

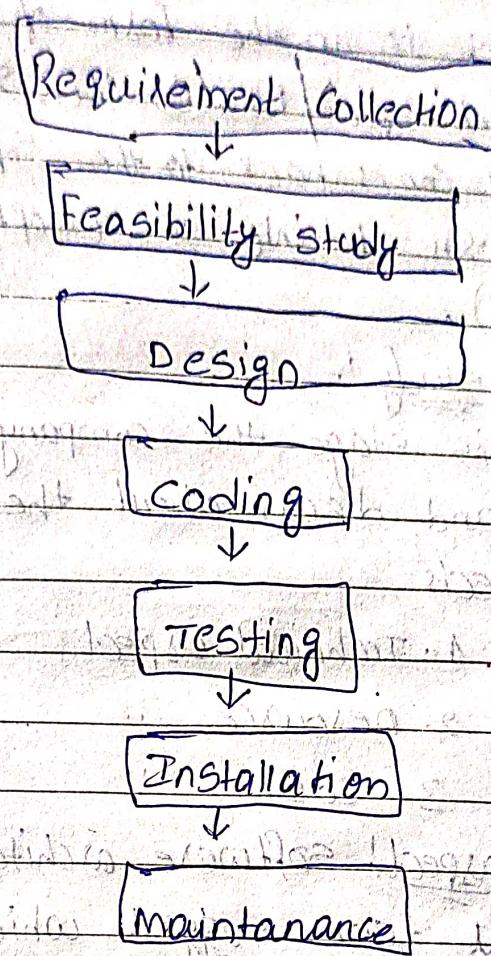
Date 16/04/21

Page 01

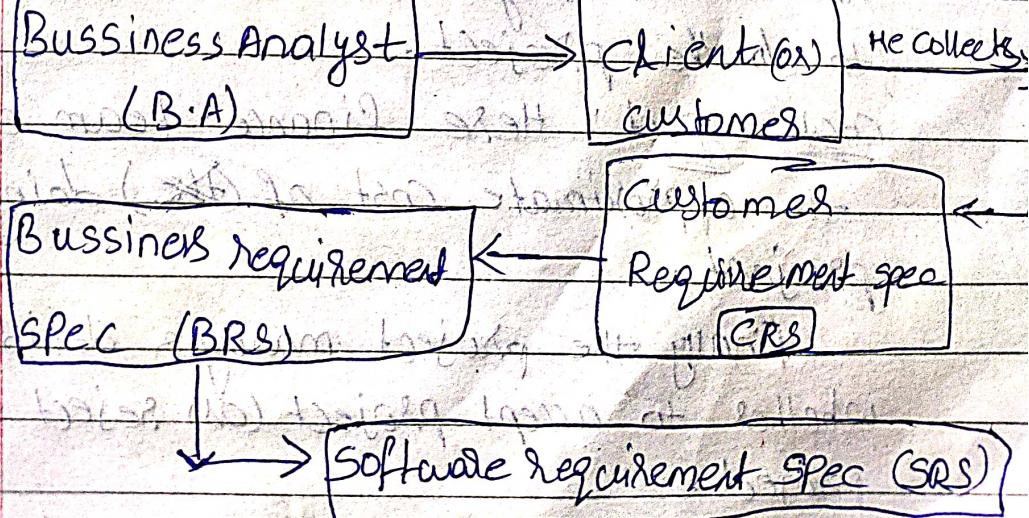
C-BEST BOOK

SDLC

It's a standard procedure we need to follow while developing an application.
It has few steps like



for example,



Requirement collection :-

In this stage the business analyst (B.A) goes to the customer and collects the requirements which is in the form of CRS (Customer requirement Spec) / BRS (Business requirement Spec) and comes back to the Software Company and converts it into SRS (Software requirement Spec).

Feasibility Study :-

In this stage the company members meet up and discuss about the following three aspects

1. Technical aspect
2. Resource ..
3. Cost

Technical aspect :- software architects and development team suggest which technology best suited to do the project.

Resource aspect :- H.R will make a list of all the things that needs to be arranged to do the project.

Cost aspect :- Here Finance team calculate the approximate cost of (**) doing the project.

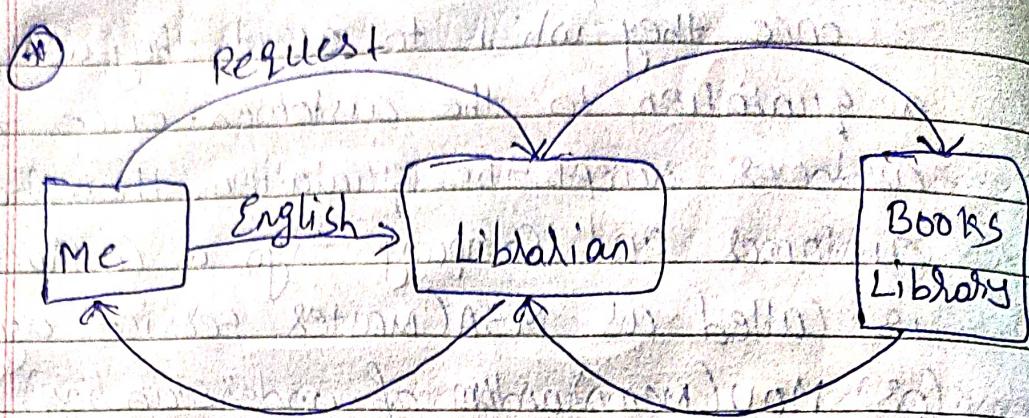
Finally the project manager decides whether to accept project (or) reject the project.

once they wish to accept, they will send a quotation to the customer. Once the customer accepts the quotation, then both customer and company sign a contract which is called as MSA (Master service agreement) (OB) MoU (Memorandum of understanding)

In the MoU they will have all the SLA (Service levelled Agrement) like

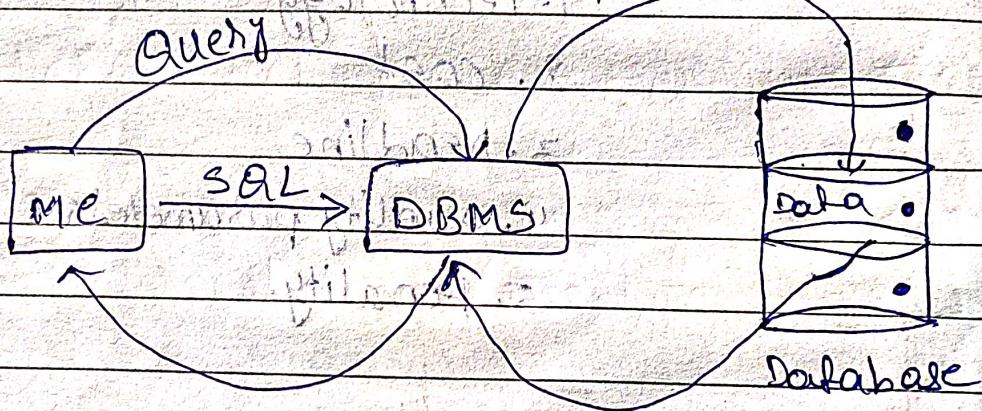
1. Technology
2. cost
3. deadline
4. quality parameters
5. penalty.

Ca



② (From English Librarian writing A-Z)

Query → Process



DATA

Date _____
Page _____
© BEST BOOK

- * A piece of useful information (0x)
- * small set of information.

What Is Database

- * place where you store data (0x)
- * IS an organised collection of information (0x)
- * Collection of one (0x) more tables.

What is DBMS

- * It is a program that stores retrieves or modifies data in the database on request.
- * Study of different techniques of design, development and maintenance of the database.

NOTE

- * First name of SQL is SEQUEL (STRUCTURED ENGLISH QUERY LANGUAGE).
- * Late they changed it to SQL (STRUCTURED QUERY LANGUAGE).

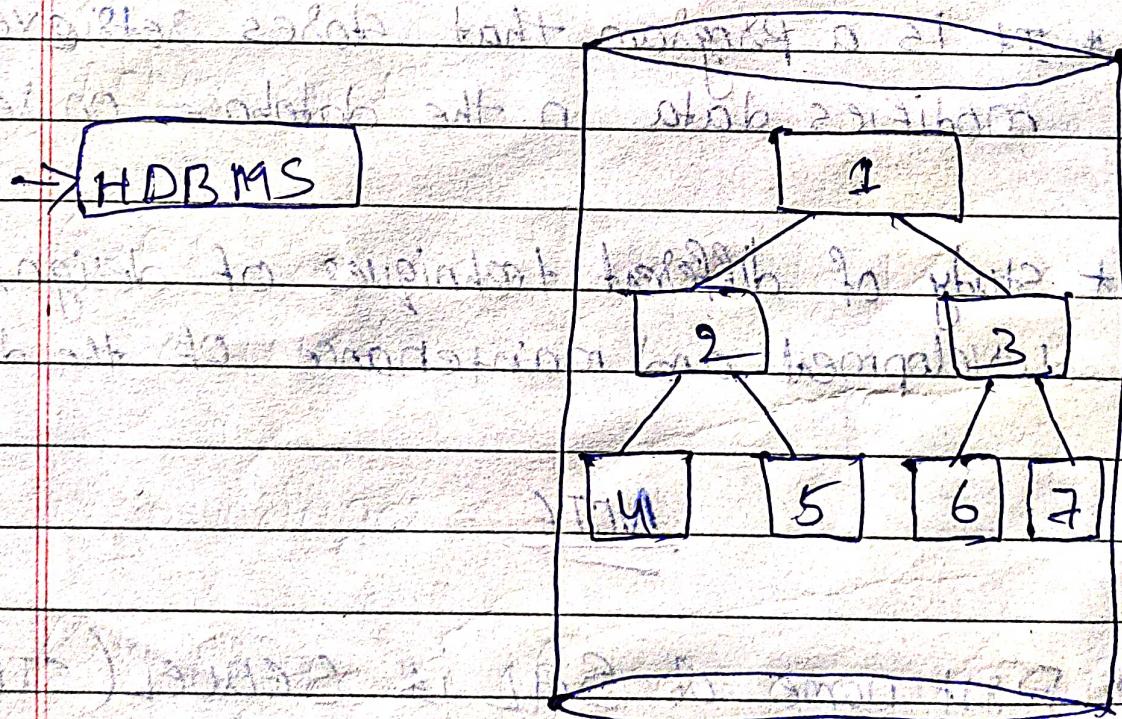
DBMS and its types

- * one of the job of DBMS is not only storing the data but also arranging the data.
- * To arrange the data we are having different types of DBMS.

1) HDBMS:-

HIERARCHICAL DATABASE MANAGEMENT SYSTEM

- * In hdbms the data is going to be stored in hierarchical manner (tree type structure).



Advantages:-

- ① If the data you are searching if it is present in first node then the time consumed for searching the data is less.



Dis Advantages:-

- ② If the data you are searching if it is present in last node then the time consumed for searching the data is more.
- ③ There is no guarantee that data will be present in the database.

After HDBMS:-

To rectify the disadvantages in hdbms they came up with another dbms called as "NDBMS".

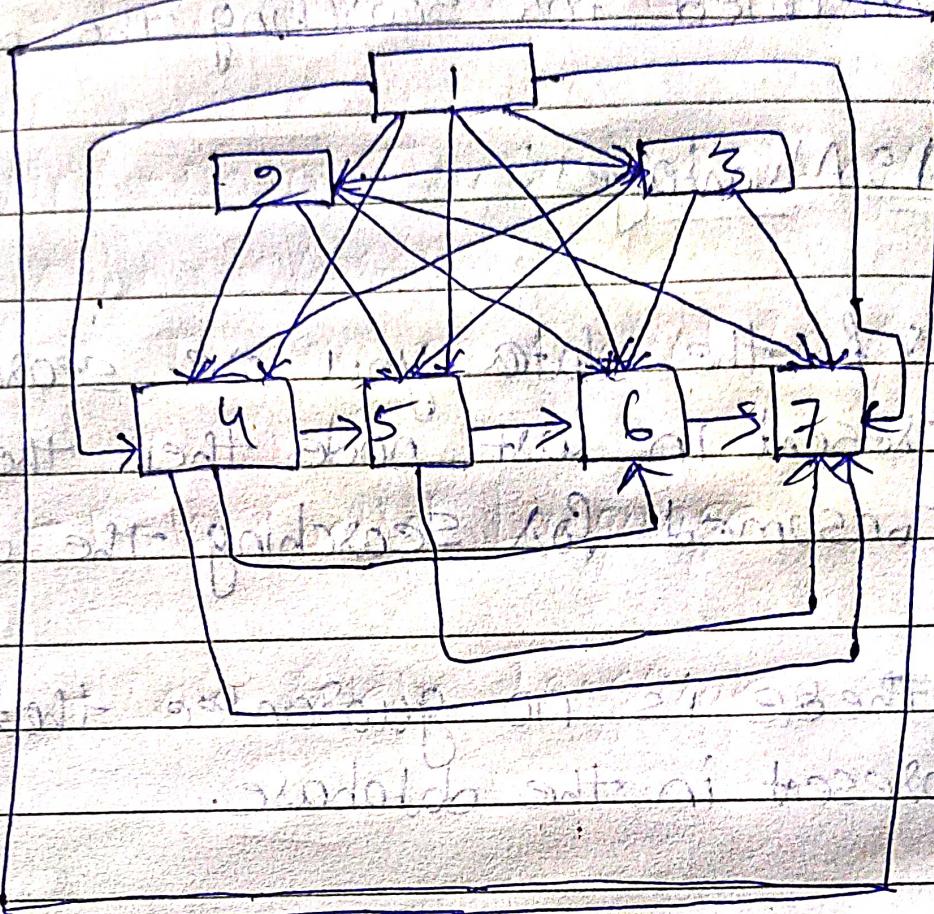
NDBMS:-

Network Data Management System

base

- ① In Ndbms the data is going to be stored in hierarchical manner along with networking structure.

→ NDBMS



Advantages

- ① All the disadvantages in dbms are rectified.

Disadvantages

- ② When the no of users are increasing the complexity of the database will be increased.

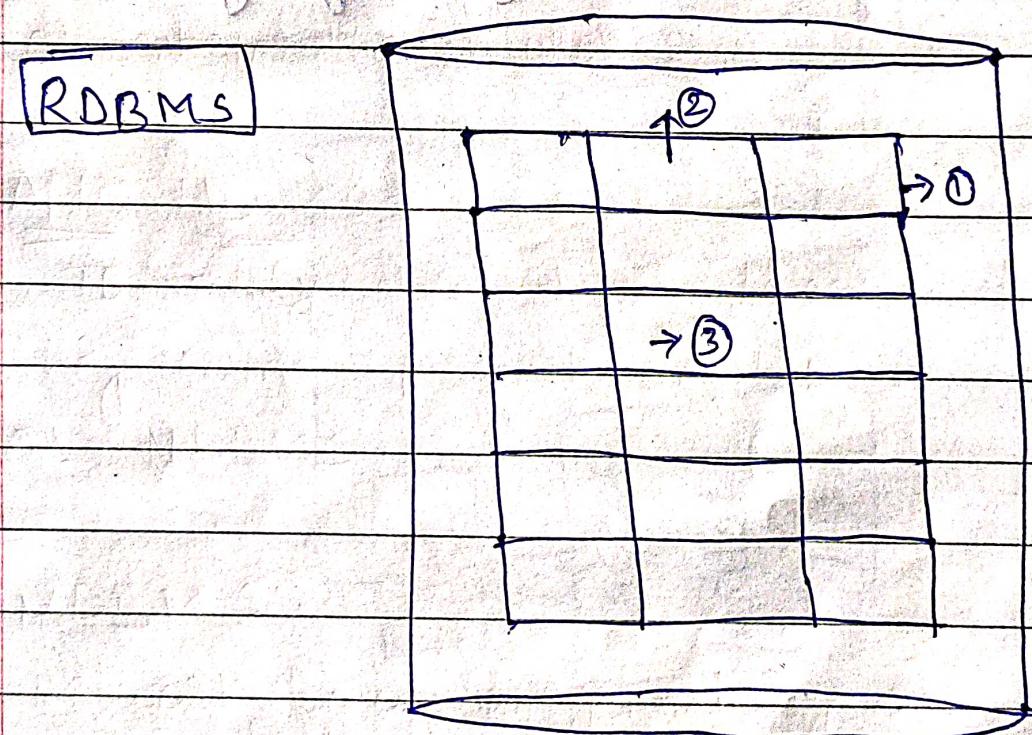
After NDBMS:

- ① Again to rectify the disadvantages in Ndbms they came up with another dbms called as RDBMS.

RDBMS

Relational DataBase Management System

- * In Rdbms the data is going to be stored in Tabular manner.



- ① The table name we call it as ENTITY
- ② Column name we will call it as ATTRIBUTE
- ③ All these are called as DATA.

S Q L T Y P E s

2 MARCH 2019

① DDL - Data Definition Language

② DML - Data Manipulation Language

* DTL/TCL - Data Transaction Language
(or)

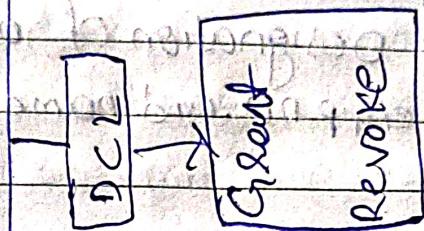
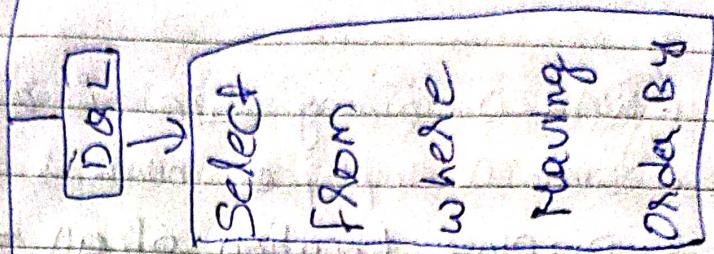
Transaction Control Language

③ DCL - Data Control Language

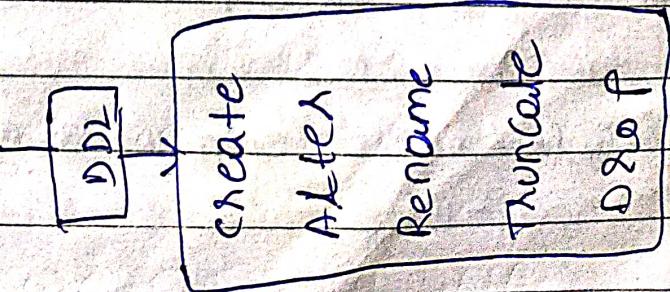
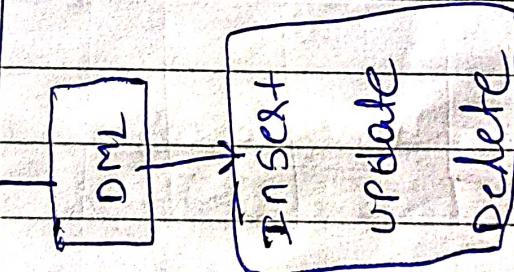
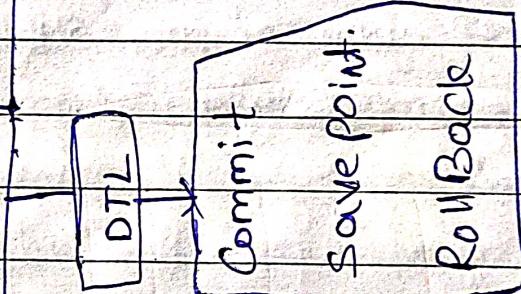
④ DAL - Data Query Language

Q <

③ <



SOL



Assignment

1. WAATO joining date of all the employees?
2. WAATO employees numbers of all the u?
3. Display Location of all the employees?
4. Display name and job of all the employees?
5. Display designation of all the employees?
6. WAATO emp no and name of all employen

2	3	4	5
6	7	8	9
0	1	2	3

2	3	4	5
6	7	8	9
0	1	2	3

Aliasing :-

Giving another name to the column name for displaying is known as aliasing.

Syntax :- (columnname as (optional) aliasname).

Ex:- Sal as salary.

Note :- If the aliasing name is having space then you need to give that aliasing name in double quotes ("").

queries:-

- ④ Select sal as "monthly salary" from emp;
- ⑤ Select empno, ename as name from emp;
- ⑥ select ename as name, sal as salary from emp;
- ⑦ Select ename, sal as salary, job as designation from emp;

Literals :-

It is a data.

3 types of literals are there,

- ① number - 420
- ② character - 'Hello'
- ③ date - '27/04/21'

In Oracle database the date format is
'DD-MM-YYYY' - '27-APR-2021'.

interview Q&A

- * select ename from emp;
- * select ename, sal;
- * select deptno, loc;
- * select ename, sal, comm, hiredate;
- * select ename, job, mgd;
- * select ename, deptno;
- * select * from emp;
- * select * from dept;
- * select ename, empno, sal, sal*1.2;
- * select ename, empno, sal, (sal*0.3) as "Hike in salary";
- * select 'employee' || ename || 'earns a monthly salary of rupees' || sal from emp;
- * select 'employee' || ename ||

venkat s18 18&A

- ① Select ename, mng from emp;
- ② Select * from emp where job = 'Salesman';
- ③ Select * from emp where job = 'Clerk' and deptno = 20;
- ④ Select * from emp where deptno = 20 and sal > 1500;

CONCAT OPERATOR :-

Concat operator is used for merging the data.
Ex:-

- ① select ename || job from emp;
- ② select 'HELLO' || ^{ename} from emp;
- ③ select 'HELLO' || ename from emp;
- ④ select ename || ' ' || job from emp;
- ⑤ write a query to display output in the below manner?

(i) SMITH IS A CLERK

→ select ename || 'IS A' || job from emp;

ARITHMETIC OPERATORS :-

+ , - , * , /

- ① select sal, sal + 500 from emp;
- ② select sal, sal * 12 from emp;

WHERE OPERATOR :-

① Select * from emp where ename = 'SMITH'

② Display all the information of ALLEN?

Select * from emp where ename = 'ALLEN';

① Display salary of James?

→ select sal from emp where ename = 'JAMES';

② Display job of scott?

→ select job from emp where ename = 'SCOTT';

③ Display department number of king?

→ select deptno from emp where ename = 'KING';

④ Display employee names of all the clerks?

→ select ename from emp where job = 'CLERK';

⑤ Display all the information of analyst?

→ select * from emp where job = 'ANALYST';

⑥ Display all the info of employees who joined on

03-dec-81?

→ select * from emp where hiredate = '03-dec-81';

⑦ Display employees who are getting sal more than 500?

→ select * from emp where sal > 500;

Arithmetic Operators:-

+ , - , * , /

Relational Operators:-

< , > , <= , >= , !=

Logical operators :-

and, or, not

Special Operators :-

between, and, in, is null, like

Between And Operator :-

when ever we have any range of values
that time we will make use of between
and operator.

Syntax :- Columnname between lowrange and
highrange

ex :- sal between 1000 and 2000

hiredate between '01-JAN-81' and
'31-DEC-81'

In operator :-

when ever we have multiple or conditions for a single column that time we will make use of "in" operator.

Syntax :- Columnname in (data)

Ex :- deptno in (10,30)

job in ('MANAGER', 'ANALYST')

IS NULL Operator :-

"IS NULL" operator is used to find out the null data present in the table.

Syntax :- Columnname is null

null data = 0 (wrong)

null data = space (wrong)

null data = null (right)

Ex :- Comm is null

mgr is null

Like operator :-

"Like" operator is a wild card operator, which is used to perform wild card operations & wild card operators:-

1. % - 0-n characters

2. _ - Single character

Syntax :- columnname like 'data'

Ex :- ename like 'S.%'

ename like '%.S'

ename like '%L.%'

ename like '%.E-%'

and so on & so on

NOT operator :-

"NOT" operator cannot be used solely, NOT operator should be clubbed with any of the special operators.

Ex :-

NOT Between And

NOT IN

IS NOT NULL

NOT LIKE

FUNCTIONS

- * Functions are used to perform operations.
- * There are two types of functions:

① Single Row Function

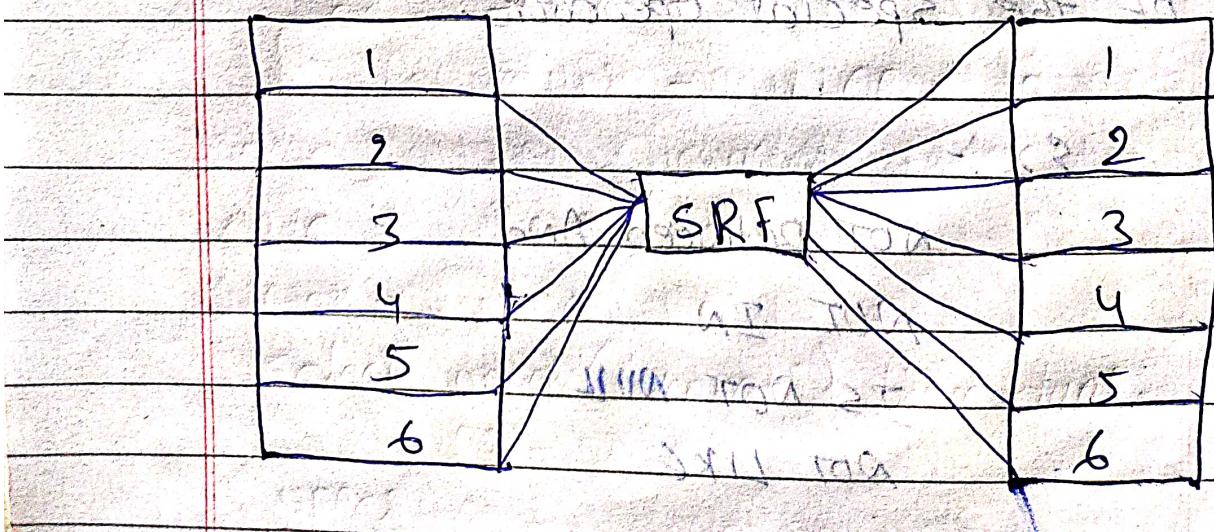
② Multi Row Function.

Single Row Function

- * These single row functions will take each row as input and produce output for each row.
- * Like that it will create a corresponding output for every row.

Ex:- `SELECT * FROM "T1"`

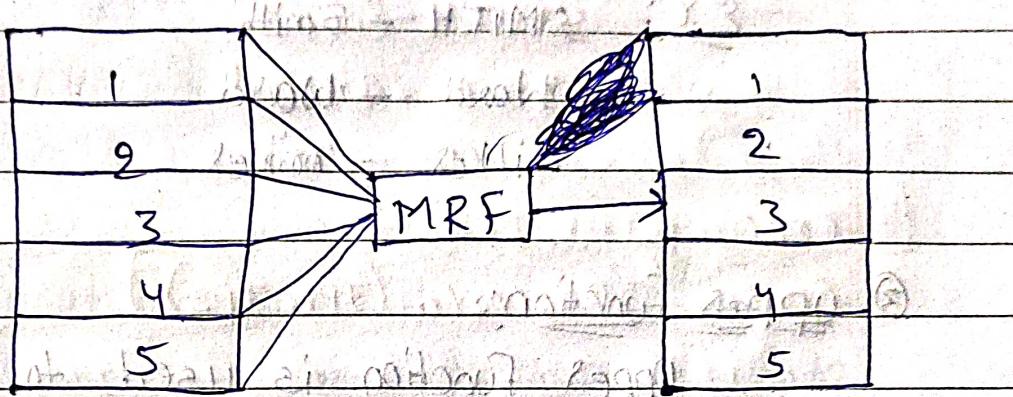
building of binds relationship



Multi Row Function :-

This function will take all the rows as input and produces exactly one output.

Ex:-



Single Row Functions :-

① Length Function :-

Length function will display no of characters present in the input.

Syntax :- Length(DATA)

: length("Smith") returns 5

Ex :- Smith - 5

Ward - 4

minch - 6

13770

MAXAV

② Lower Function :-

Lower Function is used to convert upper case characters to lower case characters and display.

Syntax :- Lower (DATA)

Ex :- SMITH - Smith

Ward - Ward

MILLER - mIller.

③ Upper Function :-

Upper Function is used to convert lower case characters to upper case characters and display.

Syntax :- Uppercase (DATA)

Ex :- smith - SMITH

mIller - MILLER

Note :-

Select upper ('venkat') from dual;

UPPER

VENKAT

④ INITCAP Function:

"INITCAP" function is used to display initial character in upper case and remaining characters in lower case.

Syntax:- INITCAP (DATA)

(DATA) → (Upper Case) + (Lower Case)

Ex :- SMITH → Smith

Ward → Ward

miller → Miller

⑤ Concat Function:

It is used to merge the data.

Syntax:- CONCAT(IP1, IP2):

Ex:- CONCAT('SMITH', 'CLERK')

SMITH CLERK

⑥ Nested Function:

Writing a function inside another function is known as nested function.

Syntax:- Function(function(DATA))

<-- Output →

<-- final output -->

Ex:-

Length(Concat('SMITH', 'CLERK'))
← SMITH CLERK →
Length(SMITHCLERK) = 10

WHAT is the output in the below manner using functions? (SMITH IS A CLERK)

SMITH - Ename

DS/A - Literal

CLERK - Job

concat(IP1, IP2)

Concat(IP1, concat(IP1, IP2))

Select Concat(ename, concat('DS/A', job))

From emp;

SMITH IS A CLERK

⑦

Replace Function :-

Replace Function is used to replace some part of Characters from the input.

Syntax :- Replace(IP1, IP2, IP3)

IP1 - DATA

IP2 - Character to replace

IP3 - Replacing character

Ex:-

replace('java', 'a', 'b') - jbv b

replace('manual', 'n', 't') - matual

replace('java', 'x', 'y') - java

replace('java', 'a') - jv

replace('testing', 'testing', 'sql') - sql

replace('corona', 'ona') - co

⑧ substring Function:

substring function is used to display some part of characters from a whole string.

Syntax:-

substr(IP1, IP2, IP3)

IP1 - DATA

IP2 - STARTING POSITION

IP3 - NO OF CHARACTERS

substr('developer', 1, 3) - dev

substr('developer', 3, 4) - velo

substr('developer', 7, 3) - per

substr('developer', -3, 3) - per

substr('developer', -5, 1) - t

substr('developer', 6, 6) - oper

subst('developer', -2) - el
Substr('developer', 4) - eloper

-9	-8	-7	-6	-5	-4	-3	-2	-1
d	e	v	e	l	o	p	e	r
1	2	3	4	5	6	7	8	9

max = 3, min = -2

IP3 - never be negative.

INSTR Function

It is used to display the position of a character.

Syntax: INSTR(IP1, IP2, IP3, IP4)

IP1 - DATA

IP2 - character to search.

IP3 - starting position

IP4 - occurrence

-9 -8 -7 -6 -5 -4 -3 -2 -1

d	e	v	e	l	o	p	e	r
1	2	3	4	5	6	7	8	9

: instr('developer', 'e', 1, 1) - 2

instr('developer', 'e', 2, 2) - 4

instl('developer', 'c', 1, 2) = 4
instl('developer', 'c', 3, 2) = 8
instl('developer', 'a', 1, 1) = 0
instl('developer', 'c', -3, 1) = 4
instl('developer', 'a', -2, 1) = 6
instl('developer', 'e', 1) = 2
instl('developer', 'v', 1, 1) = 3
instl('developer', 'v', 1) = 0

Ex :-

① select ename, instr(ename, 'A', 1, 1) from emp;

Q & A

① Display position of 2nd 'A' character in every job:

select job, instr(job, 'A', 1, 2) from emp;

② Display employee names starting with A?

select ename, from emp, where instr(ename, 'A', 1, 1) = 1;

③ Display position of no. 9 in every empno?

select empno, instr(empno, 9, 1, 1) from emp;

④ Display index number of 'l' in every employee name?

select ename, instr(ename, 'l', 1, 1) from emp;

⑤ Display employee information whose employee number starts with '7'?

select * from emp where Instl(empno,7,1) = 1;

⑥ Display employee information who joined in the month of FEB?

select * from emp where Instl(Hiredate, 'FEB',1,1) != 0;

⑩ Case Function

It is like ifElse statement in Java.

Syntax:-

Case column name

when Condition① then result①

 when Condition② then result②

 else default result

End

Ex:-

Case Job

when 'SALESMAN' then sal*2

when 'CLERK' then sal*3

when 'MANAGER' then sal*4

when 'PRESIDENT' then sal*5

else sal

END.

⑪ NVL Function : (NULL VALUE FUNCTION)

It is used to replace null data with some value.

Syntax :-

$\text{NVL}(\text{columnname}, \text{value})$

Ex:-

$\text{NVL}(\text{comm}, 0)$

`select comm, NVL(comm, 0) from emp;`

Display employee name, job, sal, total salary of all the employees? (Total salary = Sal + comm)

`select ename, sal, comm, sal + NVL(comm, 0) "Total salary" from emp;`

⑫ NVL2 Function :-

It is used to replace null data and not null data also.

Syntax :-

$\text{NVL2}(\text{comm}, \text{111}, \text{555})$

Ex:-

$\text{NVL2}(\text{comm}, \text{111}, \text{555})$

`Select comm, NVL2(comm, 111, 555) from emp;`

`select comm, NVL2(comm, comm, 555) from emp;`

Display ename, job, sal, total salary of all the employees (total salary = sal + comm) using ~~and~~
selected ename, sal, comm, NVL2(comm, comm, comm) from emp;

(or)

Select ename, sal, comm, NVL2(comm, sal + comm, sal)
from emp;

13 Systemdate Function:-

It is used to display current date that is present in the system.

Syntax :- select sysdate from dual;

SYSDATE

select SYSDATE from DUAL;

14 SystemTimestamp Function:-

It is used to display current date and time at the particular time of execution.

Syntax :-

SYSTIMESTAMP

SYSTIMESTAMP

select SYSTIMESTAMP from emp;

(15) TRUNC Function:

For this TRUNC function if you give any decimal value as input, it will remove that decimal value and it will display integer value as output.

Syntax :- TRUNC(DATA).

Ex :- TRUNC(52.3) - 52

TRUNC(49.8) - 49

TRUNC(88.5) - 88

* Select TRUNC(55.6) from dual;

TRUNC(55.6)

55

(16) Round Function:-

If you give decimal value as input it will round off that decimal value & it will display integer value as output.

Syntax :- Round(DATA)

Ex :- Round(42.2) - 42

Round(55.8) - 56

Round(95.5) - 96

Round(785.6) - 786

* Select ROUND(563.2595) from dual;

563

⑦ Mod Function:

If you give 2 values as input, it will divide those values and it will display the remainder of those values as output.

Syntax: MOD(IP1, IP2)

$$\text{Ex:- } \text{MOD}(20, 3) = 2$$

$$\text{MOD}(20, 2) = 0$$

>Select MOD(20, 2) from dual

0

Select * from emp where MOD(sal, 2) = 0

Select * from emp where MOD(sal, 2) != 0

Q & A

① Display ename, job, sal, hiredate, no of days working in the company for all the employee

Select ename, sal, job, hiredate, (sysdate - hiredate) "no of days working" from emp;

② Display all the employees information who are working in the company for more than 15700 days?

