

# D B M S

Created by :

Sangeeta Joshi

# Agenda

- DBMS

# DBMS

- A database management system (or DBMS) is essentially a computerized data-keeping system.
- Users of the system are given facilities
  - to perform several kinds of operations on such a system
    - for either manipulation of the data in the database
    - or the management of the database structure itself.
- Database Management Systems (DBMSs) are categorized according to their data structures or types.

# Database

Relational databases include the following structures:

- *Database*
  - A database is a logical grouping of data.
  - It contains a set of related table spaces and index spaces.
  - A database contains all the data that is associated with one application or with a group of related applications.
  - Ex: a payroll database or an inventory database

# Tables

## *Table:*

- A table is a logical structure made up of rows and columns.
- Rows have no fixed order,
- The order of the columns is the order specified when the table was created .
- At the intersection of every column and row is a specific data item called a value, or, more precisely, an atomic value.
- There are three types of tables:
  - A base table that is created and holds persistent data
  - A temporary table that stores intermediate query results
  - A results table that is returned when you query tables.

# Tables

| DEPTNO | DEPTNAME                     | MGRNO  | ADMRDEPT |
|--------|------------------------------|--------|----------|
| A00    | SPIFFY COMPUTER SERVICE DIV. | 000010 | A00      |
| B01    | PLANNING                     | 000020 | A00      |
| C01    | INFORMATION CENTER           | 000030 | A00      |
| D01    | DEVELOPMENT CENTER           |        | A00      |
| E01    | SUPPORT SERVICES             | 000050 | A00      |
| D11    | MANUFACTURING SYSTEMS        | 000060 | D01      |
| D21    | ADMINISTRATION SYSTEMS       | 000070 | D01      |
| E11    | OPERATIONS                   | 000090 | E01      |
| E21    | SOFTWARE SUPPORT             | 000100 | E01      |

In this table we use:

**Columns**—The ordered set of columns are DEPTNO, DEPTNAME, MGRNO, and ADMRDEPT. All the data in a given column must be of the same data type.

**Rows**—Each row contains data for a single department.

**Values**—At the intersection of a column and row is a value.

For example, PLANNING is the value of the DEPTNAME column in the row for department B01.

# Indexes

- An index is an ordered set of pointers to rows of a table. Unlike the rows of a table that are not in a specific order, an index must always be maintained in order.
- An index is used for two purposes:
  - For performance, to retrieve data values more quickly
  - For uniqueness.
- By creating an index on an employee's name, you can retrieve data more quickly for that employee than by scanning the entire table.
- Also, by creating a unique index on an employee number, DB will enforce the uniqueness of each value.

# Keys

- A key is one or more columns that are identified as such in the creation of a table or index, or in the definition of referential integrity.

## Primary key

- A table can only have one primary key because it defines the entity. There are two requirements for a primary key:
  - It must have a value, that is, it cannot be null.
  - It must be unique.

