SQL - Structured Query language which is used to interact with database is called SQL

Users of SQL

There are 3 users of SQL

1) Database developer
2) Database tester
3) DBA (Database Administrator)

=> Database developer: Developer will use SQL to create database & create table in database

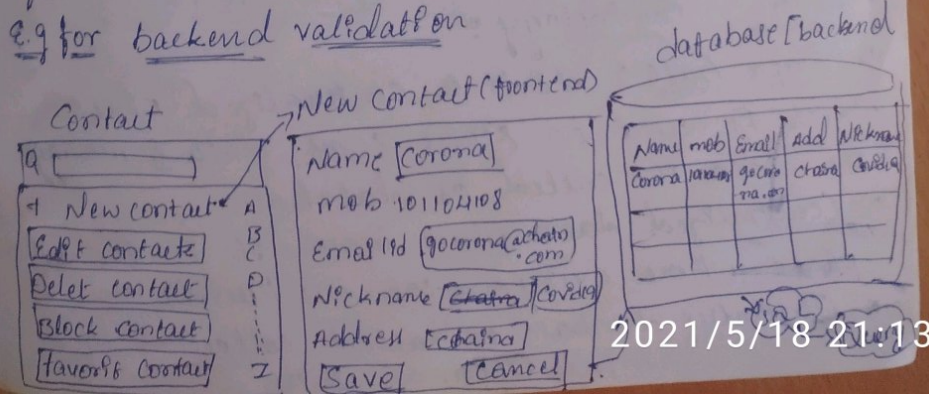
=> Database tester: Test engineer will use SQL to perform "backend validation"

=> Back end validation: whatever the data which is entered in frontend has successfully stored in database [Backend] or not. is called back-end validation.

DBA (Database Administrator):

DBA will use SQL to maintain succusfull environment in Database (succussfull environment means if any crashes (or) deasters happened, how to recover (or) backup the data & how to prevent from crashes)

E.g for backend validation



2021/5/18 21:13

Scanned by CamScanner

Importance of SQL:

* Creating Database
* Creating tables in database
* Creating views
* Inserting new data into database
* Deleting unwanted data from database.
* updating / modifying old data to new data.
* fetching / Retrieving / accessing data from D.B
* Creating stored procedure } Advance SQL (PL/SQL)
* Creating index, triggers, cursors

(PL/SQL) - Procedural language extension to SQL

History of SQL

* SQL was developed by "Raymond Boyce" in the year 1970
* Raymond Boyce is known as father of SQL
* E.F codd [Edgar frank codd] is known as co-developer of SQL * SQL was developed by "IBM" company but the current owner of SQL is "oracle"
* In Earlier days SQL was known as SEQUEL [Simple English Query Executable Language], once ANSI Standard [American National Standard Institute] acquired SEQUEL and they renamed into SQL in a year 1979"

Basic Terminology:

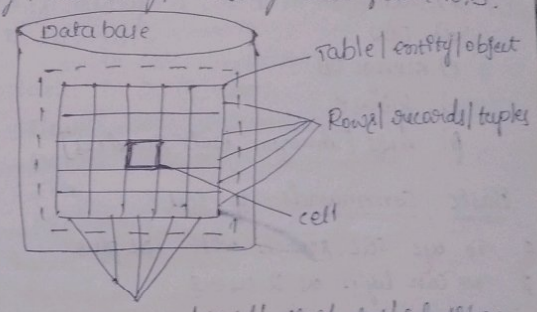
-> Data base: It is a centralized place which is used to store some data & manage the data is called as database.

-> (Centralized ~~data~~ place means it is the common place where everyone can use/access)

-> All database data will be stored in the form of tables

Each and every D.B should support the following interaction to users

-> Inserting new data into database
-> Deleting unwanted data from database
-> updating / modifying old data to New data
-> fetching / Retrieving / accessing data from D.B.



Column's | field | attributes | identifiers.

NOTE:

1) column's is also known as fields | attributes | identifiers
2) Row's is also known as Records | tuples
3) Tables is also known as ~~Tuples~~ | entity | object
4) cell is a intersection of Rows & columns.

2> RDBMS [Relational Database management System]

S/W:

a) Oracle
   2i [Internet version]
   3i
   4i
   ⋮
   9i

2021/5/18 21:13

Scanned by CamScanner

10g [Grid computing] } free version
11g
12c [cloud computing] } Paid version
18c
19c -> latest version
(feb 2020)

b) mySQL
c) SQL server
d) DB2
e) mongo DB
f) Sybase
g) AWS [Amazon web services]

Basic Commands of SQL

1. To use SQL search with SQL plus
2. we can login as 2 users
   1. Scott user
      user name: scott
      password: Oracle1234
      connected to
   2. HR user
      user name: hr
      Password: Oracle1234
      connected to
3. cl scr (space): This command is used to clear the screen
4. show user: This command is used to display loged in as which user
   e.g: show user

5. conn [connect]: This command is used to switch from one user to another user
   e.g: conn
   Enter username: Scott
   Enter password: Oracle1234
   connected to.
6. Select * from tab;
   This query is used to display no of tables in the user.
   -> tables in scott user
   BONUS
   DEPT
   EMP
   SALGRADE
   -> Tables in hr user
   REGIONS
   COUNTRIES
   LOCATIONS
7. desc emp [describe] :- This command is used to describe table structure (or) describe number of columns in a table.
   Syntax: desc table_name
   desc emp

Name	NULL?	TYPE
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

2021/5/18 21:14

Scanned by CamScanner

desc dept

Name	NULL?	TYPE
DEPT NO	NOT NULL	NUMBER(2)
DNAME		VARCHAR(14)
LOC		VARCHAR(13)

-* SQL Statements (or) SQL Languages

SQL statements / languages are use to interact with database.

-> In SQL there are 5 sub languages

1. DDL (Data defination language) -> create, alter, rename, drop, ~~term~~ truncate
2. DML (Data manipulation language) => Insert, update, delete
3. DCL (Data control language) => Grant, revoke
4. TCL (Transaction control language) => commit, savepoint, rollback
5. DQL (Data Query language) => Select

DQL (Data Query Language):

Fetching / Retrieving / accessing data from database is called as DQL

-> In DQL we have one statement i.e select.

SELECT:

Fetching / Retrieving / accessing data from specific table in database is called as select statement

=> In select we have 3 types

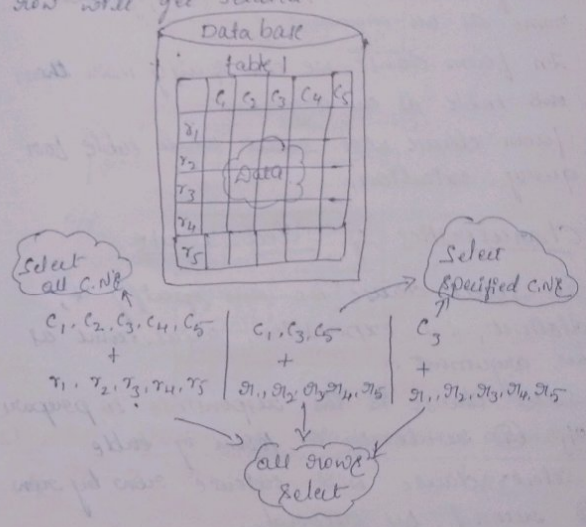
1) projection
2) ~~Sta~~ selection
3) Joins

Projection:

Fetching / Retrieving / accessing data from specific table by selecting all C.N's (or) specified (or) required number of columns is called as projection.

=> In projection we can restrict columns but we cannot restrict rows

=> In projection irrespective of column's all the row will get selected.



Syntax:

Select * / [Distinct] c.N/expression alias name -> (2)
from tablename; -> (1)

metachacter, optional

[ ] - optional
/ - or
C.N - column name

2021/5/18 21:14

Scanned by CamScanner

=> A/c to above syntax we have 2 statements i.e. select and from

=> First from statement will execute & 2nd select statement will execute.

Characteristics of from clause

* In from clause we will specify table name as an argument.
* In from clause we can specify more than one table as an argument
* from clause will select whole table for query execution.

Characteristics of select clause

* In select clause we can specify *, distinct, c.N, expression, alias_name as an argument.
* Select clause is the responsible to prepare o/p (or) result in a form of table
* Select clause will execute row by row (or) record by record.

NOTE :

S_1 : What to display -> select statement / clause

S_2 : from where to display -> from statement / clause.

Que 1. WAD [write query to display],
Student name, branch of all the students
↓ disp

Student

Sid	Sname	Per	branch
1	Monkey	70	CS
2	Sundra	35	IS
3	Sundri	40	EC
4	Pinky	99	ME
5	donkey	55	Civil

Select sname, branch → (2)
from student; → (1)

O/P

Sname	branch
Monkey	CS
Sundra	IS
Sundri	EC
Pinky	ME
donkey	Civil

2) WAD student id, student name, percentage of all the students
↓ disp

Select sid, sname, per
from student;

O/P

Sid	Sname	Per

2021/5/18 21:14

Scanned by CamScanner

3) WAD percentage, branch of all the students
↓ disp

Select per, branch
from student;

4) WAD student name of all the students
↓ disp

Select sname
from student;

5) WAD student id, student name, branch of all the students
↓ disp

Select sid, sname, branch
from students;

6) WAD student details
↓ disp

Select sid, sname, per, branch
from student;

(or)

Select *
from student;

* (Meta Character):

whenever we want to diply all cols from a table then we will go *

7) WAD employee name, designation, salary of all the employees
↓ disp

Select ename, Job, sal
from emp;

8) WAD employee number, reporting manager number, date of joining of all the employees
↓ disp

Select empno, mgr, hiredate
from emp;

9) WAD employee details
↓ disp

Select *
from emp;

10) WAD designation, commission, deptno of all the employees
↓ disp

Select Job, comm, deptno
from emp;

11) WAD commission, salary, employee number of all the employees.
↓ disp

Select comm, sal, empno
from emp;

2021/5/18 21:14

Scanned by CamScanner

12) WAD date of joining, department number, employee name, employee number of all the employees
↓ disp

select hire date, deptno, ~~empno~~ ename, empno
from emp;

13) WAD employee name of all the employees
↓ disp

Select ename
from emp;

14) WAD department details
↓ disp

select *
from dept;

15) WAD department name, location of the departments
↓ disp

Select dname, loc
from dept;

16) WAD location, department number of all the departments
↓ disp

Select loc, deptno
from dep~~artment~~;

17) WAD Region name of all the regions
↓ disp

Select Region_Name
from regions;

=> desc regions
name
REGION_ID
Region_Name

=> desc Countries
Contry_Id
Contry_name
region_Id

=> desc locations
location_Id
street_address
postal_code
city
State_Province
country_id.

18) WAD Region details
↓ disp

Select *
from regions;

2021/5/18 21:14

Scanned by CamScanner

19) WAD <u>centry name, country id</u> of the Countries
↓disp

Select country_name, country_id
from countries;

20) WAD <u>countryid, regionid</u> of all the
countries ↓disp

Select country_id, region_id
from countries

21) WAD <u>Countries details</u>
↓disp

Select *
from countries;

22) WAD <u>countryname, regionid</u> of all the
Countries ↓disp

Select country_name, region_id
from countries;

23) WAD <u>location details</u>
↓disp

Select *
from locations;

24) WAD <u>location id, street address, city</u> of the
all the locations ↓disp

Select location_id, street_address, city
from locations;

25) WAD <u>Postal code, country id, state province</u>
of all the locations ↓disp

Select Postal_code, country_id, state_province
from locations;

26) WAD <u>cityname, Postal code, location id</u> of
all the locations ↓disp

Select city, Postal_code, location_id
from locations;

<u>Alias Name</u> :

Alias name is a alternative / temporary name given to C.N / expression present in result / O/P

=> whenever we want to change C.N / expression present in result (or) O/P then we will go alias name.

=> whenever we are writting aliasname for C.N / expression it will not permanently change in original table

2021/5/18 21:14

Scanned by CamScanner

Syntax: select C.N / expression [AS] alias name → (2) (optional)
from table_name; → (1)

e.g: 1) select sal as salary
from emp;
o/p → salary
800
⋮

2) select sal salary
from emp;
o/p → salary
800
⋮

3) select sal salary, job designation
from emp;
o/p → salary designation
800 clerk
⋮

Order By clause:

=> order by clause is use to sort the o/p (3) result in ascending order / descending order then we will go order by clause.

=> By default order by clause will sort the o/p in ascending order so writting "ASC" keyword is an optional.

=> If we want to sort the o/p in descending order then writting "DESC" keyword is mandatory.

=> order by clause should be the last statement in query.

Syntax: select */[distinct] C.N / expression
alias name → (3)
from table_name → (1)
orderby C.N / expression / alias name [ASC] / DESC; → (2) (optional)

1) WAD employee details & display the o/p in ascending order based on salary.

select *
from emp
order by sal asc;
(or)
order by sal;

2) WAD employee name, designation, date of joining, employee number & display the o/p in descending order based on employee number

select ename, Job, hiredate, EmpNo
from emp
order by EmpNO DESC;

3) WAD employee name, manager number, department number, commission. display the o/p in ascending order based on manager number

select Ename, MGR, deptno, comm
from emp
order by MGR;

4) WAD employee name, employee number, annual sal & display the o/p in ascending order based on annual sal

select ename, empno, sal*12 annsal
from emp
order by sal*12; (or) order by annsal;

Expression: when ever we are writting alias name in order by clause same alias name should be present in select clause

=> An expression is a statement & it will generate result (or) o/p

=> Generally an expression is a combination of operator & operands

Operator:

1) Arithematic operator => +, -, *, /
2) logical operator => AND, OR, NOT
3) Relational operator => <, >, >=, <=, =, != etc

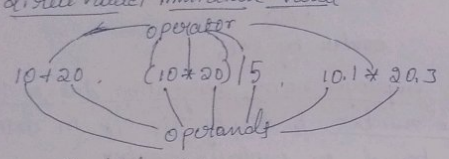
2021/5/18 21:15

Scanned by CamScanner

Operands

=> operand is a value
=> operand is classified into 2 types

1) direct value / immidiate value



2) variable / identifier

int a = 10, b=20

c = a + b

operator operand

In SQL we will use identifier/C.N as operands

sal + 1000 , sal * comm

operator operand operator operand

Literals:

Literals is a direct/immidiate value given to any of the SQL statement is called as literals.

Literals is ~~cal~~ classified into 3 types

1) Number literal
2) String/ character literal
3) Date literal

Number literal: It is a direct/immidiate value which is of type number & given to any of the SQL statement. is called as number literal

e.g: 10, 420, 840, 10.33 , 10000

String/ Character literal: It is a direct/immidiate value which is of type string/character & given to any of SQL statement is called as string / character literal.

=> Always string/character literal should be enclosed single quote (' ')

e.g: 'Soundarya', 'Nayana'

3) **Date literal:** It is a direct/immidiate value which of type date format & given to any of the SQL statement is called as date literal.

=> Date literal should be enclosed with in single quote (' ')

Oracle Date format

DD-MON-YY (or) DD-MON-YYYY

e.g: '15-Oct-1996', '15-Aug-1947', '27-FEB-95'

1) WQD employee name, employee number, salary, annual salary of all employees.

1 year sal

```
Select ename, empno, sal, sal*12 Annual salary
from emp;
```

2) WQD employee name, employee number, designation, salary, new salary with 1000 Rs increment

```
Select ename, empno, Job, sal, sal+1000 New Sal
from emp;
```

3) WQD employee name, date of joining, deptno. salary, New salary with 500 decrement

```
Select ename, hiredate, deptno, sal, sal-500 New sal
from emp;
```

4) WQD employee name, employee number, date of joining, salary, half term salary

```
Select ename, empno, hiredate, sal, sal*6 half term sal
from emp;
```

2021/5/18 21:15

Scanned by CamScanner

5. WQD employee name, designation, salary, new salary with 30% increment.

Select ename, Job, sal, sal+(sal×0.3) Newsal
from emp;

(or)

Select ename, Job, sal, sal×1.3 Newsal
from emp;

sal + sal×0.3
sal(1+1×0.3)
sal(1+0.3)
sal(1.3)
sal×1.3

6) WQD employee name, employee number, manager number, salary, comission, New comission with 20% increment.

Select ename, empno, mgr, sal, comm, comm×1.2 Newcomm
from emp;

7) WQD employeename, employee number, date of joining, salary, New salary with 30% decrement

Select ename, empno, hiredate, sal, sal-(sal×0.3) Newsal
from emp;

(or)

Select ename, empno, hiredate, sal, sal×0.7 Newsal
from emp;

sal - sal×0.3
sal(1-1×0.3)
sal(1-0.3)
sal×0.7

8) WQD employee name, designation, salary, comission New salary with 500 Rs decrement, New commission with 30% decrement

Select ename, Job, sal, comm, sal-500 Newsal,
comm×0.7 Newcomm
from emp;

9) WQD employee name, employee number, Anual salary, Anual comission. of all employees

Select ename, empno, sal×12 Annsal, comm×12 Anncomm
from emp;

10) WQD employeename, employeenumber, salary, Anualsalary, New annual salary with 10.000 increment

Select ename, empno, sal, sal×12 Annsal, (sal×12)+10000 Newsal
from emp;

11) WQD employeename, designation, salary, New annual salary with 500 Rs decrement

Select ename, Job, sal, (sal×12)-500 New-sal
from emp;

12) WQD employeeame, employee number, salary, comission, by giving 100 Rs increment to salary find annual salary

Select ename, empno, sal, comm, (sal+100)×12 annsal
from emp;

13) WQD employee name, employee number, comission, New commission. by giving 500 Rs decrement find annual comission.

Select ename, empno, comm, (comm-500)×12 ann_comm
from emp;

14) WQD employeeame, employeenumber, salary, by giving 30% increment to salary find out half term salary

Select ename, empno, sal, (sal×1.3)×6 half tem sal
from emp;

2021/5/18 21:15

Scanned by CamScanner

15) WQD employee name, employee number, salary, commission, by giving 20% increment to comission findout annual comission, by giving 10% decrement to salary findout quaterly salary.

Select ename, empno, sal, comm, (comm×1.2)×12 AnnComm
(sal×0.9)×3 quaterly_sal
from emp;

16) WQD employee name, employee number, manager number, Annual salary, Annual comission, New annual salary with 30% increment, New Annual comission with 10% decrement.

Select ename, empno, mgr sal×12 Annsal, comm×12
AnnComm, (sal×12)×1.3, New_Annsal,
(comm×12)×0.9 New_Ann Comm
from emp;

Distinct clause:

* Distinct clause is used to eliminate duplicate values present in o/p (or) result.
* Distinct clause will execute after select clause.

Syntax: Select distinct c.N/expression
from table_Name; → ①

17) WQD unique salary of all the employees

Emp

ename	job	sal
smith	clerk	3000
allen	salesman	4000
king	manager	3000
blake	clerk	5000
miller	salesman	4000

Select distinct sal
from emp; → ①

o/p: sal
3000
4000
5000

18) WQD unique designation of all the employees

Select distinct job
from emp;

o/p job
clerk
Salesman
manager

3) WQD unique supporting manager number
Select distinct mgr
from emp;

4) WQD unique annual salary of all the employees
Select distinct sal×12 annual sal
from emp;

5) WQD unique salary, unique designation of all the employees.
Select distinct sal, distinct job
from emp.

Select distinct sal, distinct job
*
Error at line 1.
X

NOTE: Any clause/statement can be written only one time in a query

⇒ A/c to above query we cannot write multiple distinct clause but we can write multiple c.Ns in a single distinct clause.

Multiple C.Ns in distinct clause

Whenever we are writting multiple c.Ns to single distinct clause distinct clause will not eliminate duplicate values in individual c.Ns instead of that it will combine multiple c.N value into a single value and then eliminate duplicate value

2021/5/18 21:15

Scanned by CamScanner

1> WAD unique designation, salary of the employees.

(a) Select distinct JOB, sal
from emp; → ①

op:

JOB	sal
clerk	3000
salesman	4000
manager	3000
clerk	5000
manager	4000

Emp

Ename	JOB	sal
smith	clerk	3000
allen	salesman	4000
king	manager	3000
blake	clerk	5000
miller	clerk	3000
queen	manager	4000

2> WAD unique MGR NO, unique designation of all the employees.

Select distinct MGR, JOB
from emp;

3> WAD unique salary, deptno, designation of all the employees.

Select distinct sal, deptno, JOB
from emp;

NOTE:

=> we cannot write any column before distinct clause

eg WAD ename, empno, unique designation.

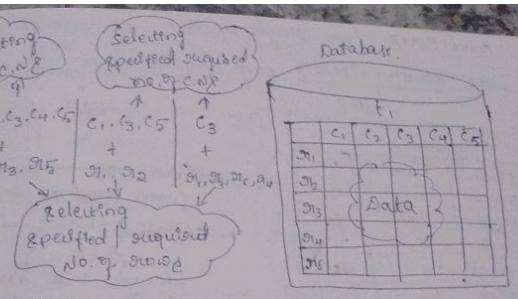
Select ename, empno, distinct JOB.

X (wrong)

* SELECTION:

Fetching / Retrieving / Accessing data from specific table by selecting all columns (or) specified or required no of C.N & also specified or required no of rows is called as selection.

=> In selection we can restrict both C.N & Rows



Syntax:

select * | [distinct] C.N/expression aliasname → (a)
from table_name → (b) ①
where <cond'n>; → ②

=> 1st from clause will excute, 2nd where clause will excute & last select clause will excute.

* Characteristics of where clause

=> where clause is used to filter the records

=> where clause it will select the record & reject the record

=> In where clause we will specify condition

Syntax of condition *

where	LHS-value	Relational operators	RHS-value;
	↓	↓	↓
	C.N / Express	<, >, <=, >=, =, !=	literals / C.N / expression

=> where clause will excute row by row (or) record by record.

=> If the specified condition is true then where clause will retaine the records. else if the specified condition is false then where clause will reject the record.

=> If the specified condition true (or) false where clause will not stop the execution, it will continue the execution for all the records present in table.

=> where clause will not excute multiple records at one shot but where clause will execute multiple

2021/5/18 21:15

Scanned by CamScanner

conditions for single record at our sho

1) WAD student details only percentage more than 60
↓ disp ↓ cond

Students

sname	branch	per
Chintu	ME	90
Pintu	CIVIL	70
Dora	CS	35
Chukki	IS	45
Kalyya	EC	65

```
Select * ---> 3
from students ---> 1
where per > 60; ---> 2
```

LHS-value (c.n) | Relat^n operator | RHS-value (number, literal)

O/P

sname	branch	per
Chintu	ME	90
Pintu	CIVIL	70
kalyya	EC	65

90 > 60 ---> T
70 > 60 ---> T
35 > 60 ---> F
45 > 60 ---> F
65 > 60 ---> T

2) WAD Studentname, percentage only those who are getting percentage less than 70
↓ disp ↓ cond

```
Select sname, per
from students
where per < 70;
```

3) WAD student details only those who are getting percentage 60.
↓ disp

```
Select *
from students
where per = 60;
```

4) WAD employee name, designation, salary only those who are getting salary more than 2000
↓ disp ↓ cond

```
Select ename, job, sal
from emp
where sal > 2000;
```

5) WAD employee details only those who are getting salary exactly 3000
↓ disp ↓ cond

```
Select *
from emp
where sal = 3000;
```

6) WAD employee name, employee number, department no only the employee no. should be more than 7500 & disply the o/p in Ascending order based on employee no.

```
Select ename, empno, deptno
from emp
where empno > 7500
order by empno;
```

7) WAD employee details only those who are working in department number 30 & disply the o/p in descending order based on salary

```
Select *
from emp
where deptno = 30
order by sal desc;
```

8) WAD employee details only the employees those who are working as salesman

```
Select *
from emp
where job = 'salesman'
```

O/P = No rows selected.

Rules:

R1: SQL Language Statement (table_name, c.n name/clause) is not case sensitive. only the values present in a table are case sensitive.

R2: whenever we are writing query by looking into table structure we should write the query not by looking into values present in a table.

2021/5/18 21:16

Scanned by CamScanner

Case manipulation function:

There are 3 types of case manipulation function.

1) upper()
2) lower()
3) initcap()

1) upper() :- This function will convert the given string into uppercase.

