

# What is Data?



- **Data** is nothing but facts and statistics stored or free flowing over a network, generally it's raw and unprocessed.
- For example: When you visit any website, they might store you IP address, that is data, in return they might add a cookie in your browser, marking you that you visited the website, that is data, your name, it's data, your age, it's data.
- Data becomes **information** when it is processed, turning it into something meaningful.

# What is a Database?



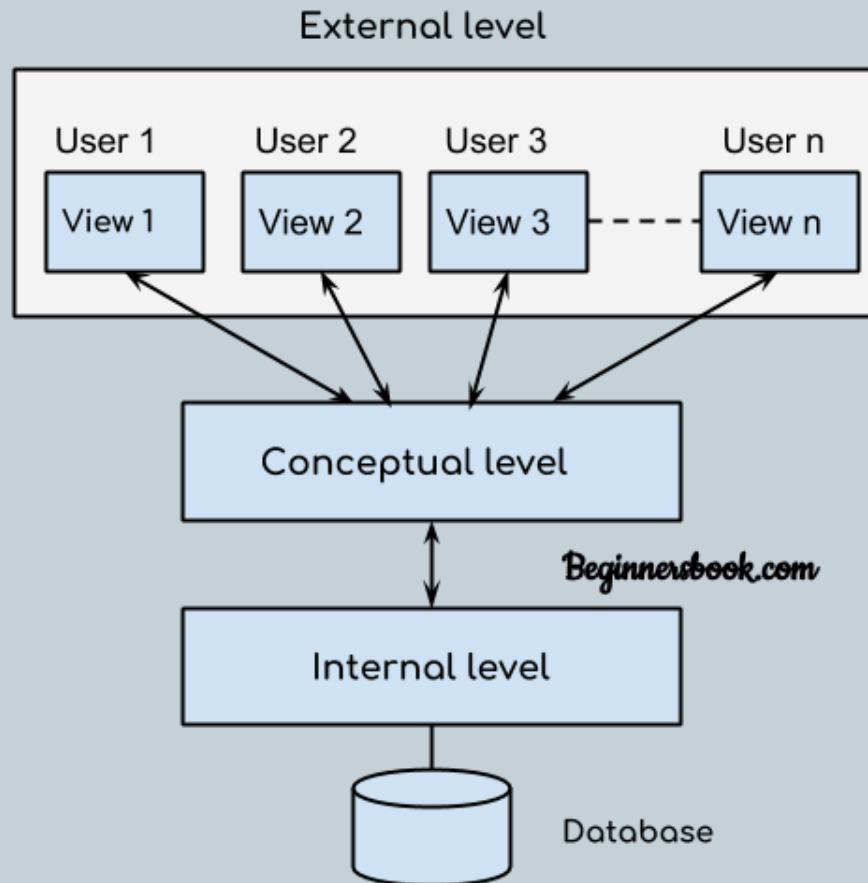
- A **Database** is a collection of related data organized in a way that data can be easily accessed, managed and updated.
- Database can be software based or hardware based, with one sole purpose, storing data.

# What is DBMS?



- A **DBMS** is a software that allows creation, definition and manipulation of database, allowing users to store, process and analyze data easily.
- DBMS provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more.
- DBMS also provides protection and security to the databases. It also maintains data consistency in case of multiple users.
- Example:
  - MySql
  - Oracle
  - SQL Server
  - Amazon SimpleDB (cloud based) etc.

# DBMS architecture



1. **External level:** several users can view their desired data from this level which is internally fetched from database with the help of conceptual and internal level mapping.
2. **Conceptual level:** The whole design of the database such as relationship among data, schema of data etc. are described in this level.
3. **Internal level:** This level describes how the data is actually stored in the storage devices. This level is also responsible for allocating space to the data.