## Foreign key

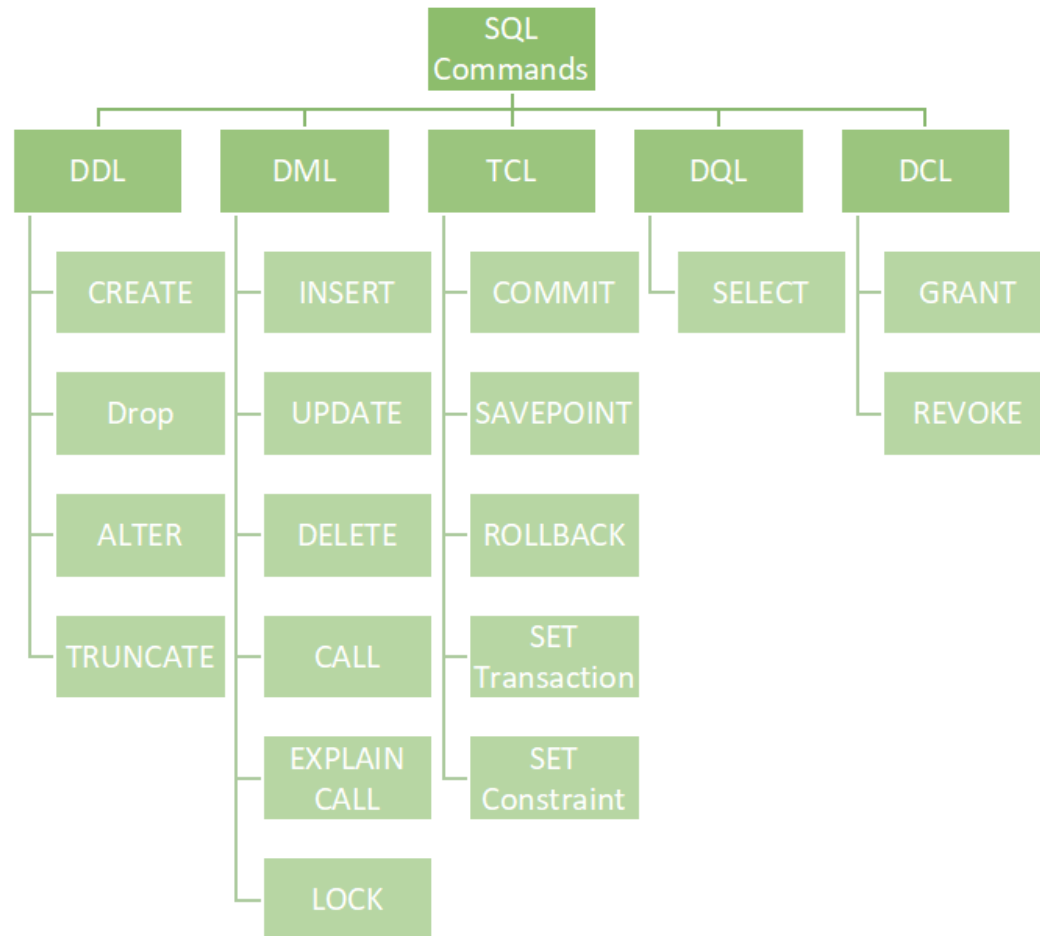
- A foreign key is a key that is specified in a referential integrity constraint to make its existence dependent on a primary or in another table.



# SQL

- Structured Query Language(SQL) is the database language
  - we can perform certain operations on the existing database
  - we can use this language to create a database.
- SQL commands are mainly categorized into five categories :
  - DDL – Data Definition Language
  - DQL – Data Query Language
  - DML – Data Manipulation Language
  - DCL – Data Control Language
  - TCL – Transaction Control Language

