SQL

It is structured query language.it language helps to insert,delete and update the data base.and also it is powerful language because that uses the simple English words.

Data:it is a collection of facts,figures and values from different source

Data base:it is a location where we stored data in certain format.

Database:

Creation :create database name(name of the database)

Deletion: delete database name(name of the data base)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aggregate functions | Comparision  functions | String  functions | Math  functions | Date  functions | Window  functions |
| Avg | COALESCE | Asci |  |  |  |
| Count | DECODE | Char |  |  |  |
| Max | NULLIF. | Concat |  |  |  |
| Minimum |  | Lower,upper |  |  |  |
| sum |  | Length |  |  |  |
|  |  | Substring |  |  |  |
|  |  | Replace,  trim |  |  |  |

| **Mathematical**  **functions** | **Description** |
| --- | --- |
| [ABS](https://www.sqltutorial.org/sql-math-functions/sql-abs/) | Returns the absolute value |
| [ACOS](https://www.sqltutorial.org/sql-math-functions/sql-acos/) | Returns the arc cosine of an argument |
| [ASIN](https://www.sqltutorial.org/sql-math-functions/sql-asin/) | Returns the arc sine of an argument |
| [ATAN](https://www.sqltutorial.org/sql-math-functions/sql-atan/) | Returns the arc tangent of an argument |
| [CEIL](https://www.sqltutorial.org/sql-math-functions/sql-ceil/),[CEILING](https://www.sqltutorial.org/sql-math-functions/sql-ceil/) | Rounds up a float to the nearest integer value |
| [COS](https://www.sqltutorial.org/sql-math-functions/sql-cos/) | Returns the cosine of an argument |
| [COT](https://www.sqltutorial.org/sql-math-functions/sql-cot/) | Returns the cotangent of an argument |
| [EXP](https://www.sqltutorial.org/sql-math-functions/sql-exp/) | Returns the  e constant (2.71828…) that raises to a power of a specified number |
| LN | Returns the natural logarithm of the argument |
| LOG | Returns the natural logarithm of the first argument |
| LOG10 | Returns the base-10 logarithm of the argument |
| LOG2 | Returns the base-2 logarithm of the argument |
| [MOD](https://www.sqltutorial.org/sql-math-functions/sql-mod/) | Returns the remainder (modulo) of a number divided by another |
| PI | Returns the value of pi which is 3.14159265358979 |
| [POWER](https://www.sqltutorial.org/sql-math-functions/sql-power/) | Returns a number raised to a power of a specified number |
| [RAND](https://www.sqltutorial.org/sql-math-functions/sql-rand/) | Returns a random floating-point value |
| [ROUND](https://www.sqltutorial.org/sql-math-functions/sql-round/) | Rounds a number to a specific precision |
| SIGN | Returns the sign of an argument |
| [SIN](https://www.sqltutorial.org/sql-math-functions/sql-sin/) | Returns the sine of an argument |
| [SQRT](https://www.sqltutorial.org/sql-math-functions/sql-sqrt/) | Returns the square root of an argument |
| [TAN](https://www.sqltutorial.org/sql-math-functions/sql-tan/) | Returns the tangent of an argument |
| [TRUNCATE](https://www.sqltutorial.org/sql-math-functions/sql-truncate/) | Truncates to a specified number of decimal places |

**Date functions:**

| **Name** | **Description** |
| --- | --- |
| [CURRENT\_DATE](https://www.sqltutorial.org/sql-date-functions/sql-current_date/) | Return the current date |
| [CURRENT\_TIME](https://www.sqltutorial.org/sql-date-functions/sql-current_time/) | Return the current time |
| [CURRENT\_TIMESTAMP](https://www.sqltutorial.org/sql-date-functions/sql-current_timestamp/) | Return the current date and time |
| [Convert date to string](https://www.sqltutorial.org/sql-date-functions/sql-convert-date-to-string/) | Use the CAST() or TO\_CHAR() function to convert a date to a string |
| [Convert string to date](https://www.sqltutorial.org/sql-date-functions/sql-convert-string-to-date/) | Use the CAST() or TO\_DATE() function to convert a string to a date based on a specified format |
| [DATEADD](https://www.sqltutorial.org/sql-date-functions/sql-dateadd/) | Add an interval to a date |
| [DATEDIFF](https://www.sqltutorial.org/sql-date-functions/sql-datediff/) | Find the difference between two dates |
| [DATEPART](https://www.sqltutorial.org/sql-date-functions/sql-datepart/) | Extract a part of a date such as a year, month, and day from a given date |
| [Extract Year from Date](https://www.sqltutorial.org/sql-date-functions/extract-year-from-date-sql/) | Extract the year from a date |
| [Extract Month from Date](https://www.sqltutorial.org/sql-date-functions/extract-month-from-date-sql/) | Extract the month from a date |
| [Extract Day from Date](https://www.sqltutorial.org/sql-date-functions/how-to-extract-day-from-date-in-sql/) | Extract the day of the month from a date |

