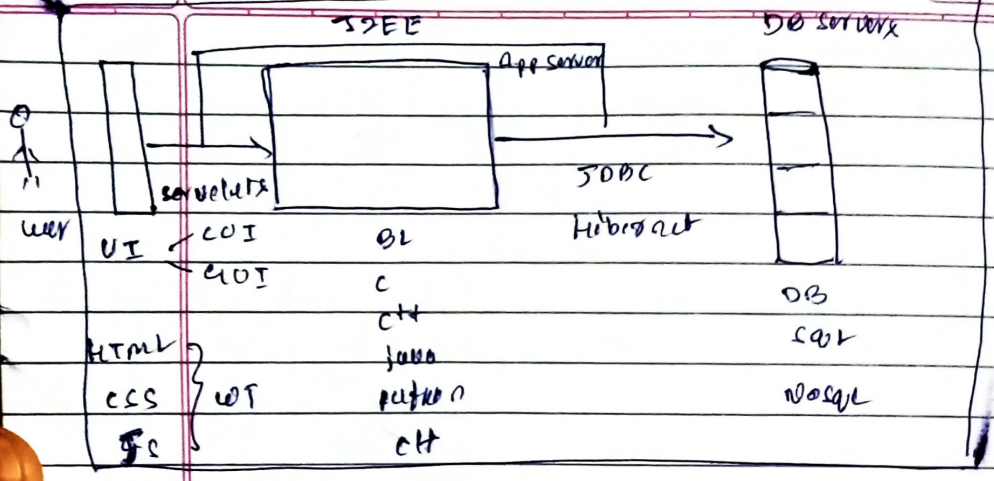
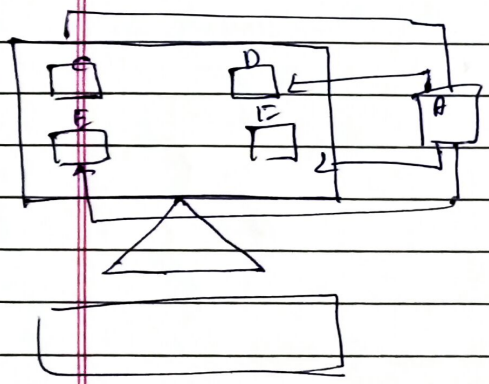
framework



-> flat file model



CUI -> ~~Character~~ Command User Interface

GUI -> graphical User Interface

HTML -> Hiper text markup language

CSS -> Cascading style sheets

JS -> Java script

BL -> Business logic

DB -> Data Base

J2EE -> Java 2 Enterprise edition

UI -> User interface -> which the user and computer system can interact using input and output devices.

Data :- A set of ... describes ...

Data classification

1) structured
2) unstructured
3) semi structured

A) structured Data ... without ...

Ex...

2) unstructured ... easily ...

Ex:- ...

3) semi struct ... help of ...

Ex:-

Data Base ... manipulation

Flat

NOTE ... most

Data :- A useful set of information or any thing which describes the real time entity is called information

Data classification

1) structured Data - RDBMS
2) unstructured Data - Non-RDBMS
3) semi structured Data - No RDBMS

1) structured Data :- the data which can be processed easily without recording is called structed data. -> Json
ex :- In amazon seeing details of products.

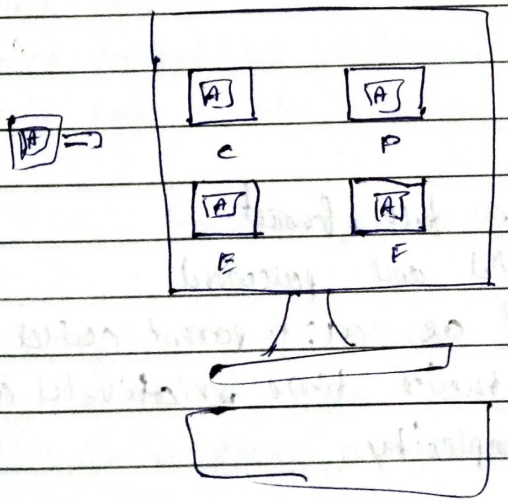
2) unstructured Data :- the data which can't be processed easily is called unstructured data.
ex :- Reading comments of product in amazon.

3) semi structured Data :- which can be processed with the help of so we key to get the result.
ex :- to open facebook initially we need to give gmail & password so that we get -> information

Data Base :- the software which deals with storage and manipulation of Data is Data Base

models of Data Base (Evolution)

1) Flat file model:

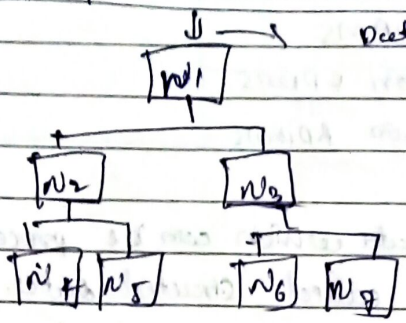


1) Data is stored in file format
2) Advantages -> Ease of Use
3) Disadvantage -> Data redundancy

NOTE:- Data redundancy -> Having the unnecessary multiple copies of the same data which leads to take more memory.

ARUN'S
PAGE NO.
DATE / /

② Hyrarchy data base model (HDBM)

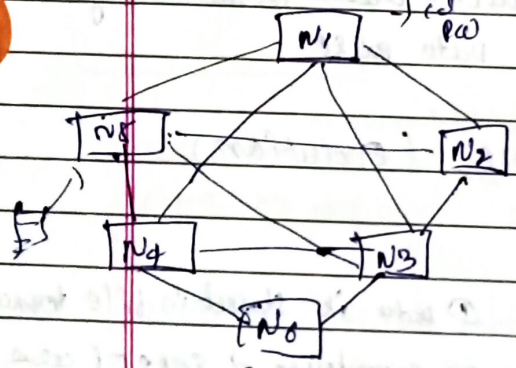


→ Data is should go & retrive from here on.

* the data should be stored in file format only
* the parential and children is m:n form
* Data storage and retrival should be from root only

Advantage :- elimination of redundency.
Disadvantage :- time consuming for data storage and retrival

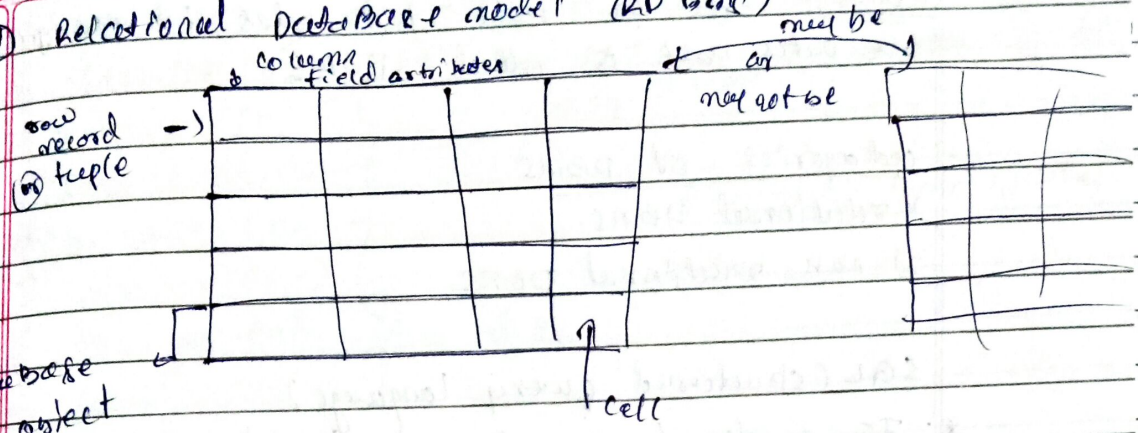
③ Network data base model :- (N-DBM)



* Data is stored in file format
* Every node has id and password.
* It is also called as m:n parent child
* Adv storage and fucur time relatively consume
* Disadvantages :- complexity

ARUN'S
PAGE NO.
DATE / /

(1) Relational DataBase model (RDBMS)



• Ex:-

	name	roll	marks	%
← one records →	A	105	40	90
	P	99	40	90

* RDB model was given by Edgar F. codd - 1971
* In RDB model the data is stored in the form of two dimensional table.
* the table is also called as database entity (or) database object
* every column in the table represents a field and hence called as Fields (or) attributes
* every Row is also called as Records or Tuple

Consistency :- giving right information

Categories of DataBase:

1) Relational Data Base (follows relational D.B. Rules)
2) Non relational Data Base (No SQL)

What is Data base management

DBM is a process of managing the data base with the help of source tools.

ARUN'S
PAGE NO.
DATE

Data type → size n=0 → once we entered the value we cannot change (length)

1) character (n) : char set
2) varchar (n) : string → here we can change (length)
3) varchar2 (n) : (' ') → within single quotes
4) Number (n) : numeric data
5) number (p,s) : (7,2)
6) Integer (n)
7) float (p,s)
8) ~~Data~~ Date
9) Boolean
10) BLOB → To store large binary values upto 4GB
11) CLOB → To store large character values upto 4GB for a cell

BLOB -> Binary large object
CLOB -> character large object

Constraints :-
not null
unique
primary key
Foreign key
Check
Default

prevent duplication

unique :- unique constraint helps in preventing duplicate values in a column.
ex :- student USN

1DB14MT056
1DB14MT001

→ no duplicates

not null :- not null constraint does not allow null in the column i.e. A value is mandatory in the column.
Ex :- in Appln there is scenario where we need to fill some blocks compulsory

name [ ]
ph no [ ]

compulsory — not null

ARUN'S
PAGE NO.
DATE / /

DBMS:- set of tools that helps in managing the data base is called DBMS

categories of DBMS
1) relational DBMS
2) non relational DBMS

SQL (structured query language)
* Is a only language used to interact with RDBMS
* SQL was given By IBM in 1973
* previously called as SEQUEL (simple or structured English query language)
* SQL helps us to maintain data integrity and consistancy.
integrity -> (data correctness) -> with the help of data types and constrains

null c [ ] 0

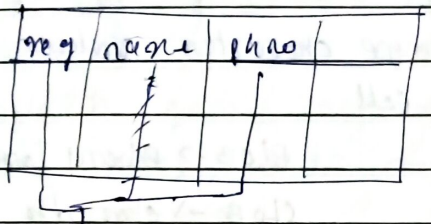
Data type : SQL commission
1) consistancy ✓
2) Availability ✓
3) scalability ✗
By data
1) consistancy ✗
2) Availability ✓
3) scalability ✓

ARUN'S
PAGE NO.
DATE / /

primary key

* primary key is used to identify a record uniquely in a table
* if we declaring a column as a primary key by default the specific column will take as unique and not null
* if a column is both unique and not null then such columns are eligible to become primary key. and that column are considered to be Candidate Keys

↓
only for developers understand.



if both are primary key the reg block will choose as primary key then phone no is alternate key.

Equ -> candidate key = primary key + alternate key

$$CK = PK + AK$$

$$\Rightarrow AK = CK - PK$$

Foreign key

* It used to create relationships across multiple tables.
* Foreign key is also called as referral integrity constrain
* Foreign key does not follow unique and not null constrain
* A primary can exist independently but a foreign key can't without corresponding primary key
* More than foreign key columns are allowed in the table

ARUN'S
PAGE NO.
DATE / /

Check

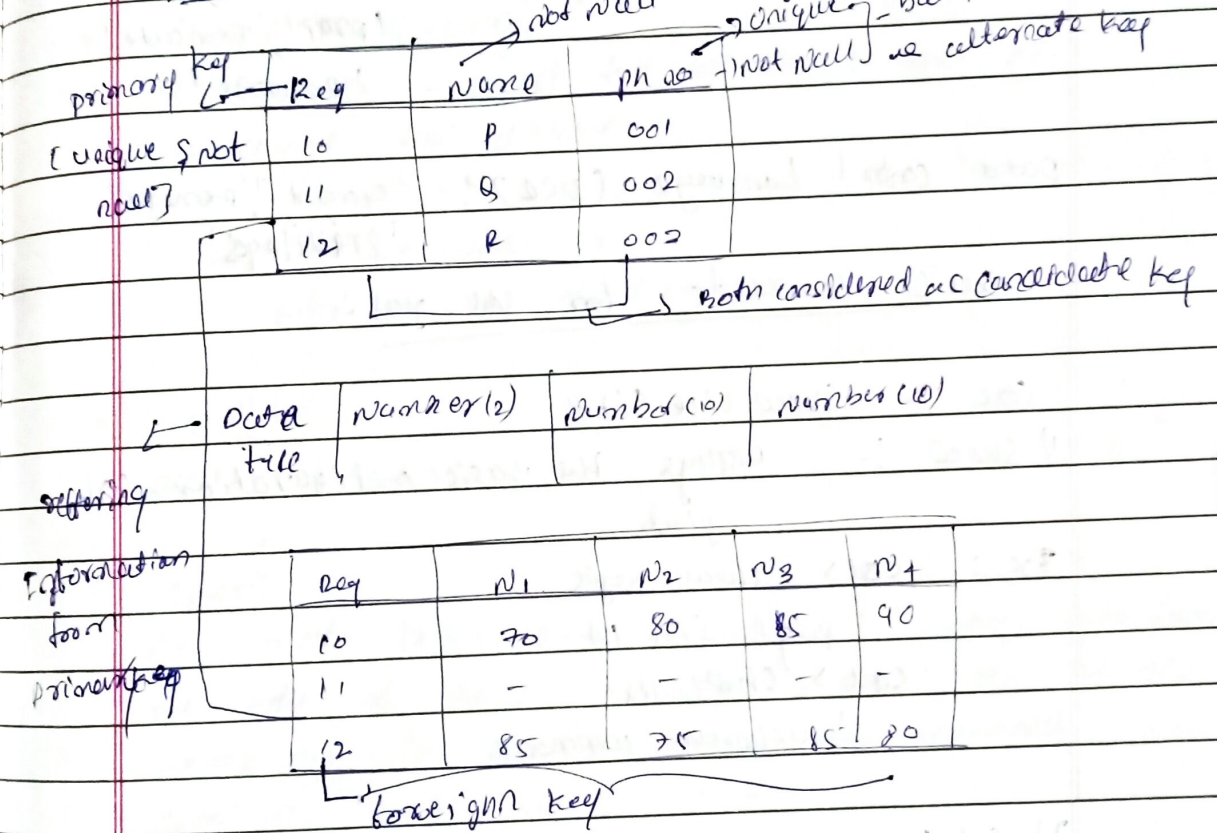
check constrain is used to apply some additional business requirement to the column

ex:- for ph no – there should be only 10 no's

Default :- * Default constrain is used to set the default value for ~~un~~defined column

* By default the default value is null (if we not assigned any value to default)

// * example for all constrains



ARUN'S
PAGE NO.
DATE / /

Sub language of sql

Data Query Language (DQL) :- statements 'select'

Data manipulate language (DML) :- 'insert' 'update' 'delete'

Transaction language (TCL) :- 'commit' 'savepoint' 'rollback'

Data Definition language (DDL) :- 'create' 'Alter' 'drop' 'Truncate' 'rename'

Data control Language (DCL) :- 'grant' 'revoke' 'privileges'

Basic commands for sql plus :

SQL > show line size

1) show :- Displays the basic configuration sql plus

Ex : SQL> show pages
page size 14
SQL> show lines
-) Unknown command.

2) set :- To set the values for sql plus no
Ex. SQL> set linesize 200
SQL> set pages 200

3) clear screen (or) cl scr
to clear screen of some data

4) Edit or ed
edit command is used to edit the immediate previously

executed query

5) / (or) r; (or) R;
command is
previously exec

6) describe or d
describe comma
specified ta
Ex : SQL> Des

7) CONNECT or
is used to
to change
Ex : 1) connec
enter us
enter p

8) exit
used to

9) spool
it used
the path of
whatever
spool off

SQL> desc
name
EMP NO
ENAME
JOB
MGR
HIRE DAT
SAL

executed query

5) / (or) r; (or) R;
command is used to re execute either immediate previously executed query (or) previously edited query.

6) describe (or) DESC
describe command is used get the description of the specified table.
ex: SQL> DESCRIBE EMP;

7) CONNECT or conn
is used to connect the one user to other (or) to change the director
Ex: 1) connect
enter user name: hr
enter password: ****

8) exit
used to exit

9) spool
it used to see the status of system spool path gives the path of the file will start pushing the commands whatever is executed after the command.
spool off will stop spooling

SQL> desc emp

name	null?	type
EMP NO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VAR CHAR2(9)
MGR		NUMBER(4)
HIRE DATE		DATE
SAL		NUMBER(7,2)

ARUN'S
PAGE NO.
DATE / /

comm		Number (7,3)
REPINO		Number (2)

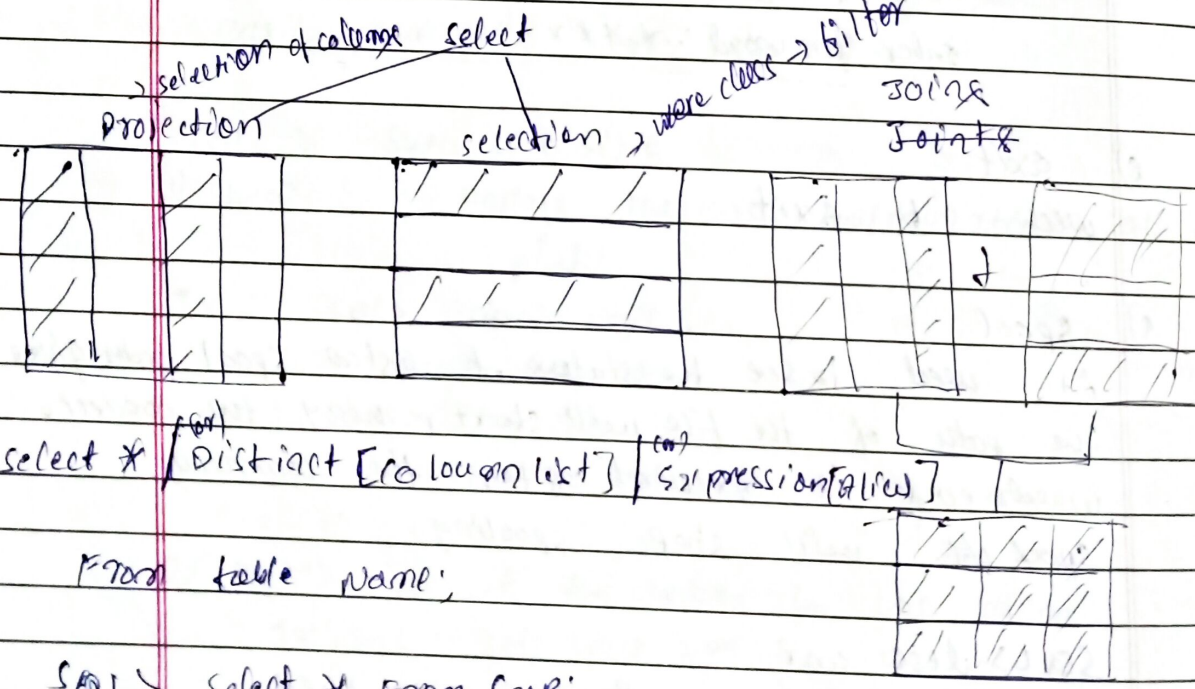
SQL> ~~sql~~ → desc dept

Name	Null?	type
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

SQL> ~~S-q-l~~ → desc sal grade

Name	Null?
GRADE	
LOSAL	
HISAL	

Data query language



select * | [col] Distinct [column list] | [col] expression [alias]

From table name;

SQL> select * From Emp;

SQL> Select ENAME, JOB from Emp;

ARUN'S
PAGE NO.
DATE / /

projection

Display employ names there HIREDATE, DEPTNO and salary of emp

SQL) select * hiredate, deptNO, salary from EMP;

SQL> select Distinct JOB from EMP;

NOTE SQL -> select Distinct JOB, Dept no from EMP;

-> if Distinct value is applied for two properties then it will give Distinct value in combination.

Error :- SQL> select Ename, Distinct JOB from EMP;

select ename, Distinct job from EMP;
*

Error at line 1:
ORA - 00936 : missing expression

Column Alias :- Giving Nick name

-> Renaming the column name while displaying result

Syntax : old name new name
old name as new name
old name "new_name"

Q. Display empNo. emp Names, their date of joining date, from the table, salary

-> SQL> select EmpNo, ENAME, JOB, hire date, salary from EMP;

PAGE NO.
DATE / /

using Alias

SQL > select empno Employee_number, ename Employee_name, Job Designation, hiredate Joining Date, sal monthly from emp;

Employee number	Employee name	Design	Join Date
2041	PAVAN	clerk	20-08-14

(Employee name: PAVAN ↓ Display from left to right)

Does not depends the size given bcz it is number display from right to left

NOTE ① for varchar and Date datatype the column name which is displayed is ~~restricted~~ to the max capacity of the column ~~districted~~ affected

Examples for all the syntax types

SQL> select sal as salary from emp;

Salary
:

(or) select sal salary from emp;

SQL> select sal ~~as~~ "salary" from emp;

Salary
:

(or) select sal ~~as~~ "salary" from emp;

select sal 'salary' from emp; → Error

Concatination operator (|| pipeline operator)

Concatination operator is used to concatenate data and literal

SQL > select ~~'hi'~~ || ename from emp

'hi', ename ↳ 'hi,' in single quotes

hi, allen

ex: SQL> select 'Hi'& || ename ~~from emp~~ || 'you are a', || Job from emp

→ no commas, if you want to give comma 'Hi,' (literal)

note there should be no open space b/w the sentence

ex: 'your_monthly_salary_is' || sal from emp

note: For new line

ex: select 'Hi', || [Enter]
'you are a' || Job || '. [Enter] → literal new line note query new line
from emp;

→ query new line not literal new line

Q: Display emp names along with their total sal
ex:- select ename, sal + ~~counter~~ comm from emp;
(+ → plus operator)

Ename	sal + comm
ALLEN	1400
WARD	1750
JONES	
MARTIN	2650

→ Here there is no value bcz it commission ~~or value is~~ not assigned it will not give the value

~~literal~~:-
A literal is an element which represents it's own meaning operation in SQL

1) Arithmetic operators

a) +
b) −
c) X
d) /

ARUN'S
PAGE NO.
DATE / /

Q-1 Display all the emp details working as salesman
=> SQL> select * from emp
where job = 'salesman';
↓ (SALESMAN)
note :- case sensitive records

Q -> Display all the emp who has salary greater than 1500;
=> SQL> select * from emp
where sal > 1500; → 1st will load.
check 2nd where condition where condition if satisfies

Q -> Display all the employees who joined after 02/04/81

SQL> select * from emp
where hiredate > 02/04/81; → error
pass like '02-may-81' → 9 rows

Q -> Display all the employees who in the dept no 20,30

SQL> select * from emp
where dept no = 20 and 30; → 0 row selected

Logical operator :-
logical operator are used to deal with multiple conditions

AND :- And operator returns true if both the condition are satisfied (true)

OR :- OR operator returns true if any one of the condition is true

NOT :- return true if the condn is false

SQL> select * from emp
where dept no = 20 OR Dept no = 30 ; → 11 records selected.

ARUN'S
PAGE NO.
DATE / /

2) Concatination operator
(||)

3) comparision operator
a) >
b) <
c) >=
d) <=
e) =
f) :=

4) IN, IS, LIKE } special operators

5) BETWEEN

6) NOT
AND } logical operators
OR
NOT

selection:-
syntax:- select distinct name, sal from emp

where condition (??)

* where class is used to filter the records based on the requirement
* the record is selected if the condition is satisfied (TRUE)
* if the condition is not satisfied (false) then the record is not selected.

one assignment
would be

ARUN'S
PAGE NO.
DATE / /

Q Display all the Employee Details who are working as salesman with salary greater than 1500

sal> select * from emp → 1 row
where job = 'salesman' and sal > 1500
'SALESMAN'

→ 1 row

Q Display both the emp Details ~~Service~~ emp Name and salary from the emp who is working a manager in dept no 30

SQL> select * from emp where job = 'SALESMAN' AND SAL > 1500;

Q who r working as clerk, salesman and manager.

sal> select * from emp where job = 'CLERK' OR job = 'SALESMAN' OR job = 'MANAGER';

Q Difference b/w primary key and foreign key

primary key	foreign key
1) uniquely identify a record in the table	1) Field in that table that is primary key in another table
2) can't accept null values	2) can accept multiple null values
3) We can have only one primary key in a table	3) more than one

sal> select * from emp
where job = 'manager' AND deptno = 30 AND sal >= 1500;
1 row selected

</html>

1	3	2	3	1
=	OR	: AND	= OR	=

OR operation take place if one condition is true

NOT(a&b)

NOTa OR NOTb

0	0	0
0	1	1 like this
1	0	1
1	1	1

AND operation take place if both condition is satisfied.

