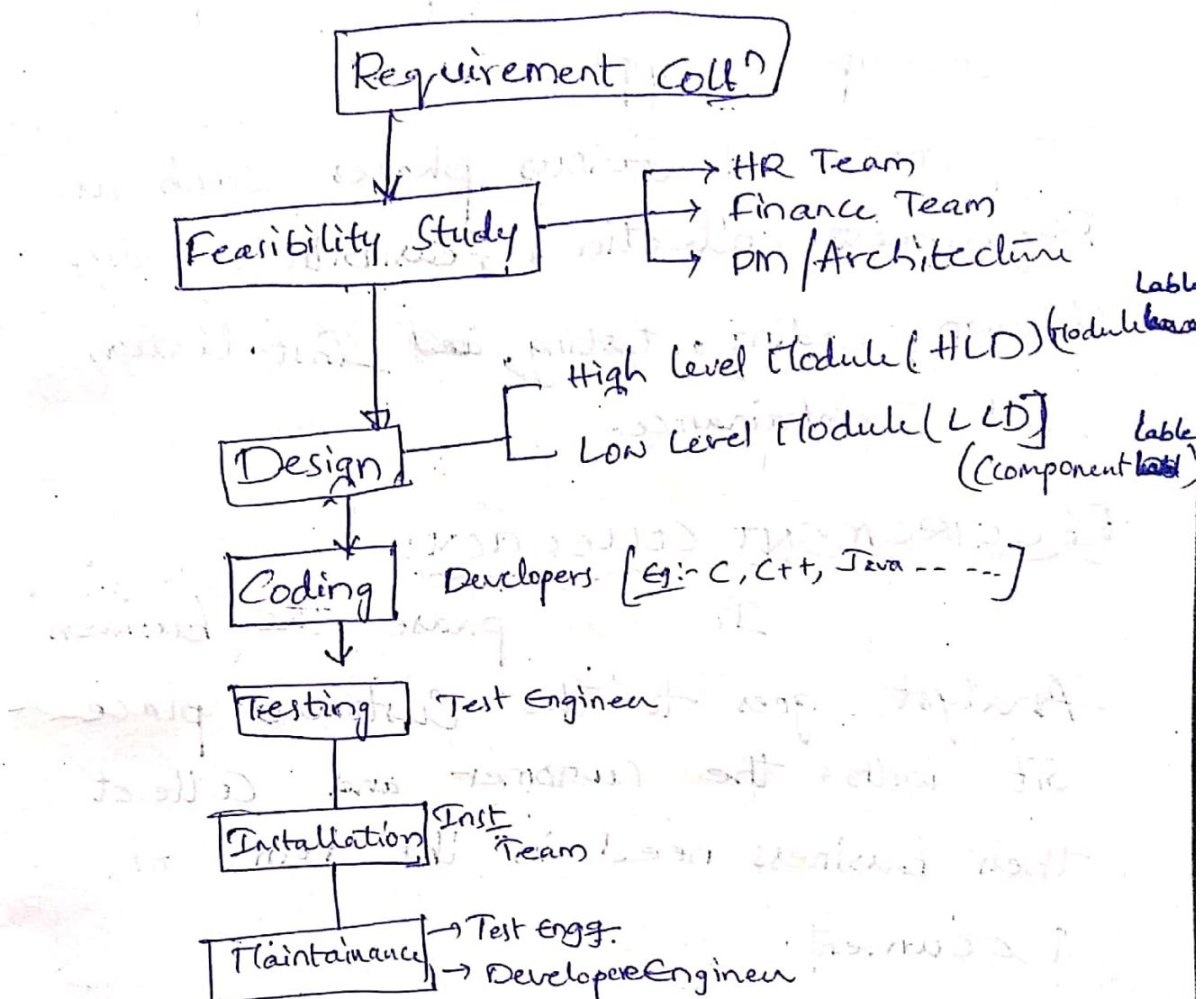
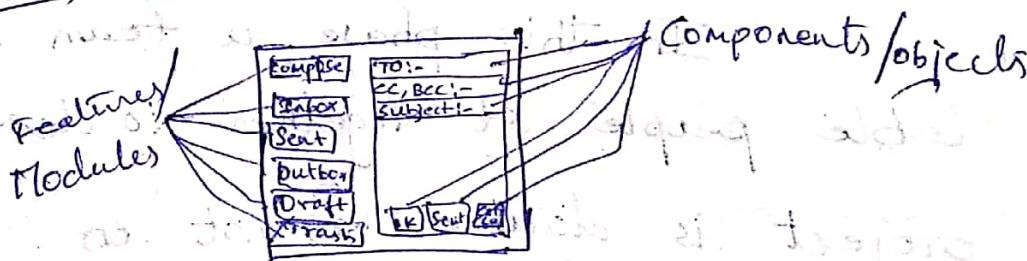


1/10/2023

① SDLC :-



Gmail :-



- SDLC Stands for Software Development Life Cycle. It is a step by step process to develop an application.
- It consists of various phases such as Requirement Collection, Feasibility Study, Design, Coding, testing, ~~and~~ Installation and Maintenance.

REQUIREMENT COLLECTION:-

In this phase the business Analyst goes to the customer's place sit with the customer and collect their business need in the form of Document.

FEASIBILITY STUDY:-

In this phase a team of high level people sit together to decide the project is doable ~~or~~ ^{or} not on the basis of some factors.

DESIGN:-

Design means blue print of

the application. Design are of two types. They are (a) High Table Design

(a) High Table Design. It is also known as module table design and it is done by the Architech.

(b) Low Table Design. It is also known as component label design and it is done by the managers.

CODING:- In this phase the developer start writing the code by choosing one programming language. This process of writing the code continue until the application is Ready.

TESTING:- Once the application is ready the test Engineers start checking the application. While checking the application the test engineers may find some defects. All such defects are hand it over to the developers. The developer does the

necessary code changes and gives to test engg. This process is continuous until the app is stable.

INSTALLATION:-

In this phase the final Stable App is moved from the Company environment to Client (or) Customer Environment.

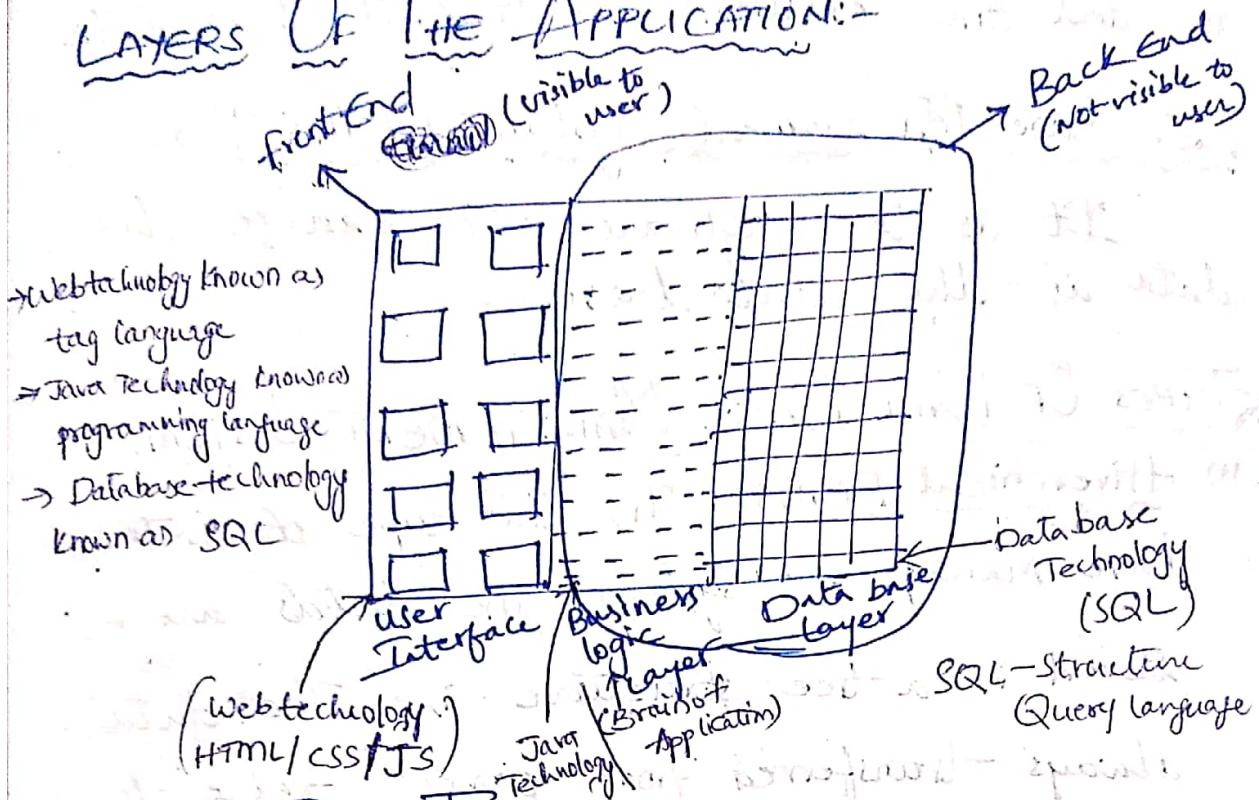
MAINTAINANCE:-

Once the application is deployed into the ~~production~~ customer's machine customer starts using it. While using the application they may face some issues.

In order to overcome such issues we go for maintenance.

8/10/23

LAYERS OF THE APPLICATION:-



DATA BASE TECHNOLOGY:-

DATA:- It is the collection of useful information which need to be saved for future references.