Value window functions

* [FIRST\_VALUE()](https://www.sqltutorial.org/sql-window-functions/sql-first_value/)
* [LAG()](https://www.sqltutorial.org/sql-window-functions/sql-lag/)
* [LAST\_VALUE()](https://www.sqltutorial.org/sql-window-functions/sql-last_value/)
* [LEAD()](https://www.sqltutorial.org/sql-window-functions/sql-lead/)

Ranking window functions

* [CUME\_DIST()](https://www.sqltutorial.org/sql-window-functions/sql-cume_dist/)
* [DENSE\_RANK()](https://www.sqltutorial.org/sql-window-functions/sql-dense_rank/)
* [NTILE()](https://www.sqltutorial.org/sql-window-functions/sql-ntile/)
* [PERCENT\_RANK()](https://www.sqltutorial.org/sql-window-functions/sql-percent_rank/)
* [RANK()](https://www.sqltutorial.org/sql-window-functions/sql-rank/)
* [ROW\_NUMBER()](https://www.sqltutorial.org/sql-window-functions/sql-row_number/)

**Select**

SELECT column\_name1, column\_name2, …  
FROM table\_name  
WHERE condition\_ expression;

**Insert**

INSERT INTO table\_name (column\_name\_1, column\_name\_2, column\_name\_3, ...)  
VALUES (value1, value2, value3, ...)

**Update**

UPDATE table\_name  
SET column\_name\_1 = value1, column\_name\_2 = value2, ...  
WHERE condition;

**Delete**

DELETE FROM table\_name WHERE condition;

**Create**

**CREATE** **Database** Database\_Name;

Create table table\_name;

**Drop;**

**DROP** **DATABASE** Database\_Name;

**Alter;**

**ALTER** **TABLE** name\_of\_table **ADD** column\_name column\_definition;

**Truncate;**

**TRUNCATE** **TABLE** Table\_Name;

**Rename;**

RENAME **TABLE** Old\_Table\_Name **TO** New\_Table\_Name;

exists

Union like

**And,or,not**

SELECT \* FROM Customers  
WHERE City=' ' OR City='München';

**Groupby**

select count(id),column\_name

from table\_name

group by column\_name;

operators

**orderby;**

SELECT \* FROM tablename  
ORDER BY columnname asc or desc;

**Alter;**

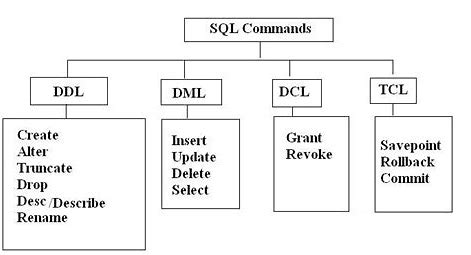
alter table\_name add column\_name varchar(10);

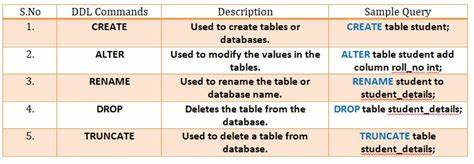
**Between**

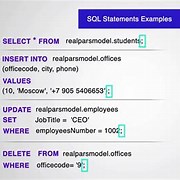
SELECT \* FROM Products  
WHERE Price BETWEEN 10 AND 20;

Joins;(inner,right,left,full)

SELECT table1.column1, table2.Column2  
FROM table1  
INNER JOIN table2 ON table1.matching\_column = table2.matching\_column;







GRANT :Used to grant(give access to) security privileges to specific database users.its mostly restrict users to insert,delete and update.