0	0	0
0	1	0 like this
1	0	0
1	1	1

SQL

Logical operators

display an Employee details such as salesmans, clerks and manager working in 20 & 30

ans → select * from EMP
2 where Job = 'clerk' or JOB = 'salesman' or Job = 'manager';

EMPNO	Name	Job	MGR	hiredate	sal	comm	DEPTNO
:		:			:		
:					:		

→ 13 rows selected

→ select * from EMP
2 where (Job = 'clerk' or JOB = 'manager') and (DEPTNO=20 or DEPTNO=30)

EMPNO	JOB	MGR	hiredate	sal	comm	DEPTNO
	clerk					20
	manager					20
						30
	manager					20
	manager					30

5 rows selected

→ select * from EMP
2 where Job = 'clerk' or Job = 'manager' and DEPTNO=20 or DEPTNO=30
all 20 and 30 selected and all clerk & manager selected
→ 10 rows selected

→ display all the salesmans and analyst details who is getting salary 1300 or greater & 3000 or less
→ select * from EMP
where (Job = 'salesman' OR Job = 'ANALYST') AND (SAL>=1300 AND SAL<=3000) → 4 row selected.

→ display the employees whose getting the sallary 1300 & 3000 Between
→ select * from EMP
WHERE SAL>=1300 AND SAL<=3000; → 8 row selected

→ display the Employee details who are getting sellary above 1300 & 3000
select * from EMP
WHERE SAL<1300 OR SAL>3000 (or) NOT (SAL>=1300 AND SAL<=3000)
(or) → 8 row selected

NOT (SAL>=1300 OR NOT SAL <=300

ARUN'S
PAGE NO.
DATE / /

-> Display employee details such as clerk and managers work in dept no 20, 30.

sol> select * from emp
where (job = 'clerk' or job = 'manager')
and (deptno = 20 or deptno = 30); => 5 rows

-> Display all the salesmans and Analysis details who is getting the salary 1300 or greater and 3000 or less

sol> select * from emp
where (job = 'ANALYST' or job = 'SALESMAN')
and (sal >= 1300 and sal <= 3000); -> 4 rows

-> Display the employees who is getting salary 1300 or more, 3000 or less

sol> select * from emp
where sal >= 1300 and sal <= 3000;

-> Display the employees who is not getting salary out of range 1300 to 3000

sol> select * from emp
where not (sal < 1300 and sal > 3000); 14 rows will selected.

where not sal >= 1300 or not sal <= 300;
6 rows

select * from emp
where (job = 'salesman' or job = 'Analyst') and
(deptno = 30 or deptno = 30) sal <= 3000
sal >= 1500 if select below 1500 also then y if it is not got

ARUN'S
PAGE NO.
DATE / /

SQL> Select * from emp
where job='manager' and deptno=30 and sal>=1500
1 row selected

SQL> Select * from emp
where job='clerk' or job='manager' or
deptno=20 or dept=30; 13 row selected

SQL> select * from emp
where (job='clerk' or job='manager') AND
(deptno=20 or deptno=30); 5 row selected

SQL> Select * from emp
where (job='clerk' or job='manager') AND
(deptno=20 AND deptno=30); No row selected

Because and operator checking 20 deptno as
30 deptno it is not possible but if we
give greater than equal it displays
Ex:- select * from emp
where deptno>=20 and deptno<=30; → 11 rows selected
then it select the 20 & 30 deptno which
employee working under 20 & 30 dept.

SQL> Select * from emp
where job='clerk' or job='manager' and deptno=20
or deptno=30; → 10 rows selected

ARUN'S
PAGE NO.
DATE / /

sql> select * from emp
where (job='Analyst' or job='salesman') And
(sal>=1500 or sal<=3000); 6 row selected

sql> select * from emp
where (job='Analyst' or job='salesman') And
(sal>=1500 And sal<=3000); 9 row selected

Bcz of in above query or operated select sal
salary 1500 select all above 1500 and 3000 select
all below 3000. Here better to use and operator
Bcz it checks both condition if satisfy
& gives result
Ex:- sql> select * from emp
where sal>=1500 OR sal<=3000;
14 row selected

sql> select * from emp
where sal>=1500 AND sal<=3000; 7 row selected

for = use or operator
for >= & <= use AND

sql> select * from emp
where not (sal>=1300 and sal<=3000); → 8 row selected

sql> select * from emp
where not sal>=1300 or not sal<=3000; → 6 row selected

sql> select * from emp
where not sal>1300 or not sal<3000;

-) Display all the employee details who is not working as salesmen & manager

PAGE NO.
DATE .. / /

- select * from emp
  NOT (Job = 'salesmen' OR Job = 'manager') → 7 rows selected
7 rows selected → NOT Job = 'salesman' AND NOT Job = 'manager';
  Job != 'salesmen' and Job != 'manager';

Special operators 26/4/19

→ Display all the employees who is not getting any commission

SQL > select * from emp
  where comm = null;
o/p → no rows selected

SQL > select * from emp
  where not comm >= 0;
  no rows selected

comparision operator fails with respect to null value

'IS' operator

IS operator use to deal with null values

SQL > select * from emp
  2 where comm is null → null without zero print

-) display all employee details who is getting commission

SQL > select * from emp
  where comm is not null

without null keyword IS operator not work

Ex SQL > select * from emp
  where job is 'manager';
  Error at line 2:

IS operator cannot used with any actual value

'IN' operator → In operator is used to deal with multiple value with respect to same column
In operator return to if the column matches any one of the value in the given list either true or else false

SQL > select * from emp
  where Job IN ('salesman', 'manager') → 7 row selected

We can use multiple arguments

select * from emp
where empno IN (10,20,30); → 1 row

SQL > select * from
where Job

Between ope
column

-) Between

① SQL >
where

② Select
where

Displa
and
and sa

→ Select
where
AND

Col
like

Syntax

the

It
operat

SQL> select * from emp
where JOB IN ('salesman', 'manager') AND DEPTNO IN (20, 30)

Between operator (it works with and operator)
column between low limit and high limit

→ Between operator is used to deal with range of values
(Same Between value => AND value)

① SQL> select * from emp
where sal between 1500 and 3000;
7 rows selected

② select * from emp
where ename between 'clerk' AND 'turner';
10 rows selected

→ display all employee details for clerks, managers and analysts who joined on or before 3rd April 81 and 3rd Dec 81 on or before.

→ select * from emp
where JOB IN ('clerk', 'manager', 'analyst')
AND HIRE DATE BETWEEN '02-APR-81' AND
'03-DEC-81';
5 rows selected

Column Like pattern
Like operator is used to evaluate a value in the pattern
Syntax column like pattern
— only one char after we can decide
_ → "0" or "1" more char
% → 0 or more char
the pattern is used printing "_ & %"

If the value is following the pattern then 'like' operator return true otherwise false
'_L' ;- it will not take because it consider as only two char has been come not using '%'
'%' → after this symbol whatever it is there not considering

ARUN'S
PAGE NO.
DATE / /

=> Display all employee details whose name is beginning with A

SQL> select * from emp;
where ename like 'A%';
2 row selected.

select * from emp
where ename not like 'A%';
12 row selected

-) Display all Employee one 'L' in its names

SQL> select * from emp
where ename like '%L%' AND ename not like '%L%L%';
for 2 'L's -) '%L%L%' -> 2 records } ending with L

(SQL) select * from emp ~~no row selected~~ where ename like '%L%'; -> 4 row
for 3 'L's -) 4 records

-) Display all employee details whose names contains atleast ~~at~~ ~~least~~ one 'L'
select * from emp; same
where ename like '~~%~~%L%' -> '%L%'; }

-) Display all employee where Job ending with 'MAN'
select * from emp
where Job like '%MAN'; -> 4 records selected
Salesman

=> Display all Employee details with name ending with L
(SQL) select * from emp } no row selected
where ename like '%L'; }

SQL> where ename like '%L' AND ename not like '%L%';
no row selected

SQL> select * from emp
where ename not like '%L';
14 records.

Functions

* Function is some specific
* depending on u defining by
* pre defined an

predefining

multi row function / group function

single row functions