SQL NOTES

1.create database ustglobal;

2.show databases; // to see databases

3.use ustglobal; // to use database

4.show tables; // to show table

5.create table student(id int, name varchar(120),email varchar(150);

6.desc student; // to show structure of table

7.drop table student // to delete table

8.drop database ustglobal; // to remove database

9. create table human(id int, name varchar(120), email varchar(150));

10. alter table human add mobileno int; //add a colomn

11.alter table human add address varchar(150);

12.alter table human drop address ; // remove colomn

13.alter table human change name humanname varchar(120);//rename colomn

14.rename table human to student; // rename table

15truncate table student; // to delete data

16.select \* from student;//to display data in table

17.

18.

19. //we can not rename database in mysql

20.// truncate only delete the data remains structure

21.//drop used to delete the whole table

22.insert into student(id,studentName)values(01,'pryanka')

23.//write a query to diplay details whose name is priya //

select \* from student were studentName='priya';

24.update student set mobileno 9638 where id=10; // update the data in table

25.delete from student wherewhere id=12; //delete data from table

25.ALIAS: used to give a table, or colomn temporary name.

[SELECT colomn\_name as new-name;]

28.//write a query to give a temporary name to bloodgroup as bg

select bloodgroup as bg from employee\_info;

29.// write a query to display employename, sal, designation,annual salary

select id, name , salary, salary\*12 as annualsalary from employee\_info;

30. // write a query to ive alias name for all colomns in table

31.DISTINCT: select distinct designation from employee\_info;

32.//write a query to display unique deptno

select distinct deptd from employee\_info;

33. if more than arg is passed to ditinct r will remove the combinatin of colomn duplicate

33.// w a q todisplay unique valueof designation aswellas deptid

SELE

34. // W a t d unique deptid,age,designation from table

select distinct depy\_id, age, designation from employye\_info;

35.operator: operator is a resrved word used primarily in sql statemnts

. arithmatic operators

. comparision operatior

. logical operators

. special operator

\*> Arithmatic operators:

used to perform some arithmatic operations [+, - ,\* ,/ ,%]

\*> Comparision operators:

checks values of two operands [=,!=,<>,<,>,>=,<=,]

\*>Logical Operators: [ and , or, not ]

AND:

A B A^B

t t t

f t f

t f f

f f f

case(1): if 1st condition is false, and operator will return false without checking 2st condition.

case(2): if the 1st condition is true then result depend upon 2nd vondition.

case(3): if both cases are true the result condition is true.

OR:

A B A|B

t t t

f t t

t f t

f f f

case(1): if the 1st and 2nd condition condition is false then it will return false ony

case(2): if the 1st condition is false then the result depends upon 2nd condition. if the 2nd condition is true then the result is true.

syntax: select \* from table-name wherecolomn-name=value or colomn-name=value;

NOT:

A ~A

t t

f t

SPECIAL OPERATORS: [in ,between]

IN:

used to evaluate the multiple comlomn name.

select \* from table-name where colomn-name IN(list of values);

between:

used to find the range of values.

select \* from table-name where colomn-name between value1 and value2 ;

not in:

not in operator is a multivalued operator

select \* from table-name where colomn-name NOT IN(list of values);

not between:

which is used to display values which are excluding the range

select \* from table-name where colomn-name not between value1 and value2;

IS operator:

used to check the null value . if the record is null then it will return the null value.

36. // w a q to display id name salary whoes salary is incremented by 5000

select id, name,salary, salary+300 as inc\_sal from employee\_info

37 // w a q t disply annual salary

select id,name, salary\*12 as annual\_sal from a

employee\_info;

38. // w a q to display daily salary

select salary /30 as daily\_salary from employee\_info;

39. // w a q to display all the records xcluding dept\_no 10

select \* from employee\_info where id!=10;

40. // w a q to display whose sala>5000

select \* from employee\_info where salary>5000;

50. // w a q to display whose sala<5000

select \* from employee\_info where salary<5000;

51. //w a q to dis whose salary>= 1500

select \* from employee\_info where salary<=1500;