Syntax : upper ('string' / C.N)

NOTE: String should be enclosed with in single quote not column name.

e.g: Select upper ('akash')
from dual;

o/p : AKASH

Select upper ('AKASH')
from dual;

o/p : AKASH

Select upper ('Akash')
from dual;

o/p :- AKASH

select upper (ename)
from emp.

o/p :- SMITH

⋮

2) lower() : This function will convert the given string into lowercase

Syntax : lower ('string' / C.N)

e.g: 1) Select lower ('AKASH')
from dual;

o/p :- akash

2) Select lower ('akash')
from dual;

o/p :- akash

3) Select lower ('Akash')
from dual;

o/p :- akash

4) Select lower (ename)
forme emp;

o/p : smith

⋮

3) initcap() : This function will convert the given string into init cap case (initial letter capital)

Syntax : init cap ('string' / C.N)

e.g: 1) Select initcap ('akash')
from dual;

o/p :- Akash

2) Select initcap ('AKASH')
from dual;

o/p : Akash

3) select initcap ('Akash')
from dual

o/p : Akash

4) select init cap (ename)
from emp;

o/p : Smith

⋮

Application of case manipulation function (or) when to go for case manipulation function

→ whenever we want to write string / character literal in where clause then it better to go case manipulation function

1) WQD employee details only the employees those who are working as salesman

Select *
from emp
where uppercase (Job) = 'SALESMAN';
(or)
where lowercase (Job) = 'salesman';
(or)
where initcap (Job) = 'Salesman';

NOTE: what type of case manipulation function writting in LHS side same type of value should be present in RHS side.

*2) WQD employee name, designation salary only those who are working as clerk & display the o/p in Ascending order based on salary

2021/5/18 21:16

Scanned by CamScanner

Select ename, Job, sal
from emp
where lower(Job) = 'clerk'
order by sal;

2) WAD employee details only for employee SMITH
↓disp ↓cond

Select *
from emp
where lower(ename) = 'smith';

3) WAD employee name, employee number, salary, annual salary, only for a employee 'Turner'

Select ename, empno, sal, sal×12 annsal
from emp
where upper(ename) = 'TURNER';

4) WAD employee name, employee number, manager no, depart number only the reporting mgr should be 7698
↓disp ↓cond

Select ename, empno, MGR, deptno
from emp
where mgr = 7698;

5) WAD employee details only the employees those where are hired after a year 1981
↓disp ↓cond

Select *
from emp
where hire date > '31-DEC-1981';
>= '01-JAN-1982';

Assignment of projection:

1) WAD name of employee.
↓disp

Select ename
from emp;

2) WAD ename & salary of all employee
↓disp

Select ename, sal
from emp;

after -> >=
before -> <=

3) WAD department name & its locations for all the depart^n
↓disp

Select dname, loc
from dept;

4) WAD name, salary, comission & date of joining of every employee
↓disp

Select ename, sal, comm, hiredate
from emp;

5) WAD employee name, his designat^n & his manager's employee number
↓disp

Select ename, Job, MGR
from emp;

6) WAD employee name, & his department number
↓disp

Select ename, deptno
from emp;

7) WAD all the details of each & every employee of the company
↓disp

Select *
from emp;

8) WAD all the details of department present in company
↓disp

Select *
from deptno;

9) WAD employee name, his employee number, his salary & also his annual salary for every employee.
↓dis

Select ename, empno, sal, sal×12 annsal
from emp;

10) WAD employee name, his employee number & salary with the hike of 30% for every employee
↓disp

Select ename, empno, sal, (sal×1.3) New_sal
from emp;

11) WAD name of employee with annual salary with a monthly bonus of 200 and provide suitable alias name.
↓disp

Select ename, (sal+200)×12 New_sal
from emp;

2021/5/18 21:16

Scanned by CamScanner

12) WQD name, salary along with annual salary of employee with yearly bonus of 1000 ↓disp

Select ename, sal, (sal×12)+1000 annsal
from emp;

13) WQ to select all the salary of employee of the salary is greater than 1500. ↓cond

Select sal
from emp
where sal > 1500;

14) WQD employee name, his hiredate, his salary & annual salary only if his annual salary greater than 10000 ↓disp ↓cond

Select ename, hiredate, sal, sal×12 annsal
from emp
where sal×12 > 10000;

15) WQD all the details of employee if the department number is 20 ↓disp ↓cond

Select *
from emp
where deptno = 20;

16) WQD all the details of employee if designation is manager ↓disp ↓cond

Select *
from emp
where lower(Job) = 'manager';

17) WQD all the details of employee only if they were hired after the year 1995 ↓disp ↓cond

Select *
from emp
where hiredate > '31-DEC-1995';

18) WQD all the details of employee only if his designation is clerk ↓disp ↓cond

Select *
from emp
where lower(Job) = 'clerk';

19) WQD department name, department number for all the department which is located in BOSTON.

Select dname, deptno
from dept
where lower(dname) = 'boston';

---

6) WQD employee details only the employees those who are hired before a year 1981 ↓disp ↓cond

Select *
from emp
where hiredate <= '31-DEC-1980';

7) WQD employee name, designation, date of joining, only the employees those who are hired after feb 1981 ↓disp ↓cond

Select ename, Job, hiredate
from emp
where hiredate >= '01-MAR-81';

8) WQD employee details these who are hired before dec 1985. ↓disp ↓cond

Select *
from emp
where hiredate <= '30-NOV-1985';

9) WQD employee name, only the employees those who are hired on 31st dec-1981 ↓cond

Select ename
from emp
where hiredate = '31-DEC-1981';

10) WQD employee name, designation, annual salary only the annual salary should be more than 10000 ↓disp ↓cond

Select ename, Job, sal×12 annsal
from emp
where sal×12 > 10000;

=> we cannot write alias name in where clause because it will exicute before select clause but we can write alias name in order

2021/5/18 21:16

Scanned by CamScanner

because order by clause will execute after Select clause.

where annual > 10000;

X

11) WAD region details only for a region name ASIA
disp cond

Select *
from regions
where upper(region_name) = 'ASIA';

12) WAD region name only the region id should be 3
disp cond

Select region_name
from regions
where region_id = 3;

13) WAD country details only if the country id is IN
disp cond

Select *
from countries
where ~~country id~~ lower(country_id) = 'in';

14) WAD country name only the region id should be less than 4
disp cond

Select country_name
from countries
where region_id < 4;

15) WAD location details only for a city bomby.
disp cond

Select *
from locations
where lower(city) = 'bomby';

16) WAD location id, city, postal code only state province should be maharastra.
disp cond

Select location_id, city, postal code
from locations
where lower(state_province) = 'maharastra';

17) WAD location details only the country id should be US & disply the O/P in Ascending order based on location id.

Select *
from locations
where lower(country_id) = 'us'
order by location_id;

18) WAD location details only for a postal code 1234
disp cond

Select *
from locations
where postal_code = 1234

19) WAD employee details only new salary with 1000 increment should be more than 10,000
disp cond

Select *
from emp
where (sal+1000) > 10,000;

20) WAD employee name, employee number, salary, commission, new salary with 30% increment only the new salary should be less than 50,000 rs
cond

Select ename, empno, sal, comm, (sal x 1.3) newsal
from emp
where sal x 1.3 < 50,000;

21) WAD employee details only the salary should be less than comission.
disp cond

Select *
from emp
where sal < comm;

* Operators

=> whenever we want to write multiple conditions in single where clause then we should go operators

=> whenever we are writing multiple condition in where clause. where clause will execute multiple condn for single record at one shot

=> operators are classified into 5 types

1) Arithematic operator -> +, -, *, /

2) Concatenation operator -> ||

Scanned by CamScanner

2021/5/18 21:17

3) Relational operator → <, >, <=, >=, =, !=/ <>
4) Special operator → IN, BETWEEN, LIKE, IS
5) Logical operator → AND, OR, NOT

Concatenation operator → (||)

It is used to combine/merge/concatenate more than one string into a single string. then we will go concatenation operator

Syntax: 'String1'/c.N || 'String2'/c.N || 'String3'/c.N || .....

e.g: 1) Select 'don' || 'key' from dual;
o/p = donkey

2) Select 'Hello ' || ename from emp;
o/p = Hello smith

1) WQD the o/p in below mentioned format
Hello 'x' your salary is 'y' and your working as 'z'
x → ename, y → sal, z → job

Select 'Hello ' || ename || 'your salary is' || sal || 'and your working as' || job
from emp;

2) WQD the o/p in below mentioned format
'Department' x 'located in' y

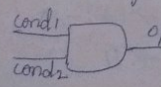
Select 'Department ' || dname || 'located in' || loc
from dept;

LOGICAL OPERATOR

=> There are 3 types of logical operator i.e And, Or, NOT

AND: when the given conditions are mandatory to exicute then we will go And operator.

Symbol



Truth table

cond1	cond2	o/p
True	True	True
True	False	False
False	True	False
False	False	False

Case 1: If condition 1 is true the o/p of and operator is depends on condition 2
If condition 2 is true then o/p is true else if condition 2 is false then o/p is false.

Case 2: If condition 1 is false whatever it might be condition 2 the o/p of And operator is false.

Syntax: where cond1 & cond2 & cond3 & .....

OR: when either of conditions are mandatory to exicute then we will go OR operator

Symbol

Truth table

Cond1	cond2	o/p
True	True	True
True	False	True
False	True	True
False	False	False

Case 1: If condition 1 is true whatever it might be the condition 2 the o/p of OR operator is true

Case 2: If condition 1 is false the o/p of OR operator is depends on condion 2
if condition 2 is true then o/p is true else if condition 2 is false then the o/p is false.

Syntax: where cond1 OR cond2 OR cond3 OR .....

1) WQD student details only the branch should be CS (cond1) the percentage (o/p) should be more than 55 (cond2)

Student

Sname	branch	per	
chirag	ME	60	✗
Masha	CS	88	✓
Pasha	IS	45	✗
Nancy	Civil	70	✗
Pollu	[illegible]	[illegible]	✗

Select * — ③
from student — ②
where lower(branch) = 'cs' and per > 55; — ①

Scanned by CamScanner

2021/5/18 21:17

cond1 cond2

me = CS → F and 60 > 55 → T → F
CS = CS → T and 98 > 55 → T → T
IS = CS → F and 45 > 55 → F → F
civil = CS → F and 70 > 55 → T → F
EC = CS → F and 66 > 55 → T → F

O/P :

Sname	Branch	per
masha	CS	98

2) WAD student name, branch only the percentage should be exactly 70 and the student name should be akash
cond1 cond2

```
Select sname, branch
from student
where per=70 and lower(sname) = 'akash';
```

3) WAD employee details only the employees those who are working as salesman & they should be working for a department 30
cond1 cond2

```
Select *
from emp
where lower(Job) = 'salesman' and deptno=30;
```

4) WAD employee name, designation, manager number only the salary should be more than 1000 and their employee number should be 7900
cond1 cond2

```
Select ename, Job, mgr
from emp
where sal > 1000 and empno = 7900;
```

5) WAD employee details only the employee name should be smith & the employee should be hired after a year 81
cond1 cond2

```
Select *
from emp
where lower(ename) = 'smith' and hiredate >= '01-Jan-81';
```

6) WAD employee details only the employees those who are working in a department number 10, 20

```
Select *
from emp
where deptno = 10 or deptno = 20;
```

NOTE :
* All relational operator (<, >, >=, <=, =, !=) will accept single value for LHS side and single value in RHS side.
e.g: where deptno = 10,20;
X
* when column name is same but values are different then it is better to go OR operator.

7) WAD employee details only those who are reporting manager number 7698, 7839
cond

```
Select *
from emp
where mgr = 7698 OR mgr = 7839
```

8) WAD employeename, salary only the employees those who are earning salary 3000, 1500, 1250
cond

```
Select ename, sal
from emp
where sal = 3000 OR sal = 1500 OR sal = 1250;
```

9) WAD employee details only the employees those who are working as clerk, salesman
cond

```
Select *
from emp
where lower(Job) = 'clerk' OR lower(Job) = 'salesman';
```

10) WAD employee details only the employee number should be 7900, 7788, 7698
cond

```
Select *
from emp
where empno = 7900 OR empno = 7788 OR empno = 7698;
```

11) WAD employee name, employee number, annual salary only the annual salary should be 10,000, 9600, 15000
cond

Scanned by CamScanner

2021/5/18 21:17

Select ename, empno, sal×12 annual
from emp
where sal×12 = 10,000 OR sal = 9600 OR sal = 15000;

12) WAD employee details only for employee smith, Turner, blake and disply the o/p in ascending order based on sal

Select *
from emp
where lower(ename) = 'smith' or lower (ename) = 'turner' or lower (ename) = 'blake'
order by sal;

13) WAD employee details only employee should be working in depart no. 30 & they should be working as clerk, salesman (cond) and (cond)

Select *
from emp
where (deptno = 30) and (lower(job) = 'clerk' or lower(job) = 'salesman');

14) WAD employee name, empno, manager no only the annual salary should be less than 20,000 and reporting manager number should be 7698, 7839 and disply the o/p in descending order based on employee number.

Select ename, empno, mgr
from emp
where (sal×12 < 20000) and (mgr = 7698 or mgr = 7839)
order by empno desc;

15) WAD employee details only the employee name should be miller, smith and they should be hired before 85

Select *
from emp
where (lower(ename) = 'miller' or lower(ename) = 'smith') and (hiredate <= '31-DEC-84');

16) WAD employee details only those who are earning commission 1400, 500 and they should be working in a department no. 10, 20, 30

Select *
from emp
where (comm = 1400 or comm = 500) and (deptno = 10 or deptno = 20 or deptno = 30);

17) WAD ~~only~~ employee name, date of joining, annual sal only the employees those who are hired on 3rd dec 81, 20th feb 81 and their employee number should be more than 7000 and disply the o/p in ascending order based on date of joining.

Select ename, hiredate, sal×12 annual
from emp
where (hiredate = '03-DEC-81' or hiredate = '20-Feb-81') and (empno > 7000)
order by hiredate;

18) WAD employee name, salary (sal) only the salary should be in a range of 1000 - 5000 (disp)

Select ename, sal
from emp
where (sal >= 1000 and sal <= 5000);

range >= and <=

19) WAD employee details only the manager number should be in a range 7600 - 7900

Select *
from emp
where (mgr >= 7600 and mgr <= 7900);

Scanned by CamScanner

20) WQD employee details only the employees those who are hired in a year 81 from 01-Jan-81 to 31-DEC-81

```
Select *
from emp
where (hiredate >= '01-Jan-81' and hiredate <= '31-DEC-81')
```

21) WQD employee details only the employees those who are hired in a month feb 81 from 01-feb-81 to 28-feb-81

```
Select *
from emp
where (hiredate >= '01-feb-81' and hiredate <= '28-feb-81')
```

22) WQD employee details only the employees those who are hired in a month feb 81 to nov 85 and they should be working as salesman

```
Select *
from emp
where (hiredate >= '01-feb-81' and hiredate <= '30-nov-85')
      and (lower (Job) = 'salesman');
```

23) WQD employee details only the employees those who are hired in a year 81 to 87 and they should be working in a department no 30

```
Select *
from emp
where (hiredate >= '01-Jan-81' and hiredate <= '31-DEC-87')
      and (deptno = 30);
```

24) WQD employeename, designation, date of joining only the annual salary should be in a range of 5000-50000 and they should be reporting to a manager no 7698, 7839

```
Select ename, Job, hiredate
from emp
where (sal*12 >= 5000 and sal*12 <= 50000) and (mgr = 7698
      or mgr = 7839);
```

25) WQD employee details only the employees those who are hired in a month dec 81 and their manager no should be in a range of 7600 to 7900

```
Select *
from emp
where (hiredate >= '01-dec-81' and hiredate <=
      '31-dec-81') and (mgr <= 7600 and mgr <= 7900);
```

26) WQD regions details only for a region asia, africa

```
Select *
from region
where lower(region_name) = 'asia' or lower
      (region_name) = 'africa';
```

27) WQD region details only for a region id 1,2

```
Select *
from region
where region_id = 1 or regionid = 2
```

28) WQD region name only the region id should be in a range of 1 to 4

```
Select region_name
from region
where region_id >= 1 and region_id <= 4;
```

29) WQD country details only for a country id In, US, CA

```
Select *
from countries
where lower(country_id) = 'in' or lower(country_id)
      = 'us' or lower(country_id) = 'ca';
```

30) WQD country id, region id only for country India, Japan

```
Select country_id, region_id
from countries
where lower(country_name) = 'India' or
      lower(country_name) = 'Japan';
```

Scanned by CamScanner

31) WQD location details only location id should be in a range of 1000 to 5000 and state province should be maharashtra.

```
Select *
from locations
where (location_id >= 1000 and location_id <= 5000)
    and (lower (state_province) = 'maharashtra');
```

32) WQD location id, street address, postal code only for a country id IN and postal code should be less than 3000

```
Select location_id, street_address, postal_code
from locations
where (lower (country_id) = 'in') and (postal_code < '3000');
```

[only] in postal code, the values present in table are considered as string literal hence should enclose within single code.

33) WQD country name only the region id should be in a range of 1 to 4 and country id should be US

```
select country_name
from countries
where (region_id >= 1 and region_id <= 4) and
    (lower (country_id) = 'us');
```

34) WQD location details only for a postal code 2703, 4523 and state province should be oxford.

```
Select * from locations
where (postal_code = '2703' or postal_code = '4523')
    and lower (state_province) = 'oxford';
```

35) WQD country id, postal code, ~~country~~ location id, street address only for a location id 1500, 2500 and postal code should be less than 5000 and display the O/P in ascending order based on country id

```
Select country_id, postal_code, location_id,
    street_address
from locations
where (location_id = 1500 or location_id = 2500) and
    (postal_code < '5000')
order by country_id;
```

Special operators:

* IN: * IN is a multivalued operator
* IN operator will accept single value in LHS side & multiple values in RHS side.
* IN will perform = comparision
* IN operator will compare single LHS value with all the values present in RHS side
* IN operator will return true if any one of the conditions or comparision is true
* IN operator will return false if all conditions or comparisions is false
* IN operator working similar to like OR operator

Syntax: LHS value IN (value1, value2, value3 ....)

LHS value → C.N/Expression; (value1, value2, value3 ....) → RHS-value.

e.g.:

10 IN (20, 10, 30)
10 = 20 → F
10 = 10 → T
10 = 30 → F
} T

10 IN (20, 30, 40)
10 = 20 → F
10 = 30 → F
10 = 40 → F
} F

2) BETWEEN:

=> when the input is in the form of range then we should go BETWEEN operator.

Syntax

where LHS_value BETWEEN lower_limit and upper_limit

LHS_value → C.N/Expression

=> Always lower limit should be less than upper limit

lower limit < upper limit

=> Internally BETWEEN operator written as

>= lower limit and <= upper limit

2021/5/18 21:18

Scanned by CamScanner

=> BETWEEN operator will works inclusive range [inclusive range means including lower limit and upper limit value]

=> BETWEEN operator will return true if all conditions or comparisions true

=> BETWEEN operator will return false if any one the condition/ comparision is false

=> BETWEEN operator will works similar to like and operator.

e.g: ① 100 BETWEEN 90 and 120

100 >= 90 and 100 <= 120

T and T → T

② 100 BETWEEN 200 and 300

100 >= 200 and 100 <= 300

F and T → F

1) WQD employee details only the employees those who are working in department number 10, 20, 30

```
Select *
from emp
where deptno IN(10,20,30);
```

NOTE: when co.N name is same but values are different then it is better go IN operator.

2) WQD employee name, employee number, salary only the employees those who are reporting to managers no 7698, 7839, 7900

```
Select ename, empno, sal
from emp
where mgr IN (7698, 7839, 7900);
```

3) WQD employee name, designation, annual salary. only the employees those who are working as salesman, clerk

```
Select ename, Job, sal*12 annsal
from emp
where lower(Job) IN('salesman', 'clerk');
```

4) WQD employee name, employee number, designation only for a employee smith, allen, turner

```
Select ename, empno, Job
from emp
where lower(ename) IN ('smith', 'allen', 'turner');
```

5) WQD employee details only the employees those who are earning salary 3000, 1250, 5000 & disply the o/p in ascending order based on empno.

```
Select *
from emp
where sal IN (3000, 1250, 5000)
order by empno;
```

6) WQD employee details only for a employee no 7000, 7698, 7902 and they should be working as salesman

```
Select *
from emp
where (empno IN(7000, 7698, 7902)) and (lower(Job)
= 'salesman');
```

7) WQD employee details only those who are earning salary in a range of 1000 to 5000

```
Select *
from emp
where (sal BETWEEN 1000 and 5000);
```

8) WQD employee details only the employees those who are hired in a year 81

```
Select *
from emp
where hiredate between '01-Jan-81' and '31-Dec-81';
```

9) WQD employee name, designation, salary, comission only those who are hired in a month feb 81 and their comission should b[illegible]

2021/5/18 21:18

Scanned by CamScanner

Select ename, job, sal, comm
from emp
where (hiredate between '01-feb-81' and '28-feb-81') and (comm in (300, 500, 1500));

10) WAD employee details only the employees those who are earning annual salary 9600, 15000, 26000 and an employee should be hired in a month feb 81 to mar 85

Select *
from emp
where ((sal x12) IN (9600, 15000, 26000)) and (hiredate between '01-feb-81' and '31-mar-85');

11) WAD employee details only for a employee smith turner, blake and they should be working for a deptno 10, 20, 30

Select *
from emp
where (lower(ename) in('smith', 'turner', 'blake'))
and (deptno in(10, 20, 30));

12) WAD employee name, designation, salary only manager number should be in a range of 7600 - 7900 and they should be hired before a month feb 85

Select ename, job, sal
from emp
where (mgr between 7600 and 7900) and
(hiredate <='31-jan-85');

13) WAD employee details only employee number should be in a range of 7000 to 7900 and they should be hired in a year 81 to 87

Select *
from emp
where (empno between 7000 and 7900) and
(hiredate between '01-jan-81' and '31-dec-87');

14) WAD employee details only the employee should be working as clerk, analyst, president and they should be reporting to mgr 7698, 7839, 7900

Select *
from emp
where (lower(job) in (clerk, analyst, president))
and (mgr in (7698, 7839, 7900));

* **LIKE:**

=> Like operator is used to perform pattern matching or wildcard search

pattern / wild card search:

It is a combination of ordinary charecter & special charecter

ordinary charecter:

It is a charecter doesnot have any special significant (there is no meaning)

Special charecter:

-> It is a charecter which has some special significant (there is a meaning)

-> There are two special charecters are present

1) percentile (%)

2) underscore (_)

1) percentile (%): It is a special charecter it will accept all charecters at any number of time

-> [A-Z] } alphabets
-> [a-z]
-> [0-9] - number
-> [$-#] - spl. charecter

min 0 - accurance
max ∞ - accurance

[except %, _]

2021/5/18 21:18

Scanned by CamScanner

2) underscore [_] : it is special character it will accept all characters at only 1 time

min 1 accurance
max 1 accurance

- [A-Z] } alphabets
- [a-z]
- [0-9] - number
- [$-#] - spl character

(except %, _)

Syntax of like: LHS_value LIKE 'Pattern/wild_card_char -ch';
↓
C.N | expression

NOTE:
→ except %_ all other characters are ordinary characters (A-Z, a-z, 0-9, $-#)

1) write patter (w.p) employee name should begin with S character
'S%'
ordinary character
special character

2) w.p employee name should ends with S character
'% S'

3) w.p employee name consists of S character [consists means any where]
'% S %'

4) wp employee name begins with 2 consigative 'S' character (one after the another)
'SS %'

5) wp employee name ends with 2 consigative 'S' character
'% SS'

6) wp employee name consists of 2 consigative S character
'% SS %'

7) wp employee name consists of 2 S character
'% S % S %'

8) wp employee name consists of SUN as substring
'% SUN %'

9) wp employee name begging 3rd character should be 'S'
'_ _ S %'

10) wp employee name last 2nd character should be S
'% S _'

11) wp employee name should be earning 4 digit salary
'_ _ _ _'

12) wp employee name consist of 5 characters
'_ _ _ _ _'

13) wp employee should be hired in a year 81
'% 81' (or) '_ _ _ _ _ _ _ 81' (or) '_ _ _ _ _ _ 81'

14) wp employee should be hired in a month feb 81
'% FEB-81'

15) wp employee should be hired in a month feb
'% FEB %' or '_ _ _ FEB _ _ _'

1) WAD employee name, designation, salary only the employee name should begins with 'a' character.

```
Select ename, Job, sal
from emp
where lower(ename) like 'a%';
```

2) WAD employee details only employee name should consists of 'a' and they should be working for department 10, 20.

```
Select *
from emp
where (lower(ename) = '%a%') and (deptno in(10,20));
```

3) WAD employee details only the designation should consists of 'man' as substring and they should earning salary 1000 to 5000

```
Select * from emp
where lower(Job) like '%man%' and (sal between 1000 and 5000);
```

2021/5/18 21:18

Scanned by CamScanner

4) WAD employee name, salary, annual salary only employee name should consists of consigative 'L' and they should be supporting to mgr 7698, 7839

```
Select ename, sal, sal*12 annsal
from emp
where (lower(ename) like '%ll%') and (mgr in (7698,
                                              7839));
```

5) WAD employee details only employee name should end with character 'n' and they should be hired before a year 85

```
Select *
from emp
where lower(ename) like '%n' and hiredate <=
                                '31-dec-84';
```

6) WAD employee details only the designation shows consists of 'es' as substring and employee name begining 2nd character should be 'l'.

```
Select *
from emp
where lower(job) like '%es%' and lower(ename) like
                                '_l%';
```

7) WAD employee name, employee number, manager no only employee name should consists of 2 l and they should be earning salary more than 1000

```
Select ename, empno, mgr
from emp
where (lower(ename) like '%l%l%') and (sal > 1000);
```

8) WAD employee details only employee number begging 2nd digit should be '8' and they should be hired in a year 81

```
Select *                                  % 81
from emp                                  DD-mon-YY
where empno like '_8%' and hiredate '%81';
```

(or)

```
where empno like '_8%' and hiredate bbw '01-jan-81'
                                   and '31-jan-81';
```

9) WAD employee details only employee should be hired in a month feb 81 & they should be earning comm less than sal

```
Select
from emp
where (hiredate like '%feb-81') and (comm < sal);
```

10) WAD employee name, employee number, date of joining only employee should be hired in a month dec and they should be working as salesman, clerk

```
Select ename, empno, hiredate
from emp
where (hiredate like '%dec%') and (lower(job) in
                    (salesman, clerk));
```

11) WAD employee details only employee name should be exactly 5 character and they should be hired in a year 81 to 87

```
Select *
from emp
where (ename like '_____') and hiredate
        between '01-jan-81' and '31-dec-87');
```

NOTE:

=> If a month or year is exactly one year (or) one month then it is better to go like operator

=> If a month or year in a form of range then it is better to go between

=> "like operator rule" = " one like operator will accept single value in LHS side and single pattern in RHS side

12) WAD employee details only employee should be hired in a month feb 81 to nov 81 & employee name last 3rd character should be 's'

```
Select *
from emp
where (hiredate between '01-feb-81' and '30-nov-81')
      and (lower(ename) like '%S__');
```

13) WAD employee name only employee should be hired from 18th nov 80 to 25-dec-87 and employee should be earning exactly 5 digit salary

2021/5/18 21:18

Scanned by CamScanner

Select ename
from emp
where (hiredate between '01-nov-80' and '25-dec-87') and (sal like '----');

14) WQD employee name only employee name should begin with 'a' charcter or 's' charecter

Select ename
from emp
where lower(ename) ~~the~~ 'a%' or lower(ename) like 's%';

15) WQD employee name only the employee name should begins with vowels (a, e, i, o, u)

Select ename
from emp
where lower(ename) like 'a%' or lower(ename) like 'e%' or lower(ename) like 'i%' or lower(ename) like 'o%' or lower(ename) like 'u%';

16) WQD employee details only the employees those who are hired in a month feb, dec and empno should be in range of 7000 to 7900

Select *
from emp
where (hiredate like %feb%) or (hiredate like %dec%) and (empno between 7000 and 7900);

17) WQD employee details only the annual sal should be exactly 5 digit and mgr last 2nd digit should be 9

Select *
from emp
where sal×12 like '----' and mgr like '%9_';

Assignment 8 (scotts and his account)

1. WQD employee name whose is not working as salesman and analyst in dept 10. (emp table)

Select ename
from emp
where job not in ('salesman', 'analyst')
and (deptno = 10);

2) WQD 1st & last name of employee excluding employees who are working in dept 50 & 80 (emp table).