DATA BASE:- It is the warehouse to store the data of the users. The data & database are stored in the form of tables.

TABLE:- It is the collection of rows and columns. The horizontal section is known as rows whereas the vertical section is known as columns.

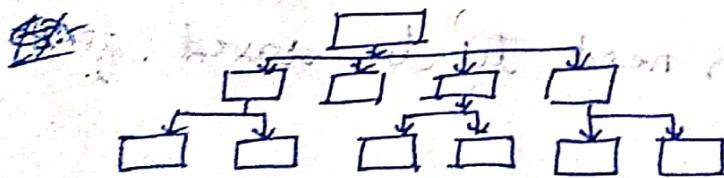
Cell:- ~~Each~~ Cell is the intersection of one row and one column. Each cell has data.

Data Base Management System (DBMS):-

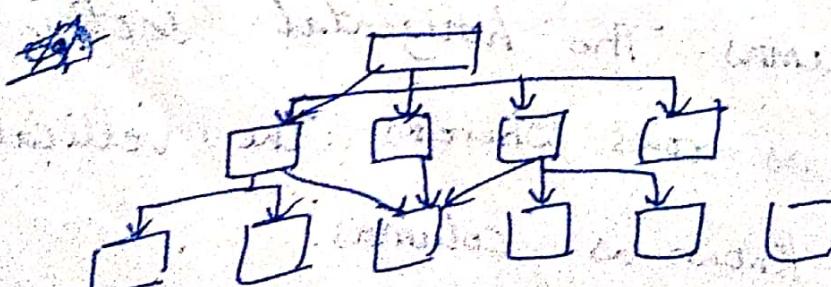
It is the software to manage the data in the data base.

* Types Of Data Base Management System:-

(1) Hierarchical DBMS:- In this type of data base management system the data's are stored in a tree structure i.e; The data always transferred from parent table to child table. It is unidirectional flow of data.

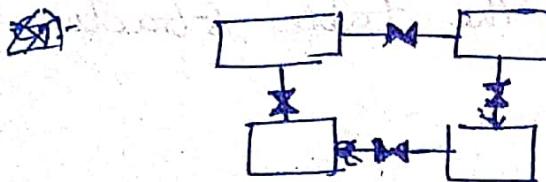


(2) Network DBMS:- In this types of data base management system the data may transfer from multiple sources to single table. It is known as tree structure.



(3) RELATIONAL DBMS:-

* In this type of data base management system the data may transfer b/w two tables with some code rules. It consume less memory as well as the execution is faster.



VENDORS AND THEIR PRODUCTS

vendors of RDBMS:-

→ Oracle → 10g/11g

→ IBM → DB2

→ S.A.P → Sybase

→ Open Source → MySQL

* ORACLE 10g/11g:- Once ~~they~~ installed 10g(11g)

by default SQL plus gets installed.

→ SQL plus gives the platform to execute the Queries.

→ The existing user name is Scott and password is tiger.

① write a Query to display the list of table present in the database

Ans:- Select *
from cat; (or) Select *
 from tab;

② write a Query to display the table structure

Ans:- Syntax: Desc ~~table name~~;

Ans:- Desc emp;

Desc ~~Dept~~ Dept;

③ write a Query to get the complete data from a table?

Ans:- Syntax → Select *

from table name;

* → Complete data
Syntax → format

Eg:- ① Select *
 from emp;

Eg:- ② Select *
 from dept;

Note:- ① SQL is not Case Sensitive. we can write the Query in upper Case (or) lower Case.

Note-2:- Every SQL Query ends with Semi Column

④ write a query to display one column data

Ans:- Sp → Select ~~one~~ column name
 from ~~table~~;

Eg:- ① Select Ename
 from EMP;

Eg:- ② Select dept no
 from emp;

Q) write a query to display two(or) more column data

Ans:- Syo:- Select column name 1, select column 2
from table;

Eg:- Select Ename, ~~Deptno~~ job
from ~~dept~~ emp;

ALASING:- Alasing is used for renaming the column name (or) table name. It is a temporary

Syntax: → For column aliasing we use 'as' as a keyword. Change
→ For table aliasing we use 'AS' as Keyword.

Eg: Select Column name as new-name / For Table aliasing
For column aliasing from table-name;

Syntax: Select col-name
from table-name newname;

Eg:- ① Select ename as name

from emp;
eg: ② Se

For multiple

Eg:- Select ename as name, sal as salary
from emp;

NOTE:- Alasing is the temporary change it

is used for. It cannot be reused. Table aliasing is used for achieving self join as well as co-related Query. Table aliasing is not visible in the output.

ADVANTAGES OF COLUMN ALASING:-

- It is used for better readability (or) understandable purpose
- It hides the behaviour of the column

* OPERATORS:- Operator is used to perform some tasks.

(a) ARITHMETIC OPERATORS:-

+	Addition
-	Subtraction
*	Multiplication
/	Division

- ① write a query to display name, salary and annual salary of the employees.

Ans:- Select ename as name, sal as salary, sal * 12 as annual_Salary
from emp;

- ② write a query to display the name and new salary of the employee which is thousand more than the actual salary.

Ans:- Select ename as name, sal as salary, sal + 1000 as new_salary
from emp;

- ③ write a query to display the name and date of joining of the employees along with the new date of joining which is one month after the actual joining

Ans:- Select ename, hire date as DOJ, hire date + 30

ENTITIES:- Entities are the real world object. Each entities are one table. Entities have various properties.

ATTRIBUTES:- Attributes are the properties of the entity. Each entity have 'n' no. of attributes. Attributes consists of similar type of data.

LITERALS IN SQL:-

In literals in SQL are of 3 types

- Numbers
- ~~charactes~~ character
- Date

(b) CONCATINATION OPERATOR:-

Concatination operator is used for combining a literal with another literal (or) ~~combi~~ ~~attribute~~ attribute to attribute (or) literal -> attribute

Select ename || ' is working as ' || Job
from emp;

eg:- Smith annual sal is 9600

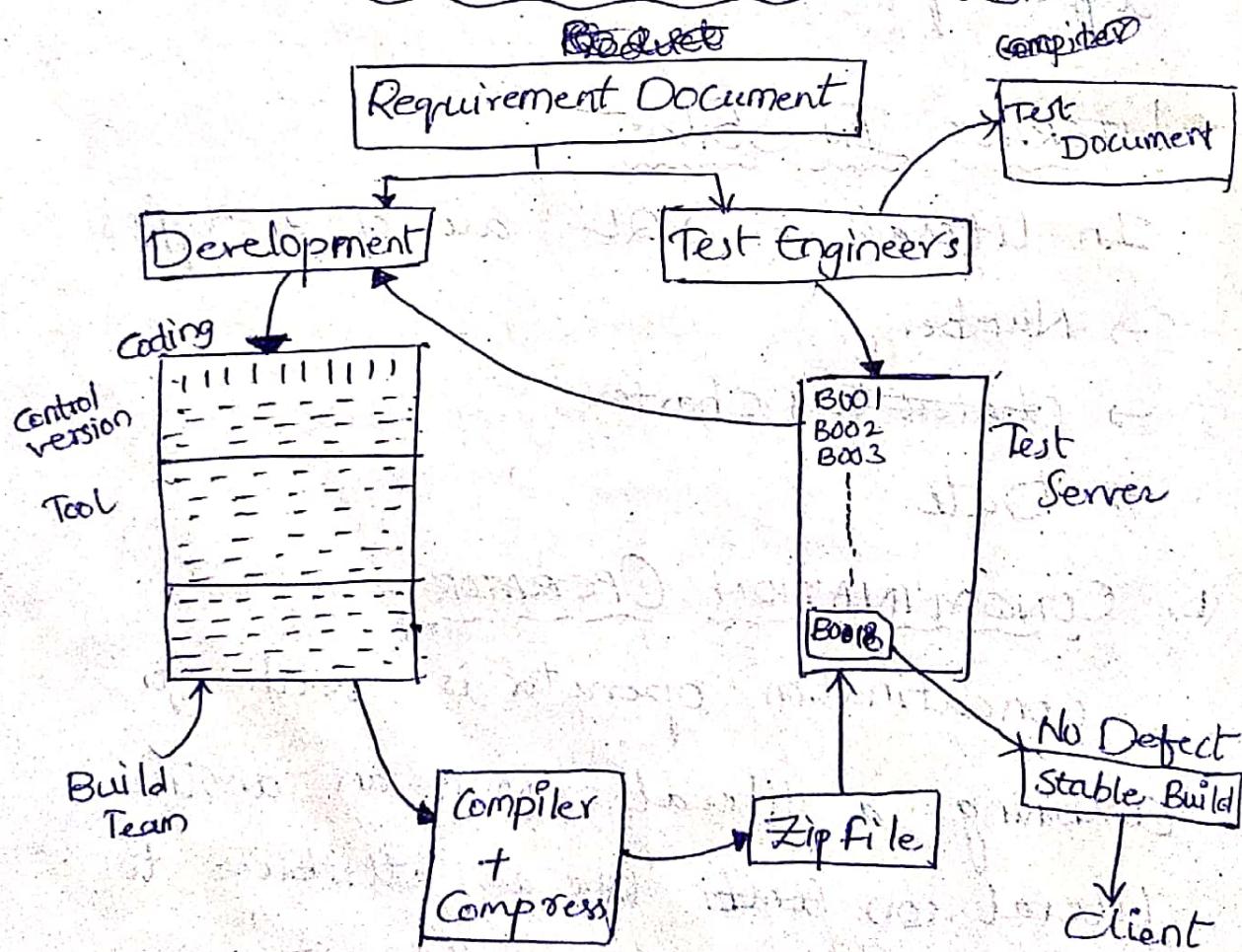
Ans:- Select ename || Annual salary is || Sal * 12
from emp;

1/10/21

BUILD:- It is a piece of code which is converted in the form of application. One release may have 'n' no. of builds. but each release is one build.