52.// w a q to display all the records who is working in deptno 20 only if salary is greater than 1000

select \* from employee\_info where sal>1100 and dept\_id=20;

53.// w a q to display all rec who is working as saleman and dob is 1995

54.//w a q to dis all the records whose salary is 100000as well as designation is analyst

55.// w a query o display id, name, emailid whoes age is 22 and whose salary is 1120

56.// w a q to dis all the records whose id not = 10,20,30

select \* from employee\_info where not id=10 and not id=20;

57.// w a query to dispay all the records of employees excluding sales man who are all working in deptNo=40

select \* from employee\_info where not(designation='salesman' and dept\_id=30);

58.//w a q to dis play all recors who is woking as a salesman for dept\_id=20 or 30

select \* from employee\_info where designation='salesman' and (dept\_id=10 or dept\_id=20);

59.// wa q to dis all emp who is working in deptno=20 only if their sal>10000 & <25000 including 10000 and 25000

select \* from employee\_info where id=20 and (salary>=10000 and salary<=25000)

60. // w a q to dis all the records who is working as a software developer and dob in 1994

61.// w a q to display employee who is working in dept 10 and 20 and 30

select \* from employee\_infi where dept\_id in(10,20,30);

62. // w a q whose designaiton is tester hr analyst

select \* from employee\_infi where designation in ('analyst','hr','tester');

63. // w a q to display who are working as a clerk in dep 20 30 40

select \* from employee\_infi where designation='clerk' and dept\_id in(10,20,30);

65.// w a q to display who are working as a clerk in dep2030 40

select \* from employee\_infi where designation='clerk' and dept\_id not in(10,20,30);

66.// w a query who are working in a dept 20 40 60 excluding salesman or analyst

select \* from employee\_infi where designation not in('clerk','salesman') and dept\_id not in(10,20,30);

67.//w a q to all the details of emp whose sal between a range of 12000 and 25000

select \* from employee\_infi where salary between 1100 and 25000;

68.//write a query to display all the records of employee who hired 2017

select \* from employee\_info where doj between '2017-01-01' and '2017-12-31';

69.//write a query to select all details of employees excluding who hired in 2019

select \* from employee\_info where doj not between '2019-01-01' and '2019-12-31';

70.//write a query to display empname salary and deptno of all emp who are working in dept 10 or 20 excluding employee whose salaray greater or equal 15000 and less than or equal to 50000

select \* from employee\_info where dept\_id in(10,20) and salary not between(15000,50000);

71.// weite a query to display annual salary if his designation is null

select \* from employee\_info where designation is null;

72.// weite a query to display annual salary if his designation is not null

select \* from employee\_info where designation is not null;

LIKE:

Like operator is used to perform pattern matching

Syntax : select \* from table\_name like ' ' ;

characteristics are of two types : 1.ordinary character type

2. special character type(wild card)

||

||

\ /

" % , \_ "

" \_ " : if want to mtch exact one character then wue use underscore

" % " : if want to match 0 or 'n' characters

73. // write a query to display a employee name whose name starts with 's'

select \* from employee\_info where name like 's%';

74.// write a query to display whose name contains only 5 character

selct \* from employee\_info where name like'\_\_\_\_\_';

75.// write a query to display name, salary,dob whose name starts with 's' and end with 's'

selct name,salary,dob from employee\_info where name like' s%s';

76.// write a query to display all infor of employee whose designation contains atleat two 'ee'.

selct \* from employee\_info where name like'%s%s%';

76//.write a quey to all employee display length is 5 and 1st char is"s'. 2nd last is 'L'

selct \* from employee\_info where name like 's\_\_\_\_l\_ ';

77.// write a query if name of employee is starts with 'r' or 'z'

selct \* from employee\_info where name like 'r%' or like 'z%' ;

78.//write a quey to all employee display whoes name 3rd char is 'c' and 6th char 'i'

select \* from employee\_info where name like'\_\_c\_\_i%';

NOT LIKE: select \* from table\_name where colomn not like ' ';

which select all the string which does not match with the given string

79. w a q to all excluding whose name starts with 'a'

selct \* from employee\_info where name not like 's%' ;