Select max(ename), min(ename)
from emp
where deptno not in (50, 80);

3) WQD the sal of kelvin & donald (emp table)

Select sal
from emp
where lower(ename) in ('kevin', 'donald');

4) WQD the department name whose location id is 1800 & 1700. (dept table)

Select dname
from dept
where location_id in (1800, 1700);

5) WQD job title for employees whose first names are steven, donald and mahesh (emp table)

Select job
from emp group by job
~~where~~ having max(ename) in ('steven, donald, mahesh);

6) WQD ph number, email id. for the employees who are working as sh_clerk, it_prog in dept 50, 30 & 80 (employees)

Select ph_number, email_id
from employees
where lower(job) in ('sh_clerk', 'it_programmer')
and deptno in (50, 30, 80);

7) WQD the dept name located in dallas & newyork

Select dname
from dept
where location

2021/5/18 21:18

Scanned by CamScanner

1200 to 2800 in dept 10,20,30

```
select * ename
from emp
where (lower(job) like '%a%a%') and (sal between
1200 and 2800) and (deptno in(10,20,30));
```

9) display all the employees who are having ~~atleast 2 A's in their~~ any reporting manager with no ending with '8' and not working as 'manager' or 'saleman' in dept 30 (emp table)

```
select * ~~ename~~ *
from emp where (mgr is not null) and
(mgr like '%8') and ((lower(job) != 'manager')
or (lower(job) != 'saleman')) and (deptno=30);
```

10) List all the employees who are not getting any comission with their designation neither 'clerk' nor 'manager' and joined in the year 81 and getting sal more than 1500

```
select * ~~ename~~ *
from emp
where (comm is null) and (lower(job) not in ('clerk',
'manager')) and (hiredate like '%81') and
(sal >1500);
```

11) Display all employees whose name starts with 'S' & does not end with vowel.

```
select * ename
from emp
where(lower(ename) like 'S%') and ((lower(ename) not like
'%a') and (lower(ename) not like '%e') and (lower(ename) not like '%i')
and (lower(ename) not like '%o') and (lower(ename) not like '%u'));
```

12) List all the employees who's salary is in the range of 1000 to 2000 dept 10,20,30 except all clerck?

```
select * ~~ename~~
from emp
where (sal between 1000 and 2000) ...
```

```
select *, ~~sal~~ sal*12 annsal
from emp
where lower(job) = 'manger';
```

14) Display all the employee who don't have commission & salary greater than 1000

```
select ~~ename~~ *
from emp
where (comm is null) and (sal >1000);
```

15) Display all the employee whose salary not in the range from 1500 to 3000.

```
select ~~ename~~ *
from emp
where sal not between 1500 and 3000;
```

16) List all employee ~~whose salary~~ with annual sal with 500 rs bonus and working as manager joined in the month of april and may.

```
select ename, (sal*12) + 500 annsal
from emp
where (lower(job) = 'manager') and (hiredate b/w
april and may)
```

17) List all the employee job with hiredate whose name starts between A & S, working in dept 30 & 20

```
select ~~ename~~ job, hiredate
from emp
whose (lower(ename) like a%s) and (deptno in(30,20));
```

18) List all the emp whose manager employee no b/w 7698. 7839 and salary ends with no. 5?

```
select ename
from emp
```

Scanned by CamScanner

```
select ename
from emp
where (deptno in (30, 10)) and (hiredate like '%81');
```

2.) Display all the employees where job has string 'na' in

```
select ename
from emp
where lower(job) like '%nan';
```

2.) select all the salesman where name has atleast 1 character 'R' whose salary is less than 1500 and who joined after 21-feb-81

```
select ~~job~~ *
from emp
where (lower(~~ename~~ job) = 'salesman') and (lower(ename)
like '%r%') and (sal < 1500) and (hiredate >
'21-feb-81');
```

2.) WAD to list all employees who has a manager & has annual salary greater than 10000 & his empid is b/w 7850 to 7950 and who works as a analyst.

```
select ~~ename~~ *
from emp
where (mgr is not null) and ((sal x 12) > 10000)
and (empno between 7850 and 7950) and
(lower (job) = 'analyst');
```

2.) List all the names of employees who are not working as 'IT programer' or 'clerk' and sal is more than 3000

```
select ename
from emp
where ((lower (job) != 'it programer') or lower (job) !=
'clerk') and (sal > 3000);
```

earning = not null
not earning = null.

```
select *
from emp
where comm = null
```

o/p -> No row selected

=> A/c to above query we are getting wrong o/p because A/c to oracle RDBMS one null is not equal to another null (null != null) to compare one null with another null we should go IS operator

Defination of IS operator

IS operator is use to compare null values

Syntax: LHS-value IS Null
↓
C.N | expression

=> IS operator will return true if LHS value is null

=> IS operator will return false if LHS value is not null

Characteristics of null: (Predefined | inbuilt)

* Null is a reserved keyword in oracle RDBMS
* Null is not representing either '0' or space, null is representing as nothing (no data)
* Null will not consume any memory allocation.
* If we perform any arthematic operation with null the result will be null only.

e.g: select 10 + null
from dual;

o/p 10 + null

) WAD employee details only the employees those who

Ename	comm
A	100
B	
C	0
D	
E	200

from emp — ①
where comm is null — ②

100 is null → F
null is null → T
0 is null → F
null is null → T
200 is null → F

O/P

Ename	comm
B	
D	

2) WAD employee details only the employees those who are not reporting to any of the manager number

```
Select *
from emp
where mgr is null;
```

NOTE:

* when C.N is same but values are different Then it is better to go In operator
* when the I/P is in a form of range then it is better to go between operator.
* whenever we need to perform pattern matching/ wild card search ~~like oper~~ it is better to go like operator
* whenever we need to compare null values then it is better to go IS operator.

NOT operator:

Not operator is use to exculde the record
except/apart from/ignoring/rejecting

Symbol

cond ▷ O O/P

Syntax:

```
where NOT cond;
```

Truth table

cond	O/P

=> NOT operator will return true

```
select *
from emp
where not deptno = 30;
      or
where deptno != 30;
      or
where deptno <> 30;
```

<> → Not equal to

2) WAD employee details except employee those who are working as saleman.

```
Select *
from emp
where lower(job) != 'salesman';
```

3) WAD employee details except an employee turner

```
Select *
from emp
where lower(ename) != 'turner';
```

4) WAD employee details except an employee who is earning salary 3000

```
Select *
from emp
where not sal = 3000;
```

5) WAD employee details apart from an employee those who are hired on 23-dec-81

```
Select *
from emp
where not hiredate = '23-dec-81';
```

Combination of special operator and NOT operator

1) NOT In operator:

when LHS value should not be equal to any of the RHS value then we should go not in operator.

Syntax LHS value NOT in (value1, value2, value3.....)

C.N/express ② ①

2) NOT BETWEEN

Scanned by CamScanner

3) NOT like operator
when LHS value should not equal RHS single pattern then we should go not like operator
Syntax: where LHS value not like pattern / wildcard search
col/exp

4) IS NOT :
when LHS value is not a null value then we will go is not operator
Syntax: where LHS value is not null;
col/exp

1) WQD employee details except an employee working in department number 10, 20
Select *
from
where dept no not in(10, 20);

2) WQD employee details except an employee working as salesman/clerk
Select *
from emp
where lower(job) not in('salesman', 'clerk');

3) WQD employee details except manger no 7698, 7839 and their salary should be less than 5000
Select *
from emp
where (mgr not in (7698, 7839)) and (sal < 5000);

4) WQD employee name, salary, annual salary, except an employee smith, turner and they should be working in deptno 20, 30
Select ename, sal, sal×12 annsal
from emp
where (lower(ename) not in ('smith', 'turner')) and (dept no in (20, 30));

5) WQD except employee details only salary should not be in a range of 1000 to 5000

6) WQD employee name, manager no, annual salary only the annual salary should not be in a range 10000 to 20000 and they should be reporting to a manager no 7698
Select ename, mgr, sal×12 annsal
from emp
where (sal×12) not between 10000 and 20000) and (mgr = 7698);

7) WQD employee details only employee name should not consists of 'a' character
Select *
from emp
where lower(ename) not like '%a%';

8) WQD employee details except those who are hired in a month feb
Select *
from emp
where hiredate not like '%feb%';

9) WQD employee details except employee hired in a year 81
Select *
from emp
where hiredate not like '%81';

10) WQD employee name only employee name should begins with consonant (except vowels)
Select ename
from emp
where (lower(ename) not like 'a%') and (lower(ename) not like 'e%') and (lower(ename) not like 'i%') and (lower(ename) not like 'o%') and (lower(ename) not like 'u%');

NOTE:
=> whenever we are writing not like operator multiple times b/w not like operator we should write and operator. If we will make it or or.

Scanned by CamScanner

select *
from emp
where (hiredate not like '%feb%') and
(hiredate not like '%dec%');

12) WQD employee details those who are earning commission

select *
from emp
where comm is not null;

13) WQD employee details those who are earning sal

select *
from emp
where sal is not null;

14) WQD employee details only the employees those who are reporting some manager number

select *
from emp
where mgr is not null;

Escape character:

The process of converting special character into ordinary character then we will go escape character.
=> Escape character can be any character which is present in key board
=> 1 escape character will convert one spl character into ordinary character.

1) WQD employee details only employee name should consists of '_' as character.

select *
from emp
where like '%!_%'
escape '!';

2) WQD employee details only the employee name should consists of '%' as character.

select *
from emp
where ename like '%!-%' (or) where ename
escape '!';

4) WQD employee details emp name should consists of 2 underscore

select *
from emp
where ename like '%!_%!_%';
escape '!';

5) WQD employee details only employee name last 2nd character should be underscore

select *
from emp
where ename like '%!_-';
escape '!';

Scanned by CamScanner

Functions is a block of statements
It will get execute whenever it is called/invoke
Is called as functions.

e.g: void Addition() → function

```
{
int a=10, b=20, c;
c = a+b;
Printf("%d", c);
}
```

→ Block of statements/ set of statements

Functions are classified into 2 types

1) user defined function.
2) pre-defined / inbuit (or) buit in function.

User defined function

These are the functions which are defined (or) developed by user based on customer requirement is called as user defined function.

=> we can change / modify user defined function whenever requirement is changed.

=> SQL will not support user defined function but C, C++, Java, python, PLSQL etc will support

Predefined function

These are the function which are automatically present s/w / applicat^n whenever the s/w (or) application is installed is called as predefined function.

=> we cannot change / modify pre-defined function

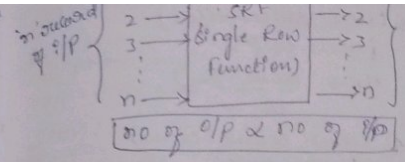
=> In SQL we will always use pre-defined funct^n

=> Predefined functions as classified into 2 types

1) Single Row function (SRF)
2) Multi Row function (MRF)

Single Row Function (SRF)

=> SRF will accept 'n' record as input and it will



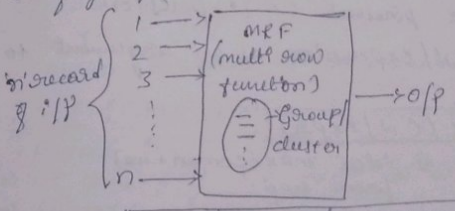
no of o/p & no of i/p

2) Multi-Row function (MRF) / Aggrigate function / Group function

=> MRF will accept 'n' record as i/p & it will create group / cluster and it will provide single record as o/p.

=> In MRF by default it will create single group & it will provide single record as o/p

=> In MRF no of o/p is directly proportional to no of groups (or) cluster



no of record & no of group/cluster

There are 5 MRF are present

1) max()
2) min()
3) count()
4) sum()
5) avg()

Count();

This function is used to count no of record given to it

=> we can pass

2021/5/18 21:20

Scanned by CamScanner

e.g.) select count(*) from emp;

o/p -> count(*)
14

select count(sal) from emp;

o/p -> count(sal)
14

3) select count(mgr) from emp;

o/p -> count(mgr)
13

4) select count(comm) from emp;

o/p -> count(comm)
4

5) select count(distinct sal) from emp

o/p -> count(distinct sal)
12

6) select count(sal*12) from emp

o/p -> count(sal*12)
14

Max(): This function is used to obtain maximum value present c.N/expression

=> we can pass c.N/expression as an argument to max funct^n

=> Syntax: $\boxed{\text{max (c.N/expr)}}$

e.g: 1) select max(sal) from emp;

o/p -> 5000

2) select max(comm+100) from emp

o/p -> 1500

NOTE:

-> We can pass string/character type of c.N as an argument to min & max funct^n it will give the o/p based on ASCII value. (A=65..... Z=91, a=97.... z=123)

e.g: 1) select max(ename) from emp

o/p -> WARD

min(): This funct^n is used to obtain minimum value present in c.N/expression.

=> we can pass c.N/expr as an argument to min funct^n

-> Syntax: $\boxed{\text{min(c.N /expr)}}$

3) select min(ename) from emp;

o/p -> ADAMS

SUM(): This function is used to obtain summation of all the values present in c.N/expression

-> we can pass c.N/expr as an argument to sum funct^n

-> Syntax $\boxed{\text{SUM (C.N/expr)}}$

e.g: select sum(comm) from emp;

o/p -> 2200

2) select sum(comm+100) from emp

o/p -> 2600

NOTE:

-> we cannot pass string/character type of c.N as an argument to sum funct^n & avg funct^n.

e.g: select sum(ename) from emp

o/p -> ~~INVERROR~~ invalid number

AVG(): This function is used to obtain avg of value present in c.N/expression

=> we can pass c.N/expr as an argument to avg funct^n

=> Syntax: $\boxed{\text{Avg (c.N/expr)}}$

formula

$$Avg = \frac{\text{sum of all the values}}{\text{Total no of values}}$$

e.g: 1) select avg(comm) from emp

o/p -> 550

2) select avg(sal*12) from emp

o/p = 24878.5714

3) select avg(ename) from emp

o/p -> invalid number

NOTE:

* all MRF will ignore null values.
* all MRF will accept duplicate values if we want to ignore ...

2021/5/18 21:21

Scanned by CamScanner

from emp

o/p → count(*) min(sal) max(comm)
15 800 1400

* All MRF will accept single argument.

e.g: select count(*), min(sal), max(comm, sal)
from emp;

o/p → Invalid no of argument.

* we cannot write combination of ordinal MRF in select statement. If we want to write then we should use Group by clause

e.g: select sal, min(sal)
from emp;

o/p: select sal, min(sal)
not a single-group group function.

* we cannot write MRF as a condition in where clause. If we want write then we should go Having clause

* MRF will execute group by group

1) WQD number of employees, min sal, maximum salary only the employees working in dept no 10, 20.

Select count(*), min(sal), max(sal)
from emp
where deptno in(10,20);

2) WQD number of employees, min annual sal, max-annual sal, min commission only the employees working as salesman, clerk.

Select count(*), min(sal×12) min_ansal, max(sal×12)
max_ansal, min(comm)
from emp
where lower(job) in('salesman', 'clerk');

3) WQD min comm, max comm. min newsal with 1000 increment, max newsal with 500 decrement only the mgr should be in range of 7600 - 7900

Select min(comm), max(comm), min(sal+1000), max
(sal-500) from emp

as a character

Select sum(sal), sum(comm), sum(sal×12), avg(sal)
from emp
where lower(job) like '%a%';

5) WQD recent hiredate, oldest hiredate only employee should be working in deptno 10, 20, 30

Select max(hiredate), min(hiredate)
from emp
where deptno in(10,20,30);

15-aug-1947 — oldest - min
31-mar-2021 → recent - max

6) WQD no of employees, oldest hiredate, min & max new sal with 30% increment only for employee no begins with '7'.

Select count(*), min(hiredate), min(sal×1.3), max(sal×1.3)
from emp
where empno like '7%';

7) WQD no of employees, total annual salary, total annual comm. only the employee name should begins with 'a' or 's' character.

Select count(*), sum(sal×12), sum(comm×12)
from emp
where (lower(ename) like 'a%') or (lower(ename) like 's%');

8) WQD no of employees, 1st employee no, last employee no only the employees those who are hired after a year so

Select count(*), min(empno), max(empno)
from emp
where hiredate >='01-jan-81';

* 9) WQD no of employees, recent hiredate & ~~oldest hiredate~~ after 1 year, oldest hiredate before 1 year. only the employees those who are earning salary.

Select count(*), max(hiredate+365), min(hiredate-365)
~~max(hiredate+365), min(hiredate-~~
from emp

2021/5/18 21:28

Scanned by CamScanner

'man' as a substring.

```
select count(*), min(sal-500), max(sal-500)
from emp
where lower(job) like '%man%';
```

Group by clause:

when to go group by clause?

* whenever we want to write combination of c.N & multi row function in select statement then we should go group by clause
* By default all multirow function will create single group and it will provide single record as o/p if we want multiple record as a o/p from MRF then we should create multiple group inorder to create multiple group we should go group by clause.

Defination of group by clause

It is used to group the record present in table based on reference

Syntax: select group by expression — (4)
└→ combination of c.N/expr/MRF
from table_name — (1)
where <cond> — (3)
group by reference — (2)
└→ c.N/expr
order by group-by-expression [ASC]/DESC; — (5)

=> Combination of c.N/expression, multirow function is called as group by expression.
=> c.N/expression present in group by clause is called as reference
=> whenever we are writing single c.N/reference in group by clause, group by clause will create main group
=> whenever we are writing multiple c.N/multiple reference

WQD

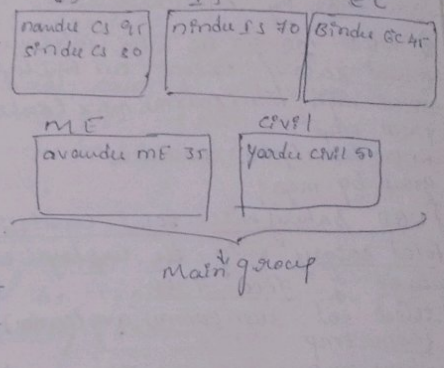
Student

Sname	branch	Per
nandu	CS	95
Nindu	SS	70
avandu	ME	35
Bindu	EC	45
sindu	CS	80
Yandu	civil	50

```
select branch, count(*) — (3)
from student — (1)
group by branch; — (2)
```

grp by expression

└→ Reference



O/P

branch	count(*)
CS	2
SS	1
ME	1
EC	1
civil	1

=> WQD branch wise min per, max per, except mechanical branch

```
select branch, min(per), max(per).
from student
where lower(branch)!='me'
group by branch;
```

3) WQD per wise no. of students, total per. only the student name should consists of 'a' character.

```
select per, count(*), sum(per)
from student
where lower(sname) like '%a%'
group by per;
```

4) WQD department wise no. of employees, min sal, max sal only those who are earning 4 digit salary

```
select deptno, count(*), min(sal), max(sal)
```

Scanned by CamScanner

of 7600 - 7900 and display the o/p ... order based on oldest hiredate.

```
select job, max(hiredate), min(hiredate)
from emp
where mgr between 7600 and 7900
group by job
order by min(hiredate) desc;
```

6) WQD manager no wise minimum annual sal, max annual salary except an employe working as president.

```
select mgr, min(sal*12), max(sal*12)
from emp
where lower(job)!='president'
group by mgr;
```

7) WQD salary wise total commission, avg. commission, total salary only the employee no should be in a range of 7000 - 7900

```
select sal, sum(comm), avg(comm), sum(sal)
from emp
where empno between 7000 and 7900
group by sal;
```

8) WQD annual salary wise no of employees, recent hiredate, oldest hiredate only the employees those who are hired in a year 81 & display the o/p in descending order based on no of employees.

```
select sal*12 annsal, count(*), max(hiredate), min(hiredate)
from emp
where hiredate like '%81'
group by sal*12
order by count(*) desc;
```

9) WQD commission wise min sal, total salary, new min-salary with 500 Rs increment only their designation should be salesman and they should working in deptno 30

```
select comm, min(sal), sum(sal), min(sal+500)
from emp
```

who are reporting to some mgr

```
select hiredate, count(*), min(empno), max(empno), max(comm), min(comm)
from emp
where mgr is not null
group by hiredate;
```

11) WQD new salary with 1000 increment wise, no of employees, recent & oldest hiredate after 6 month only the employee name should not begin with a character

```
select (sal+1000) newsal, count(*), max(hiredate+182), min(hiredate+182) from emp
where lower(ename) not like 'a%'
group by sal+1000;
```

12) WQD department wise no of employees, avg sal with 30% increment, avg comm with 100rs decrement only the employees name should be exactly 5 character & display the o/p in descending order based on new avg salary

```
select deptno, count(*), avg(sal*1.3), avg(comm-100)
from emp
where ~~lower~~ ename like '----'
group by deptno
order by avg(sal*1.3) desc;
```

Rules of Group by clause

R1: whatever the C.N present in select clause same C.N should be present in group by clause.

~~R2: whatever the C.N present in~~

```
select c1 MRF
from table_name
where <cond's>
group by c1
```

R2: whatever the C.N present in group by clause it may be (or) may not be present in select clause

Scanned by CamScanner

where < cond'n >
group by (C1)

R3: we cannot write MRF as a cond^n in where clause

```
select c1
from tablename
where <cond'n>, MRF ;
```

R4: whenever we are writting where clause & group by clause in a query first where clause will exicute based on the cond^n it will retain the record. The retained the records will be given to group by clause.

R5: Any clause is exicuting after group by clause it will exicute group by group not row by row (or) record by record.

Sub grouping:

whenever we are writting multiple c.n's / reference in group by clause, group by clause will create sub group inside the main group it is called as sub group (or)

A group created inside another group is called as sub grouping.

=> The 1st c.n present in group by clause will create main group and the next c.n/2nd c.n present in group by clause will create sub group inside the main group

=> whenever we are writting multiple ~~clause~~ c.n in group by clause, group by clause will pass both main group and sub group as an argument to select clause.