Each build is not a release

Software / Application / project / Build / Tool / Product



1/10/21 Layers Of The Application:-

TYPES OF APPLICATION:-

- Stand Alone Application
- Client Server Application
- Web Based Application

(a) STAND ALONE APPLICATION:-

Stand Alone Applications are those applications in which all the 3 layers that is UI, Business logic and data base are installed in the user machine.

In order to access such application the user need not ~~not~~ any internet connection.

Eg:- Calender, Calculator, MS office etc...

ADVANTAGES:-

- * Internet Connection is not required.
- * Execution is faster (Quick Response).
- * It is highly secure (Data hacking not possible).
- * Around the clock accessibility (any time, anywhere).
- * Easy to Access.

DISADVANTAGES:-

- * Installation is required.
- * End user machine memory is consumed.
- * No parallel usage.
- * Data Sharing is not possible.

- * Data Backup is not possible
- * ~~It's highly / secure~~

(b) CLIENT SERVER APPLICATION:-

In this type of Application the user interface layer is installed in the user machine whereas business logic and data base layer are installed in the Company machine.

In order to use this application user needs the internet connection.

eg:- whatsapp, facebook, gmail etc....

ADVANTAGES:-

- * Data Sharing is possible
- * Data Backup is possible
- * End users machine memory is consumed in less quantity.
- * Multiple users can access at a time

DISADVANTAGES:-

- * Installation is required
- * end user memory is using.
- * Execution depends on band width of the internet.
- * It is less secure as compared to Stand Alone application.

(c) WEB BASED APPLICATION:-

Web based Applications are those applications in which all the 3 layers i.e; UI, BL and DL are installed

in the companies machine and end user access it with the help of URL.

Eg: www.gmail.com, www.facebook.com, www.ola.com-----

ADVANTAGES:-

- * Installation is not required
- * end user machine memory is not consumed
- * Data Sharing is possible.
- * Data Backup is possible.
- * Multiple users can use at a time.

DISADVANTAGES:-

- * Execution depends on the band width of the internet.
- * Sometime we cannot access the Application if the Server is down.
- * Data is less secure as compared to Stand alone app.

13/10/2021

* SELECT:- Select is the statement which is used for mentioning the name of column(s) columns from which the data need to be retrieve.

FROM:- from is the keyword which is used for declaring the table name

WHERE:- Where is the clause which is used for defining the condition. Where clause will execute and retrieve the data if the condition satisfied.

Syntax:-
Statement → [Select Column name(s)]
Keyword → [from -table name]
clause → [Where Condition;]

Eg:- ③→ Select *

①→ from emp

②→ Where job = 'clerk'

① write a query to display the details of the employees whose salary is more than 1500?

Ans:- Select *
from emp
where Sal > 1500;

② write a query to details of the employee working for the department 30?

Ans:- Select *
from emp
where Deptno = 30;

③ write a query to display the details of the employees who join the Company before 1st of jan 82?

Ans:- Select *
from emp
where Hiredate < '01-Jan-82';

④ write a query to display name and annual salary of all managers

Ans:- Select ename, Sal * 12
from emp
where JOB = 'Manager';

→ Capital letters only should write

(b) RELATIONAL OPERATORS:-

=> equals

< > or ! = > not equals

> Greater than

< Less than

>= greater than equals

<= less than equals.

- ① write a query to display the details of the employees except managers.

Ans:- Select *
from emp
where job <> 'MANAGER';

- ② WAP TO ~~the display~~ the details of the employees whose salary is 3000.00 more

Ans:- Select *
from emp
where sal >= 3000;

NOTE:- SQL is not Case Sensitive. i.e; we can write the Query in upper case or lower case but the data should be as it is present in the table.

* Every SQL Query ends with Semi Column(;) .

* ORDER By:-

Order by is the keyword. It is used for retrieving the data either in Ascending order (or) Descending order

Syntax:-

- ② → Select Colname
- ① → from table-name
- ③ → Where Condition
- ④ → Order by Col-name;

Ex:- Select *
from tab
where job = 'MANAGER'
Order by Sal Desc;

① WAQTD the details of the Salesman in a Order they join the Company ?

Ans:-

```
Select *  
from emp  
Where JOB= 'SALESMAN'  
order by HIREDATE;
```

NOTE:-

(i) we can use Arthimetic operators with select statement

Relational operator → only with condition.

Arthimetic Operator



Select statement ~~only~~ and with condition,

(10/3) (c) CONCATINATION OPERATOR:-

Concatination operator is used for combining a literal with another literal or attribute.

eg-1: Select 'My name is' || ename
from emp;

eg-2: Select ename || 'is working as' || job
from emp;

① WAQTD the details of all CLERK in the emp table.

Ans:- Select *
from emp
where job = 'CLERK';

② WAQTD the name & and salary of all SALESMEN

Ans:- Select ename, Sal
from emp
where job = 'SALESMAN';

③ WAQTD the details of all employees whose salary is more than 2500.

Ans:- Select *
from emp

Where Sal > 2500;

* WAQTD The details of all employees whose salary is daily wage?

Ans:- Select ename || 'daily wage' || Sal * 12 / 365

④ WAQTD the details of the employees who joined the company after 3rd Jan 85.

Ans:- Select *
from emp
where HIREDATE > '03-JAN-85';

⑤ WAQTD the details of the employees whose annual salary is more than 10000?

Ans:- Select *
from emp
where sal * 12 > 10000;

⑥ WAQTD the details of the employees in increasing order of their Salary?

Ans:- Select *
from emp
order by sal;

⑦ WAQTD the details of the employees in the order they joined the Company?

Ans:- Select *
from emp
order by HIREDATE;

⑧ WAQTD the details of all department 20 employees?

Ans:- Select *
from emp
where deptno = 20;

(c) WAQTD the details of all clerks from department 20.

Ans:- Select *
from emp
where deptno=20 and job='CLERK';

(d) LOGICAL OPERATOR:— Logical operators are used for executing multiple conditions. various types of logical operators are,

→ AND

→ OR

→ NOT

→ AND:— "AND" operator will execute and retrieve the data if both the condition satisfy.

→ OR:— "OR" operator will execute and retrieve the data if either of the condition satisfy.

→ NOT:— "NOT" operator will execute and retrieve the data if the condition does not satisfy.

(e) WAQTD the details of all employees from department 10 having salary more than 1000?

Ans:- Select *
from emp
where deptno=10 and Sal > 1000;

(21) WAPQTD the details of all managers from department 30 joined before 22-05-86.

Ans:- Select *

from emp
where JOB = 'MANAGER' and DEPTNO = 30 and HIREDATE < '22-MAY-86';

(22) WAPQTD the details of the employees working as SALESMAN OR CLERK?

Ans:- Select *

from emp
where job = 'SALESMAN' OR job = 'CLERK';

(23) WAPQTD the details of the employees working as SALESMAN, CLERK OR ANALYST from department 20?

Ans:- Select *

from emp

where (job = 'SALESMAN' OR job = 'CLERK' OR
job = 'ANALYST') and deptno = 20;

(24) WAPQTD the details of employees from department 10 and 30 having salary more than 1200?

Ans:- (Select *)

from emp

where (DEPTNO = 10 OR DEPTNO = 30) and

SAL > 1200;

(25) WAQTD the date of joining of all clerks and managers from department 10 and 30?

Ans:- Select *
from emp

where (JOB = "CLERK" OR JOB = "MANAGER") AND
(DEPTNO = 10 OR DEPTNO = 30);

(26) WAQTD the details of the employees who are not managers?

Ans:- Select *
from emp
where not job = "MANAGER";

(27) WAQTD the details of employees who are neither ~~SALESMAN~~ nor CLERK

Ans:- Select *
from emp
where not (JOB = "SALESMAN" OR ~~JOB~~ = "CLERK");

(28) WAQTD the details of employees who are neither Salesman.

(28) WAQTD name of all clerks and managers except from dept 20.

Ans:- Select *
from emp
where (JOB = "CLERK" OR JOB = "MANAGER") AND NOT
(DEPTNO = 20);

(29) WAQTD the details of all clerks OR Salesman who joined the company after 18-01-85.

Ans:- Select *
from emp
where (JOB = "CLERK" OR JOB = "SALESMAN") AND
CHIREDATE > "18-JAN-85";

(30) WAPTD the name and deptno of employees from department 10, 20, 30 and 40.

Ans:- Select ename, deptno
from emp
where Deptno = 10 OR DEPTNO = 20 OR DEPTNO = 30
OR DEPTNO = 40;

(31) WAPTD the names of all clerks from deptno 10, 20, 30 and 50.

Ans:- Select ename
from emp
where JOB = 'CLERK' AND (DEPTNO = 10 OR DEPTNO = 20 OR
DEPTNO = 30 OR DEPTNO = 50);

15/10/21

(32) write a query to display the name, annual salary and deptno of the employees working as SALESMAN, CLERK and ANALYST from dept no 20 (or) 30.

Ans:- ~~Select ename as name, sal * 12 as annual salary~~
Select ename, Sal * 12, deptno
from emp
where (job = 'SALESMAN' OR JOB = 'CLERK' OR
job = 'ANALYST') AND (DEPTNO = 20 OR
DEPTNO = 30);

(e) Special Operators:-

IN' OPERATOR:- IN operator is used for retrieving the data from a set of values. It is also known as set operator.