FUNCTIONS:

1. FUNCTION IS A BLOCK OF CODE WHICH SHOULD PERFORM SOME

sepcified TASK.

for doing operations on data sql has many built in functions

2.FUNCTION ATTRIBUTES:

It has input arguments . In SQL has prdefined functions

1.scalar functions /single-row-func --------- upper,lower,concat

2.aggrigate functions/multi-row-func -------avg,count,max,min,sum

multi-row-func: for multiple input only one output will come

| i/p---------------------| | i/p----------------------| | ------------------- o/p i/p----------------------| |

i/p----------------------| |

single-row-func:

for each and every input output there will be correspnding output

i/p-----------------o/p

i/p------------------o/p

i/p-------------------o/p

i/p------------------o/p

80.// w q avg salary of employee

select avg(salary) as avgsal from employee\_info;

81.// wa q to avg sal, min sal,max sal

select min(sal) as minisal from employee\_info;

select max(sal) as mxsal from employee\_info;

select avg(sal) as mxsal from employee\_info;

--------------------------------------------------------------------------------------------------------------------------------

ASSAIGNMENT-1:

1.//W A Q TO DISPLAY employee working dept 20

select \* from employee\_info where dept\_id=20;

2.// W A Q TO DISPLAY employee earning more than 25000

select \* from employee where salary>2500;

3.//W A Q TO DISPLAY employee whose job is software developer

select \* from employee where designation='developer';

4.// W A Q TO DISPLAY employee whose dept\_id ia 10,20,30

select \* from employee\_info where dept\_id in(10,20,30);

5.// W A Q TO DISPLAY employee whose name starts with 's'

select \* from employee where name like 's%';

6.// W A Q TO DISPLAY employee whose name 2ndletter with 'L'

select \* from employee where name like '\_l%';

7.// W A Q TO DISPLAY employee whose name have atleat 2 'a' characters

select \* from employee where name like '%a%a%';

8.// W A Q TO DISPLAY employee whose name 2nd last letter with 'a'

select \* from employee where name like '%e\_';

9.// W A Q TO DISPLAY employee whose has exactly 5 char

select \* from employee where name like '\_\_\_\_\_';

10.// W A Q TO DISPLAY employee whose salary between 20000 and 30000;

select \* from employee where salary between 20000 and 30000;

--------------------------------------------------------------------------------------------------------------------------------

ASSAIGNMENT-2:

1.// W A Q TO DISPLAY WHOSE dept\_id=10 and software developer

select \* from employee where dept\_id=10 and designation='software developer';

2.// W A Q TO DISPLAY WHOSE dept\_id=10 and software developer and salary>5000;

select \* from employee where dept\_id=10 and designation='software developer' and salary>50000;

3.// W A Q TO DISPLAY Wall details wxcept thode whose dept\_id=10and 20 and 30

select \* from employee\_ info where dept\_id not in(10,20,30);

4.//.// W A Q TO DISPLAY Wall details wxcept thode whose job is clerk and develpoer and dept 10,20,30 with asalary oof range of 10000 and 30000

select \* from employee\_info where salary not between 10000 and 20000 and dept\_id not in(10,20,30) and designation not in('developer','clerk);

5.// W A Q TO DISPLAY Wall details wxcept whose dept 10,20,30 with salary not a range of 10000 and 20000 except salesman

select \* from employee\_info where salary not between 10000 and 20000 and dept\_id in(10,20,30) designation!='salesman';

6.// w a q to arrange employees by their salary in desending order

select \* from employee\_info order by salary;

7.// write a query to display max, min,total salry

select

select min(sal) as minisal from employee\_info;

select max(sal) as mxsal from employee\_info;

select avg(sal) as mxsal from employee\_info;

8.//write a query to display list of no.of mployees working

in dept 20

select \* from employee\_info where dept\_id=20 ;

9.//write a query to display list of employees highest sal and lowest sal earned by salesman

select max(salary),min(salary) where designation='salesman' from employe\_info;

10.write a query to display total salarry of all department

select sum(salary) from employee\_info group by dept\_id;

--------------------------------------------------------------------------------------------------------------------------------