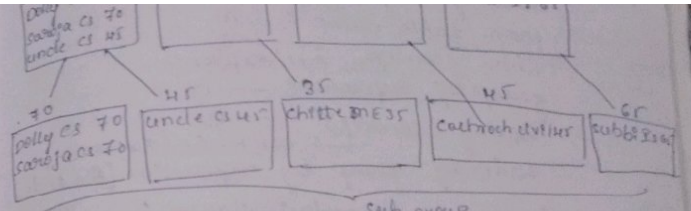
1) WQD branch wise under each percentage no of students

Student

Sname	branch	per
Dolly	CS	70
chitte	ME	35
Cockroch	civil	45
Saroja	CS	70
Subbi	IS	65
uncle	CS	45

```
select branch per count(*) — (1)
from emp — (2)
group by branch per ; — (3)
```

branch → main grp; per → sub grp



o/p :

branch	per	Count(*)
CS	70	2
CS	45	1
ME	35	1
civil	45	1
IS	65	1

2) WQD department wise under each designation min sal, max sal only those who are earning sal more than 500

```
select deptno, job, min(sal), max(sal)
from emp
where sal > 500
group by deptno, job;
```

3) WQD salary wise under each designation, under each manager no. no of employees only the employees should be working in a department no 10, 20, 30

```
select sal, job, mgr, count(*)
from emp
where deptno in (10,20,30)
group by sal, job, mgr;
```

Having Clause:

Having clause is used to filter grouped records

=> Having clause will exicute after group by clause

=> Having clause is mainly used to write MRF as a ~~function~~ condition.

2021/5/18 21:29

Scanned by CamScanner

select group by ...
from table_name —— (1)
where condition, col/expression —— (2)
group by parameters —— (3)
having condition —— (4)
order by group by expression [asc/desc]; (5)

=> Having clause is a dependent clause bcz we cannot write having clause if there is no group by clause

=> Having clause will calculate group by group

1) WAD branch wise no of student, min per, max per only min per should be more than 45

Student

Sname	branch	per
Satya	EC	70
Sunya	CS	45
Shalini	IS	80
Charath	ME	35
Kalin	Civil	45
Shreya	CS	50

```
select branch, count(*),
       min(per), max(per)
from student
group by branch
having min(per)>45;
```

→ main group

EC ✓ | CS | IS ✓ | ME | Civil

Satya EC 70	~~Sunya CS 45~~ ~~Shreya CS 50~~	Shalini IS 80	~~Charath ME 35~~	~~Kalin Civil 45~~

70>45→T | 45>45→F | 80>45→T | 35>45→F | 45>45→F

O/P

branch	count(*)	min(per)	max(per)
EC	1	70	70
IS	1	80	80

2) WAD branch wise no of students, except mechanical branch and atleast 2 student should be studing in each branch

select ...

3) WAD per wise no of student, avg per only student name should contain of 5 character & avg per should be in a range of 35-45

```
select per, count(*), avg(per)
from student
where lower(sname) like 'x%y';
group by per
having avg(per) between 35 and 45;
```

4) WAD department wise no of employees, min salary, max salary only employee should be earning 4 digit salary and min salary should be more than 500

```
select deptno, count(*), min(sal), max(sal)
from emp
where sal like '____'
group by deptno
having min(sal) > 500;
```

5) WAD designation wise no of employees only employee should be hired in a year 81 to 87 and atleast 2 employees should be working in each designation

```
select job, count(*)
from emp
where hiredate between '01-jan-81' and '31-dec-87'
group by job
having count(*)>=2;
```

6) WAD department wise, no of employees, min & max annual sal, except a deptno 30 and the min annual sal should not be less than 5000

```
select deptno, count(*), min(sal*12), max(sal*12)
from emp
where deptno
```

2021/5/18 21:20

Scanned by CamScanner

from emp
group by deptno
having (deptno != 30) and (min(sal*12) >= 5000);

=> If we want to write c.N/expr as a condn in having clause the same c.N/expr should be present in group by clause.

NOTE: If a condn wrt c.N then it is better to go where clause

2) If a condn w.r.t nRF then it is better to go having clause.

Can we write c.N as a condn in having clause?
ans: YES but the same c.N should be present in group by clause.

7) WAD mgr wise no of employees recent hiredate, oldest hiredate except those who are working as salesman and total salary should be less than total commission

Select mgr, count(*), max(hiredate), min(hiredate)
from emp
where lower(job) != 'salesman'
group by mgr
having sum(sal) < sum(comm);

8) WAD annual salary wise no of employees only employee should be working for a deptno 10, 20, 30 and avg salary should not be in a range of 100-500 and disply the o/p ascending order based on no of employees

Select sal*12, count(*)
from emp
where deptno in (10, 20, 30)
group by sal*12
having avg(sal) not between 100 and 500
order by

range should be in a year 87 ... hiredate

select sal, count(*), max(hiredate+180), min(hiredate+180)
from emp
where mgr between 7600 and 7900
group by sal
having max(hiredate) like '%87';

10) WAD commission wise total salary, average sal only the employee should be working as salesman and min commission should be less than 5000

Select comm, sum(sal), avg(sal)
from emp
where lower(job) = 'salesman'
group by comm
having min(comm) < 5000;

*11) WAD dept wise no of employees where atleast 2 clerk should be working in each deptno

Select deptno, count(*)
from emp
where lower(job) = 'clerk'
group by deptno
having count(*) >= 2;

12) WAD designation wise min commission, max com total comm. only employee name should begin with character 'a' or 's' and the oldest hiredate should be in a year 80 to 82

Select job, min(comm), max(comm), sum(comm)
from emp
where lower(ename) like 'a%' or lower(ename) like 's%'
group by job
having min(comm) ... and

2021/5/18 21:29

Scanned by CamScanner

only the employees, should be reporting to some manager no and the salary should be less than max comm

```
Select deptno, sal, max(hiredate), min(hiredate)
from emp
where mgr is not null
group by dept no, sal
having sal < max(comm);
```

sal c.N is not present in group by clause it will not execute. If we only write sal in group by clause o/p has no meaning hence sal should write in select clause

*
14) WQD mgr wise no of employees min sal, max salary, only the employee no should begin with a digit '7' and the max salary should be less than comm and disply the o/p in descending order based on min salary

```
Select mgr, count(*), min(sal), max(sal)
from emp
where emp no like '7%'
group by mgr, comm
having max(sal) < comm
order by min(sal) desc;
```

Rules of having clause:

R1: whenever we want to write c.N as cond^n in having clause same c.N should be present in group by clause

```
Select C1, MRF
from table_name
where <cond>
group by C1, C2
having MRF and C2
```

R2: whatever the MRF present in having clause it may be or maynot be present in select clause.

What is the difference b/w where & having clause

→ It will execute row by record	→ It will execute group by group
→ we cannot write MRF as a cond^n	→ we can write MRF as a cond^n
→ It will filter ungrouped records	→ It will filter grouped records
→ we can write a where clause if there is no group by clause also [independent clause]	→ we cannot write having clause if there is no group by clause [dependent clause]
→ It will execute before group by clause	→ It will execute after group by clause.
→ we can write any c.N as a cond^n in where clause	→ we can write c.N as a cond^n in having clause but the same c.N should be present in group by clause

Sub Query

⇒ Query: A query is a set of statement followed by multiple cond^{n}s which is used to interact with d.B is called as a query. (or)

A query which begins with select statement and followed by multiple cond^{n}s and terminating by semicolon(;) is called as a query.

e.g: select *|c.N
from table1
where <cond^{n}s>;

Defination of Sub Query

A query inside the another query (or)

A query written inside another query is called as a sub query.

⇒ sub query can be written in 2 clause

1) from clause
2) where clause.

Syntax of sub query in where clause

```
select *|c.N
from table1 → outer query
where <cond$^n$s> (select *|c.N → inner query/outer query
                  from table1 → inner query
                  where ...
```

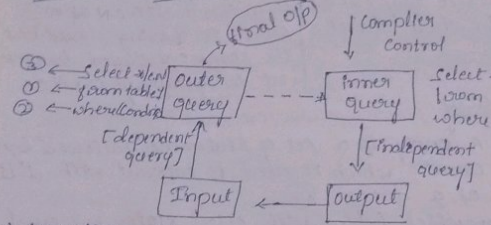
2021/5/18 21:29

Scanned by CamScanner

→ upto 255 inner queries can write

NOTE:
→ A query which is written innermost part called as inner query
⇒ A query which is written outermost part is called as outer query
⇒ Generally a sub query is a combination of inner query and outer query.
⇒ we can write upto 255 inner query so subquery is also known as nested subquery

WORKING MECHANISM OF SUB QUERY



⇒ complier will give control to inner query, inner query will execute based on the condn, ~~base~~ it will prepare the o/p
⇒ The o/p of inner query will pass as an input to outer query
⇒ outer query will execute based on the condition and the input taken from inner query it will prepare the final output.
⇒ outer query is a dependent query and inner query is a independent query.

Types of subquery
There are 2 types of subquery
1) Single row sub query
2) multi row sub query

Single row subquery:
→ If a inner query generates exactly single record as o/p then it is called as single row sub query.
→ If inner query generates exactly one record as o/p we can compare by using all relational operator (<, >, <=, >=, =, !=) and in operator.

multi row sub query
→ If inner query generates more than 1 record as o/p then it is called as multirow subquery
→ If inner query generates more than one record as o/p, we cannot compare by using relational operator so we should go IN, NOT IN, ANY, ALL operator.

when to go sub query (or) what are the cases of sub query
case 1: To determine unknown values
case 2: Linking (or) mapping multiple tables by utilizing common c.N between the tables
case 3: mapping values present in same table but in different rows (nth max and nth min value)

Case 1: To determine unknown values
1) WAD employee details only the department no should be same as allen department no → uk (sq)
(condn)

Emp

Ename	job	deptno
Smith	clerk	20
blake	salesman	10
allen	clerk	20
king	President	30
akash	salesman	10

Select * ← ③
from emp ← ① ⑤ "20" ← ④
where deptno in (select deptno from emp ← ① where (lower(ename) = 'allen'));

deptno = 20
20 = 20 → T
10 = 20 → F
20 = 20 → T
30 = 20 → F
10 = 20 → F

② smith = allen → F
blake = allen → F
allen = allen → T
king = allen → F
akash = allen → F

o/p

Ename	job	deptno
smith	clerk	20
allen	clerk	20

2021/5/18 21:30

Scanned by CamScanner

NOTE:
1) what they are asking you to disply should be written 'outerquery'.
2) To determine unknown value should be written in 'inner query'

1) WQD employee details only employee salary should be less than king salary.
dept (e.g) → cond^n; king salary → u.k (e.g)

```
Select *
from emp
where sal < (select sal
             from emp
             where lower(ename) = 'king');
```

2) WQD employee name, designation, annual salary only designation should be same as scott designation.
cond^n; scott designation → u.k (e.g)

```
Select ename, job, sal*12 annsal
from emp
where job = (select job
             from emp
             where lower(ename) = 'scott');
```

(or)

```
Select ename, job, sal*12 annsal
from emp
where lower(job) in (select lower(job)
                     from emp
                     where lower(ename) = 'scott');
```

3) WQD employee details only employee should be hired after smith hiredate.
cond^n; smith hiredate → u.k (e.g)

```
Select *
from emp
where hiredate > (select hiredate
                  from emp
                  where lower(ename) = 'smith');
```

4) WQD employee name, salary, manager number only the supporting manager no should be same as turner reporting manager no.
u.k

```
Select ename, sal, mgr
from emp
where mgr
from emp
where lower(ename) = 'turner');
```

5) WQD employee details only the date of joining should be same as miller date of joining → e.g

```
Select *
from emp
where hiredate in (select hiredate
                   from emp
                   where lower(ename) = 'miller');
```

6) WQD employee name, employee no, department no, annual salary only annual salary should be less than scott annual salary and disply the o/p in descending order based on employee no

```
Select ename, empno, deptno, sal*12 annsal
from emp
where (sal*12) < (select sal*12
                  from emp
                  where lower(ename) = 'scott')
order by empno desc;
```

7) WQD employee details only commission should be more than smith salary → e.g
cond^n

```
Select *
from emp
where comm > (select sal
              from emp
              where lower(ename) = 'smith');
```

NOTE: 1) whatever the column present in outerquery where clause it should be logically related with inner query select clause column name.

e.g: where sal (select comm ✓
where empno (select mgr ✓
where sal (select job ✗ [there is no meaning to compare]

8) WQD employee details only employee should be earning 4 digit salary and manager no should be same as blake employee no
u.k (e.g)

```
Select *
from emp
```

2021/5/18 21:30

Scanned by CamScanner

where (sal like '----') and mgr in (select empno
from emp
where lower(ename)
= 'blake');

10) WQD employee name, employee no, designation deptno except an employee those who are working as president and their salary should be same as (James) (salary)

```
select ename, empno, job, deptno
from emp
where (lower(job) != 'president') and (sal in (select sal
from emp
where (lower(ename) = 'james')));
```

11) WQD employee details only employee should be working in a department no 10, 20, 30 & employee no should be more than (miller mgr no)

```
select *
from emp
where (deptno in (10,20,30)) and (empno > (select mgr
from emp
where (lower(ename) = 'miller'));
```

12) WQD employee details only the department no. should not be same as turner deptno and commission should be more than blacke commission.

```
select *
from emp
where(deptno not in (select deptno
from emp
where lower(ename) = 'turner')) and (comm > (select comm
from emp
where lower(ename) = 'blake'));
```

13) WQD employee details only employee no should be same as smith manager no and designation should not be same as king designation.

```
select *
from emp
where empno in (select mgr
from emp
where lower(ename) = 'smith') and job not in (select job
from emp
where lower(ename) = 'king');
```

14) WQD employee details only salary should be greater than highest commission.

~~select * ename, job, sal, comm, hiredate, empno, deptno~~
~~from emp~~
~~group by ename~~

```
select *
from emp
where sal > (select max(comm)
from emp);
```

15) WQD employee details only annual salary should be less than miller annual commission and department no should not be same as smith, blake department no.

```
select *
from emp
where (sal*12) < (select comm*12
from emp
where lower(ename) = 'miller') and
deptno not in (select deptno
from emp
where lower(ename) in ('smith', 'blake'));
```

16) WQD employeename, designation, commission, only manager should be same as smith, james employee no and employee should not be hired in a year 82.

```
select ename, job, comm
from emp
where mgr in (select empno
from emp
where lower(ename) in ('smith', 'james')) and (hiredate
not like '%82');
```

17) WQD employee details only the designation should be same as (blake, king designation) and employee name should be begin with a (or) s character.

```
select *
from emp
where job in (select job
```

2021/5/18 21:30

Scanned by CamScanner

~~lower~~ where lower(ename) in ('blake', 'king')) and ((lower(ename) like 'a%') ~~and~~ or (lower(ename) like 's%'));

18) WQD employee details only employee should be hired after smith hiredate and commission should not be same as king salary.

```
Select *
from emp
where hiredate > (select hiredate
from emp
where lower(ename) = 'smith') and (comm not in
(select sal
from emp
where lower(ename) = 'king'));
```

19) WQD employeename, employee no, new salary with 1000 is increment only new salary should be same as smith, turner commission. and employee no should be in a range of 7000 to 7900.

```
Select ename, empno, (sal+1000) newsal
from emp
where (sal+1000) in (select comm
from emp
where lower(ename) in ('smith', 'turner'))and
(empno between 7000 and 7900);
```

20) WQD employee details only department no should be same as ford department no. and employee no should be same as turner mgr no. and display the o/p in descending order based on employee no.

```
Select *
from emp
where deptno in (select deptno
from emp
where lower(ename) = 'ford') and (empno in
(select mgr
from emp
where lower(ename) = 'turner'))
orderby empno desc;
```

21) WQD no of employees, min salary, max salary only annual salary should be more than king annual salary.

```
Select count(*), min(sal), max(sal)
from emp
where (sal*12) > (select (sal*12)
from emp
where lower(ename) = 'king');
```

22) WQD no of employees, second hiredate, oldest hiredate only employee should be earning some salary and they should be hired before james hiredate.

```
Select count(*), max(hiredate), min(hiredate)
from emp
where (sal is not null) and (hiredate < (select
hiredate
from emp
where lower(ename) = 'james'));
```

23) WQD employee details. only employee should be hired before recent hiredate

```
Select * from emp
where hiredate < (select max(hiredate)
from emp);
```

24) WQD employee details of 1st highest salary

```
Select * from emp
where sal in (select max(sal) / sal = (select max(sal)
from emp);
```

Preferable bcz MRF will give always single o/p

25) WQD employee name, employee no designation, date of joining of 1st oldest hiredate.

```
Select ename, empno, job, hiredate
from emp
where hiredate = (select min(hiredate)
from emp);
```

= is preferable bcz MRF will always give

2021/5/18 21:30

Scanned by CamScanner

26) WQD departmentwise no of employees, min annual salary, max annual salary only employee no should be more than smith mgr no.

```
Select deptno, count(*), min(sal*12), max(sal*12)
from emp
where empno > (select mgr
 from emp
 where lower(ename) = 'smith')
 group by deptno;
```

27) WQD designation wise, recent hiredate, oldest hiredate only commission should be less than turner salary and employee should be earning some commission.

```
select job, max(hiredate), min(hiredate)
from emp
where comm < (select sal
 from emp
 where lower(ename) = 'turner') and (comm is
   not null) group by job;
```

28) WQD mgr wise no of employees total salary, total comm, avg salary, avg commission only the department no should be same as allen, blake deptno. and avg salary should be in a range of 100 to 5000.

```
Select mgr, count(*), sum(sal), sum(comm), avg
(sal), avg(comm)
from emp
where deptno in (select deptno
 from emp
 where lower(ename) in ('allen', 'blake'))
 group by mgr
 having avg(sal) between 100 and 5000;
```

29) WQD department wise no of employees only designat should be same as (miller designation) and atleast 2 clerk should be working in each department no.

```
select deptno, count(*)
from emp
where lower(job) in (select job
 from emp
```

```
 group by deptno
 having count(*) >= 2;
```

30) WQD employeename, employee no. department no only department no should be same as the employees those who are working as salesman department no.

```
select ename, empno, deptno
from emp
where deptno in (select deptno
 from emp
 where lower(job) = 'salesman');
```

Sub-Query Case III: Linking multiple tables by utilizing common col b/w the tables.

1) WQD employee details only employee should be working in sales department. displ(emp) dept

Emp cond(emp)

Ename	job	deptno
smith	TE	30
turner	DEV	20
blake	TE	30
akash	TL	10
miller	DL	20

dept

deptno	dname	loc
10	Accounting	New York
20	Research	Dallas
30	Sales	chicago
40	operation	Boston

```
Select *  ---> (3)
from emp  ---> (1) --> (2)
where deptno in (select deptno  ---> (3)
                 from dept  ---> (1)
                 where lower(dname) = 'sales';  ---> (2)
```

outer query

"30" → inner query

deptno = 30

30 = 30 → T
20 = 30 → F
30 = 30 → T
10 = 30 → F
20 = 30 → F

accounting = sales → F
research = sales → F
sales = sales → T
operation = sales → F

o/p

Ename	job	deptno
smith	TE	30
blake	TE	30

2021/5/18 21:30

Scanned by CamScanner

2) WQD employee name, designation, only the location should consists of New' as a substring.
cond(dept)

```
select ename, job
from emp
where deptno in (select deptno
                 from dept
                 where lower(loc) like '%new%');
```

3) WQD department details only the employees those who are working as clerk.
disp(dept)
cond(emp) — deptno

```
select *
from dept
where deptno in (select deptno
                 from emp
                 where lower(job) = 'clerk');
```

4) WQD departmentname, only the mgr no should be 7698, 7839
disp(dept) — deptno — cond(emp)

```
select dname
from dept
where deptno in (select deptno
                 from emp
                 where mgr in (7698, 7839));
```

5) WQD employee details only department name should consists of a or s character.
cond (dept)

```
select *
from emp
where deptno in (select deptno
from dept
where (lower(dname) like '%a%') or (lower(dname)
  like '%s%'));
```

6) WQD department name, location only the employees those who are earning 4 digit salary.
disp(dept)
cond(emp)

```
select dname, loc
from dept
where deptno in (select deptno
from emp
where sal like '----');
```

7) WQD employee name, employee no, designation only the mgr should be in a range of 7600 to 7900 and employee should be working in a deptment research, accounting
disp(emp) condition
cond(dept) — deptno

```
select ename, empno, job
from emp
where (mgr between 7600 and 7900) and deptno in
  (select deptno
   from dept
   where lower(dname) in ('research', 'accounting'));
```

8) WQD employee details only employee should be earning salary in a range of 1000 to 5000 and they should be located in dalls, chikago location and display the o/p in descending order based on employee no.

```
select *
from emp
where (sal between 1000 and 5000) and (deptno in (select
  deptno
  from dept
  where lower(loc) in (dalls, chikago)))
  order by empno desc;
```

9) WQD department details only departname should not consists of 'a' character and employee name should consists of 2 consigative 'l' character

```
select *
from dept
where (lower(dname) not like '%a%') and (deptno
in (select deptno
    from emp
    where lower(ename) like '%ll%'));
```

10) WQD department name, location only employee should not be working in a location boston and employee should be hired in a year 81 to 87.

```
select dname, loc
from dept
where (lower(loc) != 'boston') and (deptno in
(select deptno
from emp
where hiredate blw '01-jan-81' and '31-dec-87'));
```

2021/5/18 21:31

Scanned by CamScanner

11) WQD employee details only designation should be same as smith designation and their location should begin with 'new' as a substring.

emp.deptno = dept.deptno

emp.job = dept.loc

```
Select *
from emp
where lower job in (select job
from emp
where lower(ename) = 'smith') and (deptno in (select deptno
from dept
where lower(loc) like 'new%'));
```

12) WQD employee details only sal should be less than miller commission and employee should not be working in a department operation

```
Select *
from emp
where sal < (select comm
from emp
where lower(ename) = 'miller') and (deptno in
(select deptno
from dept
where lower(dname) != 'operation'));
```

13) WQD employee name, employee no. department no only department no should be same as allen, jones employment no and location last 3rd character should be 'S'

```
Select ename, empno, deptno
from emp
where deptno in (select deptno
from emp
where lower(ename) in ('allen','jones')) and
deptno in (select deptno
from dept
where lower(loc) like '%S__');
```

14) WQD employee details only employee should be coming some commission and annual salary should be less than (jones annual commission) and employee should be located in newyork, chikago location.

```
select *
from emp
where (comm is not null) and ((sal*12) < (select (comm*12)
from emp
where lower(ename) = 'jones') and (deptno in
(select deptno
from dept
where lower(loc) in ('newyork', 'chikago'));
```

15) WQD department details only department name should consist of 2 'c' character. and salary should be same as (king, martin salary)

u.k → case1

emp → mgr2

```
Select *
from dept
where (lower(dname) like '%c%c%') and
deptno in (select deptno
from emp
~~where lower(ename) in ('king', 'martin'));~~
where sal in (select sal
from emp
where lower(ename) in ('king','martin')));
```

16) WQD department name, location only the manager no should be same as allen manager no

```
select dname, loc
from dept
where deptno in (select deptno
from emp
where mgr in (select mgr
from emp
where lower(ename) = 'allen'));
```

17) WQD department details location should not consist of 'a' character and manager no should be in a range of 7000 to 7900 and date of joining should not be same as king date of joining.

u.k

```
select *
from dept
where (lower(loc) not like '%a%') and (mgr between
7000 and 7900) and (hiredate not in (select
hiredate
from
```

2021/5/18 21:31

Scanned by CamScanner

deptno in (select deptno
from emp
where (mgr between 7000 and 7900) and
(hiredate not in (select hiredate
from emp
where lower(ename) = 'king')));

18) WAD department name, location, only comission should be same as turner comission and salary should be less than highest comission

select dname, loc
from dept
where deptno in (select deptno
from emp
where comm in (select comm
from emp
where lower(ename) = 'turner')) and sal <
(select max(comm)
from emp));

19) WAD department details only location should begin with 's' character and employee name should be exactly 5 character and employee no should be more than 'allen' manager no.

select * from dept
where (lower(loc) like 's%') and (deptno in (select deptno
from emp
where (lower(ename) like '_____')) and empno >
(select mgr
from emp
where lower(ename) = 'allen')));

20) WAD department details of 1st highest salary.

select *
from dept
where deptno in (select deptno
from emp
where sal = (select max(sal)
from emp));

21) WAD department name, location of 1st max annual salary

select dname, loc
from dept
where deptno in (select deptno
from emp
where (sal×12) in (select max(sal×12)
from emp));

22) WAD department name, location where atleast 2 clerk should be working in a department

select dname, loc
from dept
where deptno in (select deptno
from emp
where lower(job) = 'clerk'
group by deptno
having count(*) >= 2);

23) WAD department details only in each department some employee should be working → atleast 1 employee should be working

select *
from dept
where deptno in (select deptno
from emp
where mgr is not null);

(or)

(select deptno
from emp
group by deptno
having count(*) >= 1);

24) WAD no. employee ~~details~~ min salary, max salary only employee should be working in a department accounting, research.

select count(*), min(sal), max(sal)
from emp
where deptno in (select deptno
from dept
where lower(dname) in ('accounting', 'research'));

25) WAD mgr wise no of employees recent hiredate, oldest hiredate only employee should be working for a department 10, 20, 30 & they should be located in chikago.

select mgr, count(*), max(hiredate), min(hiredate)
from emp
where (deptno in (10, 20, 30)) and deptno in

2021/5/18 21:32

Scanned by CamScanner

(select deptno
from dept
where lower(loc) = 'chicago')
group by mgr;

26) WAD designation wise, no of employees min annual salary, max annual salary only the department name should be exactly 5 charecter & recent hiredate should be in a year 80

```
select job, count(*), min(sal*12), max(sal*12)
from emp
where deptno in (select deptno
from dept
where dname like '_____')
group by job
having max(hiredate) like '%80';
```

27) WAD annual salary wise no of employees only employee no should be more than smith employee no and their location last second charecter should be 'S' and atleast 1 employee should be present in each annual salary and disply the o/p in descending order based on annual salary.

```
select (sal*12) annsal, count(*)
from emp
where (empno > (select empno
from emp
where (lower(ename) = 'smith')) and (deptno in
(select deptno
from dept
where lower(loc) like '%s_'))
group by (sal*12)
having count(*) >= 1
order by ann_sal desc;
```

Drawback of case-2:

=> In subquery case 2 we can link multiple tables but we cannot fetch data from all the table.