SQ: Select ename

from emp

where DEPTNO IN (10, 20, 30, 40);

① WAPTD the details of the employees working as SALESMAN OR CLERK OR ANALYST from DEPTNO - 10 (or) 20 (or) 30.

Ans:-

Select *

from emp

Where JOB IN ('SALESMAN', 'CLERK', 'ANALYST')

AND DEPTNO IN (10, 20, 30);

② WAPTD the details of the name and Annual Salary of the clerks from deptno 10, 20, 30 and 50?

Ans:- Select ename, Sal*12

from emp

Where JOB IN ('CLERK') AND DEPTNO IN (10, 20, 30, 50);

③ WAPQTD the details of the employees whose salary is more than 1500 and less than 3500?

Ans:- Select *
from emp

where SAL BETWEEN 1501 AND 3499;

(ii) BETWEEN OPERATOR:-

Between operator is used for retrieving the data from a range of value.

NOTE:- Between operator will take the lower limit as well as upper limit

① WAPQTD the details of the Salesman, CLERK and MANAGER who join the Company after '03-JAN-81' and '05-NOV-82'.

Ans:- Select *
from emp
where job IN ('SALESMAN', 'CLERK', 'MANAGER') AND
(HIREDATE BETWEEN '03-01-81' and
'04-11-82');

② WAPQTD the details of the employees who join the Company after 11-01-82 and before 11-02-89?

Ans:- Select *
from emp
where HIREDATE BETWEEN '12-JAN-82' AND
'11-FEB-89';

(iii) LIKE OPERATOR:-

Like operator is used for retrieving a particular pattern data.

e.g:-

- ③ WAPQTD the details of the employees whose name starts with 'S'.

Ans:- Select *
from emp
where ENAME LIKE 'S%';

- ② WAPQTD the details of the employees whose name ends with 'S'?

Ans:- Select *
from emp
where ENAME LIKE '%.S';

- ③ WAPQTD the details of the employees whose name has atleast one 'A'?

Ans:- Select *
from emp
where ENAME LIKE '%.A %';

- ④ WAPQTD the details of all CLERK, SALESMAN and MANAGERS from whose name ends with 'S'.

Ans:- Select *
from emp
where JOB IN ('SALESMAN', 'CLERK', 'MANAGER')
AND ENAME LIKE '%.S';

(31) WAQTD the details of all employees whose name has 5 character.

Ans:- Select *
from emp
where ENAME LIKE '_____';

(32) WAQTD the details of all managers and salesman whose annual salary is b/w 10000 and 20000?

Ans:- Select *
from emp
where JOB IN ('SALESMAN', 'MANAGER') AND
(SAL*12 BETWEEN 10001 AND 19999);

* WAQTD the details of the employees who joined in the company in the month of December?

Ans:- Select *
from emp
where HIREDATE LIKE '%DEC%';

* WAQTD the name, salary and annual salary of the employees whose name 2nd last character is 'e'

Ans:- Select ename, sal, Sal*12 as annual_salary
from emp
where ename LIKE '%E-%';

* WAQTD the name, salary, annual salary of the employees whose name

Ans: Select ename as name, sal as salary, sal*12 as Annual Salary
from emp
where ename like '%.E-%';

(36) WAQTD the details of the employees whose name second last character is 'T'?

Ans: Select *
from emp
where ENAME LIKE '%.T-%';

(37) WAQTD the details of employees who joined in the company in the year 82.

Ans: Select *
from emp
where HIREDATE LIKE '%82%';

(38) WAQTD the details of the employees who dont have comm.

Ans: Select *
from emp
where comm NOT LIKE 'NULL';
(Or)

Select *
from emp
where comm IS NULL;

Q) WAPTD the details of the employee who joined in the month 'DEC'.

Ans:- Select *
from emp
where HIREDATE LIKE '%DEC%';

* WAPTD the details of the employee who have a commission.

A) Select *
from emp 01
where COMM is not NULL;

(in Is OPERATOR:-

"Is" is a special operator which helps us to get the ~~receiving~~ ~~value~~ ~~data~~.

to retrieve the data from null value.

18/10/21

* FUNCTIONS:-

Functions in SQL is used for performing a specific tasks. Functions are of two types. (a) Single Row function.
(b) Multi Row function

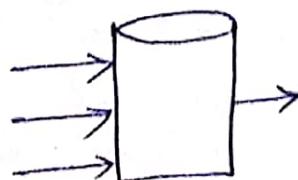
(a) SINGLE Row FUNCTION:-



In single row function we give multiple inputs and we get corresponding outputs. It is a part of Data Query Language

(b) MULTI Row FUNCTION:-

In multi row function we give multiple inputs and we get single output. It is a part of Data Query language



DUAL:- Dual is the dummy table present in the database. with the help of dual we can perform the "Individual tasks".

e.g. Select *
desc dual;
from dual;

with the help of dual table we can perform individual tasks.

(a) SINGLE ROW FUNCTION:

→ Case Manipulation function

- * Upper (Argument 1) e.g:- select upper('tarun')
from dual;
- * Lower (Argument 1) e.g:- select lower("TARUN")
from dual;
- * Initcap(Argument 1) e.g:- select INITCAP('tarun').
from dual;

* write a Query to display the employees name in lower case

Ans:- Select LOWER (ENAME) AS ~~NAME~~ NAME
from emp;

* write A Query to display the employees name along with their job in lower case.

Ans:- Select LOWER(ENAME) AS NAME, LOWER(JOB) AS JOB
from emp;

Note:- (i) Case manipulation function accepts only one argument!

(ii) Case manipulation function is applicable for character type of literal.

* WAQTD the name of employees in lower case, upper case and first letter Capital?

Ans:- Select Lower (ENAME) AS NAME, UPPER (ENAME) AS NAME, INITCAP (ENAME) AS NAME
from emp;

~~Notes on Concatenation~~

Concat Function:- Concat function is used for combining one attribute to other attribute or one attribute to other literal.

Syntax:- Concat (arg1, arg2)
from emp;

* WAQTD the

Ans:- Select CONCAT ('MY NAME IS', LOWER(ENAME))
from emp;

* WAQTD ename is a clerk
3 argument.

Ans:- Select Concat (Concat (ENAME, 'IS A'), JOB)
from emp;

* Ename is working as JOB since HIREDATE?
 | | | |
 1 2 3 4
 5 Arguments.

Ans:- Select Concat(Concat(Concat(Concat(ENAME, "IS
WORKING AS"), JOB), "SINCE"), HIREDATE)
from emp;

NESTED FUNCTION:- Function inside another
function is known as nested function.

→ Select Concat(Concat(Concat(Concat(lower(ENAME), "IS
working as"), JOB), "SINCE"), HIREDATE)
from emp;

CHARACTER MANUPULATION FUNCTION:-

i) INSTRING:

Instring is used for retrieving the
position of the character from a given
String.

Syntax: Instr(Arg1, Arg2, Arg3, Arg4)

Arg1 → literal / attribute

Arg2 → character

Arg3 → Start position

Arg4 → Occurrence

Eg-1 - Select INSTR('TARUNKUMARMANDAL', 'A', 1, 1)
from dual;

Output:- — 2

Eg-2 - Select INSTR('TARUNKUMARMANDAL', 'A', 1, 4)
from dual;
output :- — 15

Eg-3 - Maharaja

Select INSTR('MAHARAJA', 'A', 1, 3)
from dual;

output :- — 6

*WAPTD position of 'A' from the name of the
employees.

Ans:- Select INSTR(ENAME, 'A', 1, 1), ENAME
from emp;

Output:- INSTR(ENAME, 'A', 1, 1) ENAME

0 SMITH

1 ALLEN

2 WARD

0 JONES

2 MARTIN

3 BLAKE

0 CLARK

0 SCOTT

0 KING

0 TURNER

1 ADAMS

2 JAMES

* WAP TO the details of the employees whose name does not have 'A'?

Ans:- Select *

from emp

where INSTR(ENAME, 'A', 1, 1) = 0;

* WAP TO the details of the employees whose name starts with 'S'?

Ans:- Select *

from emp

where INSTR(ENAME, 'S', 1, 1) = 1;

* WAP TO the details of the employees whose name has atleast one "I"?

Ans:- Select *

from emp

where INSTR(ENAME, 'I', 1, 1) <> 0;

SUB-STRING :-

Sub string is used for retrieving a part of the string. It accepts 3 arguments.

Syntax:- Substr (arg1, arg2, arg3)

arg1 → Literal / Attribute

Arg2 → Start position

Arg3 → Length

eg:- rose merry

Select Substr('rosemary', 1, 4)
from dual;

Output:- 'ROSE'

eg-2:- rose merry

Select Substr('rosemary', -5)
from dual;

* WAQTD the details of the employees whose JOB Starts with MAN?

Ans:- Select *
from emp
where SUBSTR(JOB, 1, 3) = 'MAN';

* WAQTD the details of the employees whose JOB ends with MAN?

Ans:- Select *
from emp
where SUBSTR(JOB, -3, 3) = 'MAN';

* WAQTD the details of the employees whose JOB either starts with MAN (or) ENDS with MAN?

Ans:- Select *
from emp
where SUBSTR(JOB, 1, 3) = 'MAN' OR
SUBSTR(JOB, -3, 3) = 'MAN';

* WANTED the details of the employees whose name has atleast one vowel?