82.//wrie a q to display doj after 2016

select \* from emoloyee\_info where doj>'30-12-2016';

83.//write a q avg sal and higest sal of dept 20,30,40and 60

select max(salary),avg(salary) from employee\_info where dept\_id in(10,20,30);

84.// write a query to display all detals of employee date of 1st and last hired emploee

select min(doj),max(dob) from employee\_info;

SCALAR FUNCTIONS:

UPPER: convrts the value of a field to to uppercase

LOWER: convrts the value of a field to to lowercase

LENGTH: finds the value of field

INITCAP(): 1st char in capital rest in small not support for

myql supports in oracle

concat():Used to concatinate two strings functions.

reverse(): Used to revers the reverse a string.

85.// write a query to merge name and salary

select conacte(name,salary) from employee\_info;

86.//write a query to reverse of names

SUBSTR():

it is a func used to extract a string from given string it takes 3 arguments

SYNTAX:

SELECT substr(string,position,length) from tablename;

arg-1: string is an arg which we passed

arg-2: it is a number which specifies the position of a string.

arg-3: it just specifies the length of new string

87.// w a q to display employee name only 3 characters

select substr(name,1,3) from employee\_info;

88.// w a quetry to dispaly whose name begin with consonent

select name from employee\_info where substr(name,1,1) not in('a','e','i','o','u');

INSTR():

USED TO CHECK weather substr is present in given string or not

if a substring is not present in string then this function wil return zero. it is used to check the integer.

SYNTAX:

SELECT INSTR(STRING, SUBSTRING) FROM TABLE-NAME;

arg-1: STRING: IT IS A STRING USED TO FIND SUBSTR

arg-2: SUBSTRING: IT IS A SUBSTRING

89.// write a quey to search the position of a in the name colomn

select instr(name,'a') from employee\_info;

REPLACE:

used to replace a string in a given string

SYNTAX:

SELECT REPLACE(string,SUBSTR1,SUBSTR2) FROM TABLE-NAME;

90.// WRITE A QUERY TO REPLACE NAME 'A' WITH 'Z'

select replace(name,'a','z') from employee\_info;

91.// write a query to display no. of employees who is working in dept 30 and salary is 15000

select count(\*) from employee\_info where salary=15000 and deplt\_id=30;

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GROUP BY CLAUSE:

GROUP BY STATEENT GROUP ROWS THAT HAVE THE SAME VALUES INTO SUMMARY ROWS .

GROUP BY STATEMENT OFTEN USD WITH THE AGGRIGATE FUNCTIONS LIKE(COUNT MAX MIN AVG) TO GROUP THE RESULT-SET BY ONE OR MORE COLUMNS.

SYNTAX:

SELECT COLOMN-NAME

FROM TABLE-NAME

WHERE CONDITON

GROUP BY COLOMN-NAME;

NOTE: group by clause executes row by row. after the execution of group by the records are grouped.

\*Therefore all the clause will execute after the execution of

group by clause .

\* group by clause will excute group by.

\* GROUP BY CLAUSE IS USED WITH AGGRIGATE FUNCTIONS

92.// WRITE A Q TO DISPLAY HIGHEST SALARY FROM EACH DEPARTMENT

select max(salary) from employee\_info group by dept\_id;

93.// WRITE A QUERY TO DISPLAY NO.OF SOFTWARE WORKING IN EACH DEPARTMENT

select \* from employee\_info where designation='software' group by det\_id;

94.// WRITE A QUERY TO DISPLAY MIN SALARY IN EACH DESIGNATION

HAVING CLAUS:

HAVING CLAUSE IS OFFEN USED WITH THE GROUP By CLAUSE TO FILTER GROUPS BASED ON A SPECIFIED CONDITION.

IN HAVING CLAUSE WE CAN'T USE ONLY AN COLOMN IE,USED IN GROUP BY CLAUSE.