# SQL



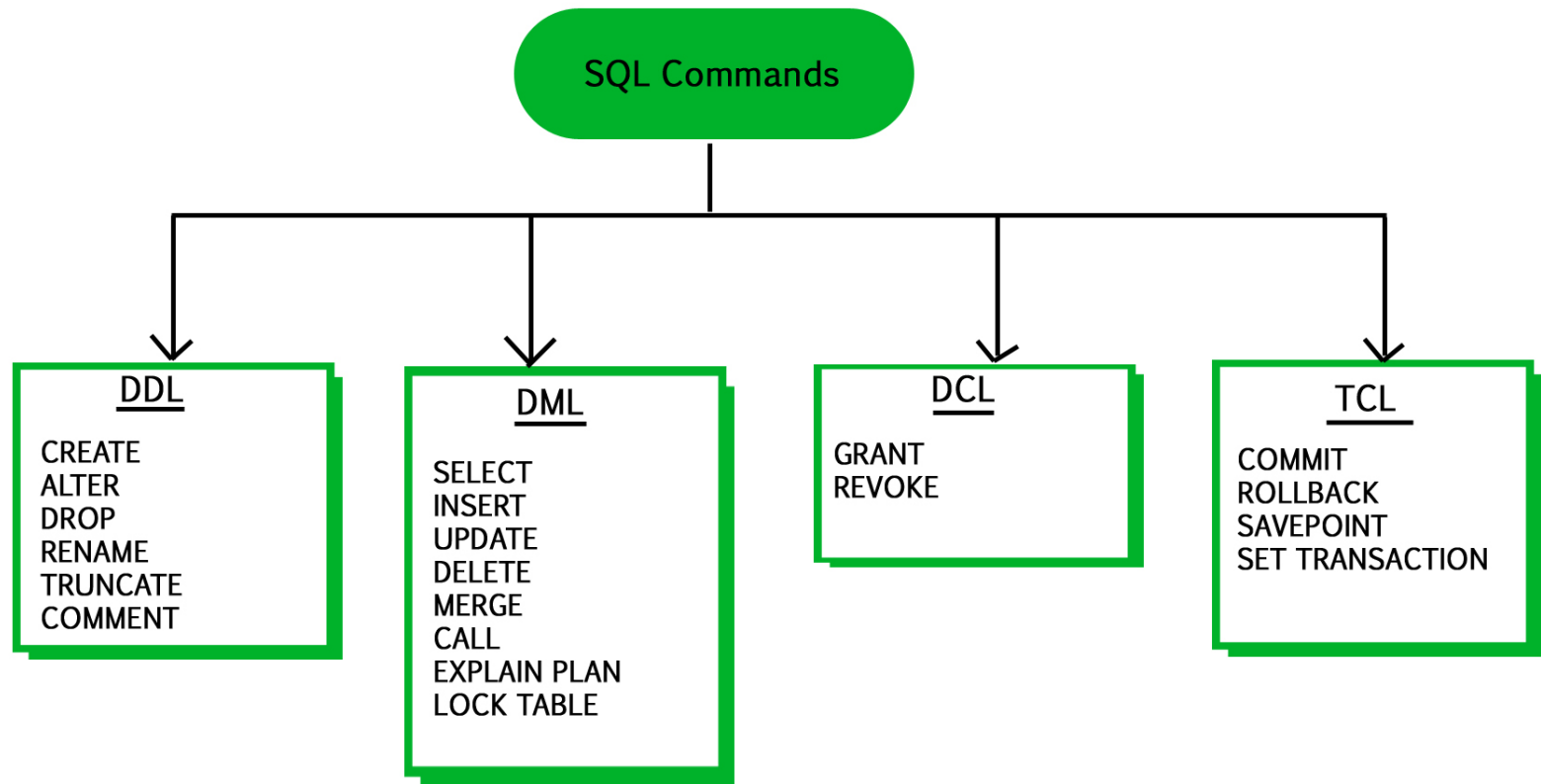
# DDL

- Data Definition Language:
  - consists of the SQL commands to be used to *define the database schema*.
  - It is used to create and modify the structure of database objects in the database.
  - DDL is a set of SQL commands used to
    - create,
    - modify
    - deletedatabase structures but not data.
  - These commands are normally not used by a general user, who should be accessing the database via an application.

# DDL

List of DDL commands:

- *CREATE*: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- *DROP*: This command is used to delete objects from the database.
- *ALTER*: This is used to alter the structure of the database.
- *TRUNCATE*: This is used to remove all records from a table, including all spaces allocated for the records are removed.
- *COMMENT*: This is used to add comments to the data dictionary.
- *RENAME*: This is used to rename an object existing in the database.



# DDL

- Create Table

```
CREATE TABLE table_name  
(  
  column_1 datatype,  
  column_2 datatype,  
  column_3 datatype,  
  ....  
);
```

# DDL

## Alter Table

- This command is used to add, delete or change columns in the existing table.

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
ALTER TABLE `myschema1`.`student` ADD COLUMN `cgpa` INT NULL  
AFTER `studentCourse`;
```

## Truncate Table

This command is used to remove all rows from the table, but the structure of the table still exists.

Syntax –

Syntax to remove an existing table.

```
TRUNCATE TABLE table_name;
```

# DDL

DROP :

This command is used to remove an existing table along with its structure from the Database.

Syntax –

Syntax to drop an existing table.