* Revoke: The REVOKE command in SQL is used to revoke or withdraw permissions that were previously granted to an account on a database object.

SQL FUNCTIONS:

ASCII :Returns the ASCII value for the specific character

CHAR\_LENGTH :Returns the length of a string (in characters)

CHARACTER\_LENGTH :Returns the length of a string (in characters)

CONCAT :Adds two or more expressions together

CONCAT\_WS :Adds two or more expressions together with a separator

FIELD :Returns the index position of a value in a list of values

FIELD(value, val1, val2, val3, ...)

Ex: SELECT FIELD("Q", "s", "q", "l") # 2

FIND\_IN\_SET :Returns the position of a string within a list of strings

FORMAT :Formats a number to a format like "#,###,###.##", rounded to a specified number of decimal places

INSERT :Inserts a string within a string at the specified position and for a certain

number of characters

Ex:SELECT INSERT("W3Schools.com", 11, 3, "no"); #(W3Schools.no)

INSTR :Returns the position of the first occurrence of a string in another string

Ex: SELECT INSTR("W3Schools.com", "3") AS MatchPosition; # 1

LCASE :Converts a string to lower-case

Ex: SELECT LCASE(“STRING”) # string

LEFT :Extracts a number of characters from a string (starting from left)

Ex: select left(“how are you”,3) # how

LENGTH :Returns the length of a string (in bytes)

LOCATE :Returns the position of the first occurrence of a substring in a string

LOWER :Converts a string to lower-case

LPAD :Left-pads a string with another string, to a certain length

LPAD(string, length, lpad\_string)

Ex: select lpad("prassu",10,"\*") as pad #\*\*\*\*prassu

LTRIM :Removes leading spaces from a string

MID :Extracts a substring from a string (starting at any position)

MID(string, start, length)

Ex: select mid("prassu",2,3) # ras

POSITION :Returns the position of the first occurrence of a substring in a string

REPEAT :Repeats a string as many times as specified

REPEAT(string, number)

Ex: select repeat(" mine",3) #mine mine mine

REPLACE :Replaces all occurrences of a substring within a string , with a new substring

REPLACE(string, from\_string, new\_string)

Ex: select replace(" PRASSU KUMAR","PRASSU","PRASANNA") as modify

# PRASANNA KUMAR

REVERSE :Reverses a string and returns the result

RIGHT :Extracts a number of characters from a string (starting from right)

RPAD :Right-pads a string with another string, to a certain length

RTRIM :Removes trailing spaces from a string

SPACE :Returns a string of the specified number of space characters

STRCMP :Compares two strings

STRCMP(*s1*, *s2*) (s1=s2,0:s1>s2,1:s1<s2,-1)

Ex: select strcmp("prasanna","prasa") # 1

SUBSTR : Extracts a substring from a string (starting at any position)

SUBSTRING : Extracts a substring from a string (starting at any position)

SUBSTRING\_INDEX : Returns a substring of a string before a specified number of delimiter occurs

TRIM : Removes leading and trailing spaces from a string

UCASE :Converts a string to upper-case

UPPER :Converts a string to upper-case

INTERVIEW QUESTIONS:

1.What is database?

It Is organized collection of data,stored and retrieved digitally from a remote or localhost.

2. what is DBMS and RDBMS?

Data base management system is a system software responsible for the creation,retrival,updation and management of database

3. What is RDBMS?

It stores data in the form of a collection of tables,and relations between the fields of the tables.

4. what is tables and fields?

Table is a collection of data stored in the form of rows and columns, again columns are classified into horizontal and vertical .the rows are called as records and the columns are called as fields

5. constraints in sql?