Lab - Query

Case 3.1 : To determine m^{th} max and n^{th} min value

1) WAD 2nd maximum salary

emp

salary
Sal
3000
4000
8000
9000
2000

All sal < 1st max sal → 2nd max

max < min >

```
select max(sal) ---- 2nd max
from emp ---- 1st max
where sal < (select max(sal) ---- 1st max
from emp); ---- (1)
```

sal < 9000
3000 < 9000 - T
4000 < 9000 - T
8000 < 9000 - T
9000 < 9000 - F
2000 < 9000 - T

op

max (sal)
8000

2) WAD 3rd maximum salary

```
select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp));
```

how to modify the query to select

1st max/min -> 0 subquery
2nd max/min -> 1 subquery
3rd max/min -> 2 subquery
4th max/min -> 3 subquery
⋮
nth max/min -> (n-1) subquery
should be write
(or)
1st max/min -> 1 MRP
2nd max/min -> 2 MRP
3rd max/min -> 3 MRP
4th max/min -> 4 MRP
⋮
nth max/min -> n MRP
should be present in query

3) WAD 2nd min salary

```
select min(sal)
from emp
where sal > (select min(sal)
from emp);
```

4) WAD 3rd oldest hiredate

```
select min(hiredate)
from emp
where hiredate > (select min(hiredate)
from emp
where hiredate > (select min(hiredate)
from emp));
```

5) WAD 3rd recent hiredate

```
select max(hiredate)
```

2021/5/18 21:33

Scanned by CamScanner

from emp
where hiredate < (select max(hiredate)
from emp
where hiredate < (select max(hiredate)
from emp));

6) WQD 2nd min annual salary
select min (sal×12)
from emp
where (sal×12) > (select min(sal×12)
from emp);

7) WQD 4th max annual salary
select max (sal×12)
from emp
where (sal×12) < (select max (sal×12)
from emp
where (sal×12) < (select max (sal×12)
from emp
where (sal×12) < (select max (sal×12)
from emp))));

8) WQD 2nd min comission
select min (comm)
from emp
where comm > (select min(comm)
from emp));

9) WQD 3rd max comission.
select max(comm)
from emp
where comm < (select max (comm)
from emp
where comm < (select max (comm)
from emp));

10) WQD 3rd min mgr number.
select min (mgr)
from emp
where mgr > (select min (mgr)
from emp
where mgr > (select min (mgr)
from emp));

11) WQD 4th max new salary with 1000 Rs increment
select max (sal+1000)
from emp
where (sal+1000) < (select max (sal+1000)
from emp
where (sal+1000) < (select max (sal+1000)
from emp
where (sal+1000) < (select max (sal+1000)
from emp))));

12) WQD 2nd max employee no
select max (empno)
from emp
where empno < (select max (empno)
from emp);

Drawback of case 3.1
1. Whenever we want to determine nth max & nth min value we should write (n-1) innerquery so query length will increase & performance will get decreased.

13) WQD employee name, designation of 3rd max salary
select ename, job
from emp
where sal = (select max (sal)
from emp
where sal < (select max (sal)
from emp));

14) WQD employee details of 3rd oldest hiredate
select *
from emp
where hiredate = (select min (hiredate)
from emp
where hiredate > (select min (hiredate)
from emp
where hiredate > (select min(hiredate)
from emp))));

15) WQD department details of 2nd min annual dep-salary
select *
from dept
where deptno in (select deptno
from emp

2021/5/18 21:33

Scanned by CamScanner

```
where (sal*12) = (select min(sal*12)
  from emp
  where (sal*12) > (select min (sal*12)
  from emp)));
```

16) WQD department name, location of 4th min commission

```
Select dname, loc
from dept
where deptno in (Select deptno
from emp
where comm = (Select min(comm)
from emp
where comm > (select min (comm)
from emp
where comm > (select min (comm)
from emp.
where comm > (select min (comm)
from emp)))));
```

17) WQD department name of 3rd oldest hiredate.

```
select dname from dept
where deptno in (Select deptno
from emp
where hiredate = (select min (hiredate)
from emp
where hiredate > (select min(hiredate)
from emp
where hiredate > (select min (hiredate)
from emp))));
```

18) WQD employee name, employee no, designation of 2nd min mgr no

```
Select ename, empno, job
from emp
where mgr = (Select min(mgr)
from emp
where mgr > (select min(mgr)
from emp));
```

*
19) WQD employee details only the salary should be less than 3rd min commission.

```
Select *
from emp
where sal < (select min(comm)
from emp
where comm > (Select min (comm)
from emp
where comm > (Select min (comm)
from emp))));
```

20) WQD department details only employee should be hired after 3rd recent hiredate

```
Select *
from dept
where deptno in (Select deptno
from emp
where hiredate > (Select max (hiredate)
from emp
where hiredate < (select max (hiredate)
from emp
where hiredate < (select max (hiredate)
from emp))));
```

Case 3.2: Mapping values present in same table but in different rows

1 set of questions

disply - manager
condn - name

empno = mgr(ename)
O.g I.g

1) WQD manager name of an employee mr. keerthi sir

disp emp

Empno	Ename	Job	mgr
01	Mr. girish sir	CEO	
02	Mr. Pradeep sir	VP	1
03	Mr. Nisha mam	Verbal	2
04	Mr. keerthi sir	Sr. QA	2
05	Mr. Pavan sir	QA	2

```
Select ename mgr_name —(5)
from emp —(1) —(2) —(3)
where empno in (Select mgr) —(3)
from emp —(4)
where lower(ename) = 'mr. keerthi sir'; (2)
```

2021/5/18 21:33

Scanned by CamScanner

empno = Q
1 = Q → P
2 = 2 → P
3 = 2 → P
4 = 2 → P
5 = 2 → P

mr. Prabhakar str = mr. keerthi str → P
mr. Pradeep str = mr. keerthi str → P
mr. Nitha mam = mr. keerthi str → P
mr. keerthi str = mr. keerthi str → P
mr. Pavan str = mr. keerthi str → P

o/p: mgr.name
mr. Pradeep str

2) WQD manager name of an employee 'smith'
dep (mgr.name) cond (ename)

```
select ename mgr.name
from emp
where empno in (select mgr
                from emp
                where lower(ename) = 'smith');
```

3) WQD manager name, manager designation manager sal of an employee turner.
dep dep cond

```
select ename mgr.name, job mgr.job, sal mgr_sal
from emp
where empno = (select mgr
               from emp
               where lower(ename) = 'turner');
```

4) WQD manger details of an employee miller
dep cond

```
select *
from emp
where empno = (select mgr
               from emp
               where lower(ename) = 'miller');
```

5) WQD manager name, manager designation manager department of an employee blake, james.
dep cond

```
select ename mgr.name, job, deptno
from emp
where empno in (select mgr
                from emp
                where lower(ename) in ('blake','james'));
```

6) WQD depart name, location of blake manager.
dep [dept table] cond [emp table]

deptno ← common c.N
deptno → case 3.1
case 2 → + ← case 3.1

```
select dname, loc
from dept
where deptno in (select deptno
                 from emp
                 where empno = (select mgr
                                from emp
                                where lower(ename) = 'blake'));
```

7) WQD department details of scott, adams manager
dep (dept) cond (emp)

deptno → case 2 → + ← case 3.1

```
select *
from dept
where deptno in (select deptno
                 from emp
                 where empno in (select mgr
                                 from emp
                                 where lower(ename) in ('scott','adams'));
```

8) WQD manager manager of an employee 'mr. Pavan str'
dep cond

```
select ename mgr.name
from emp
where empno in (select mgr
                from emp
                where empno in (select mgr
                                from emp
                                where lower(ename) = 'mr. Pavan str'));
```

9) WQD manager, manager of an employee miller
dep dep cond

```
select ename mgr.name
from emp
where empno in (select mgr
                from emp
                where empno in (select mgr
                                from emp
                                where lower(ename) = 'miller'));
```

10) WQD manager manger manager of an employee adams
dep cond

```
select ename mgr.name
from emp
where empno in (select mgr
                from emp
                where empno in (select mgr
                                from emp
                                where empno in (select mgr
                                                from emp
                                                where lower(ename) = 'adams'))));
```

2021/5/18 21:33

Scanned by CamScanner

11) WAD department details of James manager manager
display cond

```
select *
from dept
where deptno in (select deptno
from emp
where empno in (select mgr
from emp
where empno in (select mgr
from emp
where lower(ename) = 'james')));
```

12) WAD manager name, manager salary, manager annual sal of an employee clark, ford
display cond

```
select ename mgr.name, sal, ~~an~~ (sal x 12) annsal
from emp
where empno in (select mgr
from emp
where lower(ename) in ('clark', 'ford'));
```

13) WAD employee name those who are reporting to a manager mr. Peradup sir
cond

```
select ename
from emp
where mgr in(select empno
            from emp
            where lower(ename) = 'mr. peradup sir');
```

display mgr	= empno (mgr-name) (cond)
o.q	s.q

14) WAD employee name those who are reporting to a manager blake

```
select ename
from emp
where mgr in (select empno
from emp
where lower(ename)= 'blake');
```

display ename	cond mgr
mgr	empno (mgr-name)
d.q	s.q

15) WAD employee details those who are reporting to a manager king, ford → cond
display

```
select *
from emp
where mgr in (select empno
from emp
where lower(ename) in ('king', 'ford'));
```

16) WAD department details of an employee those who are reporting to a manager scott

```
select *
from dept
where deptno in (select deptno
from emp
where mgr in (select empno
from emp
where lower(ename) = 'scott'));
```

17) WAD employee name, employee no, designation of an employee those who are reporting to a manager turner, miller.

```
select ename, empno, job
from emp
where mgr in (select empno
from emp
where lower(ename) in('turner', 'miller'));
```

18) WAD department details only the employees those who are reporting to a manager adams, blake

```
select *
from dept
where deptno in (select deptno
from emp
where mgr in (select empno
from emp
where lower(ename) in ('adams', 'blake'));
```

19) WAD employee name only the employee those who are reporting to scott manager

```
select ename
from emp ~~empno~~ mgr
where ~~mgr~~ in (~~empno~~ select ~~empno~~ empno ~~mgr~~
from emp
where ~~mgr~~ empno in (select ~~empno~~ mgr
from emp
where lower(ename) = 'scott));
```

Draw back of sub query

→ In sub query case-2 we can link multiple tables but we cannot fetch data from all the tables

→ In case 3.1 to determine nth max & nth min

2021/5/18 21:33

Scanned by CamScanner

we should write (n-1) inner query so query length will get increased & performance will be decreased.

Any and All operator

→ Single LHS with multiple RHS = comparision then we will go for IN operator.

→ Single LHS with multiple RHS != comparision then we will go for not in operator.

* ~~when~~ NOTE:

when we want to compare single LHS with multiple RHS and we want to perform <, >, <=, >= comparision then it is better to go any and ALL operator.

ANY:

* Any is a multivalued operator
* ANY operator will accept single value in LHS side & multiple values in RHS side
* ANY operator will compare single LHS value with all the values present in RHS side.
* ANY operator will return true if any one of the condition or comparision is true.

→ ANY operator will return false if all the conditions or comparision is false.

* ANY operator will works like or operator.

Syntax:

where LHS_value relational operator ANY(value1, value2,...);

LHS_value → C.N/expr

relational operator: >ANY >=ANY =ANY <ANY <=ANY != ANY

RHS_value → value1, value2,...

e.g:

10 >ANY (20, 10, 30)

10 > 20 → F
10 > 10 → F
10 > 30 → F
} F

30 > ANY (90, 20, 50)

30 > 90 → F
30 > 20 → T
30 > 50 → F
} T

NOTE:

* = ANY and IN operator both are same because = ANY will perform = comparision and IN operator also will perform = comparision.
* ANY and IN operator both are different because ANY will perform all relational operator comparision but IN will perform only = comparision.

All operator:

* ALL is a multivalued operator
* ALL will accept single value in LHS side & multiple values in RHS side.
* All operator will compare single LHS with all the values present in RHS side.
* All operator will return true if all the condition comparision is true.
* All operator will return false if any one of the condition or comparision is false
* All operator will works like AND operator.

Syntax

LHS value relational operator All (value1, value2,...)

LHS value → C.N/expr

relational operator: >ALL >=ALL =ALL <ALL <=ALL != ALL

RHS_value → value1, value2,...

e.g:

100 > ALL (200, 300, 400)

100 > 200 → F
100 > 300 → F
100 > 400 → F
} F

100 > ALL (90, 30, 40)

100 > 90 → T
100 > 30 → T
100 > 40 → T
} T

100 > ALL (90, 300, 400)

100 > 90 → T
100 > 300 → F
100 > 400 → F
} F

When to go to any & ALL operator:

→ waiting ANY & ALL operator it depends upon the question.

1) WQD employee details only the employee salary should be less than any of the miller, turner salary. (lik cond)

```
Select *
from emp
where sal < any (Select sal
from emp
where lower(ename) in ('miller', 'tromer'));
```

2) WQD employee details only employee salary should be more than all smith, james salary (lik cond)

```
Select *
from emp
where sal > all (select sal
from emp
where lower(ename) in ('smith', 'james'));
```

2021/5/18 21:34

Scanned by CamScanner

NOTE

* = ALL and IN operator both are not same. ALL will perform AND comparision IN will perform OR comparision.

Which are the multi values operators?

IN, NOTIN, ANY, ALL

*SET operators:

Set operator is use to combine multiple query o/p into single o/p

1. union: union operator will combine multiple query o/p into single o/p without any duplicate.

e.g: A = {1, 2, x, y, z}
B = {5, 2, z, a, b}
A∪B = {1, 2, x, y, z, 5, a, b}

=> Select ename
from emp
where lower(job) = 'salesman'
union
select ename
from emp
where mgr = 7698;

O/P Ename
ALLEN
JAMES
MARTIN
TURNER
WARD.

2. Union ALL: Union all it will combine multiple query o/p into single o/p with duplicate

e.g: A = {1, 2, x, y, z}
B = {5, 2, z, a, b}
A∪B = {1, 2, x, y, z, 5, 2, z, a, b}

=> Select ename
from emp
where lower(job) = 'salesman'
Union all
Select ename
from emp
where mgr = 7698;

O/P
ALLEN
WARD
MARTIN
TURNER
ALLEN
WARD
MARTIN
TURNER
JAMES

3. Intersect: intersect is use to combine multiple query o/p into ~~into~~ single o/p only comon values

e.g: A = {1, 2, x, y, z}
B = {5, 2, z, a, b}
A∩B = {2, z}

=> Select ename
from emp
where lower(job) = 'salesman'
intersect
select ename
from emp
where mgr = 7698;

O/P: Ename
ALLEN
MARTIN
TURNER
WARD

4) MINUS: Minus operator will disply the o/p a value which is not present in another set.

e.g: A = {1, 2, x, y, z}
B = {5, 2, z, a, b}
A-B {1, x, y}
B-A {5, a, b}

=> Select ename
from emp
where lower(job) = 'salesman'
minus
Select ename
from emp
where mgr = 7698;

O/P No row selected

=> Select ename
from emp
where mgr = 7698
Select ename
from emp
where lower(job) = 'salesman';

O/P: ~~lower~~ ename
James

2021/5/18 21:34

Scanned by CamScanner

JOINS :-

=> Joins is a special possibility of select statement
=> By using joins we can fetch multiple tables info (information) simultaneously is called as joins.
NOTE: Special possibility means by select statement select statement means we can write multiple tables c.N in single select statement.
=> Oracle RDBMS has classified into 5 types of joins
1. Cartesian join / cross join
2. Inner join / Equi join / Natural join
3. Non-Equi join
4. Self join
5. Outer join
   a) Left outer join / Left join
   b) Right outer join / Right join
   c) Full outer join / Full join.

1) Cartesian join / cross join:
=> Cartesian is cross product of multiple tables.
=> Cartesian join is working on a principle of "cross product".
Cross product: A value from one table with all the values present in another table is called as cross product.
-> In cartesian join we will not specify "joining cond".
-> Cartesian join will result in both valid & invalid records.

e.g: A = {1, 2, 3}

B = {x, y, z}

A×B = {1x, 1y, 1z}
{2x, 2y, 2z}
{3x, 3y, 3z}

Syntax: Oracle syntax

```
select *|c.N's
from table1, table2 ....;
```

ANSI syntax

```
Select *|c.N's
from table1 cross join table2 ....;
```

1) Name employee name, department name of all the employees

emp

Ename	job	deptno
Smith	clerk	10
Allen	salesman	30
martin	clerk	20
Blake	salesman	30
Akash	analyst	10

dept

deptno	dname	loc
10	Accounting	Newyork
20	Research	Dallas
30	Sales	chicago
40	operation	Boston

oracle syntax

```
select ename, dname -> ②
from emp, dept; -> ①
```

o/p:

Ename	dname
Smith	Accounting
Smith	Research
Smith	Sales
Smith	operations
~~Smith~~ Allen	Accounting
Allen	Research
Allen	Sales
Allen	operations
⋮	⋮

Ansi syntax

```
Select ename, dname
from emp cross join dept;
```

5 × 4 = 20 records as o/p
5 records valid
15 records invalid

2) Select ename, grade
from emp, salgrade; -> No common c.N between the tables.

3) Select ename, dname, grade
from emp, dept, salgrade;
↓
Joining more than 2 tables

NOTES:
* Cartesian join will result in both valid & invalid records but when compare to valid records invalids records are more hence it is not a preferred type of join.
* Cartesian join can apply any kind table (with (or) without common c.N)

2021/5/18 21:34

Scanned by CamScanner

* cartesian join will result in both valid & invalid records because we are not specifying any joining condition.

*

2. Inner join / Equijoin / Natural join:

=> Joining the multiples tables by "specifying joining cond^n" the tables is called as inner join. (or)

=> Joining multiple tables by "utilizing common c.N" b/w the tables is called as inner join.

=> Inner join will result in only "valid records".

=> In inner join we will specify "joining condition".

=> Inner join can applicable only a table consists of common column.

Joining condition:

Joining the multiple tables by utilizing common c.N b/w the tables is called as joining condition.

Syntax.

where table1. common c.N = table2. common c.N;

Oracle syntax

Select * / c.N's
from table1, table2
where <joining cond> and <cond's>;

<table1. cn = table2. cn>

cn → common column

ANSI syntax:

Select * / c.N's
from table1 [inner] join table2
on <joining cond> → <table1. cn = table2. cn>
where <cond's>;

ON is a key word not clause.

1. WAD employee name, department name of all the employees

emp

Ename	job	deptno
Smith	Clerk	10
Allen	Salesman	30
Martin	Clerk	20
Blake	Salesman	30
Akash	Analyst	10

dept

deptno	dname	loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

Oracle syntax

Select ename, dname → (3)
from emp, dept → (1)
where emp. deptno = dept. deptno → (2)

ANSI syntax

Select ename, dname → (3)
from emp inner join dept → (1)
on emp. deptno = dept. deptno → (2)

Training

emp. deptno = dept. deptno

10 = 10 → T	30 = 10 → F	20 = 10 → F	30 = 10 → F
10 = 20 → F	30 = 20 → F	20 = 20 → T	30 = 20 → F
10 = 30 → F	30 = 30 → T	20 = 30 → F	30 = 30 → T
10 = 40 → F	30 = 40 → F	20 = 40 → F	30 = 40 → F

10 = 10 → T
10 = 20 → F
10 = 30 → F
10 = 40 → F

o/p

Ename	dname
Smith	Accounting
Allen	Sales
Martin	Research
Blake	Sales
Akash	Accounting

Difference b/w cartesian and inner join

cartesian join	inner join
* o/p will consists of both valid & invalid records	* o/p will consists of only valid records
* In cartesian join we will not specify any joining cond^n	* In inner join we will specify the joining cond^n.
* It is applicable for a table with/without common column	* It is applicable only a table consists of common column

Q WAD employee name, designation, salary, department name, location only the employees should be earning 4 digit salary

emp dept c.N cond

Oracle syntax

Select ename, job, sal, dname, loc
from emp, dept
where emp. deptno = dept. deptno and sal like '____';

ANSI syntax

Select ename, job, sal, dname, loc
from emp join dept
on emp. deptno = dept. deptno

2021/5/18 21:35

Scanned by CamScanner

where sal like '----';

3) WAD employee name, salary, annual salary, department name only location should begin with 'new' as a sub string.

Select ename, sal, (sal*12) annual, dname
from emp, dept
where emp. deptno = dept. deptno and lower(loc) like 'new%';

4) WAD employee name, employee no, manager no, department name, location only the employee should be hired in a year 81 and their department name should contain 'a' constantly 'c' character.

Select ename, empno, mgr, dname, loc
from emp, dept
where emp. deptno = dept. deptno and hiredate like '%81' and lower(dname) like '%c%';

5) WAD employee name, salary, new salary with 1000 increment, department name only the department name should not contain 'a' character & salary should be less than 5000.

Select ename, sal, (sal+1000) new sal, dname
from emp, dept
where (emp. deptno = dept. deptno) and (lower(dname) not like '%a%') and (sal < 5000);

6) WAD employee name, employee no, designation, department details only the employee should be earning non commission.

Select ename, empno, job, dept.*
from emp, dept
where (emp. deptno = dept. deptno) and (comm is not null);

NOTE:

1) whenever we want to write combination of emp and * in select clause it is always better to write table_name.*

7) WAD employee details, department name, location, only the employee should not earning salary in a range of 1000 to 5000.

Select emp.*, dname, loc
from emp, dept
where (emp. deptno = dept. deptno) and (sal not between 1000 and 5000);

8) WAD employee details, annual salary, department name only the employee should reporting to a manager id 7600 to 7900

Select emp.*, sal*12 annual, dname
from emp, dept
where (emp. deptno = dept. deptno) and (mgr between 7600 and 7900);

9) WAD employee details, department details, only the employee should be working in chicago, boston location

Select empdept.* (or) Select *
from emp, dept
where (emp. deptno = dept. deptno) and (lower(loc) in ('chicago', 'boston'));

10) WAD employee name, employee no, designation, department details only the employee should be working in department no 10, 20.

Select ename, empno, job, dept.*
from emp, dept
where (emp. deptno = dept. deptno) and (emp. deptno in (10, 20));

NOTE:

whenever we are writing common column in where clause as a condition select clause it is always better to write table_name. common column name display in

8) WAD employee name, employee no, designation, department no, department name only the employee should be hired in a year 81 & display the o/p in ascending order based on employee no.

Select ename, empno, job, emp.deptno, dname
from emp, dept
where (emp. deptno = dept. deptno) and (hiredate like '%81')
order by empno;

NOTE:

whenever we are writing common column in order by clause whatever the table_name written with common column in select clause same thing should be written in order by clause.