```
DROP TABLE table_name;
```



# D M L

## Data Manipulation Language

- SQL that deals with *manipulation of data* present in the database
- It is the component of the SQL statement that controls access to data and to the database.
- Basically, DCL statements are grouped with DML statements.

## List of DML commands:

- *INSERT* : It is used to insert data into a table.
- *UPDATE*: It is used to update existing data within a table.
- *DELETE* : It is used to delete records from a database table.
- *LOCK*: Table control concurrency.
- *CALL*: Call a PL/SQL or JAVA subprogram.
- *EXPLAIN* PLAN: It describes the access path to data.

# DML

1. *INSERT INTO table\_name VALUES (value1, value2, value3);*

2. If we want to insert values in the specified columns then we use the following query:

*INSERT INTO table\_name (column1, column2, column3)  
VALUES ( value1, value2, value3); table\_name:*

3. *Using SELECT in INSERT INTO Statement:*

We can use the SELECT statement with INSERT INTO statement to copy rows from one table and insert them into another table.

**Inserting all columns of a table:**

*INSERT INTO first\_table SELECT \* FROM second\_table;*

Inserting specific column values:

*INSERT INTO first\_table(names\_of\_columns1)*

*SELECT names\_of\_columns2 FROM second\_table;*

# DML

To insert multiple rows in a table using Single SQL Statement:

```
INSERT INTO  
table_name(Column1,Column2,Column3,  
.....)
```

```
VALUES (Value1, Value2,Value3,.....),
```

```
(Value1, Value2,Value3,.....),
```

```
(Value1, Value2,Value3,.....),
```

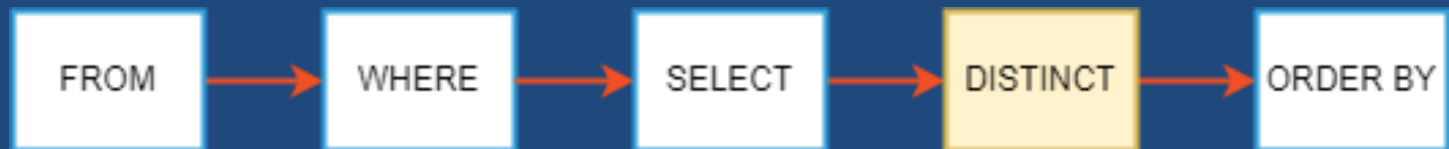
```
..... ;
```

# select

- Use the ORDER BY clause to sort the result set by one or more columns.
- Use the ASC option to sort the result set in ascending order
- DESC option to sort the result set in descending order.
- The ORDER BY clause is evaluated after the FROM and SELECT clauses.
- In MySQL, NULL is lower than non-NULL values
- Use the MySQL DISTINCT clause to remove duplicate rows from the result set returned by the SELECT clause.

# Select distinct

When executing the SELECT statement with the DISTINCT clause, MySQL evaluates the DISTINCT clause after the FROM, WHERE, and SELECT clause and before the ORDER BY clause:



# Select Limit

- The LIMIT clause is used in the SELECT statement to constrain the number of rows to return.
- The LIMIT clause accepts one or two arguments.
- The values of both arguments must be zero or positive integers.

# Select Limit

```
SELECT n FROM t  
ORDER BY n  
LIMIT 3, 4;
```

|     |             |
|-----|-------------|
| 1   |             |
| 2   |             |
| 3   |             |
| 4   | OFFSET 3    |
| 5   |             |
| 6   | ROW_COUNT 4 |
| 7   |             |
| ... |             |

# Select & Limit

- By default, the SELECT statement returns rows in an unspecified order. When we add the LIMIT clause to the SELECT statement, the returned rows are unpredictable.
- Therefore, to ensure the LIMIT clause returns an expected output, we should always use it with an ORDER BY clause



# Where clause

The WHERE clause allows to specify a search condition for the rows returned by a query.

the syntax of the WHERE clause:

```
SELECT  
    select_list  
FROM  
    table_name  
WHERE  
    search_condition;
```

The search\_condition is a combination of one or more expressions using the logical operator *AND, OR and NOT*.

In MySQL, a predicate is a Boolean expression that evaluates to TRUE, FALSE, or UNKNOWN.

# comparison operators in WHERE clause.

| Operator | Description  |
|----------|--|
| =        | Equal to. You can use it with almost any data type.                    |
| <> or != | Not equal to   |
| <        | Less than. You typically use it with numeric and date/time data types. |
| >        | Greater than.  |
| <=       | Less than or equal to  |
| >=       | Greater than or equal to   |

# Where clause

- Use the WHERE clause to filter rows by a condition.
- MySQL evaluates the WHERE clause after the FROM clause and before the SELECT and ORDER BY clauses.

# DML Update

The basic syntax of the UPDATE query with a WHERE clause is as follows –

```
UPDATE table_name  
SET column1 = value1, column2 = value2...., columnN = valueN  
WHERE [condition];
```

# D C L (Data Control Language)

- DCL includes commands mainly deal with
  - rights,
  - permissions,
  - and other controls of the database system.
- List of DCL commands:

*GRANT*: This command gives users access privileges to the database.

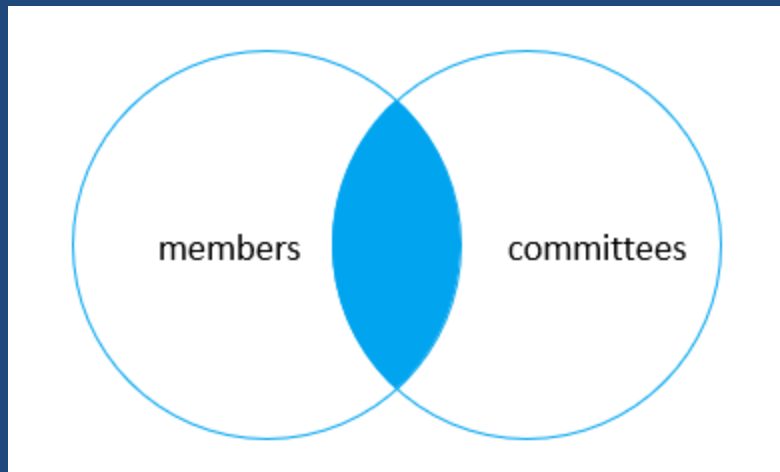
*REVOKE*: This command withdraws the user's access privileges given by using the GRANT command.

# JOINS

- A relational database consists of:  
multiple related tables linking together using common columns i. e. *foreign key* columns.  