NOTNULL: restricted the null values into the column

CHECK: Verifies that all values in a field satisfies a condition

DEFAULT: automatically assigns a default value if no value is has been specified for the field

UNIQUE: Ensures unique values to be inserted in the fields

INDEX: it is a field providing faster retrival of records

PRIMARY KEY: it identifies each record in table

FOREIGN KEY: Ensures referential integrity for a record in another table

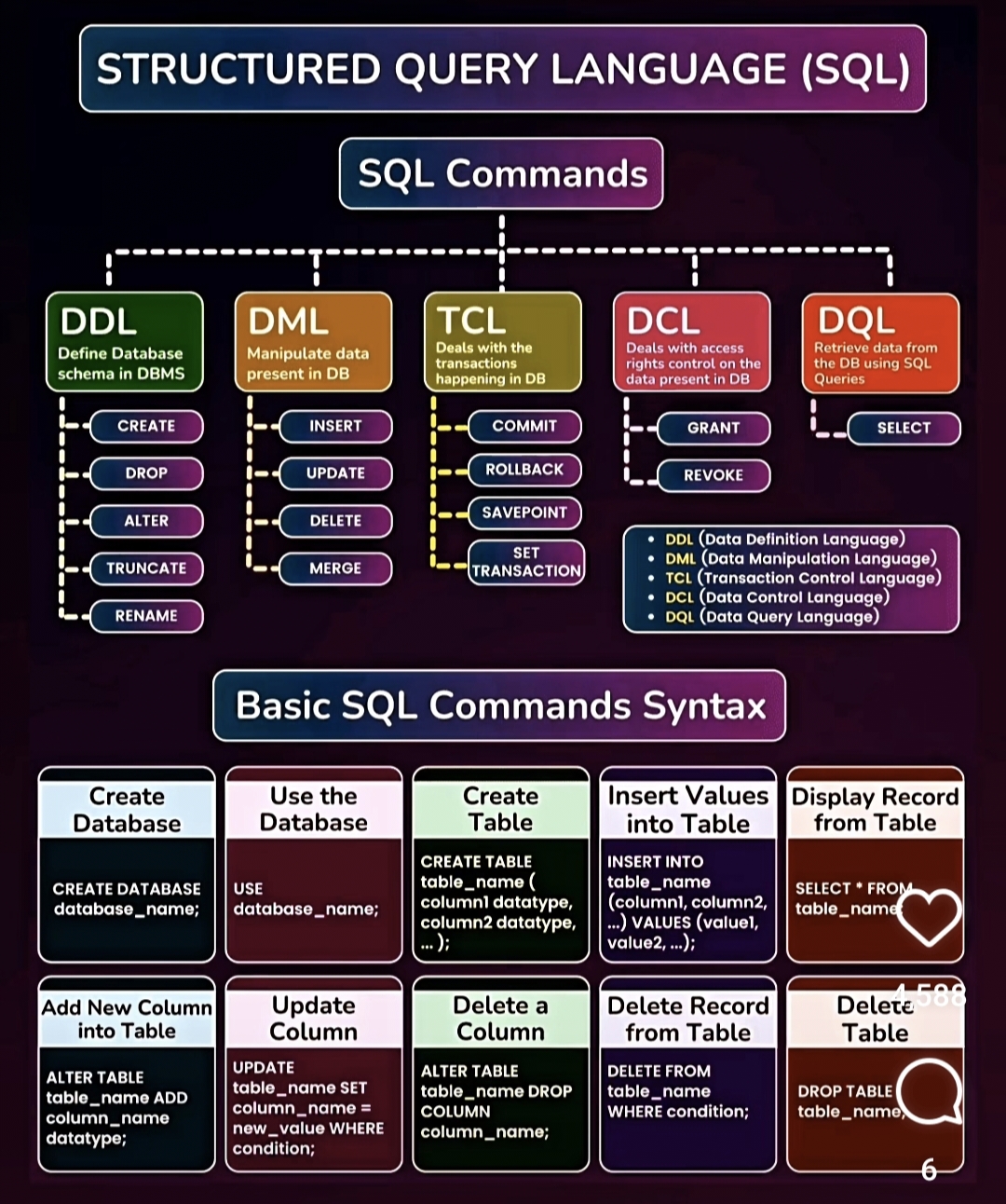
6. What is join and its types?

It is a clause used to combine records from 2 or more tables in a SQL database.

Inner join,left join, right join,full join

7. what is query?

It’s a request for data or information from database table or combination of table





ADVANCED SQL:

1.Subqueries: A subquery is a select query that is enclosed inside another query. The inner select query is usually used to determine the results of the outer select query.