Select * from emp where (sysdate - hiredate) > 15700;

(3) Display ename, sal, job, hiredate, years of experience of all the employees?

select ename, sal, hiredate, round((sysdate - hiredate)/365) "YOE" from emp;

Note: Replace round as trunc for another method.

select ename, sal, hiredate, trunc((sysdate - hiredate)/365) "YOE" from emp;

(4) Display enames whose hiredate first digit is equal to years of experience last digit

select ename, hiredate, round((sysdate - hiredate)/365) as years from emp where substr(hiredate, 1, 1) = substr((sysdate - hiredate)/365), -1, 1);

(5) Display employees information whose last but one character in ename is similar to middle letter of the joining month?

select * from emp where substr(ename, -2, 1) = substr(hiredate, 5, 1);

⑯ Orders by Statement:

Orders by statement is used to arrange the data inside the database.

Syntax: order by Column name / Condition

Ex: order by ename;

If you want it in descending order then mention "DESC" keyword after the Column

order by ename DESC;

order by sal * 12;

① Select * from emp order by ename;

② Select * from emp order by ename DE

③ Select ename, sal, sal * 12 from emp

④ order by sal * 12;

⑤ Select ename, sal, sal * 12 "Annual Salary"
from emp order by "Annual Salary";

⑥ Select * from emp order by deptno;

⑦ Select * from emp order by deptno, ena

⑧ Select * from emp order by deptno, en
desc;

O & A

1. Display the list of employees joined on
01-may-81, 03-Dec-81, 17-Dec-81, 29 in ascending
order of seniority.

Select * from emp where hire date
(01-may-81, 03-Dec-81, 17-Dec-81) order by
hire date;

- 2) Display employees in ascending order of
designation who joined after second half
year of 1981?

Select * from emp where hire date >
'31-Dec-1981';

- 3) Display empno, ename, job, hire date, years
experience of all the managers and arrange
them in ascending order of their experience

Select empno, ename, job, hire date, Round
(Systdate - hire date)/365) "YOE"

from emp

where job = 'MANAGER'

Order by YOE;

Multi Row Function

multi row functions are "max, min, sum, avg, count".

- (1) select max (sal) from emp;
- (2) select min (sal) from emp;
- (3) select sum (Sal) from emp;
- (4) select Avg (Sal) from emp;
- (5) select count (*) from emp;
- (6) select count (Comm) from emp;

Q & A

- ① Display no. of employees working as analyst?
- select count (job) from emp where job = 'ANALYST';

- ② Display highest salary earned by clerk?
- select max (sal) from emp where job = 'CLERK';

- ③ Select total salary of department no 10 employees?
- select sum (sal) from emp where deptno = 10;

- ④ Display average salary of department no 20 employees?

select Avg (sal) from emp where deptno = 20;

⑤ Display least salary earned by manager?

select min(sal) from emp where job = 'MANAGER';

⑥ Display total salary earned by all the employees who joined in the year of 81?

select sum(sal) from emp where hiredate like '%.81';

⑦ Display no of employees working in deptno 10?

select count(*) from emp where deptno=10;

⑧ Display no. of employees working in each department?

select count(*) from emp group by deptno;

GROUP BY STATEMENT:-

It is used to group the records or used to divide a single table into multiple tables.

Syntax:- Group by Columnname / condition.

Group by deptno

① Select deptno, count(*) from emp group by deptno;

① Display no. of employees working in each job
= do select job, count(*) from emp group by job;

② Display maximum salary earned in each department?

= do select max(sal), deptno from emp group by deptno;

③ Display no. of employees joined in each year
= do select ~~job~~ substr(hiredate, -2), count(*) from emp
~~where~~ group by substr(hiredate, -2)

④ Display no. of employees in each job, select
only those jobs where no. of employees
are 3 and above?

= do select job, count(*) from emp group by job having count(*) >= 3;

Having statement:

It is used to provide condition for group by.

Syntax: Having condition

NOTE:-

- ① Having should be used only for group by.
- ② without group by statement we can't use having.

A&A

- ① Display total salary of each department only if the total salary exceeds 9000?

select deptno, sum(sal) from emp group by deptno having sum(sal) > 9000;

- ② Display average salary of each job except Analyst and select only those jobs whose average salary exceeds 1000 and arrange them according to job?

select job, avg(sal)

from emp

where job != 'ANALYST'

Group by job

Having avg(sal) > 1000

Order by job;

Date _____
Page _____
* BEST BOOK

order of usage:-

select
from
where
group by
having
ordered by

order of execution:-

From

Group by

Having

Select

ordered by.

Sub Queries

Writing a query inside another query is known as Sub Queries.

Q & A

- ① Display employees information whose salary is more than smith's salary?

Select * From emp,

where sal > (select sal from emp where ename = 'smith');

- ② Display employees information who are working in same department as James?

Select * From emp where deptno =

(select deptno from emp where ename = 'JAMES');

- ③ Display employees information who/are working in same department joined on the same date of Ford?

Select * From emp

where hiredate = (select hiredate from emp where ename = 'FORD');

④ Display employees information who are working in same job of miller?

Select * from emp
where job = (select job from emp where
ename = 'MILLER');

TYPES OF SUBQUERIES

1. Single row subqueries:

2. Multi row subqueries:

one-one $\rightarrow \exists_1, \exists_2, \exists_1 \exists_2$

one-many \rightarrow Any, All, \exists

Ex of any $\rightarrow \exists = \text{any}(10, 20, 30)$

Ex of All $\rightarrow \exists = \text{all}(10, 20, 30)$

① Display employees information whose deptno is same as any manager?

Select * from emp where deptno = Any(select deptno from emp where job = 'MANAGER');

② Display employees information who are working in same department as analyst?

Select * from emp where deptno = All(select deptno from emp where job = 'ANALYST');

③ Display employees whose joining date is same as James?

Select * from emp where hiredate =

(select hiredate from emp where ename='James');

④ Display location of Smith?

Select location from dept

where Deptno = (select Deptno from emp where
ename = 'Smith');

⑤ Display James department name?

Select Dname from dept where Deptno =

(select Deptno from emp where ename = 'James');

⑥ Display all the employees located in newyork?

Select * from emp

where Deptno = (select Deptno from
dept where Loc = 'New York');

⑦ Display location of all the salesman?

Select * from dept

where Deptno = 11 (select Deptno from
emp where Job =
'SALESMAN');

⑥ Display maximum salary employee name?

select * from emp

where sal = (select max(sal) from emp);

⑦ Display employees names who gets highest salary in their respective jobs?

select * from emp

where sal = (select max(sal) from emp)

group by job;

⑧ Display employee names who earns least salary in each department?

select * from emp

where sal = (select min(sal) from emp)

group by depno;

⑨ Display employee information whose salary is more than all the Analysts?

select * from emp

where sal > all (select sal from emp

where job = 'Analyst');

⑩ Display employees information whose salary is less than avg salary of each job?

select * from emp

where sal < all (select avg(sal) from emp

group by job);

(11) Display employees information who are working in Research department?

Select * from emp

where deptno = (select deptno from dept
where dname = 'Research');

(15) Display Second highest salary?

Select max(sal) from emp

where sal < (select max(sal) from emp);

(16) Display ename, job, sal whose salary is more than miller salary and job is similar to adams?

Select ename, job, sal from emp

where sal > (select max(sal) from emp where

ename = 'MILLER') and job =

(select deptno from dept where

(select job from emp where

ename = 'ADAMS'));

(17) Display employees information whose joining date is similar to scott and belongs to new york loc

Select * from emp

where hiredate = (select hiredate from emp where
ename = 'SCOTT'); and

deptno = (select deptno from dept

where loc = 'New York');

(15) Display employees information of second highest salary?

Select * from emp

where Sal = (select max(Sal) from emp where
< (select max(Sal) from emp));

Table Aliasing:-

Giving another name to the table name is known as Table Aliasing.

Syntax :- Tableename Aliasname

Ex :- EMP E

④ Select E.Ename from emp E

④ Select M.Ename from emp E (x)

④ Select M.Ename from Emp m

④ Select M.Ename name from emp m

JOINS

Joins are used to join two (or) more tables and display as a single table.

1. EQUIJOIN

For joining of two (or) more tables if we make use of "=" operator then it is known as Equijoin.

- ① Select * from emp, where deptno = deptno (x)
- ② Select * from emp, where $\overset{\text{Dept}}{\text{emp.deptno}}$ = $\overset{\text{Dept}}{\text{dept.deptno}}$;
~~Select * from emp;~~ $\overset{\text{Dept}}{\text{dept.deptno}}$;
- ③ Select * from emp E, Dept D where
 $E.\text{deptno} = D.\text{deptno};$

Q & A

- ① Display ename, Dname for all the employees?

select ename, dname from emp, dept where
emp.deptno = dept.deptno;

- ② Display ename, deptno, location for all employees?

select ename, emp.deptno, loc from emp, dept
where emp.deptno = dept.deptno;

- ③ Display ename, dname of all the salesmen?

select ename, dname from emp, dept where
emp.deptno = dept.deptno and job = 'SALESMAN';

② Non-Equi Joins:

For the joining of two (or) more tables if we don't make use of "=" operator then it is known as Non Equi Join.

③ Select * from emp, salgrade where sal between losal and hsal

A & A

① Display employee names and grades of all the employees?

Select ename, grade from emp, salgrade where sal between losal and hsal;

② Display name, salary, grades of all the employees?
Select ename, sal, grade from emp, salgrade where sal between losal and hsal;

③ Display all the information of grade 4 employees?
Select * from emp, salgrade where sal between losal and hsal and Grade = 4;

④ Display information of all the salesman?

Select * from emp, salgrade where sal between losal and hsal and JOB = 'SALESMAN';

⑤ Display employee information of Grade 3 and 4?

Select * from emp, salgrade where sal between losal and hsal and grade in (3,4);

- ⑥ Display Ename and grades of employees whose salary is more than ALL(En.Sal)

Select * from emp, salgrade where sal between Losal and Hosal and sal > (select sal from emp where Ename = 'ALLCA1');

- ③ Cross Joins / Cartesian Join :-

For joining of two (or) more tables if we don't provide any join condition then it is known as Cross Join. It will by default perform Cross Join.

* Select * from emp, ~~and~~ Dept;

* Select * from emp, Salgrade;

- ④ self Join

Joining the table again with it self is known as self join.

EMP \leftrightarrow Employee, Managers

EMP \rightarrow Employee (E) \rightarrow MGR

EMP \rightarrow MANAGER (M) \rightarrow EMPNO

* Select * from emp E, Emp m where

E.MGR = M.EMPNO;

Q & A

- ① Display all the employee names and their manager's name
select E.Ename, M.Ename from emp E, emp M
where E.mgr = M.empno;

- ② Display (all the) Ename, sal, manage name, manager sal
select E.Ename, E.Sal, M.name "manager name"
M.Sal "manager sal" from emp E, emp M
where E.mgr = M.empno;

- ③ Display all the employees information whose salary
and their manager's salary is same?
Select * from emp E, emp M where E.mgr =
M.empno and E.Sal = M.Sal;

- ④ Display employee information whose employee and
manager joined on same date?
Select * from emp E, emp M where E.mgr =
M.empno and E.hiredate = M.hiredate;

- ⑤ Inner join

It is used to display the common information
present in the table.

Select * from emp, dept where emp.dept
= Dept.Deptno;

⑥ Outer Join:

Outer Join will display inner join information and it will display left out data from tables. Outer Join concept is divide into 3 types:

1. Left Outer Join / Left Join

2. Right Outer Join / Right Join

3. Full Outer Join / Full Join

① Left Outer Join:

It will display inner join information and left out data from left side table.

>Select * from emp,dept

where emp.deptno = dept.deptno (+)

② Right Outer Join:

It will display inner join information and left out data from Right side table.

③ Select * from emp,dept,

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

④ Full Outer Join:

will display inner join information and left out data from left side table and Right side table.

⑤ In the oracle database Full outer join is not

⑧ It is just a theoretical concept.

ANSI (American National Standard Institution)

⑨ Formal we can use this.