2021/5/18 21:35

Scanned by CamScanner

12) WAD employee details, department name only the department name should be exactly 5 characters.

```
Select emp.*, dname
from emp, dept
where (emp.deptno = dept.deptno) and (dname like
'_____');
```

13) WAD employee details, department details, except employee working in department no 30 & the employee name should consists of 'a' character

```
Select *
from emp, dept
where (emp.deptno = dept.deptno) and (emp.deptno !=
30) and (lower(ename) like '%a%');
```

14) WAD employee name, commission, annual salary, only department name only salary should be less than commission, and location should consists of new as a substring

```
select ename, comm, (sal*12) annsal, dname
from emp, dept
where (emp.deptno = dept.deptno) and (sal < comm)
and (lower(loc) like '%new%');
```

15) WAD employee name, employee no, designation, department details only employee should be earning 4 digit salary.

```
Select ename, empno, job, dept.*
from emp, dept
where (emp.deptno = dept.deptno) and (sal like '___');
```

16) WAD employee name, department no, department name. only the employee should be hired in a year 81 to 87

```
Select ename, emp.deptno, dname
from emp, dept
where (emp.deptno = dept.deptno) and (hiredate
        between '01-jan-81' and '31-dec-87');
```

17) WAD employee details, department details only employee should be earning commission more than 500 Rs.

```
Select *
from emp, dept
where (emp.deptno = dept.deptno) and (comm >
                500);
```

18) WAD employee details only employee should be working in a department 'accounting', 'research'.

```
Select emp.*, dname
from emp, dept
where (emp.deptno = dept.deptno) and (lower(dname
-e) in ('accounting', 'research'));
```

* when display from more than 2 table then we will join
* when display from one table & cond^n from another table then also we can write quarter by using joins

19) WAD department details only employee should be working as analyst in a department no 10, 20

```
Select dept.*,
from emp, dept
where (emp.deptno = dept.deptno) and
        (lower(job) = 'analyst') and
    (dept.deptno in (10, 20));
```

20) WAD employee name, employee no, new salary with 1000 as increment only employee should not be working in a location chicago

```
Select ename, empno, (sal+1000) new sal,
from emp, dept
where (emp.deptno = dept.deptno) and
        (lower(loc) != 'chicago');
```

21) WAD employee name, employee no, annual salary, department name, location, only the salary should be same as smith salary.

```
Select ename, empno, (sal*12) annsal, dname, loc
from emp, dept
where (emp.deptno = dept.deptno) and
        (sal in (select sal
            from emp
            where lower(ename) = 'smith'));
```

2021/5/18 21:35

Scanned by CamScanner

22) WAD employee details, department details only the designation should be same as miller designation.

```
Select *
from emp, dept
where (emp.deptno = dept.deptno) and
      (job in (Select job)
       from emp
       where lower(name) = 'miller'));
```

23) ~~WAD departmentname which has~~ Oracle and Ansi syntax to join more than two tables

Oracle syntax:

```
Select * / c.n'S
from table1, table2, table3, table4.....
where table1.cn = table2.cn and
      table2.cn = table3.cn and
      table3.cn = table4.cn .... and <condns>
```

Ansi syntax:

```
Select * / c.n'S
from table1 inner join table2
on table1.cn = table2.cn
inner join table3
on table2.cn = table3.cn
inner join table4....
on table3.cn = table4.cn....
where <cond'S>;
```

Oracle Syntax:

In oracle syntax to join 'n' no of tables we will write 'n' table names in single from clause and (n-1) joining condn in single where clause.

Ansi Syntax:

In Ansi syntax to join 'n' no of tables we will write 'n' no of table name in (n-1) from clause and (n-1) joining condn in (n-1) no of 'on' keyword

24) WAD region name, country name, city of all the locations.

region → regions; country name → countries; city → location; regions + countries → dep

Oracle Syntax:

```
Select region_name, country_name, city
from regions, countries, locations
where regions.region_id = countries.region_id and
      countries.country_id = locations.country_id;
```

Ansi Syntax

```
Select region_name, country_name, city
from regions inner join countries
on regions.region_id = countries.region_id
inner join locations
on countries.country_id = locations.country_id;
```

Table aliasing:

```
Select region_name, countries_name, city
from regions r, countries c, locations l
where r.region_id = c.region_id and c.country_id =
      l.country_id;
```

NOTE

⇒ whenever we are writting table_aliasing in a query throughout the query i.e in Select clause and where clause same table_aliasing should be used.

25) WAD regions details, country name, city, postal code only the location id should be in a range of 1000 to 5000

```
Select r.*, country_name, city, postal_code
from regions r, countries c, location l
where (r.region_id = c.region_id) and (c.country_id =
      l.country_id) and (location_id between 1000
      and 5000);
```

In regions, countries, location region_id & country_id are common column between

⇒⇒ If once we use table_aliasing in from clause, throughout the query we can write table aliasing only if we write table name once after the table_aliasing the o/p will show error message.

2021/5/18 21:35

Scanned by CamScanner

3) WAD region name, country id, location details only the city name should contain 'a' character

```
Select region_name, c.country_id, l.*
from regions r, countries c, locations l
where r.region_id = c.region_id and c.country_id =
    l.country_id and lower(city) like '%a%';
```

4) WAD regions details, country details, cityname only the country id should be 'in', 'us'

```
Select r.*, c.*, city
from regions r, countries c, location l
where r.region_id = c.region_id and c.country_id =
l.country_id and lower(c.country_id) in
('in', 'us'));
```

5) WAD regions details, country details, location details only the region id should be in a range of 1 to 5

```
Select *
from regions r, locations l, countries c
where r.region_id = c.region_id and c.country_id =
    l.country_id and c.region_id between
1 and 5;
```

Why inner join is also known as equi join?

Since we are joining multiple tables and specifing equal to(=) operator in the joining condn so inner join is known as equi join.

When to go for innerjoin?

1) when we want to disply from more than 1 table
2) when disply 1 table & condn another table

Non-Equi join

* joining the multiple tables without specifying equal to(=) operator in the joining condn is called as Non-equi join. (or)
* joining the multiple tables when there is no common column b/w the tables is called as Non-equi join.
* in Non-equi joining condn except equal to(=) operator we will specify other all relational operator

When to go non-equi join:

case1:- when there is no common column b/w the tables

case2: If there is no common column but atleast there should be relevant column b/w the tables (c.in can be different but datatype & size should be same).

1) WAD employee name, designation, salgrade details only the salary range between losal and hisal

```
Select ename, job, sal
from emp, salgrade
where sal between losal and hisal;
```

joining condn

2) WAD employee name, employee no, department no, grade only salary should be less than losal and they should be working in department no. 10, 20.

```
select ename, empno, deptno, grade.
from emp, salgrade
where (sal < losal) and (deptno in (10, 20));
```

joining condn — column condn

3) WAD employee details, salgrade details only the commission should not be in a range of lowsal and high sal

```
select *
from emp, salgrade
where comm not between losal and hisal;
```

Assignment on 3 tables joining

6) WAD region name, country name, locations details only the country id in 'us'.

```
select region_name, country_name, l.*
from regions r, countries c, locations l
where (r.region_id = c.region_id) and (c.country_id =
    l.country_id) and lower(c.country_id) in ('in', 'us'));
```

7) WAD region name, country name, ~~locations~~ details, street adress ~~only~~ postal code, city name only for the region id 2, 3.

```
Select region name, c.*, street_address, postal_name, city
from regions r, countries c, locations l
where (r.region_id = c.region_id) and (c.country_id
    = l.country_id) and (r.region_id in (2, 3));
```

8) WAD region details, country id, location id, postal code, only for the city bombay, Tokyo and disply the o/p in descending order based on location id.

2021/5/18 21:36

Scanned by CamScanner

Select r.*, c.country_id, location_id, postal_code
from regions r, countries c, location l
where (r.region_id = c.region_id) and (c.country_id = l.country_id) and (lower(city) in ('bombay', 'tokyo'))
order by location_id desc;

9) WAD region details, country details, location details except for country india, china and their street address should consists of 'c' character

Select r.*, c.*, l.*
from regions r, countries c, locations l
where (r.region_id = c.region_id) and (c.country_id = l.country_id) and (lower(country_name) not in ('india','china')) and (lower(street_address) like '%c%');

10) WAD country details, location details only for the postal code exactly 4 digit and their state province should not begin with 'm' character.

Select c.*, l.*
from countries c, location l
where (c.country_id = l.country_id) and (postal_code like '----') and (lower(state_province) not like 'm%');

11) WAD region_id, country_id, location_id, postal code, city name except for region europe, asia and their street address beggining third number should be 2

Select r.region_id, c.country_id, location_id, postal_code, city
from regions r, countries c, locations l
where (r.region_id = c.region_id) and (c.country_id = l.country_id) and (lower(region_name) not in ('europe', 'asia')) and (street_address like '__2%');

12) WAD country_details, cityname, state province only for the state province oxford, maharashtra and they should have postal code.

Select c.*, city, state_province
from countries c, locations l
where (c.country_id = l.country_id) and (lower(state_province) in ('oxford', 'maharashtra')) and (postal code is not null);

13) WAD region details, location details only for the postal code 3095, 2901.

Select r.*, l.*
from regions r, locations l
where

14) WAD country details, location details, region details only the street address should consists of 3rd as a substring.

Select c.*, l.*, r.*
from regions r, locations l, countries c
where (r.region_id = c.region_id) and (c.country_id = l.country_id) and (lower(street_address) like '%3rd%');

Outer join

Joining the multiple tables by utilizing common column b/w the table is called as outer join

⇒ outerjoin is a combination of inner join and invalid records

$\boxed{\text{outer join} = \text{inner join (valid record)} + \text{invalid record}}$

⇒ outer join will result both valid & invalid records

⇒ outer join will result valid records from all the tables and invalid records from specific table.

⇒ Based on invalid records outer join has classified into 3 types

a) Left outer join / Left join
b) Right outer join / Right join
c) Full outer join / Full join

Left outer join / Left join

It is a combination of inner join and a record from left table which doesnot have any matching pair in right table is called as LOJ (Left outer join).

(or)

valid records from all the tables and invalid records from Left table is called as LOJ

Oracle syntax:

Select *|c.n'c
from table1, table2
where table1.cn = table2.cn (+) and c.n(condition);

2021/5/18 21:36

Scanned by CamScanner

where (+) ---> Invalid records

ANS syntax

Select *|c.n'e
from table1 left [outer] join table2
on table1.cn = table2.cn
where <cond'e>;

D. WAD employee name, department name only the employee should not working any of the department.

Left table emp

empno	ename	deptno
1	Smith	10
2	Allen	30
3	Blake	20
4	Miller	10
5	Shashi	

Right table dept

deptno	dname	loc
10	accounting	newyork
20	research	dallas
30	sales	chicago
40	operations	boston

Oracle syntax:

Select ename, dname —(5)
from emp, dept —(1)
where emp.deptno = dept.deptno(+); —(2)

ANS syntax

Select ename, dname —(3)
from emp left outer join dept —(1)
on emp.deptno = dept.deptno —(2)

op	ename	dname
inner join (valid record)	Smith	accounting
	Allen	sales
	Blake	research
invalid record (left table)	Miller	accounting
	Shashi	

} left outer join

Training

10 = 10 → T	30 = 10 → F
10 = 20 → F	30 = 20 → F
10 = 30 → F	30 = 30 → T
10 = 40 → F	30 = 40 → F
20 = 10 → F	10 = 10 → T
20 = 20 → T	10 = 20 → F
20 = 30 → F	10 = 30 → F
20 = 40 → F	10 = 40 → F
Null = 10 → F	
Null = 20 → F	
Null = 30 → F	
Null = 40 → F	

Right outer join / Right join

→ It is a combination of inner join and a record from right table which does not have any matching pair in ~~right~~ left table is called as ROJ (Right outer join).

(or)

→ valid record from all the table & invalid record from right table is called as ROJ

Oracle syntax

Select *|c.n'e
from table1, table2
where table1.cn(+) = table2.cn and c.n'e <cond'e>;

ANS syntax

Select *|c.n'e
from table1 right [outer] join table2
on table1.cn = table2.cn
where <cond'e>;

D. WAD employee name, department name only for the department no employees are working.

Left table emp

empno	ename	deptno
1	Smith	10
2	Allen	30
3	Blake	20
4	Miller	10
5	Shashi	

Right table dept

deptno	dname	loc
10	accounting	new york
20	research	dallas
30	sales	chicago
40	operations	boston

Oracle syntax

Select ename, dname —(3)
from emp, dept —(1)
where emp.deptno(+) = dept.deptno; —(2)

ANS syntax

Select ename, dname
from emp, dept
where emp.deptno = dept.deptno

10 = 10 → T	20 = 10 → F	30 = 10 → F
10 = 30 → F	20 = 30 → F	30 = 30 → T
10 = 20 → F	20 = 20 → T	30 = 20 → F
10 = 10 → T	20 = 10 → F	30 = 10 → F
10 = Null → F	20 = Null → F	30 = Null → F

40 = 10 → F	
40 = 30 → F	
40 = 20 → F	
40 = 10 → F	
40 = Null → F	

op	ename	dname
	Smith	accounting
	Miller	accounting
	Blake	research
	Allen	sales

} inner join (valid record)

	ename	dname
invalid record (right table)		operations

→ Right outer join

Full outer join / Full join:

* It is a combination of LOJ and ROJ is called as FOJ

(or)

* valid record from all the table & invalid records from all the table is called as FOJ

~~Ora~~ Oracle syntax

Select *|c.n'e
from table1, table2
where table1.cn = table2.cn(+)
union
Select *|c.n'e
from table1, table2
where table1.cn(+) = table2.cn;

2021/5/18 21:36

Scanned by CamScanner

ANSI syntax:

Select *[c.n'S]
from table1 full [outer] join table2
on table1.cn = table2.cn
where <cond's>;

1) WAD employee name, department name, only the employees not working for any of the department and only for the department no employees are working.

Left table

emp

empno	ename	deptno
1	smith	10
2	allen	30
3	Blake	20
4	miller	10
5	shashi	

Right table

dept

deptno	dname	loc
10	Accounting	Newyork
20	Research	dallas
30	sales	chicago
40	operations	boston

Oracle syntax

Select ename, dname
from emp, dept
where emp.deptno = dept.deptno(+)
union
Select ename, dname
from emp, dept
where emp.deptno(+) = dept.deptno;

ANSI syntax

Select ename, dname
from emp full outer join dept
where emp.deptno = dept.deptno

O/P

ename	dname
smith	accounting
Allen	sales
Blake	research
miller	accounting
shashi	operations

Inner join (valid record)

invalid record (left table)

invalid record (Right table)

Full outer join

NOTE:

while we are performing outer join if there is no invalid records then we will get valid record as o/p

SELF JOIN:

* Joining same table to itself is called as self join.
* Except self join in other all types of join we will specify more than 1 table name but in self join we will join same table to itself.

=> Based on joining cond self join as 2 cases:

case1: Joining same table by utilizing same c.n

syntax

Select *[c.n'S]
from ~~table1, table2~~ table1 t1, table1 t2
where t1.c1 = t2.c1 and <cond's>;

case2:

Joining same table by utilizing different c.n

Syntax:

Select *[c.n'S]
from table1 t1, table1 t2
where t1.c1 = t2.c2 and <cond's>

1) WAD employee name, manager name of all the employees.

(employee table) emp a

empno	ename	job	mgr
1	girish	CEO	
2	Pradeep	VP	1
3	keerthi	QA	2
4	Nisha	Verbal	2
5	Tanush	Dev	2

(manager table) emp b

empno	ename	job	mgr
1	girish	CEO	
2	Pradeep	VP	1
3	keerthi	QA	2
4	Nisha	Verbal	2
5	Tanush	Dev	2

Select a.ename, b.ename mgr name —③
from emp.a, emp.b —→①
where a.mgr = b.empno; —→② —→ joining cond (case-2)

null = 1 → F 1=1 → T 2=1 → F 2=1 → F 2=1 → F
null = 2 → F 1=2 → F 2=2 → T 2=2 → F 2=2 → T
null = 3 → F 1=3 → F 2=3 → F 2=3 → F 2=3 → F
null = 4 → F 1=4 → F 2=4 → F 2=4 → F 2=4 → F
null = 5 → F 1=5 → F 2=5 → F 2=5 → F 2=5 → F

O/P

ename	mgr name
Pradeep	girish
keerthi	Pradeep
Nisha	Pradeep
Tanush	Pradeep

2) WAD employee name, employee designation, employee salary, manager sal of all the employees.

Select a.ename, a.job, a.sal, b. sal mgr.sal
from emp a, emp b
where a.mgr = b.empno;

2021/5/18 21:36

Scanned by CamScanner

3) WAD employee name, employee department no. manager name, manager annual salary only for the employee designation should consists of 'er' as substring.

```
select a.ename, a.deptno, b.ename, mgrname, b.(sal)
mgr annsal
from emp a = emp b
where a.mgr = b.empno and lower(a.job) like '%er%';
```

4) WAD employee details, manager name, manager commission only the employee salary should be greater than manager sal

```
select a.*, b.ename mgr_name, b.comm mgr_comm
from emp a, emp b
where a.mgr = b.empno and a.sal > b.sal;
```

5) WAD employee name, employee no. employee designation, manager name, manager new salary with 1000 increment only employee should be hired before manager hiredate and disply the o/p ascending order based on employee no.

```
select a.ename, a.empno, a.job, b.ename mgr_name
b.(sal+1000) mgr_new_sal
from emp a, emp b
where a.mgr = b.empno and a.hiredate < b.hiredate
order by a.empno desc;
```

6) WAD employee details, manager name, manager deptno, manager designation, only manager and employee designation should be same.

```
select a.*, b.ename mgr_name, b.deptno mgr_deptno
b.job mgr_job
from emp a, emp b
where a.mgr = b.empno and a.job = b.job;
```

7) WAD employee name, employee salary, employee manager no, manager details, only employee name & manager name should consists of 'a' character.

```
select a.ename, a.sal, a.mgr, b.*
from emp a, emp b
where (a.mgr = b.empno) and (lower(a.ename) like
'%a%') and (lower(b.ename) like '%a%');
```

8) WAD employee name, employee designation, manager name, manager salary only employee should be working in a department no 20, 30

```
select a.ename, a.job, b.ename mgr_name, b.sal
mgr_sal
from emp a, emp b
where (a.mgr = b.empno) and (a.deptno in (20,30));
```

9) WAD employee details, manager name, manager designation only manager should be earning annual salary less than employee annual commission.

```
select a.*, b.ename mgr_name, b.job mgr_job
from emp a, emp b
where (a.mgr = b.empno) and (b.(sal*12) < a.(comm*12));
```

10) WAD employee name, empno, employee deptno, manager name, manager deptno only employee deptno should not be same as manager deptno.

```
select a.ename, a.empno, a.deptno, b.ename mgr_name
b.deptno mgr_deptno
from emp a, emp b
where (a.mgr = b.empno) and (a.deptno not in
(b.deptno));
```

11) WAD employee name, employee commission, manager details only employee should be hired on the same date of manager hiredate.

```
select a.ename, a.comm, b.*
from emp a, emp b
where (a.mgr = b.empno) and (a.hiredate = b.
hiredate);
```

Case 1

1) WAD employee name Those who are earning same sal

~~WAD to get duplicate sal.~~

```
select a.ename, a.sal
from emp a, emp b
where a.sal = b.sal and a.empno != b.empno;
```

WAD duplicate sal.

```
select sal
from emp
group by sal
having count(*) > 1;
```

2021/5/18 21:36

Scanned by CamScanner

3) WAD employee name, who is working for same department as

```
select a.ename, a.deptno
from emp a, emp b
where a.deptno = b.deptno and a.empno != b.empno;
```

Determining unknown values By using Self join -

1) WAD employee details only the employee department no should be same as 'martin' deptno

```
select a.*
from emp a, emp b
where a.deptno = b.deptno and lower(b.ename) =
      'martin';
```

2) WAD employee details salary should be more than smith salary and designation should be same as clark designation

```
select *
from emp a, emp b, emp c
where a.sal > b.sal and lower(b.ename) = 'smith'
      and a.job = c.job and lower(c.ename) = 'clark';
```

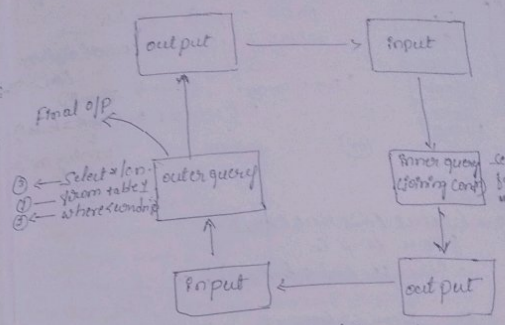
* In self join 1 unknown is considered as 1 table, as the unknown values increases ~~tables~~ no of tables are also increases.

Co-Related Sub Query: [CRSQ]

* CRSQ is a advance version of subquery.
* CRSQ is working on a principle of "sub-query" and "joins".
* In CRSQ innerquery and outerquery will be present.
* In CRSQ innerquery is depends on outerquery o/p and outerquery is depends on innerquery o/p
* In CRSQ innerquery and outerquery both are inter dependent on each other because of specifying joining cond" in innerquery.

max <
min >
N =
N-1 no equal

Working mechanism of Co-Related Sub Query.



=> compiler will give control to outerquery, outerquery will exicute partially for 1st time, and it will prepare the o/p

=> The o/p of outer query will pass an i/p to innerquery. Innerquery will exicute for the 1st time completely. Based on condition and i/p taken from outerquery innerquery will prepare the o/p.

* The o/p of innerquery will pass as an input to outerquery, outerquery will exicute completely for the 2nd time. Based on the condition and i/p taken from innerquery, outerquery will prepare final o/p.

NOTE:

* Innerquery will exicute only one time & it will exicute completely [select, from & where]
* outer query will exicute 2 times, 1st time it will exicute partially [only from clause] and 2nd time it will exicute completely [where & select clause]

Logic

	outer query	inner query (joining cond)
max value	where N	= where table1.cn <= table2.cn
	where N-1	= where table1.cn < table2.cn
min value	where N	= where table1.cn >= table2.cn
	where N-1	= where table1.cn > table2.cn

2021/5/18 21:37

Scanned by CamScanner

1) W&D 3rd max salary [n=3 (or) n-1=2] "outer query"

emp a	emp b
sal	sal
6000	6000
5000	5000
3000	3000
2000	2000
1000	1000

```
select sal ← (3)
from emp a — (1)
where 3 = (select count(distinct sal) → (2)
          from b — (1)
          where a.sal <= b.sal — (2)
```

"inner query"

forming count

Tracing

```
select sal
from emp a
where 2 = (select distinct (sal)
           from emp b
           where a.sal < b.sal);
```

Tracing:

6000 <= 6000 → T	5000 <= 6000 - T	3000 <= 6000 → T
6000 <= 5000 → F	5000 <= 5000 - T	3000 <= 5000 - T
6000 <= 3000 → F	5000 <= 3000 - F	3000 <= 3000 - T
6000 <= 2000 → F	5000 <= 2000 - F	3000 <= 2000 - F
6000 <= 1000 → F	5000 <= 1000 - F	3000 <= 1000 - F
3 = 1 → F	3 = 2 → F	3 = 3 → T

2000 <= 6000 → T	1000 <= 6000 → T
2000 <= 5000 → T	1000 <= 6000 → T
2000 <= 3000 → T	1000 <= 6000 → T
2000 <= 2000 → T	1000 <= 6000 → T
2000 <= 1000 → T	1000 <= 6000 → T
3 = 4 → F	3 = 5 → F

O/P

sal
3000

NOTE:
Writing distinct clause in inner query of CRSQ is mandatory, if we not specify distinct clause, we will get wrong O/P.

