored.
* A collection of all these rows and columns is called a **Table**.
* A **Table** is only one of the many object types that a database can have.

**What is RDBMS:**

Data is divided into different tables and parent-child relationship is established in-between them.

Advantages of RDBMS include,

1. Accuracy (duplication is avoided,)
2. Flexibility,
3. Collaborative,
4. Trusted DBMS and
5. Security (restricted access to users.)

Relational Databases use “entity-relationship” logic to establish the relation between various entities.

**What is an entity?**

An entity is the smallest unit containing a meaningful set of data.

**Pluggable Database (a.k.a Multitenant Architecture):**

Pluggable database concept was introduced in 12c. Pluggable databases have all the attributes of a DB such as its own users, objects, applications, etc. These pluggable databases will be a part of a container database which acts like the “root.” A container database does not store any object (only the metadata such as configuration files) and it must contain at least one pluggable database.

**Database Objects:**

These are categorized into 2 sections:

1. Schema Objects
2. Non-schema objects

**Schema Objects:**

A schema is a collection of logical structures of data or objects. Schema Objects are the logical structures created by the users.

A database user has only one schema with the same name as the username. **Example:** A “hr” user has only one schema named “hr.”

Some Schema Objects are,

**1.Tables:**

Tables are the basic units of database to store data, formatted with columns and rows.

**2.View:**

A view is a virtual table that provides access to a subset of columns or some restricted rows from one or more tables. Views are SQL scripts that have a name like a table and when we query from a view, the view executes that query and retrieves the data. Views act like a real table but they do not take up any physical space for returning the data. Example:

Creating a view:

CREATE VIEW high\_salaries

SELECT department\_id, MAX(salary) salary

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID;

Using the view:

SELECT \* FROM high\_salaries;

**3.Constraints:**

Constraints are rules to restrict the entry of invalid data into the tables.

**4.Index:**

Indices are used to improve the speed of retrieval of data from tables. Indices know the physical addresses of data in disks and goes to these blocks directly. They work automatically and cannot be run manually.

**5.Sequence:**

Sequences are the database objects that generate unique integers and are generally used for primary key values (i.e., unique values for a row.) The major difference between sequences and other databases is that sequences can be accessed by multiple users/tables.

**6.Synonym:**

A synonym is an alias for a database object that references to the original object. Using synonyms, one can query other users’ objects without writing its schema name. **Example:**

Let’s say a user named Gnanam creates a schema named Employees. If we want to access this schema, we need to write “Gnanam.Employees”. But, if Gnanam creates a public synonym named ‘employees,’ then we can simply retrieve it by writing “employees.”

**7.Materialized view:**

Unlike View, a Materialized View has a real table with real values in it that are filled by a SQL query. This real table is truncated and refilled in a specific time frequency.

Sometimes a view query might take a long time to retrieve the required data. During this scenario, we can use the Materialized view in order to store the data from a view query and retrieve the data from the real table in the future. This increases the performance noticeably in most cases.

**8.Functions and Procedures:**

**PL/SQL** is **P**rocedural **L**anguage extension to **SQL**. It extends the capabilities of SQL (PL/SQL = SQL + Procedural Programming.) Using PL/SQL, we can have variables, control structures such as if-structure, loops, triggers, I/O operations, etc.

Functions and procedures are sets of compiled codes that are written just once and are called upon whenever required. SQL and PL/SQL codes can be used together in functions and procedures.

A Function **returns an output**, where as a Procedure **does not return an output**.

**9.Triggers:**

Triggers are compiled program units that are stored in a database that are executed with some specific event like insert, update, delete, create, etc. and are used to perform other operations BEFORE, ON or AFTER the main operation.

**10.Packages:**

Packages are the schema objects that are compiled and stored in the database and consist of logically grouped SQL and PL/SQL codes, variables, cursors, etc. in it, to perform one or more than one operation by using its functions or procedures. Packages have functions and procedures and you use these functions or procedures to perform certain operations.

Packages can include many functions and procedures together and you can call and use any of them by writing "package\_name.function\_name" or "package\_name.procedure\_name".

**11.Database Links:**

These are the connections in between two physical database servers that are used to call upon the objects of other databases.

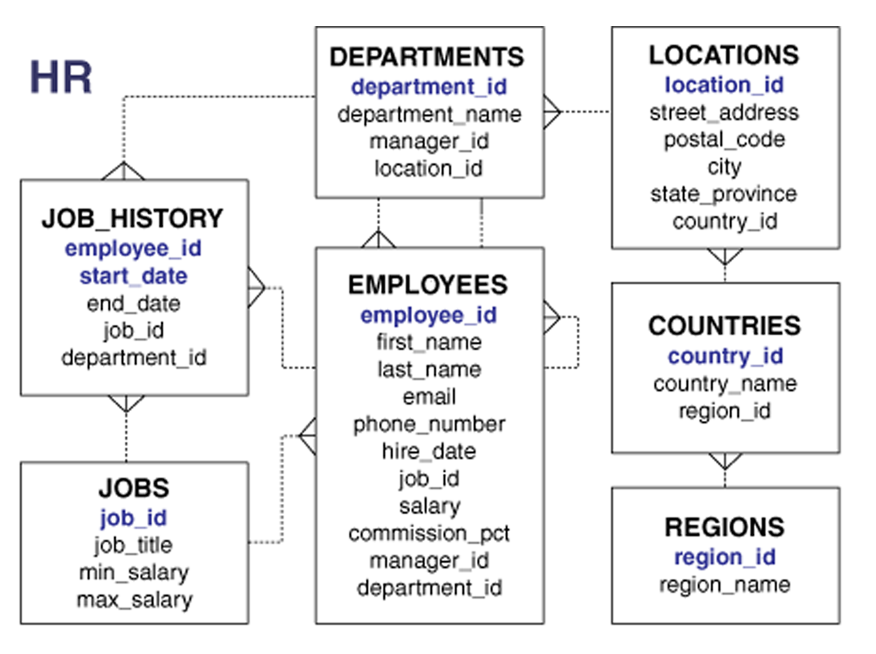
**Non-Schema Objects:**

These are objects that are stored in a database but are not placed under a schema. Some examples are directories, parameter files, roles, tablespaces, users, etc. Except for roles, all the others belong to database administration.

**The HR Schema used in the course:**

**What is Schema?** – A Schema is a logical set of all the objects of a user. A user can have only one schema hence, schemas can also be referred to as users. A schema can contain tables, views, triggers, constraints, etc.

**HR Schema** is a schema as well as a user that will be used in this course. Below is the entity-relationship (ER) diagram for the HR schema.



The column names that are **blue** in color are the **primary keys** for those tables. Primary keys are the IDs for each record and they are unique. These primary keys are used to establish the connections with other tables.

**What is SQL?** – **S**tructured **Q**uery **L**anguage is a language used to interact with the database.