1.Find the employees whose salary is greater than the average salary

select emp\_name, dept, salary from employees where salary > (select avg(salary) from employees);

**2.Stored procedures**: A stored procedure is an SQL code that you can save so that you can use the code over and over again. So, if you want to write a query repeatedly, save it as a stored procedure and then call it to execute it.

* Ex: Find the players who have scored over six goals.

delimiter &&

create procedure top\_players()

begin

select name, country, goals

from players where goals>6;

End &&

delimiter ;

call top\_players();

**3. Triggers**: A trigger is a special type of Stored Procedure that runs automatically when an event occurs in the database server.

There are mainly three types of triggers in SQL, they are:

1. Data Manipulation Trigger
2. Data Definition Trigger
3. Logon Triggers

Ex: For this, you will create a simple students' table that will have the attributes - student roll number, age, name, and student marks. Before you insert records in the table, you will have to check if the marks are less than 0. If true, the trigger will set the marks to 50.

Now, create a student table.

create table student

(st\_roll int, age int, name varchar(30), mark float);

delimiter //

create trigger marks\_verify\_st

before insert on student

for each row

if new.mark < 0 then set new.mark=50;

end if; //

insert into student

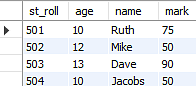
values(501,10,'Ruth', 75.0),

(502,12, 'Mike', -20.5),

(503, 13, 'Dave', 90.0),

(504, 10, "Jacobs", -12.5);

select \* from student;



1. **Views** : Views are actually virtual tables that do not store any data of their own but display data stored in other tables.

Create a simple view by selecting specific columns from the table.

Ex: create view cust\_details

as

select customerName, phone, city

from customers;

select \* from cust\_details;

1. Rename: SQL allows you to rename a view.

Syntax: rename lastname to newname;

1. Display: To view all the views

Syntax: show full tables where table\_type = 'VIEW';

1. Delete: To delete a view, you can use the drop command.

Syntax: drop view cust\_details;

**5.Windows Function**: Windows Functions in MySQL are useful applications in solving analytical problems and have become an essential part of advanced SQL.

EX: Find the total combined salary of the employees for each department from the employee table.

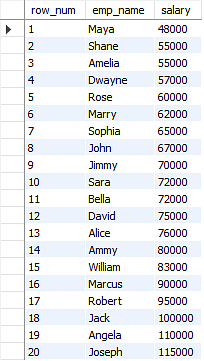
select emp\_name,age,dept,

sum(salary) over (partition by dept) as total\_salary

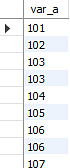
from employees;

6**.Row number:** it gives a sequential integer to every row within its partition.

Ex: select row\_number() over(order by salary) as row\_num,

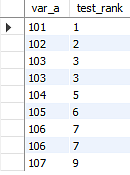
emp\_name, salary from employees order by salary;

**7.Rank function**: The Rank Function assigns a rank to a particular column. There are gaps in the sequence of rank values when two or more rows have the same rank.

create table demo1(var\_a int);on

select var\_a,

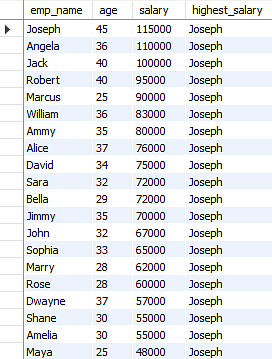
rank() over (order by var\_a) as test\_rank

from demo1;

**8.First Value**: The first value function provides the output value of the specified expression w.r.t. the first row in the window frame.

select emp\_name, age, salary, first\_value(emp\_name)

over (order by salary desc) as highest\_salary from employees;

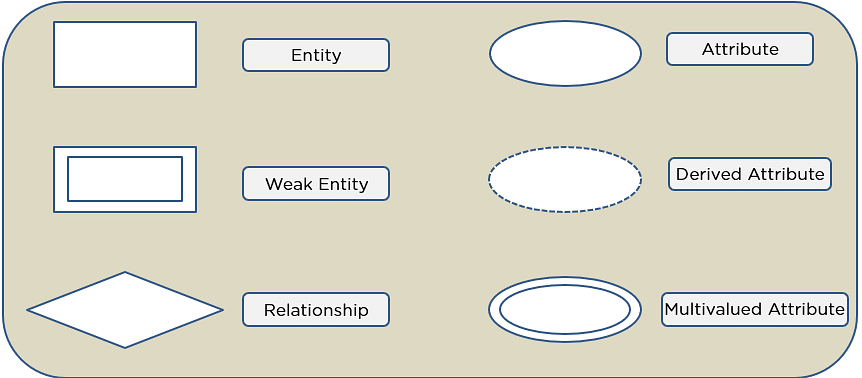


**ERDBMS**

ER diagram: its represents the relationship between the entities to be stored in data base. It acts as a framework created with specialized symbols for the purpose of defining the relationship between the database entities. ER diagram is created based on three principal components: entities, attributes, and relationships.