Ans:- Select *
from emp
where INSTR(ENAME, 'A', 1, 1) <> 0 OR INSTR
(ENAME, 'E', 1, 1) <> 0 OR INSTR(ENAME, 'I', 1, 1) <> 0
OR INSTR(ENAME, 'O', 1, 1) <> 0 OR INSTR(ENAME, 'U', 1, 1) <> 0

* WANTED the details of the employees whose name has double "A" or double "L" or double "L"?

Ans:- Select *
from emp
where INSTR(ENAME, 'LL', 1, 1) <> 0;

* WANTED the details of the employees whose name has atleast 2L's?

Ans:- Select *
from emp
where INSTR(ENAME, 'L', 1, 2) <> 0;

LENGTH FUNCTION:- Length function is used for retrieving the length of the characters (or count of the characters present in the string)

Syntax:- LENGTH (arg1)
from emp;

~~arg1~~ arg1 → literal / attribute

Eg-i: Select length ('TARUN')
from Dual;

Eg-II: Select ~~ename~~ length (ename), ename
from emp;

REPLACE FUNCTION:-

Replace is used for Replacing the character(s) characters from a given string with a new character.

Syntax:- Select replace (arg1, arg2, arg3)

from table name;

Arg1 → literal / Attribute

Arg2 → character(s) characters which need to be replaced

Arg3 → new character(s) new characters

Eg-i: Select replace ('JAVA', 'J', 'M')

from Dual;

Eg-ii: Select replace (JOB, 'MAN', 'WOMAN')

from emp;

PAD: Pad is also a character manipulation function which is used for uniforming the size of the string. It is of two types

- (i) LPAD
- (ii) RPAD

Syntax: Select LPAD (arg1, arg2, arg3) (or LPAD(arg1, arg2, arg3)
from table name;

arg1 → Literal / Attribute

arg2 → Size (or) Length

arg3 → replacing character

eg-1 Select LPAD (ENAME, 10, '-')
from emp;

Output:- Smith

Allen

eg-2 Select RPAD (ENAME, 10, '_')
from emp

Output:- Smith -----
Allen -----

Trim: Trim is used to remove the character from left hand side (or) right hand side.

eg-1 → Select TRIM (TRAILING 'A' FROM 'AJJA')
from Dual;

Output:- TRI
AJJ

Select TRIM (LEADING 'A' FROM 'AAAAAJJA')
from Dual;

Output TRI

JJA

Select TRIM (TRAILING 'A' from 'AJJAHJH')
from Dual;

Output - TRIM (TR
--- ---
AJJAHJH

Select TRIM (LEADING 'A' FROM 'AAAAAJJA')
from Dual;

Output - TRI

JJA

Select TRIM (LEADING 'A' FROM 'AAAAAJJA')
from dual

Output → JJA

NUMBER MANIPULATION FUNCTION:-

(1) Mod Function:- Mod function is used for retrieving the modulus value (or) remainder value.

Syntax:- Select MOD(arg1, arg2) arg-1 → Literal/Attrib
 from table-name; arg-2 → Number

e.g:- Select MOD (29, 3)
 from Dual;

Output:- 2

eg-2:- Select MOD(Sal, 3)
from emp;

(b) TRUNC:- Trunc is used for removing the Decimal value.

Syntax:- Select Trunc(arg1, arg2)
from tablename;

eg-1:- Select Trunc(234.76567)
from Dual;

output:- (234) ~~.76567, 2~~ ②

eg-2:- Select Trunc(234.76567, 3)
from Dual;

output:- 234.765

(c) ROUND:- Round is used for round offing the value

Syntax:- Select round(arg1, arg2)
from tablename;

eg-1:- Select round(234.76567)
from dual;

output:- 235

eg-2:- Select round(234.76567, 3)
from dual;

output:- 234.766

MULTI Row Functions

Max ()

Min ()

Avg ()

Count ()

Sum ()

Function	Number	Character	Date
Max()	✓	✓	✓
Min ()	✓	✓	✓
Avg ()	✓	✗	✗
Count ()	✓	✓	✓
Sum ()	✓	✗	✗

(i) WANTED - the details of the maximum Salary

Ans:- Select Max(Sal)
from emp;

(ii) WANTED The Average Salary?

Ans:- Select Avg(Sal)
from emp;

⑪ WAPTD the max salary for each Department?

Ans Select Max (Sal), DEPTNO
from emp

Group by Dept No;

GROUP-BY:- Group by in SQL is used for dividing the single table into multiple Groups. Using Groupby we can retrieve multiple output from a single table with multirow function.

Syntax:-

Select → ③
 from → ①
 Group by → ②

(4B) WAQTD the details of the lowest - Paid employee from each Department?

-Ans:-

~~Select sum(SAL), JOB
from emp
Group by JOB;~~

Select min(SAL), Deptno
from emp
Group by Deptno;

(19) WAP/TD the Count of the employees present in each JOB Group?

Ans:- Select sum(SAL),JOB
from emp
Group by JOB;

Select count(ename), job
from emp
group by job;

(45) WAP/TD the dept which has more than 3 employees.

Ans:-

Select count(ename), dept no
from emp

group by dept no

~~where~~ having count(ename) > 3;

NOTE :- we cannot perform multirow function with 'where' clause.

HAVING:- Having is a clause with the help of having clause we can perform multi row function.

Syntax:- Having → clause.

Execution order:-

(1) → Select

(2) → from

(3) → Group by

(4) → having

(5) → Order by

- * WHERE CLAUSE:- Where clause will go to each row and check for the condition.
If the condition satisfies
 - Where clause can be used without groupby
 - using where clause we cannot perform multi row function

HAVING CLAUSE:- Having clause will go to each and every group and check for the condition.

- Having clause cannot be executed (or) used without groupby.
- using where clause we can perform with multi row function

- * WAP TO find the average salary for each designation and display if the average salary of that designation is more than the average salary of all employees?

Ans:- Select job as Designation, avg (sal)
from emp
group by job
having avg (sal) > (select avg (sal) from emp);

(46) WAQTD the total salary of each job and select only if the total salary is less than 7000?

Ans:- Select Sum(Sal), job
from emp
Group by job
Having Sum(Sal) < '7000';

(47) WAQTD the total salary of all salesman and clerks?

Ans:- Select Sum(Sal), job
from emp
Group by job
Having job in ('SALESMAN', 'CLERK');

(48) WAQTD the average salary of each job?

Ans:- Select Avg(Sal), job
from emp
Group by job;

⑭ WAP/TD The name and job of highest paid clerk?

Ans:- Select ename as name, job
from emp

Where sal = (Select max(sal) from emp group
by job having job = 'CLERK');

⑮ WAP/TD the details of first clerk joined the Company?

Ans:- Select *

from emp

Where HIREDATE = (Select MIN(HIREDATE) from emp
group by ^{JOB} having job = 'CLERK');

63)
Select *
from emp

where HIREDATE = (Select min(HIREDATE) from emp
where job = 'CLERK');

Sub-Query:-

Whenever the Query is indirect query we take help of Sub Query. The Subquery consists of two parts, they are inner query and Outer query. The output of the inner query is the input for the outer query.

NOTE:-

The attribute name present in the select statement of inner query and where clause of outer query should have same data-type (as) literal.

- (52) WAQTD the details of the employees whose salary is more than Smith?

Ans:-

Select *

from emp

Where Sal > (Select sal from emp where ename = 'SMITH')

- * WAQTD the details of the employees who join the company before Martin as salesman

Ans:-

Select *

from emp

Where HIREDATE < (Select HIREDATE
from emp where ename = 'MARTIN') AND
JOB = 'SALESMAN'

- (53) WAQTD -the details of the employees who joined the company after Ward?

Ans:-

Select *

from emp

Where HIREDATE > (Select hiredate from
emp where ename = 'WARD');

(4) WAQTD the details of the clerks who joined in the same department as Turner?

Ans:- Select ~~ename~~ &
from emp

where deptno = (Select deptno from emp where
ename = 'TURNER') and
job = 'CLERK');

(5) WAQTD the name and Salary of the employees who joined the Company before Smith as Salesman?

Ans:- Select ename as name, sal as salary
where from emp

where hiredate < (Select hiredate from emp
where ename = 'SMITH') and job = 'SALESMAN');

(6) WAQTD the location of the employees who joined the Company in same department of Smith?

Ans:- Select loc as location
from Dept

where Deptno = (Select Deptno from emp
where ename = 'SMITH');

Note:- Sometime the data need to be retrieved from one table by taking reference of other table.

In order to achieve it atleast one Common attribute is compulsory between the two tables.

(57) W.A.Q.T.D the employees details whose designation is similar to that of employee number 7902?

Ans:- Select job as Designation

from emp

where JOB = (Select job from emp where
empno = "7902");

(58) W.A.Q.T.D the details of the department details of the employees whose department is similar to that of Turner?

Ans:-

Select *

from dept

where deptno = (Select deptno from emp
where ename = "TURNER");

(59) W.A.Q.T.D the details of the employees working in the same department of MILLER and job is similar as Scott?

Ans:- Select *

from emp

where deptno = (Select deptno from emp where
ename = "MILLER") AND JOB = (Select
job from emp where ename = "SCOTT")

* WAQTD the details of the employees working for newyork location?

Ans:-

~~Select loc as location
from emp
where deptno = (select deptno from dept where loc = 'NEWYORK')~~

from emp
where deptno = (select deptno from dept where loc = 'NEWYORK')
o/p of inner query is more than 1
but output of outer query is 1

Select loc as location

from emp

where deptno = (select deptno from dept where loc = 'NEWYORK');

(60)

WAQTD the details of the employees who joined the company after Smith and name start with 'S'?