Because of this, data in each table is incomplete from the *business perspective*.
- That's why *joins* come into the play.
- A join is a method of linking data between one (self-join) or more tables based on values of the common column between the tables.

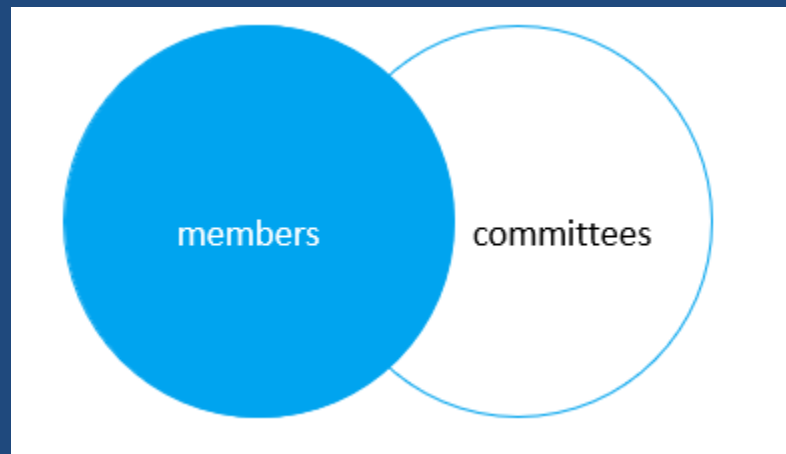
# Inner Join

- The inner join clause joins two tables based on a condition which is known as a join predicate.
- inner join clause includes only matching rows from both tables.



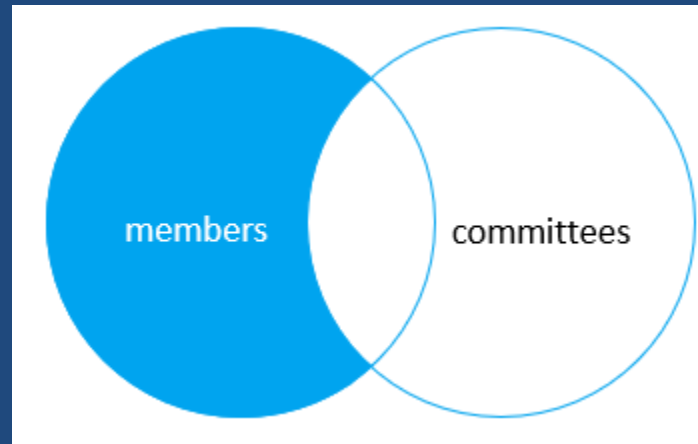
# LEFT JOIN

- left join selects all data from the left table whether there are matching rows exist in the right table or not.
- In case there are no matching rows from the right table found, the left join uses NULLs for columns of the row from the right table in the result set.



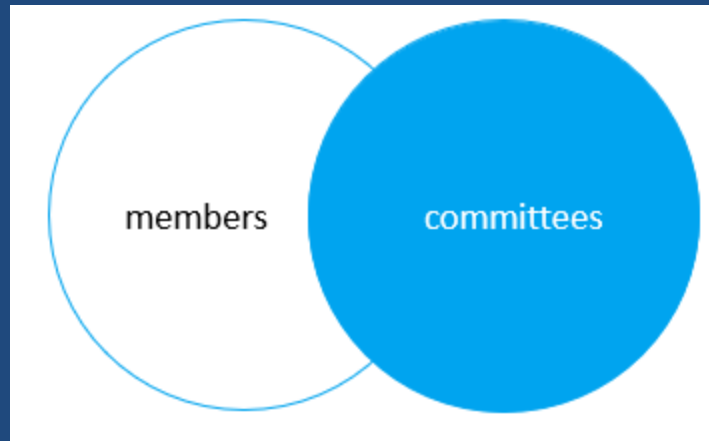


# Left Join

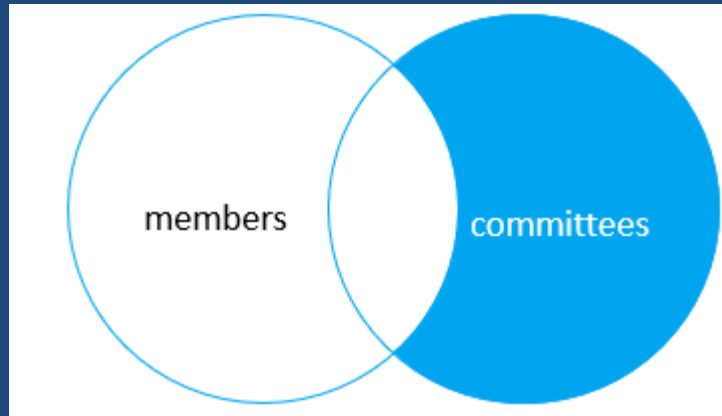


# Right Join

- The right join clause selects all rows from the right table and matches rows in the left table.
- If a row from the right table does not have matching rows from the left table, the column of the left table will have NULL in the final result set.



# Right Join



# Cross Join

- Unlike the inner join, left join, and right join, the cross join clause does not have a join condition.
- The cross join makes a Cartesian product of rows from the joined tables. The cross join combines each row from the first table with every row from the right table to make the result set.
- Suppose the first table has  $n$  rows and the second table has  $m$  rows. The cross join that joins the tables will return  $n \times m$  rows.