2) W&D 3rd min salary

[n=3
n-1=2]

```
select sal
from emp a
where 3 = (select count(distinct sal)
           from emp b
           where a.sal >= b.sal);
```

3) W&D 2nd oldest hiredate

```
select hiredate
from emp a
where 2 = (select count(distinct hiredate)
           from emp b
           where a.hiredate >= b.hiredate);
```

4) W&D 4th max annual salary

```
select (sal x 12) annsal
from emp a
where 4 = (select count (distinct sal*12)
           from emp b
           where a.(sal x 12) <= b.(sal x 12);
```

5) W&D 3rd minimum annual salary

```
select (sal x 12) annsal
from emp a
where 3 = (select count (distinct sal x 12)
           from emp b
           where a.(sal x 12) >= b.(sal x 12);
```

6) W&D 4th recent hiredate

```
select hiredate
from emp a
where 4 = (select count (distinct hiredate))
from emp b
where a.hiredate <= b.hiredate);
```

7) W&D 7th maximum new salary with 1000Rs increment

```
select (sal+1000) new sal
from emp a
where 7 = (select count (distinct sal+1000)
           from emp b
           where a.(sal+1000) <= b.(sal+1000));
```

8) W&D 2nd max comission.

```
select comm
from emp a
where 2 = (select count (distinct comm)
           from emp b
           where a.comm <= b.comm);
```

2021/5/18 21:37

Scanned by CamScanner

9) WAD 3rd minimum commission.

```
select comm
from emp a
where 3 = (select count (distinct comm)
from emp b
where a.comm >= b.comm);
```

10) WAD 10th min half term salary.

```
select (sal x 6) half_term_sal
from emp a
where 10 = (select count (distinct sal x 6)
from emp b
where a.(sal x 6) >= b.(sal x 6));
```

11) WAD 4th min mgr number

```
select mgr
from emp a
where 4 = (select count (distinct mgr)
from emp b
where a.mgr >= b.mgr);
```

12) WAD 3rd recent hiredate after 6 months

```
select (hiredate + 182)
from emp a
where 3 = (select count (distinct hiredate + 182)
from emp b
where a.(hiredate + 182) <= b.(hiredate + 182));
```

13) WAD 2nd, 5th, 7th max salary.

```
select sal
from emp a
where (select count (distinct sal)
from emp b
where a.sal <= b.sal) in (2, 5, 7);
```

14) WAD 3rd, 10th, oldest hiredate

```
select hiredate
from emp a
where (select count (distinct hiredate)
from emp b
where a.hiredate >= b.hiredate) in (3, 10);
```

15) WAD first 5 max salary

```
select sal
from emp a
where (select count (distinct sal)
from emp b
where a.sal <= b.sal) in (1,2,3,4,5); (or) = all (1,2,3,4,5);
(or)
between 1 and 5;
(or)
<= (5);
```

16) WAD first 3 annual salary

```
select (sal x 12) annual sal
from emp a
where (select count (distinct (sal x 12))
from emp b
where a.(sal x 12) <= b.(sal x 12)) between 1 and 3;
```

17) WAD 2nd, 6th, 8th, max new commission with 500 decrement

```
select (comm - 500) new comm
from emp a
where (select count (distinct (comm - 500)
from emp b
where a.(comm - 500) <= b.(comm - 500)) in (2, 6, 8));
```

18) WAD employee details of 3rd min salary.

```
select *
from emp a
where 3 = (select count (distinct sal)
from emp b
where a.sal >= b.sal);
```

19) WAD employee name, designation, department no of 10th min annual salary.

```
select ename, job, deptno, (sal x 12) annual
from emp a
where 10 = (select count (distinct (sal x 12)
from emp b
where a.(sal x 12) >= b.(sal x 12));
```

20) WAD employee name, employee no. department name of 2nd min commission.

```
select ename, empno, dname
from emp a, dept
where (a.deptno = dept.deptno) and (2 = (
select count (distinct comm)
from emp b
where a.comm >= b.comm));
```

2021/5/18 21:37

Scanned by CamScanner

21) WQD department details of 2nd min mgr no

```
select dept.*
from emp a, dept
where (a.deptno=dept.deptno) and (2 = (select
  count (distinct mgr)
  from emp b
  where a.mgr >= b.mgr));
```

Sub query case 2 concept

```
select *
from dept
where deptno in (select deptno
from emp a
where 2 = (select count(distinct mgr)
from emp b
where a.mgr >= b.mgr));
```

22) WQD employee name, designation, salary, department details of first 5 minimum salary.

```
select ename, job, sal, dept.*
from emp a, dept
where (a.deptno = dept.deptno) and (select count
(distinct sal)
from emp b
where a.sal >= b.sal) <= 5;
```

23) WQD department name, location of first, 5th, 7th oldest hiredate

```
select dname, loc
from dept, emp a
where (a.deptno = dept.deptno) and (select count (distin
ct hiredate)
from emp b
where a.hiredate >= b.hiredate) in (1,5,7));
```

24) WQD employee details, dept name of first 3 minimum annual commission.

```
select emp.*, dname
from emp a, dept
where (a.deptno = dept.deptno) and (select count(dis
tinct comm*12)
from emp b
where a.(comm*12) >= b.(comm*12)) in (1,2,3));
```

Pseudo columns (False column):

Pseudo columns are the false columns which are not actually present in table but it will reflect in the o/p whenever we are specifying in select statement. Is called as pseudo columns

table

C_1	C_2	C_3	rowid	rownum

original column. — Pseudo column (false column)

* Each and every table in a database will contain of pseudo columns
* A/c to Oracle RDBMS, pseudo columns are classified into 2 types.
  1. Rowid
  2. Rownum.

Rowid

=> It is a unique and permanent value given to each and every record present in table.
=> Rowid is a 18 bit Alphanumeric value given to each and every record present in a table.
=> Rowid represents physical address of each and every record. [physical address means database name, table name, column name]
=> Rowid will be automatically generated by insert stmt.
=> Rowid is a static [constant]

Rownum:

=> Rownum is a unique and temporary value given to each and every record present in table.
=> Rownum is a sequential integer value given to each and every record present in table.
=> Rownum represents total no of records.
=> Rownum will be automatically generated by select statement
=> Rownum is a dynamic [

Whenever we want to fetch intermediate record from the table then we will go for Pseudo column

2021/5/18 21:37

Scanned by CamScanner

bottom record. { where N = where table1.rowid <= table2.rowid
where N-1 = where table1.rowid < table2.rowid

top >
bottom <
N = 1
N-1 no equal

1) WAD top 1st employee record.

```
select *
from emp a
where 1 = (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid);
```

2) WAD bottom 3rd employee record.

```
select *
from emp a
where 3 = (select count (distinct rowid)
from emp b
where a.rowid <= b.rowid);
```

NOTE: writting distinct is optional in pseudo c.n bcz rowid is unique.

3) WAD top 3rd employee name, designation, dept.no

```
select ename, job, deptno
from emp a
where 3 = (select count (rowid)
from emp b
where a.rowid >= b.rowid);
```

4) WAD top first 5 employee details

```
select *
from emp a
where (select count (rowid)
from emp b
where a.rowid >= b.rowid) <= 5;
```

5) WAD employee name, date of joining, commission of bottom 5th employee

```
select ename, hiredate, comm
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid <= b.rowid) <= 5);
```

6) WAD employee name, annual salary manager no of top 10th employee.

7) WAD employee details of bottom 3 employee details

```
select *
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid <= b.rowid) in (1,2,3));
```

8) WAD department name, location of top 5th employee

```
select dname, loc
from emp a, dept
where (a.deptno = dept.deptno) and (5 = (select count(rowid)
from emp b
where a.rowid >= b.rowid));
```

9) WAD department details of bottom 5th, 7th, 10th employee

```
select dept* from dept, emp a
where (a.deptno = dept.deptno) and (select count
(rowid)
from emp b
where a.rowid <= b.rowid) in (5,7,10));
```

10) WAD employee name, designation, salary, department name, location of top 2nd, 3rd, 8th employee

```
select ename, job, sal, dname, loc
from emp a, dept
where (a.deptno = dept.deptno) and (select count
(distinct rowid)
from emp b
where a.rowid >= b.rowid) in (2,3,8));
```

11) WAD employee details, department details of 4th employee

[If there is top/~~given~~ bottom is not specified, then by defalut it is considered as top]-*

```
select emp.*, dept.*
from emp a, dept
where (a.deptno = dept.deptno) and (4 = (select
count (distinct rowid)
from emp b
where a.rowid >= b.rowid));
```

Scanned by CamScanner

```
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid) != (select min (rownum)
                              from emp);
```

13) WAD employee details except last employee.

```
select *
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid) != (select max (rownum)
                              from emp);
```

14) WAD employee details except 1st & last employee

```
select *
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid) not in ((select max (rownum)
from emp), (select min (rownum)
                              from emp));
```

15) WAD employee details of 1st & last employee

```
select *
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid) in ((select max (rownum)
from emp), (select min (rownum)
                              from emp));
```

*16) WAD 1st half off the employee details

```
select *
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid) <= (select count (*)/2
                              from emp);
```

*17) WAD 2nd half off the employee details

```
select *
from emp a
where (select count (distinct rowid)
from emp b
where a.rowid >= b.rowid) > (select count(*)/2
```

=> The process of storing wrong data/ invalid data inside the table in database is called as data integrity. (02)

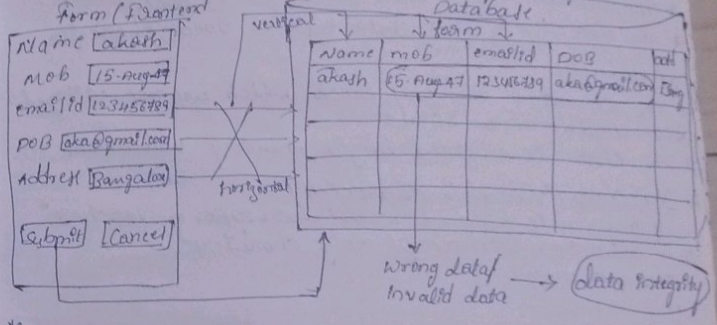
=> The process of checking correctness and accuracy of data inside the table in database is called as data integrity.

=> The process of data integrity will avoid of storing wrong data/ invalid data.

=> The process of data integrity will ensure we are storing only valid data.

=> The process of data integrity can be achieved through assigning "data types and constraints" to the column present in table.

=> RDBMS will support data integrity concept.



* Data Types:

=> Data types represent what type of data should accept or store inside the c.m in table is called as data types.

=> a column in the table can be assigned with max of 1 data type.

=> In the table each and every column should be mandatory to assign with data type.

=> In oracle we have 'n' no of data types are present but we prefer only 4 types.

1. char
2. varchar
3. number

(extra data type
BLOB (Binary large object) -> image,

2021/5/18 21:38

Scanned by CamScanner

1) Char [charecters]:

If any c.n is assigned with "data type of char" then that "c.n will accept only charecters".

Charecter means
A - Z
a - z
0 - 9
$ - #

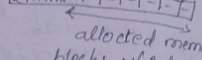
Syntax
char(size)

e.g:

	ename
10	akash
10	abcd123
10	xyz@123
10	123@#$
10	123abcd!@#$
10	Ram
Fixed	;

char(10)

ename



allocted memory blocks which are not utilized will be considered as "space".

* The default size of char data type is one (1)
* The max size of char data type is 2000 charecter

Drawbacks
* The allocated memory block which are not utilized will considered as "space".
* The char is a fixed length memory allocation.

2) Varchar [variable charecter]

If any c.n is assigned with "data type of varchar" then that "c.n will accept only charecters".

Charecter means
A - Z
a - z
0 - 9
$ - #

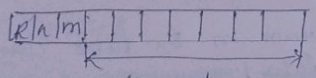
Syntax
varchar(size)

e.g:

	ename
5	akash
7	abcd123
7	xyz@123
6	123@#$
	123abcd!@#$ invalid
3	Ram
	;

varchar(10)

ename



allocated memory blocks which are not utilized will be considered as "null"

variable length

* The allocated memory block ... 4000 charecters

NOTE:
* From Oracle 10g onwards they have introduced a new data type that is varchar2 which accept max 10000 charecters.

What is the difference b/w char and varchar datatype

Char	Varchar
=> It is a fixed length memory allocation	=> It is a variable length memory allocation
=> allocated memory blocks which are not utilized will be considered as "space"	=> allocated memory blocks which are not utilized will be considered as "null"
=> Default size is one (1)	=> There is no default size.
=> max size is 2000	=> max size is 4000

3) Date:

If any c.n/attribute is assigned with date datatype then that c.n will accept only data format.

Oracle date format

DD-mon-YY (or) DD-Mon-YYYY

e.g

DOB
15-AUG-47
15-OCT-96
abcd
123456
$$@$

date

Invalid data

4) Number:

=> If any c.n/attribute is assigned with a datatype as number then it will accept only number/digit.

Number/digit
[0-9]

Syntax
number (precision, scale)

Case 1:

2021/5/18 21:38

Scanned by CamScanner

accept.

e.g:

empno
1234
,
abcde
123b
ab1b
10.1
-90
0
⋮

→ Number(4) → Precision

invalid data

-9999 to +9999

here -9 is assign for 1 block

Case 2:

number (precision, scale)

* Scale represents decimal.
* precision represents max number digits can accept including decimal.
→ here we can write upto 38 precision.

number(38)

e.g:

Per
35.1
90.12
9.123
123.15
123.13
100.12
100
100.00
-99.1
123.1
⋮

→ number(4,2) → scale → Precision

invalid data.

→ 100.00 is considered as 100.0 only hence it is valid.

note

-9999 to +9999 range.

Constraints:

⇒ Constraints are the set of rules (or) validation given to the column to restrict of storing wrong data in a column is called as constraints.

Constraints are classified into 5 types

1. NOT NULL:

⇒ Not Null is a constraint given to column. If a column is assigned with a NOT NULL then it will not accept Null values. that i.e. it is a mandatory or compulsory to enter some data.

which column (or) attribute can be assigned with NOT Null constraints:

⇒ The column is mandatory (or) compulsory for a table/entity for those column's will assign NOT NULL constraints.

NOTE:

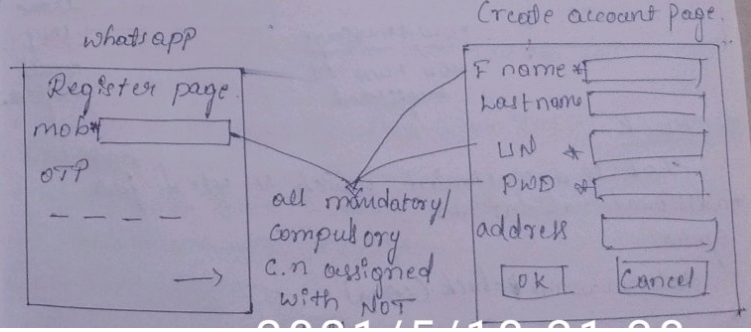
→ If a column is assigned with a constraint as NOT NULL it will not accept Null value but it will accept duplicate values.

e.g: ~~Real time example~~

Sname	USN	branch	Per	DOB	gender	mob	address	email	year
Varchar (20)	Varchar (20)	Varchar (20)	number (4,2)	date	Varchar (10)	number (10)	Varchar (20)	Varchar (20)	int (4)
N.N	N.N	N.N	X	N.N	N.N	N.N	N.N	N.N	X

N.N → Not null

Real time example.



Scanned by CamScanner

column is assigned with a unique then it will not accept duplicate values that is it will not accept a value which is already present (or) existing.

which column (or) attribute can be assigned with unique constraints

→ If any column/attribute which accept only distinct values for those column will assign with unique constraints.

NOTE:

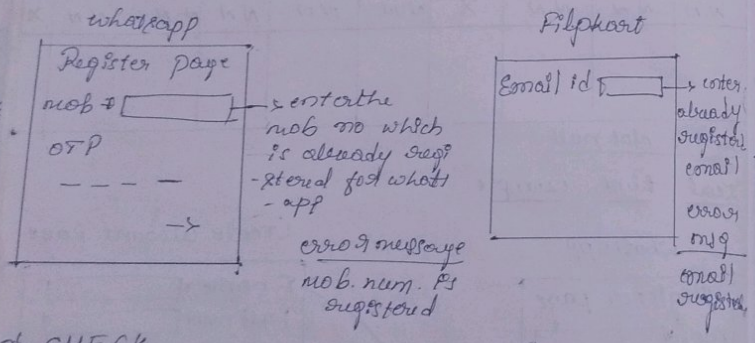
→ If a column is assigned with a constraint as "unique" it will not accept duplicate values but it will accept NULL values.

e.g:

Student

Sname	USN	branch	PCT	DOB	gender	mob	address	email	% of
varchar (20)	varchar (10)	varchar (10)	number (4,2)	date	varchar (10)	number (10)	varchar (10)	varchar (20)	number (4)
	UNQ					UNQ		UNQ	

Real time example



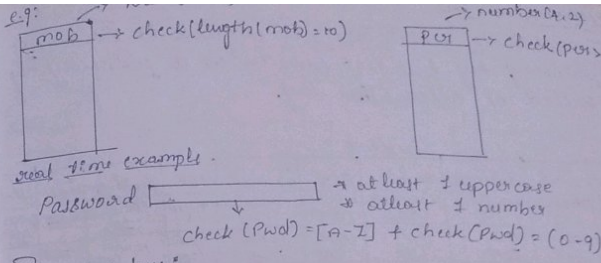
3) CHECK

Check is a constraint which is used for additional validation

Syntax

column_name check (cond)

e.g:



Primary key:

→ Primary key is a constraint given to column which is used to identify or represent a record uniquely.

which column (or) attribute can eligible to become Primary key:

* A column (or) attribute which is "mandatory" and it should consists of "distinct value" that is combination of "NOT NULL" and "UNIQUE" constraint for those column will eligible to become primary key.

Characteristics of primary key:

* PK is a combination of NOT NULL and UNIQUE constraint.
* If a column is assigned with PK constraint then that column will not accept "NULL" and "duplicate values."
* In a table more than one column can eligible to become a PK but only one column can be considered as a PK.
* In a table PK is not mandatory but it is highly recomended.
* If a table has PK then that table is called as strong entity.

→ If a table ~~has PK~~ doesn't have PK then that table is called as week entity.

Candidate key:

The column's which are all eligible to become a

Scanned by CamScanner

Alternative key

The columns which are all eligible to become a primary key but not choosen as a primary key is called as alternative key.

e.g

Student

Name	mob	aadhar	email	pan	branch	tel	gender	dob	USN
varchar (20)	number (10)	varchar (20)	varchar (20)	number (4,2)	varchar (20)	num (4)	varchar (20)	date	varchar (20)
NN UN	NN UN	NN UN	NN UN	NN UN	NN UN	NN UN	NN UN	NN UN	NN UN

=> According to above example "mob, email, USN" are the candidate key

* If we choosen "mob" -> PK then "email, USN" -> alternate key.
-> If we choosen "email" -> PK then " mob/USN -> —u—
-> If we choosen "USN" -> —u— " email, mob" -> —u—

5) Foreign key:

Foreign key is a referential integrity constraint, it is a column/attribute which belongs to another table.

Why we should use foreign key:

* It is used to establish the relationship b/w the tables. (the relationship is called as works for)

Characteristics of foreign key:

* The table in which foreign key is present is called as child table / slave table.
* The table to which actually foreign key is belong is called as parent table / master table
* In table we can have more than one column as foreign key.
* If a column is assigned with foreign key then that column will accept both NULL and duplicate values but it will not accept the value which is not present in parent table column
* only a PK column can be considered as FK.

example.

establishing relationship (works for)

emp

empno (PK)	ename	job	sal	deptno (FK)
1	Harshitha	manager	50000	30
2	Geha	Lead	40000	10
3	Chandan	Co-operator	25000	50
4	Keerthi	S.W.E	80000	40
5	Akash		25000	
6	Shasrath	Sr Engg	35000	
7	Deepa	Counselor	45000	30

Child table / slave table

dept

deptno (PK)	dname	loc
10	HR	Raj
20	HR	Raj
30	Counsellor	BSR
40	teaching	Raj
50	Attendance	Raj

Parent / master table

FK will not accept a value which is not present in parent table(dept)c.n(deptno)

example for more than one FK in a table:

Customer

cid (PK)	cname	add

Product

Pid (PK)	Pname	Price

order details

oid (PK)	cid (FK)	Pid (FK)

Single-Row Function

=> SRF will accept 'n' record as i/p and it will provide 'n' as record as o/p
=> SRF will execute row by row [record by record]
=> we can write single row function in where clause
=> we can pass c.n/expression/literals as an argument to SRF
=> SRF are nested
=> SRF are classified into 5 types

1) character function
2) Number function
3) General function
4) Date function.
5) Conversion function.

1) Number function:

1. mod():- This function is used to obtain modules of given number.

Syntax: mod(m,n) -> modules of m and n [m/n]

2021/5/18 21:39

Scanned by CamScanner

o/p → mod (10, 2)

1) WAD employee details, who is getting even salary.

```
Select *
from emp
where mod(sal, 2) = 0;
```

```
Select *
from emp
where mod(sal, 2) != 0
```
→ odd digit salary

2. round(): This function is used to round of the given number

Syntax: round(argument)

e.g: → select round(49.49) from dual;
o/p → 49

Select round(49.51) from dual;
o/p → 50.

NOTE:
* If decimal number >= 5 then it will round of to next no.
→ If decimal < 5 then it will round of to same no.

3) Trunc(): This function is used to round of the given number always to the lowest / same number.

Syntax: trunc(argument)

e.g: Select trunc(49.51) from dual;
o/p → 49

Select trunc(49.49) from dual;
o/p → 49.

NOTE:
* If decimal number is >=5 / <=5 then it will round of to same number only.

4) power(): This function will used to obtain m to the power n

Syntax power(m,n)

e.g: Select power(2, 3) from dual;
o/p → 8

$\Rightarrow 2^3 = 8$

5) SQRT(): This function is used to obtain square root of given number.

Syntax: sqrt(m)

e.g: Select sqrt(4) from dual;
o/p → 2

$\Rightarrow \sqrt{4} = 2$

2) General Function

ename	sal	comm	total_sal
smith	800		

→ A/c to above query we will get wrong o/p because if we perform any arithmatic operation with null the result will be null only. to handle null values we will go "NVL()"

Definition of NVL(): NVL() is use to handle null values.

Syntax: NVL(arg1, arg2)

→ If argument 1 (arg1) is null value then NVL() will return the value present in arg2
→ If argument 1 is not a null value then NVL() will return arg1 it self.

1) WAD ename, sal, comm, total sal (sal + comm) of all the employee

```
Select ename, sal, comm, sal + nvl(comm, 0) total_sal
from emp;
```

2) NVL2 function: It is also used to handle null values.

Syntax: NVL2 (arg1, arg2, arg3).

→ If arg1 is null
→ If arg1 is not a null value then NVL2() will return the value present in arg2.

1) WAD employee name, salary, commission, total salary only if a employee earning commission give 30% increment to commission & find total salary.

```
Select ename, sal, comm, (sal + NVL2(comm, comm*1.3, 0))
from emp;
```
total_sal

NOTE:
In NVL & NVL2 both function arg1 is comparing.

3) Case(): Case() is similar to if else statement.

```
Select case
when cond1 then 'result1'
when cond2 then 'result2'
when cond3 then 'result3'
```

2021/5/18 21:20

Scanned by CamScanner

o/p as Poor salary else if the salary in a range of 1501 to 2500 then display o/p as ok-ok salary else if the salary in a range of 2501 to 3500 then display the o/p as average salary else display the o/p as richest person.

```
select ename, sal, case
from when sal between 500 and 1500 then 'Poor sal'
when sal between 1501 and 2500 then 'ok-ok salary'
when sal between 2501 and 3500 then 'avg salary'
else 'richest person'
end from emp;
```

Palindrome string

```
select ename, case
when ename = reverse(ename) then 'Palindrome string'
else 'not a palindrome string'
end from emp;
```

3) Date()

1) sysdate: It is used to obtain system current date.

Syntax: sysdate

1) WAD employee name, Date of joining, no of experiance of all employees

```
select ename, hiredate, round ((sysdate - hiredate)/365)
from emp;
```

exper

2) systimestamp: It will obtain the system current date along with time.

Syntax: systimestamp

e.g: select systimestamp
from dual;

o/p: 13-may-21 03.43.45.642000 PM +05:30

3) months_between(): It is used to obtain no of months b/w two dates.

Syntax: months_between (Date1, Date2)

e.g select months_between (sysdate, '12-may-20')
from dual;

o/p :- 12.053501

e.g: select ename, hiredate, round (month_between (sysdate, hiredate)/12) exper

Syntax: last_day (date)

e.g: select last_day(sysdate)
from dual;

o/p: 31-may-21

4) extract: This function is used to extract date comp-nent

Syntax: Extract (Day/month/year from date)

e.g: select extract (Day from sysdate)
from dual;

o/p: 13

e.g: select extract (month from sysdate)
from dual;

o/p → 5 → may month

e.g: select extract (year from sysdate)
from dual;

o/p → 2021

1) WAD employee details only the employee should be hired in a month of febaraary without using the operator.

```
select *
from emp
where extract (month, hiredate) = 2 from hiredate) = 2;
```

2) WAD employee details, employee should be hired in feb-81

```
select *
from emp
where extract (month from hiredate) = 2 and
      extract (year from hiredate) = 1981;
```

NOTE: when we are extracting month, it will extract month in number & when we are extracting year & it will extract in 4 digit [eg: 1981]

4) Conversion(): ~~This function is used to~~

1. Case manipulation function:
   a. upper()
   b. lower()
   c. initcap()
2. to_char(): This function is used to convert date format to character format

Syntax:

Scanned by CamScanner

→ select to_char (sysdate, 'd')
from dual;
o/p → 5

DD: This model is used to obtain a day in number
e.g: select to_char (sysdate, 'dd')
from dual;
o/p → 13 → Today date

Dy: This model is used to obtain a day begining 3 alphabet
e.g: select to_char (sysdate, 'dy')
from dual;
o/p :- thu → Thursday

DAY: This model is used to obtain a day in complete alphabet.
e.g: select to_char (sysdate, 'day')
from dual;
o/p → thursday.