SYNTAX:

SELECT COLOMN-NAME

FROM TABLE-NAME

WHERE CONDITION

GROUP-BY COLOMN-NAME

HAVING CONDITION

ORDER OF EXECUTION :

HAVING CLAUSE EXCUTES AFTER GROUP BY CLAUSE THEREFORE WE WILL CHECK THE CONDITION AFTER GROUPING

SINCE IT EXECUTES AFTER EVERY GROUP IT WILL CHECK CONDITION

95.//WRITE A Q TO DISPLAY DEPT NUMBER WHICH HAS ATLEAST FOUR WORKING EMPLOYEES.

select dept\_id from employee\_info group by dept\_id having count(dept\_id)>4;

96.// W A Q TO DISPLAY NO. OF DEPTNO. WHICH HAS TOTAL SALARY > 20000

select dept\_id,sum(salary) from employee\_info having sum(salary)>20000;

97.// W A TO ALL DEPT WHOES AVG SALARY >20000 EXCLUDING THE EMPLOYEE WHOSE NAME BEGINS WITH S

select name,dept\_id,avg(salary)>2000 from employee\_info where name not like('s%') group by dept\_id having avg(salary)>20000;

ORDER BY CLAUSE :

THE ORDER BY CLAUSE KEYWORDIS USED TO SORT THE RESULT SET IN ASCENDING ORDER OR DESCENDING ORDER

THE ORDER BY CLAUSE SORTS THERECORDS IN ASCENDING ORDER ORDER BY DEFAULT. TO SORT SORT IN DESCENDING ORDER WE USE THE SESC KEYWORD

SYNTAX:

SELECT COLOMN1,COLOMN2,....FROM TABLE-NAME

ORDER BY COLOMN1,COLOMN2,....ASC|DESC;

OREDR OF EXECUTION: from>WHERE>groupby>having>select>ORDER BY

note: Order by class excecute after the execution of

order by class by default should be the last claue to be written in the syntax always

If the group by clause used order by clause we can use aggrigate function or multi row function.

only the colomn name which is use the group by clause.

98.// WRITE A QUERY TO DISPLAY ALL THE RECORS SALARY SHOULD BE ASCENDING ORDER

select \* from employee\_info order by salary;

99.// write a query TO DISPLAY NAME, DESIGNATION,DOJ OF ALL EMPLOYEE WHO ARE WORKING IN EAITHER ANY ONE OF THE DEPARTMENTS 10,20,50,60 SORT THE RECRDS IN DESCENDING OREDER BY THEIR SALARY..

SELECT NAME, DESIGNATION, DOJ FROM EMPLOYEE\_INFO WHERE DEPT\_ID IN(10,20,30,50,60) ORDER BY DEPT\_ID DESC;

100.// WRITE A QUERY TO DISPLAY DEPT\_ID ALONG WITH NO. OF EMPLOYEES IN IT.

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

101.// WRITE A Q TO DSPLAY WHOSE DESIGNATION IS TESTER AND SOFTWSRE DEVELOPER

select \* from employee\_info where designation ('tester','developer');

SUB-QUERY:

A QUERY WITHIN ANOTHER QUERY IS CALLED AS SUBQUERY. A SUBQUERY ALSO CALLED AS INNER-QUERY.

SUB QUERY IS ANOTHER WAY TO FETCH THE DATA FROM MULTIPLE TABLES

WHATEVER THE INNER QUERY SHOULD BE ENCLOSED WITHIN PARANTHESIS

SUB-QUERIES ARE CLASSIFIED INTO TWO CATEGERIES

1.SINGLE-ROW SUBQUERY

2.MULTI-ROW SUBQUERY

NOTE:

INNER-QUERY IS THE 1ST EXECUTABLE AND IT TAKES THE INPUT FROM THE OUTER -QUERY.

OUTPUT OF INNER QUERY TAKES THE INPUT FROM INNER-QUERY.THEN THE FINAL RESULT FROM THE OUTER QUERY.

WHEN TO USE SUB-QUERY MAIN CLAUSE??

\*1> THE GIVEN CONDITION CONTAINS UNKNOWN VALUE THEN WE CAN USE SUB-QUERY TO OBTAIN THE UNKNOWN VALUE.

\*2> IF THE COLOMN TO BE SELECTED AND THE CONDITION TO BE FILTERED A GIVEN TWO DIFFERENT TABLES WE CAN USE A SUB-QUERY.

1.SINGLE-ROW SUBQUERY:

IF THE SINGLE VALUE OPERATORS SUCH AS EQUAL TO,LESS THAN,GREATER THAN,LESS THAN,NOT EQUAL TO(=,!=,<,>)ARE USED

WE CALL IT AS 'SINGLE-ROW-SUBQUERY'.

SINGLE ROW SUB QUERY MUST AND SHOULD RETURN A SINGLE RECORD

SYNTAX:

SELECT COLOMN1,COLOMN2,.....FROM TABLE WHERE COLOMN1 OPERATOR

(SELECT COLOMN FROM TABLE WHERE EXPR1=VALUE);

2.MULTI-ROW SEUBQUERY:

WE CALL IT AS "MULTI-ROW SUBQUERY"

MULTI-ROW SUBQUERY CAN RETURN ONE OR MORE NUMBER OF ROWS (INNER QUERY CAN RETURN ANY NO. OF ROWS)

SYNTAX:

SELECT COLOMN1,COLOMN2,.....FROM TABLE WHERE COLOMN1 OPERATOR

(SELECT COLOMN FROM TABLE WHERE EXPR1=VALUE);

103.//WRITE A Q TO DISPLAY WHO IS WORKING FOR RESEARCH DEPARTMENT

select name from employee\_info where dept\_id =(select dept\_id from department where deptname='research');

104.// WRITE A QUERY TO DISPLAY EMPLOYEE WHO IS WORKING IN LOCATION WHICH HAS ATLEAST ONE 'A' CHARACTER IN ITS NAME

select location from department where dept\_id in(select dept\_id from employee\_info where name like('%a%'));

105.// WRITE A QUERY TO OBTAINE 2ND MAX SALARY FROM EMPLOYEE\_INFO TABLE

select max(salary) from employee\_info where salary <(select max(salary) from employee\_info);

106.//WRITE A QUERY TO DISPLAY THE 3RD MAX SALARY

select max(salary) from employee\_info where salary <(select max(salary) from employee\_info where salary <(select max(salary) from employee\_info)) ;

JOINS:

JOINS IS USED TO FETCH THE DATA FROM TWO ARE MORE TABLES

IT IS USED FOR COMBINING COLOMN FROM TWO OR MOE TABLES BY USING VALUES COMMON COLOMN TO HOST THE TABLES.

JOIN IS A KEYWORD USED IN SQL QUERIES FOR JOINING TWO OR MORE TABLES

MINIMUM REQUIRED CONDITION FOR JOINING TABLES IS AT LEAST NO. OF TABLES SHOULD BE TWO.

TYPES OF JOINS:

\*1> CARTESIAN JOIN/ CROSS JOIN:

RETURNS ONLY THE CARTESIAN PRODECT OF ROWS FROM THE TABLE IN JOIN

RECORD FROM ONE TABLE IS MERGED WITH EACH RECORD WITH THE ANOTHER TABLE OR JOIN TWO TABLES WILL MERGE WITH EACH AND EVERY RECORD WITH THE RIGHT TABLE

IN ANOTHER WORDS IT GIVES US COMBINATIONS OF EACH ROW OF FIRST TABLE WILL ALL RECORDS IN SECOND TABLE

SYNTAX:

SELECT COLOUMN-NAME-LIST

FROM TABLE-NAME1 CROSS JOIN TABLE-NAME1;

107.// WRITE A QUERY TO DISPLAY ALL THE RECORDS OF CRATESIAN JOIN

**Inner join:**

* Joining two tables with the help of joint colomn is known as inner-join.

select colomn\_name table\_name inner join table2\_name on condition

* common data present in both the table is displayed in the i ner join.

**outer-join :**

There are 3 types of joins.

1.Left-outer join.

2.Right-outer join.

3.Full-outer join.

**1.Left-outer join:**

1.in the ges the inner join + matched record of left table

2.the record do not have pair(matched) in the opposite table with respect to joins it will returns the total table.