Ans:- Select *

from emp

where HIREDATE > (select hiredate from emp

where ename = 'SMITH') AND ENAME

LIKE 'S %';

(61)

WAQTD the name of the employees who neither work in the department of Smith or Miller?

Ans:-

Select * ~~from emp~~

from emp

~~where deptno not in (select deptno from emp where ename = 'SMITH' or ename = 'MILLER')~~

where not deptno = (select deptno from emp

where ename = 'SMITH' and ename = 'MILLER');

Types of SubQuery

- (A) Single row Sub Query:- If the output of the inner query is one (or single), It is known as Single row Sub Query.
- (B) multi row Sub Query:- If the output of the inner query is more than one it is known as multi row Sub Query.

Eg:- Select *

→ from emp

→ where Deptno in (Select Deptno from dept
where loc in ('NEWYORK', 'DALLAS'));

* WAP TO find the details of all clerks and analyst working in DALLAS and NEWYORK?

Ans:- Select *

from emp

where Deptno in (Select deptno from dept Where

* JOINS:- Joins in SQL is used for retrieving the data from more than one table.

→ It is a part of Data Query Language.

TYPES OF JOIN:-

① Cross / Cartesian Join

② Inner Join

→ Equi Join

→ Non equi Join

→ Self join

③ Outer Join

→ left outer

→ Right outer

→ full outer

① CROSS / CARTESIAN JOIN:- In this type of join each row of one table is getting combined with each row of another table.

Syntax:- Select column_name(s)

from table 1, table 2;

Eg:- Select *

from Dept, EMP;

* WAP TO retrieve the details of the employee along with the department details for employees working for newyork location?

Ans:- Select *

from emp, dept

where ~~emp.deptno = dept.deptno~~ and loc = 'NEW YORK';

Note - In real time Industry we do not go for Cross / Cartesian Join. Since in Cross join it retrieve the match record as well as unmatch record.

② INNER JOIN :- It is used for retrieving the match record from the existing tables.

These are of 3 types

- (a) Equi join
- (b) Non-equi join
- (c) Self join

(a) EQUI JOIN :- Whenever there is the common attribute between the two tables we go for equi join.

* WAP TO retrieve the details of the employees along with their department details. ?

Ans:- Select *

from emp, dept

where emp.deptno = dept.deptno;

② WAPTD the employees who belong to the research department?

Ans:- Select *

from emp, dept

where (emp.deptno = dept.deptno) and

(Dname = 'RESEARCH');

③ WAPTD the name ~~of the~~ and the department name of the employees.

Ans:- Select ename as name, Dname as dept-name

from emp, dept;

where emp.deptno = dept.deptno;

④ WAPTD the name and the department details of the employee?

Ans:- Select ename as name, dept.*

from emp, dept

where emp.deptno = dept.deptno;

⑤ WAPTD the employees details with their loc?

Ans:- Select loc as location, emp.*

from emp, dept

where emp.deptno = dept.deptno;

⑥ WAPTD the details of employees with department details?

Ans:- Select emp.* , Dept.*
from emp, Dept

Where emp.deptno = dept.deptno;

⑦ WAPTD the name, location, hiredate and salary of all clerks and managers

Ans:- Select ename as name, loc as location, sal as salary, hiredate
from emp, dept

Where (emp.deptno=dept.deptno) and (JOB in ('CLERK',
'MANAGER'));

27/10/21

⑥ NON-EQUI JOIN:- It is a type of inner join.

→ In this type of join we can retrieve the data from Two (or) more table without having a common column.

→ Generally we use operators like "IN", "BETWEEN", "LIKE", etc.... (relational operators except "=").

→ for example b/w emp table and Salgrade table there is no ~~common~~ common column.

→ Hence we go for non-equi join

* WAQTD The name of ~~the~~ employees along with their salary and salary grade?

Ans: Select ename as name, sal as salary, Grade from emp, salgrade
Where sal between losal and hisal;

* WAQTD the name of the employees along with their grade information?

Ans: Select ename as name, ~~grade~~ Salgrade.*
from emp, salgrade
Where sal between losal and hisal;

* WAQTD the name of the employees along with their job, salary and grade for department no 10, 20 with a salary in grade 1, 2 or 3.

Ans: Select ename as name, sal as salary, job, grade
from emp, salgrade
Where sal between losal and hisal and deptno
in (10, 20) and grade in (1, 2, 3);

(Q2) WAQTD the employees details along with their department details who is earning more than SMITH?

Ans: Select *
from emp, dept
where (emp.deptno = dept.deptno) and & Sal > (select
sal from emp where ename = 'SMITH');

⑥ WAPTD the employees details of the employees from the research and accounting department?

Ans:- Select *

from emp,dept

where Dname in ('Research', 'ACCOUNTING') AND
emp.deptno = dept.deptno;

TING)

AND

⑦ WAPTD the employees who are not working in Dallas?

Ans:- Select *

from emp,dept

where not loc in ('DALLAS') and ~~emp.deptno~~

emp.deptno = dept.deptno

⑧ WAPTD the details of the employees with location ~~or~~ who don't have commission.

Ans:- Select *

from emp,dept

where comm is Null and emp.deptno = dept.deptno

⑨ WAPTD the details of the employees from Grade 1 and grade 2?

Ans:- Select emp.* , ~~sal~~ grade

from emp, Salgrade

where grade in (1,2) and sal between losal and

~~highsal~~ ;

(3) WAPTD the details of all salesmen and managers from grade 1 and grade 2?

Ans:- Select job, grade
from emp, salgrade

where job in ('SALESMAN', 'MANAGER') AND Grade
in (1, 2) and Sal between losal and hisal;

(4) WAPTD the details of the employees from dallas and delhi who are not salesman (or) clerk. using
efjoin

Ans:- Select *

from emp, dept

where loc in ('DALLAS', 'DELHI') and not job
in ('SALESMAN', 'CLERK') AND
emp.deptno = dept.deptno;

* WAPTD name of the employees along with their department details and grade of the salary?

Ans:- Select ename as name, dept.* , grade
from emp, dept, sal grade
where emp.deptno = dept.deptno and Sal between
losal and hisal;

(C) Self JOIN:- Whenever we want to
combine the data from table itself we
go for self join.

→ In order to achieve self join we take
the help of table aliasing.

* WAQTD the name of the employees along with their managers' name?

Ans:- Select e.ename as name, m.ename as manager_name
from emp e, emp m
where e.mgr = m.empno;

(75) WAQTD the details of the employees of the employees along with their managers name?

Ans:- Select e.* , m.ename as manager_name
from emp e, emp m
where e.mgr = m.empno;

(76) WAQTD the name of the employees and department ^{no} name along with their managers name and department ~~name~~ deptno.

Ans:- Select e.name as

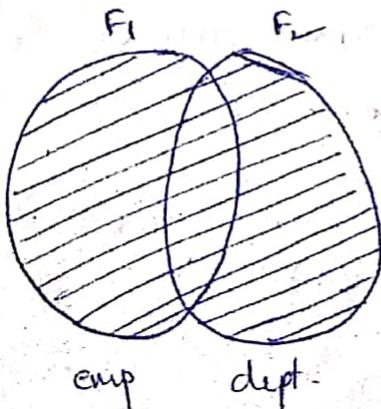
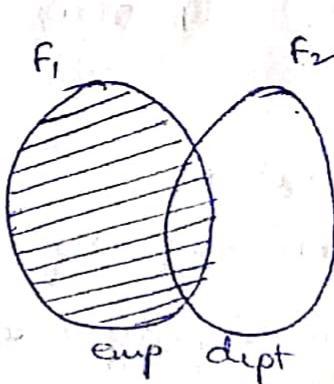
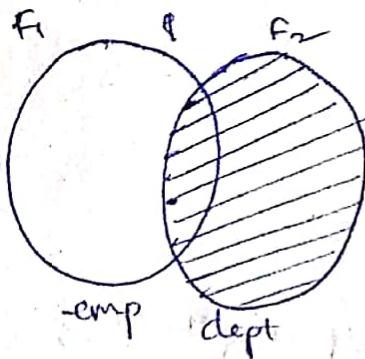
Q1/2
* WAP/TD the details name of the employees earning the same salary?

Ans:- Select e.ename as name1, e.Sal as salary, m.ename as name2
from emp e, emp m
Where e.Sal = m.Sal and e.ename <> m.ename;

* WAP/TD the name of the employee along with their department no and manager name along with their department no?

Ans:- Select e.ename as employee, m.ename as manager,
e.deptno, m.deptno
from emp e, emp m
Where e.mgr=m.empno;

Outer JOIN:-



Outer join is used to get the complete match data from one table and unmatched data from another table.