MM: This model is used to obtain a month in digit.
e.g: select to_char (sysdate, 'mm')
from dual;
o/p → 5

MON: This model is used to obtain a month in begining 3 alphabet
e.g: select to_char (sysdate, 'mon')
from dual;
o/p :- may

Month: This model is used to obtain a month in complete alphabet
e.g. select to_char (sysdate, 'month')
from dual;
o/p :- may

YY: This model is used to obtain a year last 2 digit
e.g: select to_char (sysdate, 'yy')
from dual;
o/p → 21

YYYY: This model is used to obtain a year complete 4 digit
e.g: select to_char (sysdate, 'yyyy')
from dual;
o/p: 2021

YEAR: This model is used to obtain a year in complete alphabet
e.g: select to_char (sysdate, year)
from dual;
o/p :- twenty twenty-one

5) Character function
1) length()

o/p : 9

2) WAD only the employee name should be exactly 5 characters without using like operator.
select *
from emp
where length (ename) = 5;

2) Reverse(): This function is used to obtain reverse the given string
Syntax: reverse('string'/column')
e.g: select reverse ('soumalaya')
from dual;
o/p: 'ayalamuos'

2) select ename, reverse (ename)
from dual;

3) Concat(): This function is used concatinate the given string
Syntax: concat(arg1, arg2)
e.g: select concat ('don', 'key')
from dual;
o/p → donkey

NOTE: concat() function will combine only 2 arg. at a time.

2) select concat('don', concat ('key', 'nitish'))
from dual;
o/p → donkey nitish

1) WAD the o/p in the below mentioned format
'Hello' x 'your' 'salary' is' y 'and working as' z
x → ename, y → sal, z → job
select concat ('hello', concat (ename, concat ('your salary is', concat (sal, concat ('and working as', job)))))
from emp;

4) Replace(): This function is used to replace old character with new character
Syntax replace (column, 'string', 'oldch', ['new ch'])
NOTE:
arg3 → New character is optional.
e.g: select replace ('jspiders', 'j', 'q')
from dual;
o/p :- qspiders
→ select replace ('jspiderj', 'j', 'q')
from dual;
o/p → qspiderq

NOTE:

2021/5/18 21:39

Scanned by CamScanner

⇒ SUBSTR(): Full form → substring.

This function is used to extract character/sub string from the given string.

Syntax: ~~(ent'str~~ substr(ch'string, start_pos[length])

e.g: -ve ← - - - - ~~left to right to~~ left to right

-5 -4 -3 -2 -1
A L L E N
1 2 3 4 5

+ve - - - - → ~~Right to~~ left to Right

substr('allen', 1, 3) → all
substr('allen', 2, 3) → lle
substr('allen', 1, 1) → a
substr('allen', 4, 3) → en
substr('allen', 5, 3) → n
substr('allen', 1) → allen → If we not mention the length it will extract complete string
substr('allen', 3) → len
substr('allen', -3, 2) → le
substr('allen', -4, 5) → llen

NOTE:

⇒ In substr, if start position is +ve (or) -ve Extraction will happen from left to right only.

⇒ If we are not specifing length substr will extract till last character

1) WAP employee name only the employee name should begins with 'a' character.

```
Select ename
from emp
where lower(substr(ename, 1,1)) = 'a';
```

or

```
where substr(lower(ename), 1,1) = 'a';
```

2) WAP employee name, only employee name should be ending with character S

```
Select ename
from emp
where substr(lower(ename), -1) = 's';
```

(or)

```
where substr(lower(ename), length(ename)) = 's';
```

3) WAP employee ename, only employee name should begins with vowels

```
Select ename
from emp
where substr(lower(ename), 1,1) in ('a','e','i','o','u');
```

→ For vowels

```
where substr(lower(ename), 1,1) not in ('a','e','i','o','u');
```

→ for consonants

4) WAP employee details only employee 1st character should be same as designation 2nd character.

```
Select *
from emp
where substr(lower(ename), 1,1) = substr(lower(job), 2, 1);
```

5) WAP employee details only name begining 3rd character should be 'a'.

```
Select *
from emp
where substr(lower(ename), 3,1) = 'a';
```

6) WAP the employee name, in such way 1st half in upper-case and 2nd half in lower case.

```
Select ~~ename~~
from emp
~~where~~ upper(substr(ename, 1, length(ename)/2)) || lower(
  substr(ename, length(ename)/2 +1))
from emp;
```

6) INSTR(): → (In string) (or) (In the string)

This function is used to search the character (or) substring from the given string.

Syntax: INSTR(ch'string, 'char/'ss', [S_pos, occurence])

~~space~~ → space, starting pos, no of times

e.g: ← - - - - - -ve right to left

-5 -4 -3 -2 -1
A L L E N
1 2 3 4 5

- - - - - - → +ve left to right

instr('allen', 'l', 1, 1) → 2
instr('allen', 'l', 1, 2) → 3
instr('allen', 'l', 2, 1) → 2
instr('allen', 'l', 2, 2) → 3
instr('allen', 'l', 2, 3) → 0
instr('allen', 'l', 1) → 2
instr('allen', 'l', 4) → 0
instr('allen', 'l') → 2
instr('allen', 'l', -3, 1) → 3
instr('allen', 'l', -3, 2) → 2
instr('allen', 'l', -3, -3) → 0
instr('allen', 'l', -3) → 3

NOTE:

* In ~~substr~~ instr if starting position is +ve extraction will happen from left to Right
* In Instr if starting position is -ve extraction will

2021/5/18 21:40

Scanned by CamScanner

happen from right to left but index will be display in +ve.

* An instr if we are not specifying starting position & occurance by default starting position will be 1 & occurance will be 1

Subquery in From clause

1) WQ to disply 1st max without aggregate function.

```
select sal
from (select sal
      from emp
      where order by sal)
where rownum = 1;
```

2) 1st min sal

```
select sal
from (select sal
      from emp
      order by sal desc)
where rownum = 1;
```

3) select sal, ename
from (select ~~emp~~ sal
from emp
order by sal desc)

} wrong query ∵ only sal is present in from clause inner query.

=> o/p -> error

Normalization.

=> Normalization is a database design technique (table design technique) which organize the data without any redundancy (duplicate data) and dependency. (or)

=> Normalization is a process of splitting a large table into several smaller table and establishing a relationship b/w table is called as normalization

=> E.F codd is father of normalization & he proposed a "Theory of normalization" which consists of several normal form

1 NF ANF BCNF (Boyce codd normalization) -> 3.5 NF
2 NF 5NF
3NF 6NF

It is also know as 3.5NF

1 NF: Each and every cell in a table should consists of single value in a table.

Student

Sname	gender	Age
Akash	male, female	15
Amith	male	20, 10

wrong

Sname	gender	Age
Akash	male	15
Akash	female	15
Amith	male	20
Amith	male	10

Conclution: If a cell consists of multiple values we need to split into multiple rows.

-> Each and every record should be unique in a table

Student

Sname	Gender	Age
Supriya	female	20
Supriya	female	20

Conclution - If a table consists of same record then to identify uniquely we will add a primary key column.

2NF:- Table should be in 1NF

* A table should consists of primary key
* A table can consists of "Functional Dependency"
* A table should not consists of "partial Dependency".

Functional Dependency

=> all non-primary key columns is depending on a single primary key column is called as functional dependency.

Partial Dependency.

=> A non-primary key column is depending on another non-primary key and ~~col~~ inturn it is depending is on PK column is called as Partial dependency.

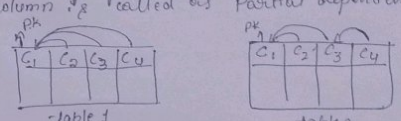
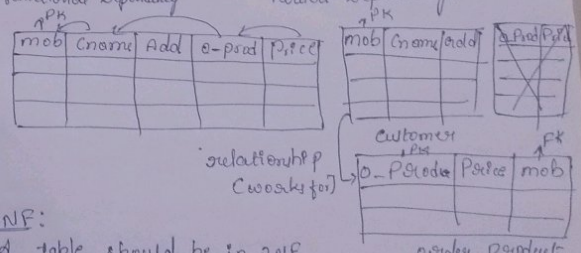


table 1 — functional Dependency

table 2 — Partial Dependency.

e.g

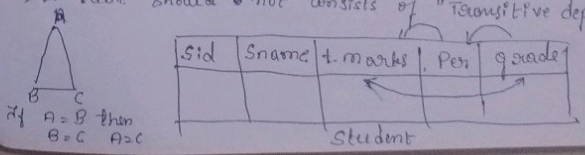


relationship (works for)

3NF:

=> A table should be in 2NF

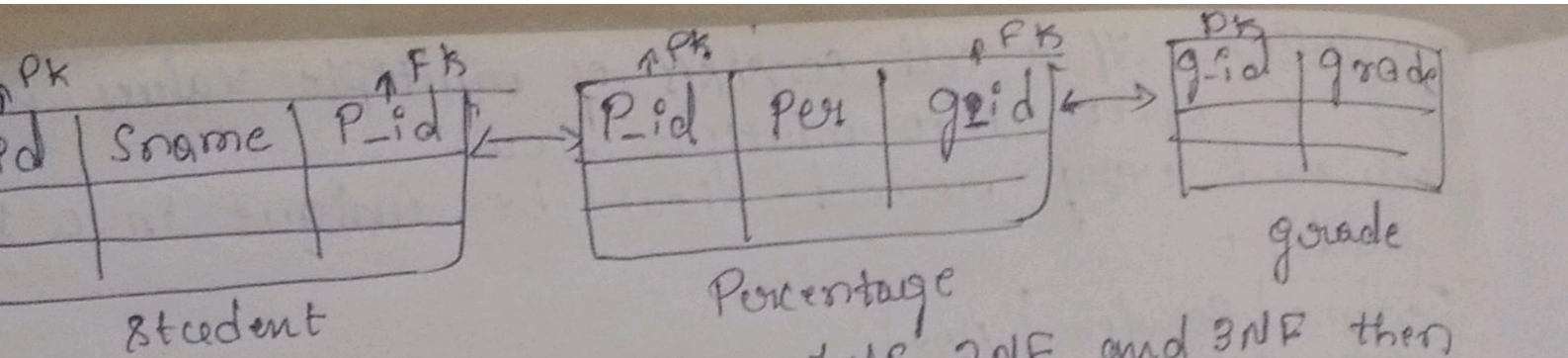
=> A table should not consists of "Transitive dependency".



If A = B then B = C A = C

2021/5/18 21:40

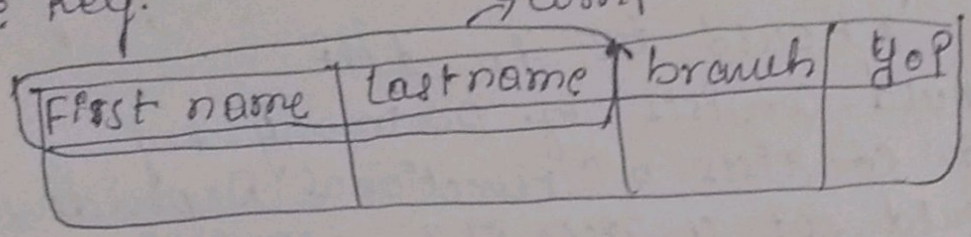
Scanned by CamScanner



E: If a table is satisfying 1NF, 2NF and 3NF then le is in normal form [No dependency, No redundency].

mposite key: Combination of multiple column and identi ying each and every record uniquely is called as mposite key.

g:



2021/5/18 2

Scanned by CamScanner

SQL Statement / languages

We have 5 different languages in SQL

1. Data Definition Language

1. Create: It is used to create a table or an object

Syntax:

```
Create table tablename(
colname1 datatype not null/null,
colname2 datatype not null/null,
:
Constraint constraint_name Unique(colname),
Constraint constraint_name primary key(col
-name),
Constraint constraint_name check (condn).
Constraint constraint_name Foreign key
(colname) References parent table(colname)
);
```

2. Alter: Alter statement is used to modify the table structure

1> To add a column

```
Alter table tablename
Add column_name datatype constraint;
```

2> To drop a column

```
Alter table tablename
drop column column_name;
```

3> To Rename a column

```
Alter table tablename
Rename column old column_name to new colum
-n_name;
```

3. Drop: Drop is used to delete the table permanent-ly from a db.

```
Drop table tablename;
```

from oracle 10g onwards we have a concept of recycle bin. ~~when~~ we can get back the table from recycle bin by using flashback statement.

Scanned by CamScanner

2021/5/18 21:40

Flash back table tablename to before drop:
we can also delete the table from recycle bin by using purge statement.
Purge table tablename;

4) Truncate: ~~table~~ tablenames
Truncate statement is used to delete all the records from table permanently
Truncate table tablename;

5) Rename: rename statement is used to change table name permanently.
Rename oldtablename to newtablename;

Data manipulation Language

1) Insert: Insert statement is used to add a record in a table
Insert into tablename values (v1, v2, v3...);
Insert into table name (col1, col2,.) values(v1,v2);

2) Update: update statement is used to modify the record present in a table
Update tablename
set col1 = val1, col2 = val2 ...
where <condn>;

3) Delete: Delete statement is used to delete the records present in a table [specific record]
Delete from tablename
where <condn>;

Transaction Control language

1) Rollback: roll back statement is used to undo the DML changes.
Roll back;

2) Commit: Commit statement is used to save the

back statement
position.
Savepoint savepoint_name;
Rollback to savepoint_name;

Data Control language (DCL)

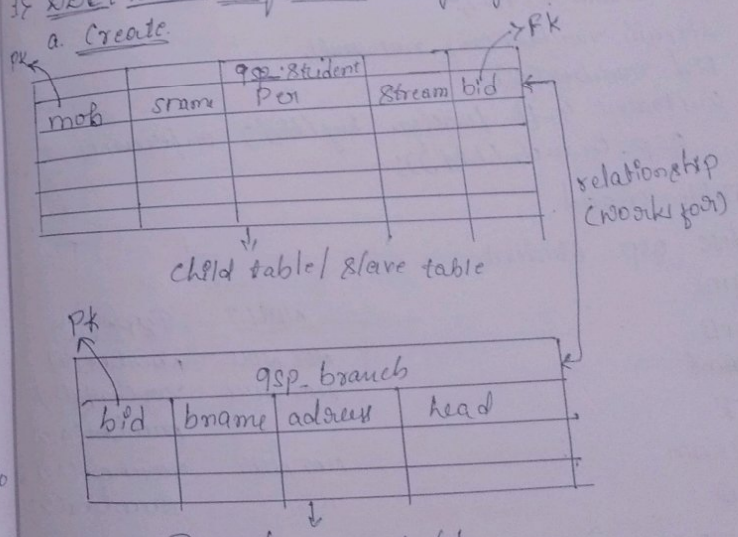
1. Grant: It is used to give permissions to the user. Grant sql statements on tablename to username;

2. Revoke: It is used to get back the permissions from the user Revoke sql statements on tablename from username;

SQL statements / SQL Language: [execution part]

1) DDL [Data Definition Language]

a. Create



Scanned by CamScanner

Create table gsp_branch (
bid number(5) Primary key,
bname varchar(20) not null,
address varchar(20) not null,
head varchar(20));

Table Created

desc gsp_branch

name	Null?	Type
BID	NOT NULL	Number(5)
BNAME	NOT NULL	varchar2(20)
ADDRESS	NOT NULL	varchar2(20)
HEAD		varchar2(20)

Create table gsp_student (
mob number(10) Primary key,
Sname varchar(20) not null,
Per number(4,2),
Stream varchar(20) not null,
bid number(5),
constraint q_fk foreign key(bid) references
gsp_branch(bid));

Table Created.

desc gsp_student

Name	NULL?	Type
MOB	NOT NULL	Number(10)
SNAME	NOT NULL	varchar2(20)
PER		Number(4,2)
STREAM	NOT NULL	varchar(20)
BID		Number(5)

b. ALTER:

Case 1: To add a new column.

desc gsp_student

Name
MOB
SNAME
PER
STREAM
BID

alter table gsp_student
add email varchar(20) not null;

Table altered.

desc gsp_student

Name
MOB
Sname
Per
Stream
bid
email.

Case 2: To drop/delete a column

alter table gsp_student
drop column stream;

Table altered

desc gsp_student

Name
mob
Sname
Per
Bid
email

Case 3: To Rename a column.

alter table gsp_student
rename column per to cgpa;

Table altered.

desc gsp_student

Name
mob
Sname
Cgpa
Bid
email

2021/5/18 21:41

Scanned by CamScanner

NOTE : Any changes/modification done where. by using alter statement i.e w.r.t c.n

Ex. drop:-

desc demo

name

Sname

⇒ select * from demo;

Sname

a

b

⇒ drop table demo;

Table dropped

⇒ desc demo

error: object demo doesnot exist

⇒ select * from demo;

error : table or view doesnot exist

Falsh back

⇒ flash back table demo to before drop;

flash back completed.

⇒ desc demo

name

Sname

⇒ select * from demo;

Sname

a

b

Purge:

⇒ drop table demo;

Table dropped.

⇒ Purge table demo;

Table purged.

⇒ flashback table demo to before drop;

~~flash back complete~~

flash back table demo to before drop

error - object not in recycle bin.

⇒ desc demo

error - object demo doesnot exist.

⇒ select * from demo;

error - table/view doesnot exist.

Truncate

⇒ desc demo

name

Sname

⇒ select * from demo;

Sname

a

b

⇒ truncate table demo;

Table truncated.

⇒ desc demo

name

Sname

⇒ select * from demo;

no rows selected.

* Diff b/w Drop & Truncate

Drop	Truncate
⇒ It is use to delete whole table permanently	⇒ It is used to delete all records permanently.
⇒ Table structure will not be present.	⇒ Table structure will be present.
⇒ Flashback & purge will support	⇒ flashback & purge will not support

Rename:

⇒ desc demo

name

Sname

⇒ rename demo to sample;

Table renamed.

⇒ desc demo

error object demo doesnot exist

⇒ desc sample

name

Sname.

what is difference b/w Rename & Alias name

Rename	Alias name
* It is used to change the table name permanently	* It is used to change c.n/expression for the o/p present
* It is a permanent changes	* It is a temporary changes

2021/5/18 21:41

Scanned by CamScanner

2) DML (Data Manipulation Language):-

1) Insert:

=> select * from sample:
no rows selected
=> insert into sample values ('akash', 'cs'); -> approach 1
1 row created
=> insert into sample (sname, branch) values ('amith', 'cs') -> 2nd approach
1 row created
=> insert into sample values (&sname, &branch); -> 3rd approach (knowledge purpose)
enter value for sname: 'queen'
enter value for branch: 'eee'

-> / [backward slash]
enter value for sname: 'sourth'
enter value for branch: 'me'
=> select * from sample;

sname	branch
akash	cs
amith	ps
queen	eee
sourth	me

approach - 1 -> we should enter the all column's values
approach - 2 -> we can enter values for required columns
approach - 3: we should enter the values for all the column's in the same order as in the structure

=> insert into sample values ('akash');
error -> not enough values
=> insert into sample values ('akash', 'cs', 90);
error -> too many values.

2) Update:

=> select * from sample;

sname	branch
akash	cs
amith	ps
queen	eee

=> update sample
set branch = 'me'
where lower(sname) = 'queen';
1 row updated.

=> select * from sample:

sname	branch
akash	cs
amith	ps
queen	me

=> update sample
set sname = 'amith.p', branch = 'eee'
where lower(sname) = 'amith';
1 row updated
=> select * from sample;

sname	branch
akash	cs
amith.p	eee
queen	me

=> update sample
set branch = 'me';
3 rows updated
=> select * from sample;

sname	branch
akash	me
amith.p	me
queen	me

3). Delete

=> delete from sample
where lower(sname) = 'queen';
1 row deleted
=> select * from sample:

sname	branch
akash	me
amith.p	me

=> delete from sample;
2 rows deleted.
=> select * from sample
no row selected.

-* Diff b/w drop & delete

Drop	Delete

2021/5/18 21:41

Scanned by CamScanner

* Diff b/w drop & Delete

Drop	Deleted
=> It is used delete whole table	=> It is used to delete specific record
=> table structure will not be present	=> table structure will be present.
=> we cannot write where clause	=> we can write where clause
=> we can get back the dropped table by using flashblack statement	=> we can get back the deleted record by using rollback statement.
=> It is a permanent change	=> It is a temporary changes
=> It is belongs to DDL language	=> It is belong to DML Language

* Difference b/w truncate & delete

Truncate	Delete
=> It is used to delete all records	=> It is used to delete a specific record.
=> we cannot write where clause	=> we can write where clause
=> ~~we can~~ It is a permanent changes	=> It is a temporary changes
=> It is belongs to DDL language	=> It is belongs to DML language.
=> we cannot get back the deleted records	=> we can get back the deleted records by using rollback.

3> DCL [Data Control Language)

1> Grant

=> show user
User is "SCOTT"
=> Select * from regions;
error -> table (or) view doesnot exist
=> conn
~~Enter username:~~ hr Login as hr user.
=> ~~Select * from~~ show user
User is "hr"
=> Select * from regions;

o/p Region_id	Region_name
6 ...	dubai ...

=> grant select, insert on regions to scott;
grant succeeded.
=> Login as scott user
=> show user
user is "SCOTT"
=> Select * from hr.regions;

region_id	region_name
6	dubai
⋮	⋮

=> insert into hr.regions values (8, 'USA');
1 row created.
=> select * from hr.regions;

region_id	region_name
6	dubai
⋮	⋮
8	USA
⋮	⋮

=> Delete from hr.regions;
=> error -> insufficient privileges
=> ~~Log~~

2> Revoke

=> Login as hr user
show user
user is "HR"
=> Revoke select, insert on regions from scott;
Revoke succeeded.
=> Login as scott user
=> show user
user is "scott"
=> Select * from hr.regions;
error -> view (or) table doesnot exist

Views :-

=> Views is like a table but it is not a table.
=> views will be created by selecting specified/required no of c.n's is called as views.
=> A view will be created by a table is called as target table.
=> Views are classified into 2 types

2021/5/18 21:41

Scanned by CamScanner

1. Simple view
2) complex view

1) Simple view:
* A view which is created by single target/target table is called as simple view.

Syntax:
Creat view view_name
as
select c.n'e
from table;

e.g:
=> create view emp_view
as
select ename, sal, job
from emp;
view created
=> select * from emp_view;

o/p ename	sal	job
smith	800	clerk
:	:	:

2) complex view
=> a view which is created by more than one table/target table is called as complex view
=> In complex view we will specify joining cond.

Syntax:
Create view view_name
as
select c.n'e
from table1, table2
where <joining cond>;

e.g:
=> creat view comp_view
as
select ename, job, dname, loc
from emp, dept
where emp.deptno = dept.deptno;
view created
=> select * from comp_view;

o/p ename	job	dname	loc
clark...	manager...	Ac...	New York...

=> Grant select on emp_view to hr;
grant succeeded.
=> conn
enter user-name: hr
enter password:
connected
=> select * from scott.emp_view;

ename	sal	job
smith	800	clerk
:	:	:

4. TCL [Transaction Control Language]:-

NOTE:
* any changes/modification done w.r.t column by using alter statement
* any changes/modification done w.r.t row by using Dml language.
* any changes/modifications done by using DDL language is a permanent changes.
* any changes/modifications done by using Dml language is a temporary changes. To make it as permanent changes we should use TCL language.

1) Commit:
e.g: => insert into sample values('a');
1 row created
=> insert into sample values('b');
1 row created
=> select * from sample;

o/p sname
a
b

=> commit;
commit completed.

2) Roll back:
=> insert into sample values('a');
1 row created
=> insert into sample values('b');
1 row created
=> select * from sample;

sname
a
b

2021/5/18 21:41

Scanned by CamScanner

=> rollback
roll back completed.
=> select * from sample;
no rows selected.
=> insert into sample values ('a');
1 row created
=> insert into sample values('b');
1 row created
=> select * from sample;
Sname
a
b
=> commit;
commit complete.
=> select * from sample;
Sname
a
b
=> roll back
roll back complete
=> select * from sample;
Sname
a
b

NOTE: After Dml changes (insert, delete, update) if we commit [save] and then if we do roll back, it will not undo any Dml changes.

3) Save point:
=> insert into sample values ('a')
1 row created
=> insert into sample values ('b')
1 row created
=> insert into sample values ('c')
1 row created
=> select * from sample;
Sname
a
b
c
=> Save point a;
Save point created.
=> update sample
set sname = 'd'
where sname = 'b';
1 row updated
=> select * from sample;
Sname
a
d
c
=> savepoint b;
Savepoint created.
=> delete from sample
where sname = 'c';
1 row deleted.
=> select * from sample;
Sname
a
d
=> save point c;
Save point created.
=> Roll back to c;
Roll back completed.
=> select * from sample;
Sname
a
d
=> Roll back to b;
Rollback complete.
=> select * from sample;
Sname
a
d
c
=> roll back to a;
roll back complete.
=> select * from sample;
Sname
a
b
c.

2021/5/18 21:42

Scanned by CamScanner