**Syntax:**

Select \* from table1 left outer join table2 on condition.(ANSI)

Select \* from table1\_name table2\_name where table1 colomn\_name = tabe2 colomn\_name(\*);

**Right\_outer join:**

* It gives the inner join +un matched record of right table

Select \* from table1 left outer join table2 on condition. (ANSI)

**Full outer join :**

Full outer join gives inner join as well as all the records from both the tables who do not have pairs .

**Key attributes:**

* A key attribute is used to find an attribute using which uniquely identifies a record in a table.
* All the attributes except key attributes are known as non-key attributes.

**Primary key attributes :**

* A key attributes which is chosen to be main attribute to determine the record uniquely. In the table.

**Non-primary key attributes:**

* All the attributes except primary key attributes are known as primary key attributes.

**Composite key:**

* The combination of two or more attributes that determines the attribute uniquely is called composite key

**Foreign-key:**

* The foreign key is used to relate the two tables . the foreign key in which it is present is called as child table. And the pk is present is called as the parent key.
* **Functional dependency:**
* In a functional dependency a relation that exists such that an attributes determines another attribute uniquely is called as the functional dependency.

**Types of functional dependency**

**Total dependency:**

* If all the attributes in a relation is determined by a key attribute is known as the total functional dependencies.
* Example: R1-> {a, b, c, d} where R1 is a relation a-> b ,c, d {b,c,d are fully dependent on a }

**Partial dependency:**

A relation is said to be transitive dependency if

* It consists of composite key attributes they exists the dependency such that the attribute can be determined by the another attribute which is a part of the composite key
* Example : R1 -> {a,b,c,d} ,(a,b)-> c,d where c and d are partially dependent.
* **Transitive dependency:**
* A relation is said to be transitive functional dependency if there exists a relation such that an attribute is determined by non key attributes which in turns determines by key attributes.
* Example R1->{a,b,c,d}

a->b;

b->d;

a->d

Where b is dependent on a, d is dependent on b. In that way d is dependent on a.

**Normalization:**

* Redundancy
* Reputation
* Due to these draw backs we are using normalization.

Types of anomaly

Insert.

Update.

Delete.

The process of decomposing the table into smaller table by removing the redundancy and anomalies by identifying dependencies.

The process of reducing the table into a normal form is called as normalization.

* A state of a table without redundancy is called as normal form.
* **TYPES OF NORAML FORMS:**
* 1NF
* 2NF
* 3NF
* 4NF

**1NF:**

* Table is said to be in 1st normal form. If it satisfies the following condition table should have the following rows.
* Every cell in the row should be single(atomic) value.

**2NF:**

THE table should be 1nf

The table should not have a partial dependency(composite key)

**Note:**  If the table consists of partial functional dependency the attribute which are responsible are removed from the table.

**3NF:**

The table in the 3nf should satisfy the following conditions

* It should follow 2nf
* The table should not have transitive functional dependency**.**

**Note:**  1NF, 2NF and 3NF these are based on functional dependency of relational schema.

**ER-Relationship**:

* ER-diagram describes their structure of database with the help of diagram which is known as the ER-diagram.
* It is the blue print of database that can be later use to implement the database.
* **Component of ER-diagrams:**

Entity: Represents by the rectangle.

Attribute: Represented by eclipse.

Relationship: Represented by diamond.

**Views:**

* Views are logical or virtual tables that can be created on the existing table.
* Views does not occupy memory
* Syntax:

1. Create view view\_name

As Select \* from table;

1. Select \* from view\_name

**Note:** Any table operation for form or view base table will got reflected.

**TCL:**

Transaction control language.

**Roll back**:

Rollback;

**Commit:**

Set autocommit =0; //1st we have to give autocommit=0

Then only we can perform the o

operations of rollback.

Savepoint works with rollback

Rollback to a;

**Stored procedure:**

* **A stored procesure is a collection of group of sql statements.**
* Stored in the database ‘s data called from either a rectangle program another stored procesure

**Syntax:**

**DELIMETERS $$**

**Create procedure <procedure Name>**

**As**

**Begin**

**<SQL statement>;**

**End $$**

**DELIMETER;**