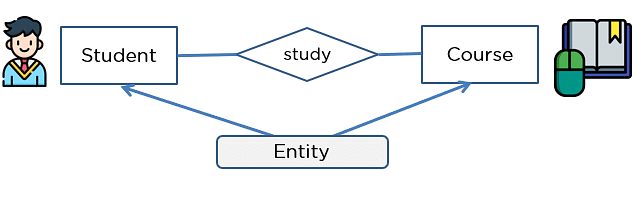
ER MODEL: An Entity-Relationship Model represents the structure of the [database](https://www.simplilearn.com/tutorials/sql-tutorial/create-mysql-database) with the help of a diagram. ER Modelling is a systematic process to design a database as it would require you to analyze all data requirements before implementing your database.

Symbols:



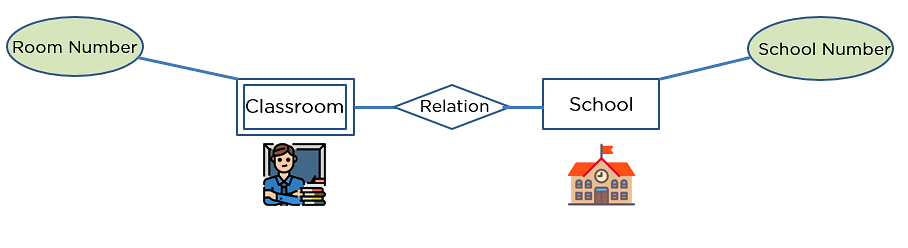
Components in ER Diagram:

1.Entities: it can be either a living or non-living organisms. In a study course,student and courses are the entities.

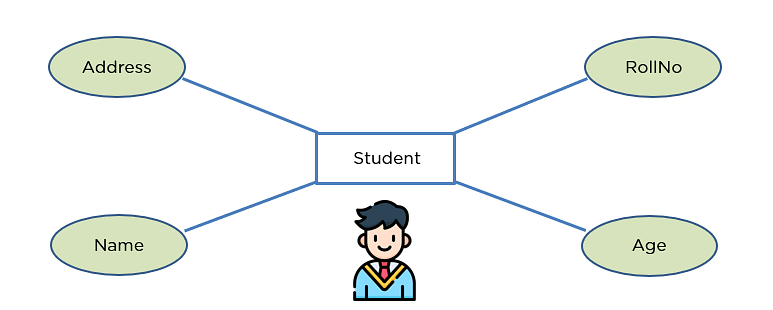


* Weak entity: a entity makes reliance over another entity is called weak entity(double rectangle)

In the example below, school is a strong entity because it has a primary key attribute - school number. Unlike school, the classroom is a weak entity because it does not have any primary key and the room number here acts only as a discriminator.



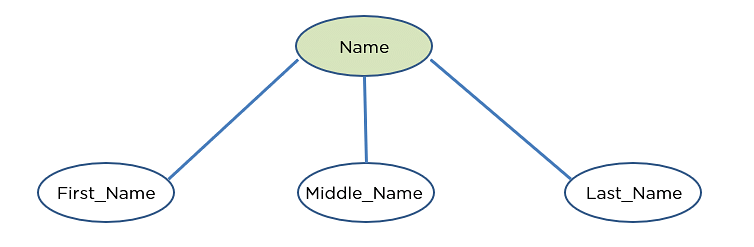
2.Attribute: it exhibits the properties of an entity. attributes are address, rollno, name and age