Outer join is used for match record from one table as well as unmatched record from another table.

eg:- ① - left outer join example

Select *

from emp, dept

where emp.deptno(+) = dept.deptno;

Output:- 15 rows

eg:- ② - right outer join example

Select *

from emp, dept

where emp.deptno = dept.deptno(+);

Output:- 14 rows

eg:- ③

Select *

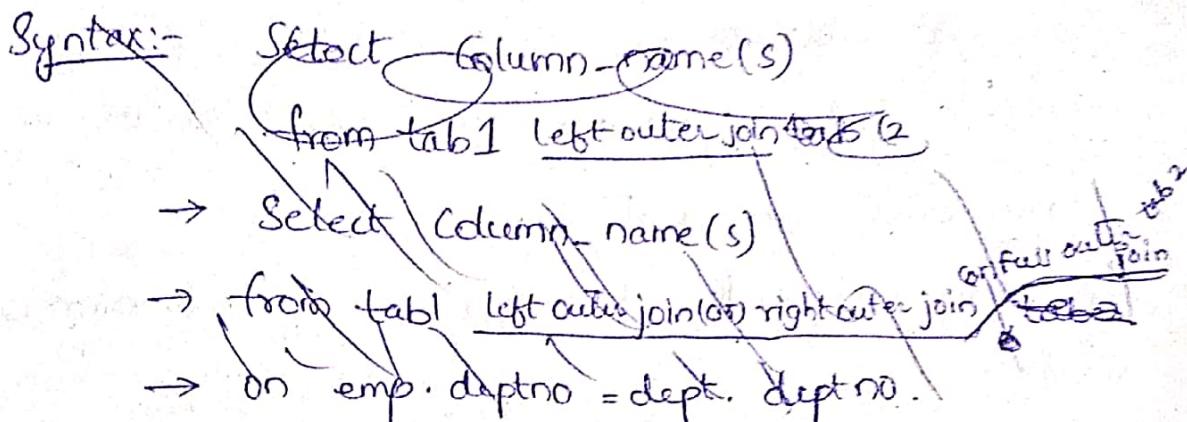
from emp, dept

where emp.deptno ~~=~~(+) = dept.deptno(+);

Output:- predicate may reference only one outer join table.

NOTE:- we cannot use more than one predicate(+)

* A.N.S.I:- American National Standard Institute.



e.g:- Select *

```
from emp left outer join dept
on emp.deptno = dept.deptno;
```

Syntax:-

```
→ Select Column_name(s)
→ from tab1 left outer join on right outer join or full outer join tab2
→ on emp.deptno = dept.deptno
```

* FULL OUTER JOIN:- It will take match record from both the tables and take left over data from Both tables.

* LEFT OUTER JOIN:- It will take match record from both the tables and take right side remaining data from right side table.

* RIGHT OUTER JOIN:- It will take match record from both the tables and take left side remaining data from left side table.

* Rownum (Row Number) :-

- (i) It is the dummy column present with all the tables.
- (ii) The data present in this Column is numeric in nature.
- (iii) By default the value of rownum is 1.

* WAP TO the first row of data from emp table?

Ans:-
Select *
from emp
where rownum=1;

* WAP TO the first five rows of data from emp table?

Ans:-
Select *
from emp
where rownum < 6;

* WAP TO the first ten rows of data from emp table?

Ans:-
Select *
from emp
where rownum < 11;

Co-related Sub Query:-
* WAP TO the third row of data from emp.?

Ans:- Select *

from (Select rownum as rn, emp.* from emp)
Where rn=3;

* WAP TO the Second, fourth, Sixth ^{row of} data from emp?

Ans:- Select *

from (Select rownum as rn, emp.* from emp)
Where rn in (2, 4, 6);

* WAP TO the last row of data from emp table?

Ans:- Select *

from (Select * from emp order by rownum desc)
Where rownum=1;

10/23

* WAP TO the last three rows of data from emp table?

Ans:- Select *

(execute only data) from (Select * from emp Order by rownum Desc)
where rownum in (1, 2, 3);

(Q3)

Select *

(execute only data) from (Select * from emp order by rownum Desc)
Where rownum <= 3;

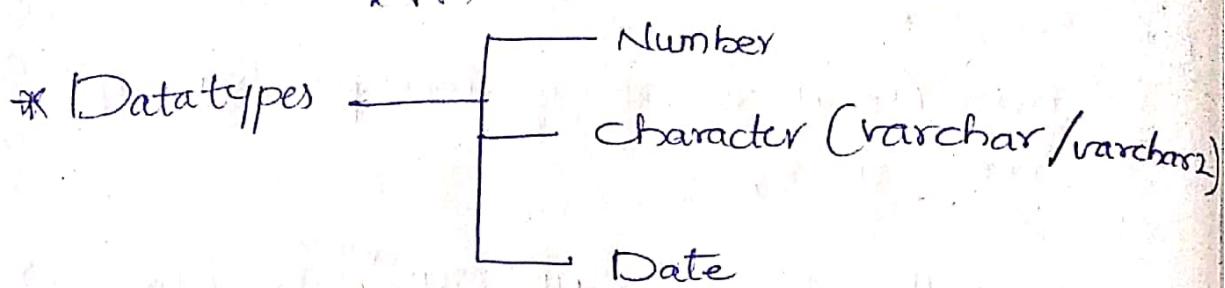
(Q4)

Select *

(execute with row number) from (Select rownum as rn, emp.* from emp)
Where rn > (Select count (ename) from emp)-3;

DDL

Data Definition Language



The difference between varchar and varchar2 is ~~is~~

- varchar accepts only 2000 characters.
- varchar2 accepts only 4000 characters.

NOTE:-

From SQL 10g onwards it by default takes "varchar2".

Example:-

(entity) \rightarrow Students

ID	Name	DOJ	Stream

Data type/
Size/
Constraint.

* CONSTRAINT: - i) Constraint in SQL is used for
Setting rules and restrictions for the columns
(or) table.

ii) With the help of constraint we can avoid
incorrect data insertion.

iii) Various types of constraint are

- (a) Unique Constraint.
- (b) Not Null Constraint
- (c) Primary Key Constraint
- (d) Foreign Key
- (e) Candidate Key
- (f) Alternate Key
- (g) Composite Key
- (h) Check
- (i) Default

(a) UNIQUE CONSTRAINT: - Unique constraint restrict
the duplicate data insertion, but it allows
the null values.

CONSTRAINT:-

b) NOT NULL CONSTRAINT: - Not null constraint
avoids (or) restrict the null values but it
allows the duplicate values.

* Difference between Unique and not null constraint?

UNIQUE CONSTRAINT

- i) It restricts duplicate Data
- ii) It allows the Null values

NOT NULL CONSTRAINT

- i) It allows Duplicate Data
- ii) It restricts the Null values.

(c) Primary Key Constraint:- primary key constraint does not allow the null value as well as duplicate value.

(ii) It is also known as unique identifier Attribute.

(iii) In a table we can have only one column as primary key constraint.

(iv) Generally the data present in the primary key column is always numeric in nature.

(d) FOREIGN KEY CONSTRAINT:-

(i) Foreign key constraint is also known as referential integrity constraint.

(ii) The attribute present in the one table is referred to the attribute present in the another table. i.e; The foreign key of one table may consider as primary key of another table.

(iii) In a table we can have multiple columns as foreign key.

(iv) It allows the duplicate values as well as null values.

* Difference B/w Primary key and foreign key:-

Primary Key

- (i) It does not allow the null values as well as duplicate values.
- (ii) It is known as unique identifier Attribute.
- (iii) The primary key may present one column in It acts as the parent table.

Foreign Key

- (i) It does not allow the null values as well as duplicate values.
- (ii) It is also known as referential integrity constraint.
- (iii) The Foreign key may present multiple columns.
- (iv) It acts as a child table.

(e) CANDIDATE KEY CONSTRAINT

- (i) Candidate key constraint has the properties of primary key. i.e; The Candidate key satisfies the behaviour of primary key.
- (ii) In a table we can have multiple Candidate key.

(f) ALTERNATE KEY:-

ii) Alternate Key constraint has the properties of primary key but are not treated as primary key.

(g) COMPOSITE KEY:-

When two (or) more columns are combined together to derive the properties of primary key is known as Composite key.

(h) CHECK:-

Check Constraint is used for applying extra validation to the attribute.

(i) DEFAULT:-

Default Constraint is used for inserting default value to the attribute.

* Data Definition Language:-(i) Data definition language is used for making permanent changes in the database.

(ii) The various commands in the data definition language are,

(a) Create

(b) Alter

(c) Rename

(d) Drop

(e) Truncate

(a) CREATE:- Create is used for creating a table into the database.

→ Using "Create" we can create the database too.

How to Create a table:-

Syntax:-

Create

→ Create database db-name

→ Create table table-name

(col-name) datatype(size) constraint, col 2 datatype
(size) constraint, ...)

e.g:- Create table Students-qsp

(id number(4) primary key, name varchar 2(15), doj date,
date);

(b) ALTER:- Alter is used for doing changes in the existing table structure.

→ Using Alter we can do the following changes.

- * Add
- * Rename
- * Modify
- * Drop.

* Add:- Add is used for adding a new column.

Syntax:-

Alter table table_name

add (col-name datatype (size) constrain,--);

Eg:-

alter table students_qsp

add (stream varchar 2(18));

* RENAME:-

Rename is used for renaming the column name with a new name.

Syntax:-

alter table table-name

rename column (column_name to New Name);

Eg:-

alter table students - qsp

rename column DOJ to DOA;

MODIFY:-

Modify is used for modifying the datatype Size or constrain of the existing columns.

Syntax:-

Alter table table-name

modify (col-name new datatype (new size) new
constrain);

Eg:- Alter table ~~table~~ students - gsp
modify (name varchar 2(20) not null);

A) DROP:-

Drop is used for removing the column along with the data from the table.

Syntax:- alter table table-name
drop column (column-name);

Eg:- alter table ~~table name~~ students - gsp
drop column doa;

B) RENAME:-

Rename is used for renaming the table name

Syntax:-
Rename table-name to new name

Eg:- rename students - gsp to qspiders - students1;
existing

* NOTE:- When creating a table with data if we pass the false condition the table may created "but data ~~wont insert~~"

* What happens to Create a duplicate table with data?

Ans:- Create table employees14
as (select * from emp);

Syntax:- Create table table-name
as (select ~~columns~~ * from table name);

* What to create a duplicate table from the existing tables without data?

Ans:- Create table employees16

as (Select * from emp where 1=2);

Syntax:- Create table table-name

as (Select columnname from table-name
where condition);

Eg:- Create table employees16;

as (Select * from emp where job='MANAGER');

(d) DROP:-

Drop is used for removing the complete table along with the data and table structure.

Syntax:- Drop table table-name;

Eg:- Drop table employees16;

(e) TRUNCATE:-

Truncate is used for removing the complete data from the table where as table structure retains the same.

Syntax:- Truncate table table-name;

Eg:- Truncate table employees14;

* Difference b/w Drop and truncate?

<u>Ans:</u>	<u>DROP</u>	<u>TRUNCATE</u>
	i) we can remove the complete data along with the table structure	i) we can remove the complete data and the table structure remains the same.
	ii) Drop will goes to the recycle bin	ii) Truncate will not goes to the recyclebin
	iii) we cannot use this for further references	iii) we can use this for further references

* DATA MANIPULATION LANGUAGE (D.M.L)

- Data Manipulation language is a temporary change to the database.
- It is a sub language under SQL.
- With the help of DML we can insert the data into the table, modify the data as well as remove the data from the table.
- The various commands under DML are

(a) Insert

(b) Update

(c) Delete

21/11/2021

(a) INSERT:-

Insert is used for inserting the data into the table.

Syntax:-

Insert into table-name

(col1, col2, -----);

value (v₁, v₂, -----);

Eg:-

insert into qspiders-students

(id, name, phone-no, doj)

values (1, 'RAJKUMAR', 7207509660, '10-SEP-2021');

* WAP to insert multiple rows of data in a single Query

Insert All into qspiders-students
values (2, 'RAJKUMARI', 7699044654, '10-SEP-2021')
into Q Spiders-students
values (3, 'RANI', 7699044677, '15-SEP-2021')
into Q Spiders-students
values (4, 'KANI', 7699044543, '15-SEP-2021')
Select * from dual;

* How to insert ~~multiple~~ rows of data in a

Single Query:-

FOR Multiple row Insertion:-

Syntax:-

Insert all into table-name
(col1, col2, col3, ---)
values (v1, v2, v3, ---)
into table-name
values (v1, v2, v3, ---)
into table-name
values (v1, v2, v3, ---)
into table-name
values (v1, v2, v3, ---)

Select * from dual;

For Single row Insertion:-

Syntax:-

Insert into table-name
(col1, col2, ---)
values (v1, v2, ---);

* Commit: - It is used for making temporary change to permanent change.

* UPDATE:

→ update is used for updating the data present inside the table.

→ It work in row wise (Tuple)

Syntax:-

```
update table-name  
Set col-name = value  
Where Condition;
```

Eg:- update QSPIDERS_STUDENTS
Set Name = 'RAJA'
Where ID = 5;

* DELETE:

→ Delete is used for deleting the data present in the table.

→ It works in row wise

Syntax:-

```
Delete table-name  
Where Condition;
```

Eg:- Delete QSPIDERS_STUDENTS
Where ID = 5;

TRANSACTION CONTROL LANGUAGE (T.C.L):

- T.C.L → Commit
- Roll Back
- Save Point

(a) COMMIT:- Commit is used for making temporary change to permanent change.

(b)

Commit is used for making DML command Change to permanent change

Syntax:- Commit;

(b) ROLL BACK:-

Roll Back is used to "go back to previous Transaction" before commit.

Syntax:- Roll back;

Eg:- delete qspiders-students
where id in (1, 2, 3);

Output:- 3 rows deleted

Select *
from qspiders-students;

Output:- 1 row

roll back;

(e) Save Point

Save point acts as a break point between the commit and roll back.

Eg: 14 rows

Delete 2 rows

Commit

12 rows

~~10 rows~~

Delete 2 rows.

10 rows

Save point A;

Delete 2 rows

8 rows

Save point B;

Delete 3 rows

Save point C;

5 rows

Roll back to A;

Data Control Language (D.C.L):

D.C.L:-

- (a) Grant
- (b) Revoke
- (c) flashback
- (d) Purge.

(a) GRANT:-

Grant is used to grant permission to various credentials.

Syntax:-

Grant Select on table-name to credential;

Eg:- grant select on qspiders-students to hr;

Output:- Grant Succeeded.

(b) REVOKE:-

Revoke is used for taking the Condition of permission back from the Credentials.

Syntax:-

Revoke Select on table-name from credential;

Eg:- revoke select on qspiders-students from hr;

Output:- revoke succeeded.

(c) FLASHBACK:-

Flashback is used for retrieving the table back from the recyclebin.

Syntax:-

flashback table table-name to before table;

Eg:- drop table qspiders_students;

output:- Table dropped.

flashback table qspiders_students to before drop;

output:- flashback complete.

(d) PURGE:-

Purge is used for deleting the data permanently from recycle bin.

Syntax:-

Purge recycle bin;

Purge table table-name;

STUDENTS

* NORMALIZATION:-
3/11/2021

- i) Normalization is a process of organising the data in the database
- ii) Normalization is used for minimising the redundancy from a table as well as dependency of data.
- iii) using Normalization we can divide the larger table into smaller table and link them using some relationship.
- iv) with the help of normalization we can reduce the undesirable characteristics such as insertion, deletion and modification anomalies.

STUDENTS						Department					
ID	NAME	Contact	Degree	Stream	DOJ	Branch ID	ID	Name	HOD	CONTACT	FACULTIES
1	Dinga	7202509660	Btech	CSE	12-Jun-21	10	CSE	K-RANI			RAJA MAHARAJA RANI

Types of Normalization:-

- (a) First Normal Form (1NF)
- (b) Second Normal form (2NF)
- (c) Third Normal Form (3NF)
- (d) Boyce Code Normal Form (BCNF)

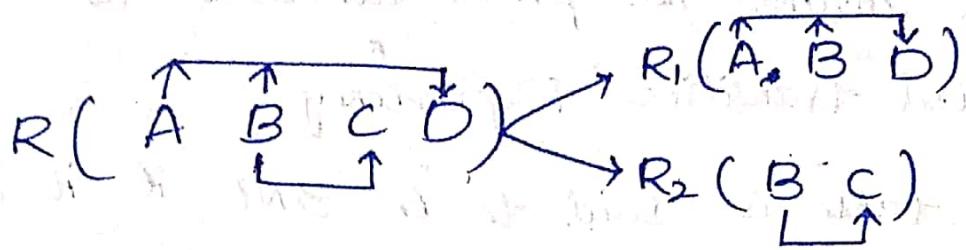
- (a) ~~First Normal Form (1NF)~~ :- ~~If each cell consists of single data~~
- ~~If each cell consists of single data~~ If each cell consists of single data
- (b) mono atomic data is called as 1NF.

Eg:-

DEPARTMENT					
ID	Name	HOD	Contact	Faculties	
10	CSE	K-Rani	720 7501660	Raja Maharaja Rani Maharani	It does not belong to 1NF

A table is said to first normal form when each cell of the table consists of ~~cell~~ mono atomic data (or) single data

(b) SECOND NORMAL FORM (2NF):-



$(A+B) \rightarrow (A B C D) \rightarrow$ Candidate key

$(A, B) \rightarrow$ Prime Attribute

$(C, D) \rightarrow$ Non Prime Attribute

alone define the

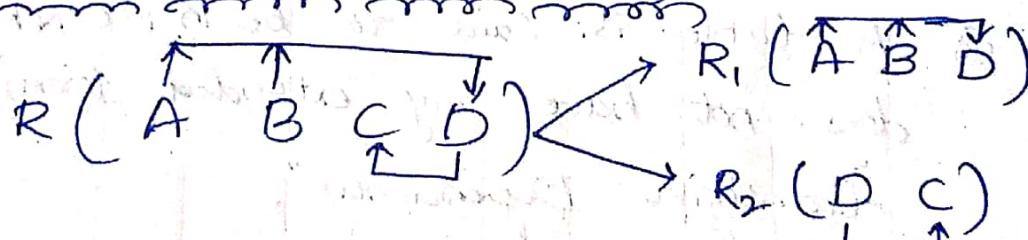
If Prime Attribute \rightarrow Non Prime Attribute

Partial then it is called Partial Dependency.

(i) When a prime attribute is deriving a non prime attribute it is known as Partial dependency.

(ii) A table is said to Second normal form if does not have any partial dependency.

(c) THIRD NORMAL FORM (3NF):-



$(A+B) \rightarrow (A B C D) \rightarrow$ Candidate key

$(A, B) \rightarrow$ Prime Attribute

$(C, D) \rightarrow$ Non prime attribute.

- If non prime attribute alone defining the another non prime attribute is called "transitive Dependency".
- A table is said to be 3NF if it does not have any transitive Dependency.

④ Boyce Code Normal form(BCNF) :-



$(A+B) \rightarrow \text{Candidate Key } (A, B, C)$

$(A, B) \rightarrow \text{Prime attributes}$

$C \rightarrow \text{Non-Prime Attribute}$

(i) When a non prime attribute is driving an another prime attribute it is known as extended form of "transitive Dependency".

(ii) A table is said to be in BCNF if does not have any extended form of "transitive Dependency".

(iii) When a non prime attribute is driving an prime attribute. Such non prime attributes are known as superkey.

(iv) In a table no non-prime attribute
should be a Superkey.