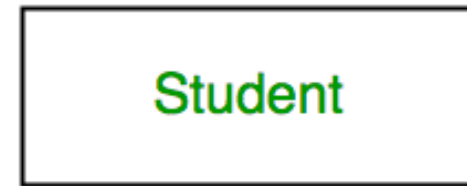
# DBMS - Data Models



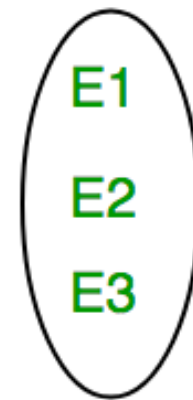
- Data models define how the logical structure of a database is modeled.
- Data Models are fundamental entities to introduce abstraction in a DBMS.
- Data models define how data is connected to each other and how they are processed and stored inside the system.

# Definitions

- An Entity is an object of Entity Type and set of all entities is called as entity set. e.g.; E1 is an entity having Entity Type Student and set of all students is called Entity Set.
- In ER diagram, Entity Type is represented as:



Entity Type



Entity Set

# Definitions



- **Attribute:** Attributes are the **properties which define the entity type**.
- For example, Roll\_No, Name, DOB, Age, Address, Mobile\_No are the attributes which defines entity type Student.
- In ER diagram, attribute is represented by an oval.





# Definitions

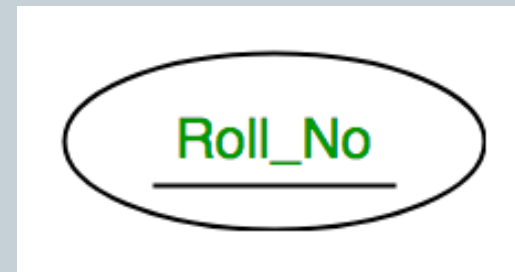


## Key Attribute –

The attribute which **uniquely identifies each entity** in the entity set is called key attribute.

For example, Roll\_No will be unique for each student.

In ER diagram, key attribute is represented by an oval with underlying lines.



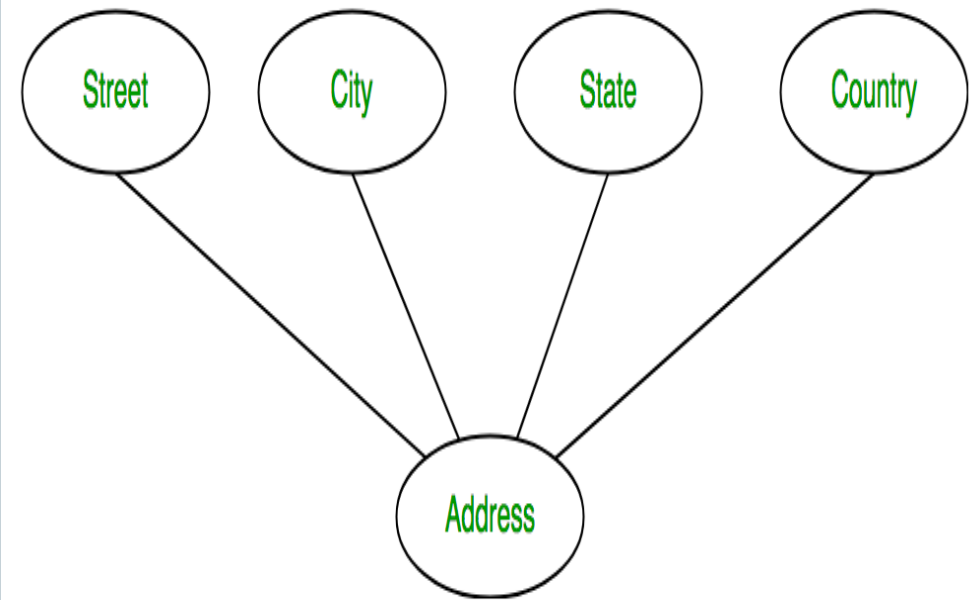
# Definitions

## Composite Attribute –

An attribute composed of many other attribute is called as composite attribute.