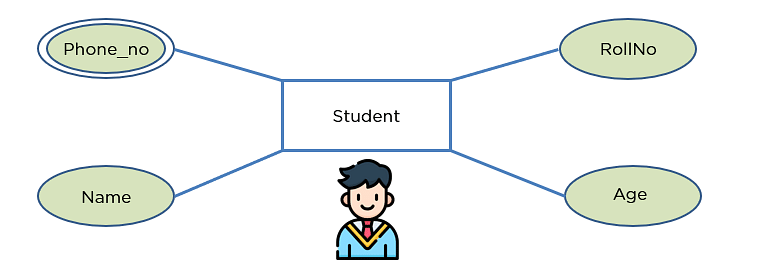


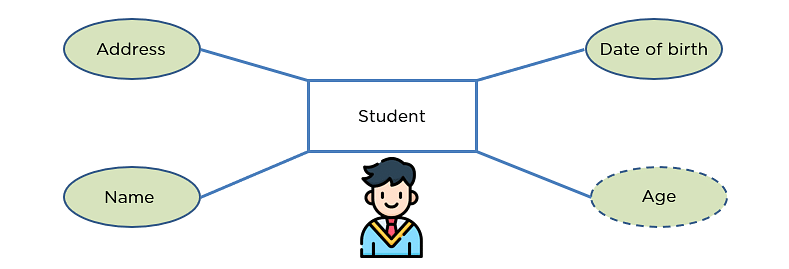
* Key attribute: it is uniquely an identifies the entity from entity set

Ex: For a student entity, the roll number can uniquely identify a student from a set of students.

* Composite Attribute: an attribute composed of several other attributes

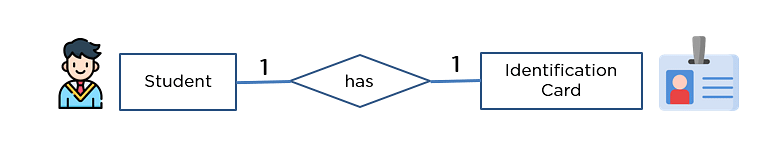


* Multivalued attributes: some attributes can posses over one rule, those attributes are called multivalued attributes(phone\_no).
* Derived attributes:an attribute can be derived from other attribute of the entity is known as derived attributes(age)

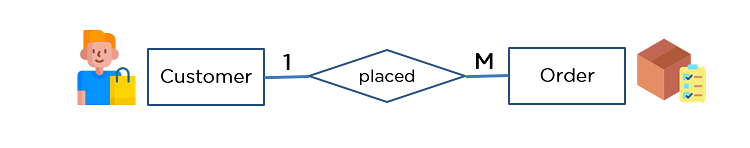


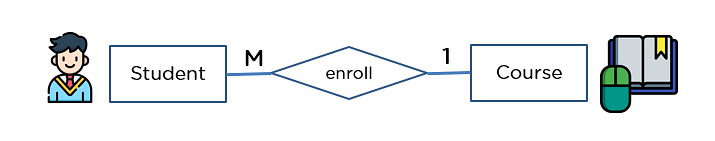
3.Relationship: its shows the relation between the two or more entities.

* **One-one:** when a single element of entity associated with a single element of another entity

 Ex: a student has only one identification card and an identification card is given to one person.

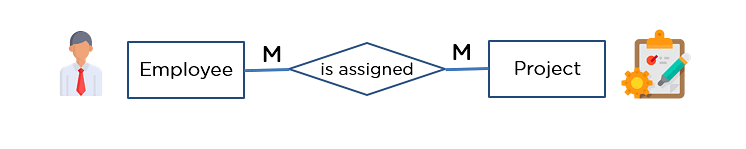
* **One-many**: when a single element associated with more than one element of another entity.

 Ex: a customer can place many orders, but an order cannot be placed by many customers.

* Many-one: When more than one element of an entity is related to a single element of another entity, then it is called a many-to-one relationship.

Ex: students have to opt for a single course, but a course can have many students.

* Many-many: When more than one element of an entity is associated with more than one element of another entity.

 Ex: you can assign an employee to many projects and a project can have many employees.

**SQL INJECTION**

Sql injection: it is code based vulnerability that allows attackers to read and access sensitivity. A successful SQL injection attack can badly affect websites or web applications using relational databases such as MySQL, Oracle, or SQL Server.it helps to beware of attackers

Types of sql injections:

