Operating System File:

It is an stream of characters or bytes of information stored at one location.

Used to store the information in a PC.

It is managed by "File System" of OS.

OS File Disadvantages :

1. No Datatype Support

2. It will not support automatic Error Handling

3. It will not support Dynamic memory allocation

4. It support limited information

5. It provides poor security

6. It will not support Multiple users

7. It will not support Multiple platforms

8. Data retrievals and manipulations are time taking process.

Data : It is a collection of " Raw facts " from which conclusions can be drawn.

Database : It is a collection of " Related and Meaningful information " stored centrally at one location .

DataBase Management System (DBMS) :

It is a collection of programs working together to control Data Manipulations (add , change ,remove) , Retrievals (Read) and sharing on Database.

DBMS : Summary

It supports to Store information with Security.

It allows to manipulate(add,change,remove) , retrieve and share info.

DBMS Modals :

1. Hierarichal Modal (HDBMS)
2. Network Modal (NDBMS)
3. Relational Modal (RDBMS)
4. Object Oriented RDBMS Modal (ORDBMS)

1. Hierarichal Modal (HDBMS)

* Developed in early 1960.
* Used to represent the information in inverted tree structure (Hierarichal Tree).
* Data retrievals and manipulations are very faster compared to OS file.

Ex : IBM - IMS (Information Management System)

Hierarichal Tree Structure:

Disadvantage: Data Redundancy (duplication of data)

2. Network Modal (NDBMS)

* Developed in mid of 1960's.

* Used to represent the information using a physical link .

* No data redundancy like HDBMS.

* Data retrievals and manipulations are very faster compared to OS file and HDBMS.

Ex : Dbase , Foxpro

NDBMS MODAL :

Disadvantage:

* Supports Limited information .

* Provides Poor security .

DBMS Features :

* Supports Limited information .

* Provides Poor security .

* It will not support multiple users and

multiple platforms .

* It will not support E.F.Codds rules (maximum 4/12 rules supported).

* It will support limited levels of relations.(7 levels)

Ex: Dbase , Foxpro , Ms-Access , Oracle 6.0

3. Relational Model : (RDBMS)

Developed in early 1968

Features :

1. Information is represented in Rows and columns.

2. Intersection of row and column gives single value

3. Supports " NULL " values (Empty values)

4. Supports E.F.Codds Rules (min 6 / 12 Rules)

5. Supports " ACID " properties

A - Automicity , C - Consistency , I - Isolated , D - Durability

6. Supports Data Integrity Rules (Constraints)

7. Supports Mutiple users and Multiple platforms

8. Supports Unlimited data

9. Supports to store any type of information

(numbers , characters , strings , Text , date , time, Images and Files).

10. Provides High Security

11. No data redundancy like HDBMS

12. No physical link like NDBMS

13. Relations are maintained Logically thru Code.

14. Supports any Levels of Relations

15. Data retrievals and Manipulations are very faster compared to Os file and other Dbms models.

Ex : Oracle 7.x [7.0, 7.1, 7.2, 7.3]

Sql server , Sybase , My sql , DB2 , Teradata , Ingres

4. Object Oriented RDBMS (ORDBMS)

Supports to design the database objects implementing OOPS features .

Used to design the objects as per user specification.

Ex: Oracle 8.0

Oracle Versions

* Oracle 6.0 - It is an DBMS Tool

It will not support client - server architecture

* Oracle 7.x - (7.0 / 7.1 / 7.2 / 7.3)

It is an Modal RDBMS Tool

It supports client - server architecture.

* Oracle 8.0

It supports OOPS Features.(ORDBMS)

Used to develop the database components as per the user requirement .

* Oracle 8i : It is in_built with JAVA

Supports to execute Java programs with in Oracle server or client systems.

* Oracle 9i : Supports Advanced Features of JAVA.

Supports XML . Improved internal architecture related to Java .

* Oracle 10g :

Supports Grid Technology - used to store EJB components directly into Database .

* Oracle 11g /12c: Supports more advanced features related to DBA & other technologies .

O -- Oakridge **R** -- Relational **A** -- Analytical

C -- Computing **L** -- Logical **E** -- Engine

Oracle 9i Installation:

Oracle 9i --- 3 Cd's

1. copy software into PC

2. Open Cd1(disk1) and Dbl click on setup/autorun file in install/win32 folder

3. Choose

Oracle Enterprise edition

Oracle Server

Oracle Home directory (choose default path)

E:\oracle\ora92\BIN

Typical Installation

Provide Global SID name : oracle/orcl (Database name)

for rest of the options choose default values

4. Provide the passwords for "System" and "Sys" (DBA) Users.

Win XP + sp3 / 2003 -- Oracle 9i / Developer 6i (32 bit)

Win 7,8 / Vista -- Oracle 10g for vista (64 bit)

Linux --- Oracle for Linux (9i, 10g , 11g)

RDBMS --- Store

Security

Manipulations (add ,change ,remove)

Retrieve (Read)

Share

Oracle Data types:

Data type represents the type of information stored in memory location. 2 types.

i. **Simple Data types**

Automatically provided by oracle.

ii. **Composite Data types**

Defined by user in PL/SQL code.

ex: pl/sql records , pl/sql Tables ,Objects , collections

Simple Data types:

1. **Number(p,s)** P -- precision S -- scale

used to represent numeric information

maximum limit for precision is 32 digits (7.x) / 38 digits (8.0)

ex: empno number(4)

basic number(7,2) --- 12345.67

roll number(3)

fee number(5)

2. **char(N)**

used to represent character information.

maximum limit is 256 (7.x)/ 2000 bytes (8.0).

* Fixed length character datatype supports Static memory allocation.

ex: NAME char(20) -- 'ram mohan' -- 20

'sridhar' -- 20

sex CHAR(1)

pan_no char(10)

passport_id Char(10) pin cHaR(6)

3. varchar / varchar2(N)

used to represent character information.

maximum limit is 2000 (7.x)/ 4000 bytes (8.0).

* Varrying length character datatype supports dynamic memory allocation.

ex: name varchar2(20) -- 'ram mohan' -- 9

'sridhar' -- 7

email varchar2(100)

address varchar2(200)

description varchar(4000)

varchar - E.F.codd -- SQL

varchar2 -- Oracle -- SQL * Plus

Note:

* character values are case sensitive

* any information in single quotes (' ') is character

Ex: '11/07/15' '4:30 pm' 'AOSPK2165E' '*123#'

'1-11-110/A' ' ' 'ram@gmail.com' '22222'

4. Date

used to represent date and time information.

memory alloted is 7 bytes.

standard format of Oracle date and time is dd-mon-yy hh:mi:ss .

ex: birthday date

anniversary date

5. Long

used to represent numbers or characters

maximum limit is 2 GB.

ex: description long

6. Raw(n)

used to represent Images

maximum limit is 256 (7.x)/ 2000 bytes (8.0).

ex: photo raw(1000)

7. long raw

used to represent images upto 2 GB

ex: picture long raw

8. LOB's --- Large Objects [8.0]

used to represent huge loads of data

maximum limit is 4 GB.

4 types of LOB's are supported

i. CLOB -- represent characters

ii. NCLOB -- represent other language text

iii. BLOB -- represent Images

iv. BFILE -- represent OS Files (Address of file)

Nchar , Nvarchar2 - support other languages

SQL (Structured Query Language)

It is an 4th generation Language.

It is used in the development of every RDBMS tool.

It is developed by E.F.Codd using C Language (90%)

and Low Level Languages (10%).

It is used to develop solutions with 1 line code.

Oracle --- SQL * PLUS

Sql Server --- Transact SQL

Sybase --- Interact SQL

SQL * PLUS :

It is an enviroment provided by Oracle

corporation to work with Oracle DB.

It consists of SQL and Non-SQL commands.

SQL in SQL * PLUS :

SQL stmts further divided into 5 sub languages

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Data Querying Language (DQL / DRL)
4. Data Control Language (DCL)
5. Transaction Control Language (TCL)

I. Data Definition Language (DDL)

used to define database objects.

Database Object: Automatically stored into the software whenever it is defined.

ex: Table , View, Procedure , Function ... etc

Table :

Used to represent the information in every RDBMS tool. It represents data in columns and Rows.

A Table can have minimum 1 column maximum 1000 columns. A Table can have unlimited Rows.

DDL : i) create ii) Alter iii) Rename [8.0] iv) Drop v) Truncate

create : used to make new database objects.

syntax:

create table <tablename>

(column1 <datatype>,

column2 <datatype>,

column3 <datatype>,

.....

.....

column N <datatype>);

Naming convention :

a-z, A-Z, 0-9 , _ (under score)

max 30 characters

duplicates not allowed

; -- SQL stmt Terminator

Examples:

create table dept (deptno number(2), dname varchar2(20), loc varchar2(20));

create table student (roll number(3), sname varchar2(20),

course varchar(20), fee number(5));

create table emp (empno number(4), ename varchar2(20),

sex char(1), sal number(10,2), hiredate date, deptno number(2));

describe emp

desc dept

desc student

*** describe /desc** -- gives structure of table.

It is non-sql stmt supports in sql * plus only.

ii) **Alter :**

used to change the structure of existing table.

4 options

a) Add b) Modify c) Rename column[9i]

d) Drop column [8i]

Alter -- **add**

Used to add new columns to existing table.

alter table student add(phone number(10),email varchar2(50));

alter table emp add job varchar(20);

Alter -- modify

used to change the structure of existing columns.

*supports to increase / reduce column size, change data type

alter table emp modify(ename varchar2(50), sal number(16,2));

alter table student modify sname varchar2(50);

alter table emp modify sal number(8,2);

alter table student modify roll char(6);

Note: changing data type or reducing column size

is supported only if table is empty.

Alter - Rename column [9i]

Used to change existing column name.

only 1 column allowed at a time.

alter table dept rename column dname to dept_name;

alter table dept rename column loc to location;

Alter - Drop column [8i]

Used to remove existing column from table.

alter table emp drop column job;

alter table student drop (phone ,email);

Design at least 5 tables on each Domain :

BANK , HOSPITAL , INSURANCE , RAIL_RESERVATION , College/School/University.

iii. **Rename [8.0]**

used to change existing table name.

> rename emp to employ;

> rename dept to department;

desc emp -- error

desc employ

iv. **Drop**

used to remove the table permenantly from database along with data.

> drop table emp;

> drop table student;

> drop table dept;

> desc emp -- error

> desc dept -- error

II. Data Manipulation Language (DML)

Used to manipulate the information in Table.

DML: Insert , Update , Delete

Insert : Used to add new rows to existing table.

Syntax I:

insert into <table name> values (list of values);

Ex:

INSERT into DEPT values(10,'SALES','MUMBAI');

insert into student values(101,'RAM','oracle',2100);

insert into emp values(1001,'SRIDHAR','M',30000,'14-jul-15',10);

Syntax II: supports to insert values into selected columns of table.

insert into <table name>(column list) values (list of values);

Ex:

>insert into dept(dname,deptno) values('ADMIN',30);

>insert into student(roll,sname) values(102,'HARI');

>insert into emp(empno,ename,sex) values(1002,'KRISHNA','M');

* Rest of the columns are filled with NULL values (Empty values). NULL - No Value

Null : key word

used to represent empty values into table column.

supports with all types of data.

Using Null keyword :

insert into dept values(40 , 'TECH' , NULL);

insert into emp values(1003, 'RAJESH', 'M', null, null, 20);

Substitution Operator: & [&character]

-- used to accept the values from key board.

* -- valid for that particular stmt only.

-- valid in sql * plus only

insert into EMP(empno,ename,sex) values (&empno,'&ename','&sex');

Substitution Operator: &&

valid for compleate session.

>insert into student values (&roll,'&sname','&&course',&fee);

SQL Operators:

Arthematic	**Relational**	**Logical**	**Special**
+	=	and	in
-	!= or <>	or	between
*	>	not	like
/	<		is null
	>=		
	<=		

Where -- clause

used to specify the conditions while manipulating or

retrieving data from table.

used with Update , Delete and Select statements.

ii> **Update :** used to change existing rows in table.

Syntax:

update <table name> set <column1> = value

[, column2 = value , column3 = value , ... where <condition>];

Ex:

> update emp set sal = sal + 1000;

> update student set course = 'oracle 11g';

```
> update emp set sal = sal + sal * .12 where sal < 10000;

> update student set fee = fee + 1000 where roll = 102;

> update dept set loc = 'BANGLORE'where deptno = 30;


>insert into emp(empno,ename,sex) values(1005,'NAVEEN','M');

>Update emp set sal = 12000, hiredate = '27-jul-15' ,deptno = 10
  where empno = 1005;
 update dept set loc = 'HYDERABAD';
 insert into dept(loc) values ('vijayawada');
          null    null    vijayawada
 update dept set deptno = 60 ,dname = 'EXPORT' where loc = 'vijayawada';
  60  EXPORT  Vijayawada
```

iii> Delete :

Used to remove rows from existing table.

Syntax:

```
delete from <table name> [ where <condition> ];

Ex:

delete from emp;

delete dept;                    -- High Level Delete

delete from student;

desc student

delete from emp where sal > 80000;

delete from dept where loc = 'HYD';

delete from student where course = 'Java';

drop table student;

desc student  -- error
```

Note:

* DML operations can be cancelled.

* DDL cannot be cancelled. (Automatically saved)

III. Data Querying Language (DQL / DRL)

Used to retrieve the information from database for read only purpose.

*** SELECT** statement

Syntax:

Select < column list > from < table name >

[where < condition >

group by < columns >

having < condition >

order by < columns >] ;

Ex:

select empno, ename, sal from emp;

select roll, sname, fee from student;

select dname, loc from dept;

select * from emp;

select * from dept;

select * from student;

* -- represents all columns in table

select empno , ename, sal from emp where sal < 10000;

select roll, sname, fee from student where fee >= 1000;

select dname, loc from dept where deptno = 30;

Where - Clause Examples

select * from emp where ename = 'RAM';

select * from emp where job != 'MANAGER';

select * from emp where deptno = 20 and job = 'ANALYST';

select * from emp where deptno = 30 or job = 'SALESMAN';

select * from emp where empno = 101 or empno = 103

or empno = 105 or empno = 107 or empno = 109;

select * from emp where sal >= 10000 and sal <= 20000;

select * from emp where hiredate >= '1-jan-15' and hiredate <= '31-dec-15';

Special Operators :

Used to improve the performance of Oracle

while retrieving or manipulating data.

-- 4 operators

1. IN / Not IN
2. Between / Not Between
3. Like / Not Like
4. is null / is not null

1. In / Not in

Used to pick one by one value from given list of values. Supports with all types of data .

Examples:

select * from emp where empno in (101,103,105,107,109);

update emp set sal = sal + 5000 where ENAME IN ('RAM','RAVI','ANIL');

delete from emp where hiredate in ('10-jan-15' , '10-feb-15' , '10-mar-15');

select * from emp where empno Not in (101,103,105,107,109);

update emp set sal = sal + 2000 where ename Not in ('RAM','RAVI','ANIL');

2. Between / Not Between

Used to pick the values with in the specified range.

Supports with Numbers and Date values only.

Between is an Inclusive operator - It includes range limits in output.

Not Between is an Exclusive operator - It eliminates range limits from output.

Examples:

select * from emp where sal between 10000 and 20000;

select * from emp where hiredate between '1-jan-14' and '31-dec-14';

select * from emp where sal not between 10000 and 20000;

where sal >= 10000 and sal <= 20000 -- between

where sal < 10000 or sal > 20000 -- not between

between ---- >= and <=

Not between ---- < or >

3> Like / Not Like

Used to search for a pattern in given input.

supports with character data only.