For example, Address attribute of student Entity type consists of Street, City, State, and Country.

In ER diagram, composite attribute is represented by an oval comprising of ovals.



# Definitions

## Multivalued Attribute –

An attribute consisting **more than one value** for a given entity.

For example, Phone\_No (can be more than one for a given student).

In ER diagram, multivalued attribute is represented by double oval



# Definitions

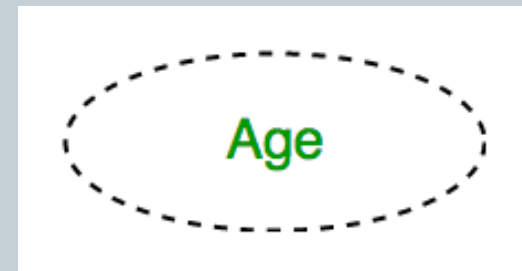


## Derived Attribute –

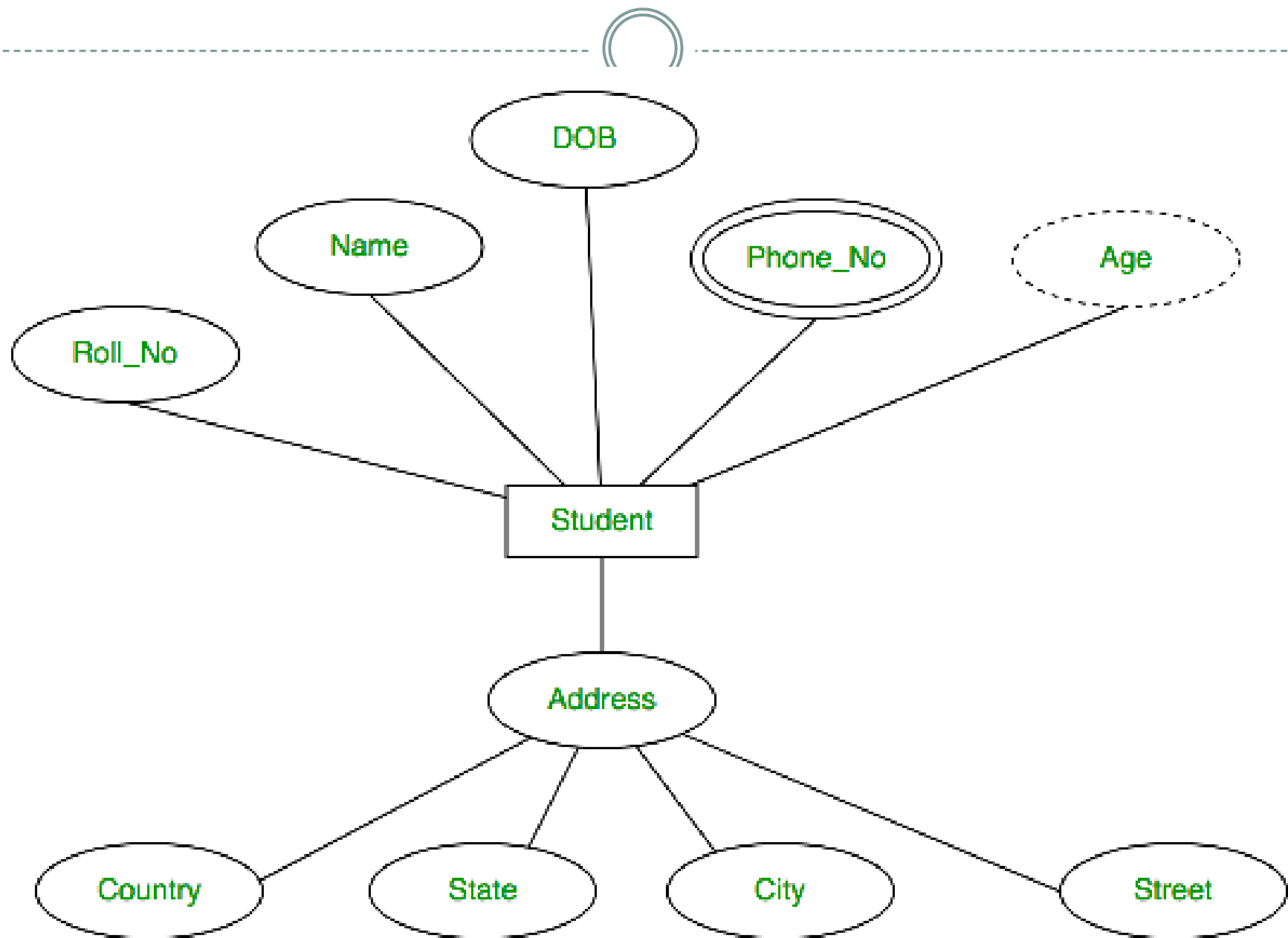
An attribute which can be **derived from other attributes** of the entity type is known as derived attribute.

e.g.; Age (can be derived from DOB).

In ER diagram, derived attribute is represented by dashed oval.

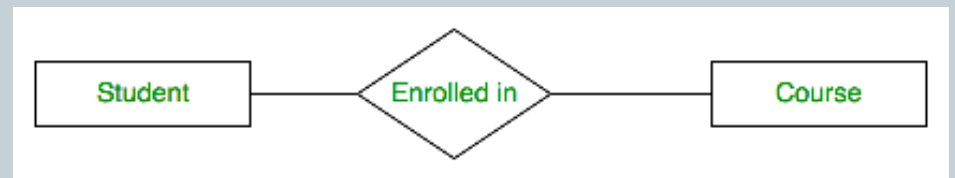


# Entity type Student



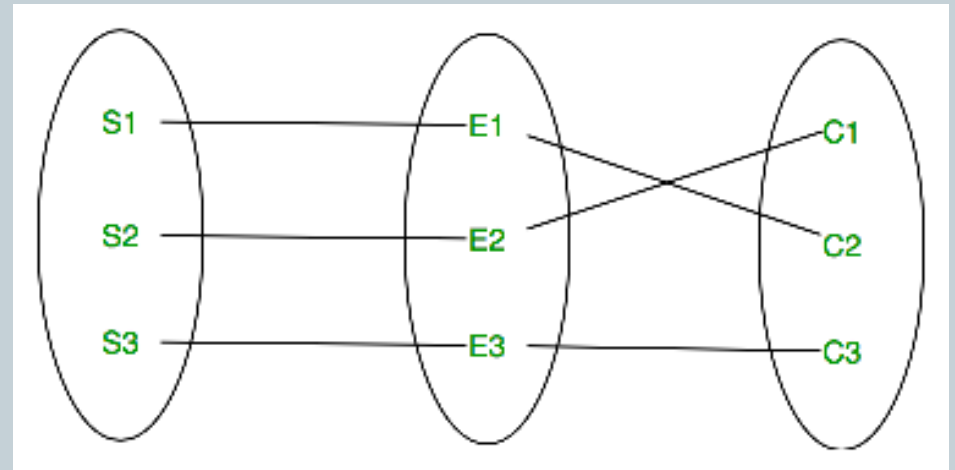
# Definitions

- **Relationship Type** : A relationship type represents the **association between entity types**. For example, 'Enrolled in' is a relationship type that exists between entity type Student and Course.
- In ER diagram, relationship type is represented by a diamond and connecting the entities with lines.



# Definitions

- **Relationship Set:**
- A set of relationships of same type is known as relationship set.
- The following relationship set depicts S1 is enrolled in C2, S2 is enrolled in C1 and S3 is enrolled in C3.



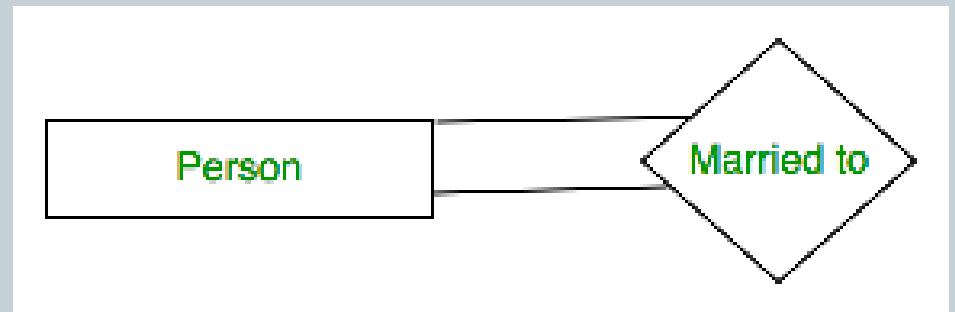
# Degree of a relationship set:



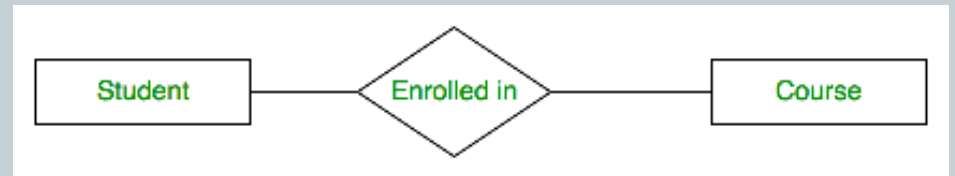
- The number of different entity sets **participating in a relationship** set is called as degree of a relationship set.
  - Unary Relationship
  - Binary Relationship
  - N-ary Relationship



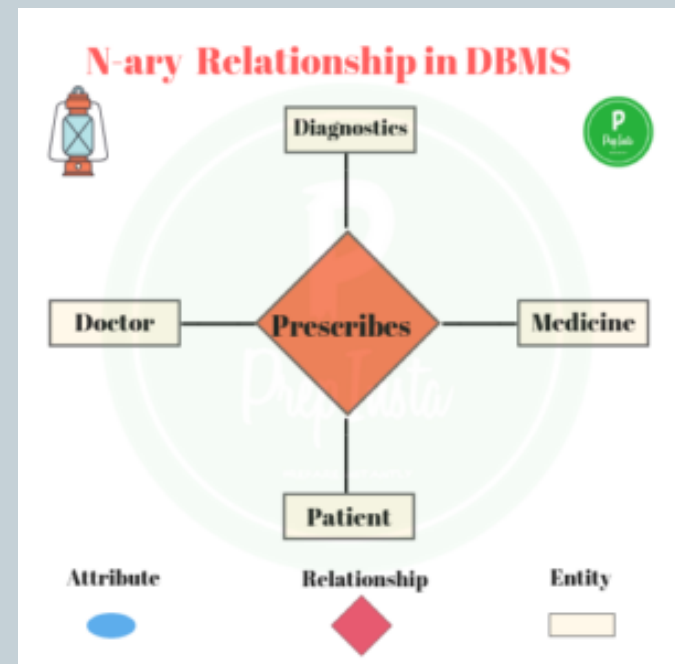
- **Unary Relationship –**  
When there is **only ONE entity set participating in a relation**, the relationship is called as unary relationship. For example, one person is married to only one person.



- **Binary Relationship –**  
When there are **TWO entities set participating in a relation**, the relationship is called as binary relationship.
- For example, Student is enrolled in Course.



- **n-ary Relationship –**  
When there are n entities set participating in a relation, the relationship is called as n-ary relationship.

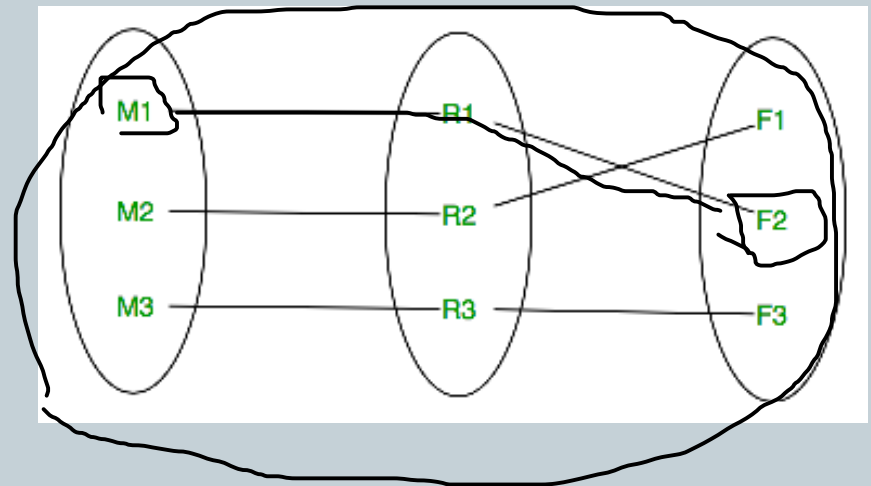
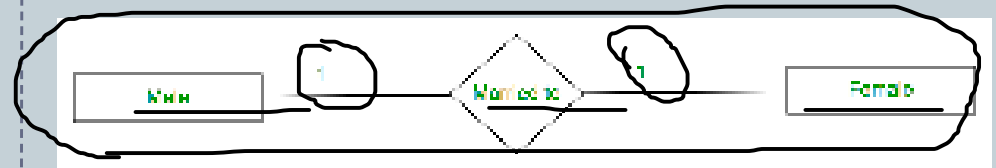


# Cardinality:

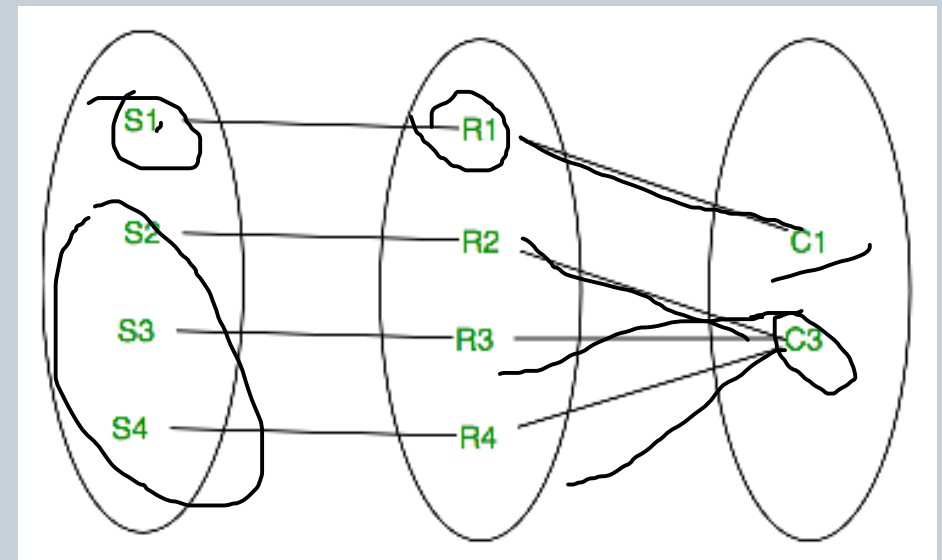


- The number of times an entity of an entity set participates in a relationship set is known as cardinality.
  - One to one
  - Many to one
  - Many to many

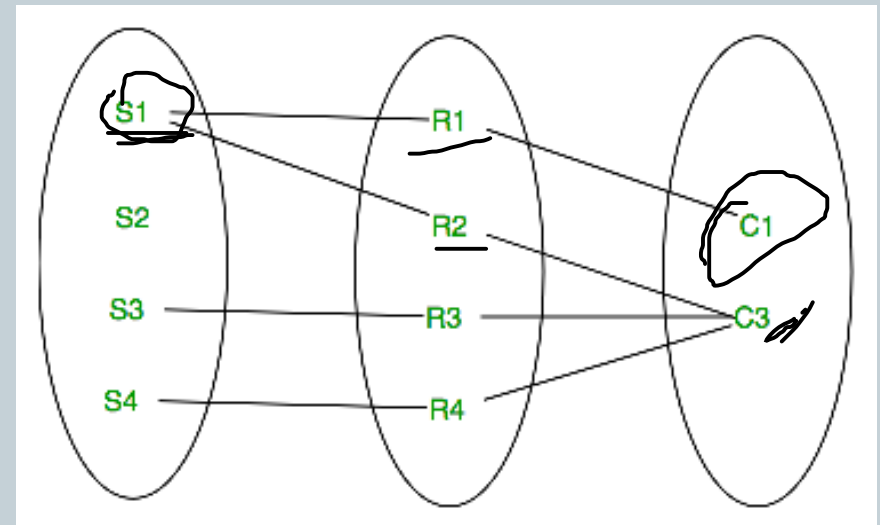
- **One to one** – When each entity in each entity set can take part **only once** in the relationship, the cardinality is one to one. Let us assume that a male can marry to one female and a female can marry to one male.
- So the relationship will be one to one.



- **Many to one** – When entities in one entity set **can take part only once in the relationship set** and entities in other entity set **can take part more than once in the relationship set**, cardinality is many to one.
- Let us assume that a student can take only one course but one course can be taken by many students.
- So the cardinality will be  $n$  to  $1$ .
- It means that for one course there can be  $n$  students but for one student, there will be only one course.



- **Many to many** – When entities in all entity sets can **take part more than once in the relationship** cardinality is many to many.
- Let us assume that a student can take more than one course and one course can be taken by many students.
- So the relationship will be many to many.



# Participation Constraint:



- Participation Constraint is applied on the entity participating in the relationship set.
- **Total Participation** – Each entity in the entity set **must participate** in the relationship. If each student must enroll in a course, the participation of student will be total. Total participation is shown by double line in ER diagram.
- **Partial Participation** – The entity in the entity set **may or may NOT participate** in the relationship. If some courses are not enrolled by any of the student, the participation of course will be partial. The diagram depicts the 'Enrolled in' relationship set with Student Entity set having total participation and Course Entity set having partial participation.



# Introduction to SQL



- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.
- Types of DB language

# Data Definition Language



- **DDL** stands for **Data Definition Language**. It is used to define database structure or pattern.
- It is used to create schema, tables, indexes, constraints, etc. in the database.
- Using the DDL statements, you can create the skeleton of the database.
- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

# Data Definition Language



- **Create:** It is used to create objects in the database.
- **Alter:** It is used to alter the structure of the database.
- **Drop:** It is used to delete objects from the database.
- **Truncate:** It is used to remove all records from a table.
- **Rename:** It is used to rename an object.

# Data Manipulation Language



- **DML** stands for **Data Manipulation Language**. It is used for accessing and manipulating data in a database. It handles user requests.
- **Select**: It is used to retrieve data from a database.
- **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 all records from a table.

# Data Control Language



- **DCL** stands for **Data Control Language**. It is used to retrieve the stored or saved data.
- The DCL execution is transactional. It also has rollback parameters.
- (But in Oracle database, the execution of data control language does not have the feature of rolling back.)
- **Grant**: It is used to give user access privileges to a database.
- **Revoke**: It is used to take back permissions from the user.

# Transaction Control Language



- TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.
- **Commit:** It is used to save the transaction on the database.
- **Rollback:** It is used to restore the database to original since the last Commit.

# *Oracle Built-In Data types*



- character datatypes

- CHAR
- NCHAR
- VARCHAR2 and VARCHAR
- NVARCHAR2
- CLOB
- NCLOB
- LONG

- NUMBER datatype

- DATE datatype

- binary datatypes

- BLOB
- BFILE
- RAW
- LONG RAW

Datatype	Description	Column Length and Default
CHAR ( <i>size</i> )	Fixed-length character data of length <i>size</i> bytes.	Fixed for every row in the table (with trailing blanks); maximum size is 2000 bytes per row, default size is 1 byte per row. Consider the character set (one-byte or multibyte) before setting <i>size</i> .
VARCHAR2 ( <i>size</i> )	Variable-length character data.	Variable for each row, up to 4000 bytes per row. Consider the character set (one-byte or multibyte) before setting <i>size</i> . A maximum <i>size</i> must be specified.
LONG	Variable-length character data.	Variable for each row in the table, up to $2^{31} - 1$ bytes, or 2 gigabytes, per row. Provided for backward compatibility.
NUMBER ( <i>p</i> , <i>s</i> )	Variable-length numeric data. Maximum precision <i>p</i> and/or scale <i>s</i> is 38.	Variable for each row. The maximum space required for a given column is 21 bytes per row.
DATE	Fixed-length date and time data, ranging from Jan. 1, 4712 B.C.E. to Dec. 31, 4712 C.E.	Fixed at 7 bytes for each row in the table. Default format is a string (such as DD-MON-YY) specified by NLS_DATE_FORMAT parameter.



# SQL Table



- SQL Table is a collection of data which is organized in terms of rows and columns.
- In DBMS, the table is known as relation and row as a tuple.
- Table is a simple form of data storage.
- A table is also considered as a convenient representation of relations.

# EXAMPLE



MP_ID	EMP_NAME	CITY	PHONE_NO
1	Kristen	Washington	7289201223
2	Anna	Franklin	9378282882
3	Jackson	Bristol	9264783838
4	Kellan	California	7254728346
5	Ashley	Hawaii	9638482678

# Operation on Table



- Create table
- Drop table
- Delete table
- Rename table

# Create table



- SQL create table is used to create a table in the database.
- To define the table, you should define the name of the table and also define its columns and column's data type

- **Syntax:**

```
create table "table_name"  
("column1" "data type",  
"column2" "data type",  
"column3" "data type",  
...  
"columnN" "data type");
```



- CREATE TABLE EMPLOYEE (
- EMP\_ID INT NOT NULL,
- EMP\_NAME VARCHAR (25) NOT NULL,
- PHONE\_NO INT NOT NULL,
- ADDRESS CHAR (30),
- PRIMARY KEY (ID)
- );

DESC EMPLOYEE;

# SQL INSERT Statement



- The SQL INSERT statement is used to insert a single or multiple data in a table. In SQL, You can insert the data in two ways:

**SYNTAX: INSERT INTO TABLE\_NAME VALUES (value1, value2, value 3, .... Value N);**

○ Ex: INSERT INTO EMPLOYEE VALUES (6, 'Marry', 'Canada', 600000, 48);

# SQL SELECT Statement



- In SQL, the SELECT statement is used to query or retrieve data from a table in the database. The returns data is stored in a table, and the result table is known as result-set.

SELECT column1, column2, ...

FROM table\_name;

SELECT \* FROM table\_name;

SELECT EMP\_ID FROM EMPLOYEE;

# SQL DELETE table



- In SQL, DELETE statement is used to delete rows from a table.
- We can use WHERE condition to delete a specific row from a table.
- If you want to delete all the records from the table, then you don't need to use the WHERE clause.

**DELETE FROM table\_name WHERE condition;**

**DELETE FROM EMPLOYEE**

**WHERE EMP\_ID = 3**





- <https://sqliteonline.com/>
- <https://paiza.io/projects/H4Nn1G5Fwms49YBztSMSXA?language=mysql>
- [https://www.tutorialspoint.com/execute\\_sql\\_online.php](https://www.tutorialspoint.com/execute_sql_online.php)