It uses 2 special characters (Meta characters)

i) **%** -- represents zero or more characters

ii) **_** (under score) -- represents one character

* List the employees whose name begins with 'S' .

select ename, sal, job from emp where ename like 'S%';

o/p: SANDEEP SITA S SHYAM SRIDHAR

* List the employees whose name ends with 'S' .

select ename, sal, job from emp where ename like '%S';

o/p: JAMES ADAMS SRINIVAS S

* List the employees whose name begins with 'S'/'s'

```
select ename, sal, job from emp where ename like 'S%' or ename like 's%';
```

* Employ Name begins with 'S'/'s' or ends with 'S'/'s'

```
select ename, sal, job from emp  where ename like 'S%' or ename like 's%'
or ename like '%S' or ename like '%s' ;
```

o/p: SIVA sravan JAMES adams

* List the employees whose name begins and ends with 'S' .

```
select ename, sal, job from emp  where ename like 'S%S';
```

o/p: SRINIVAS SAS SahaS SS

* List the employees whose name got 5 Letters .

```
select ename, sal, job from emp  where ename like '_ _ _ _ _' ;
```

* List the employees whose name got 2nd letter as I

```
select ename, sal, job from emp where ename like '_ I%' ;  ---  SITA  SIVA
```

* List the employees whose name got LL .

```
select ename, sal, job from emp  where ename like '%LL%' ;
```

MILLER ALLEN LLOYD DOLL

* female names ends with A or I

```
select ename, sal, job,sex from emp
where sex = 'F'  and (ename like '%A' or ename  like '%I');
```

cond1 -- sex = 'F'

cond2 -- (ename like '%A' or ename like '%I')

```
select * from emp where ename not like 'S%';
```

* List the employees whose name got last 5 Letters as REDDY / RAO .

```
select ename, sal, job from emp where ename like '%REDDY';  '%RAO'
```

**** Searching for meta characters [_ , %]**

Using "escape option" we can search for meta characters in given input.

place '\' before meta characters(% , _) to convert as normal character.

* List the employees whose names having '_' (under score)

select ename, sal , job from emp where ename like '%_%' escape '\' ;

ANIL_KUMAR

KIRAN_KUMAR

SANDEEP_REDDY

*** search for emails with '_' .**

select ename, sal , job, email from emp where email like '%_%' escape '\' ;

ram_kumar@gmail.com

*** search for emails with out '_' .**

select ename, sal , job, email from emp where email NOT like '%_%' escape '\' ;

ravi909@gmail.com

* List the employees whose name got '%' symbol.

select ename, sal , job from emp where ename like '%\%%' escape '\' ;

ANIL%KUMAR

KIRAN%KUMAR

SANDEEP%REDDY

NULL values :

It is an undefined and uncomparable value.

It is not equal to zero or space.

It will not occupy any memory.

It is represented by NULL key word and displayed as space .

It is an standard value supported by every RDBMS tool defined by E.F. Codd.

Any arthematic operation with null value gives null.

* It provides unique treatment for all types of data.

4. is null / is not null :

Used to compare null values

supports with all types of data.

* List the employees who r not having commission.

```
select ename , sal , comm from emp  where comm is null;
```

* List the employees who r having commission.

```
select ename , sal , comm from emp  where comm is not null;
```

* Assign job as Executive if job is null.

```
update emp set job = 'EXECUTIVE' where job is null;
```

* Assign dept as 10 if deptno is null.

```
update emp set deptno = 10 where deptno is null;
```

* Assign hiredate as today if hiredate is null.

```
update emp set hiredate = '16-jul-15' where hiredate is null;
```

* Removing column value

```
update dept set loc = null where deptno = 20;
update emp set comm = null where empno = 7900;
delete from emp where empno = 7900;
```

Examples using different operators:

```
select empno, ename, sal, comm, deptno from emp
where sal > 20000 and comm is null and deptno in(20,30);
update emp set sal = sal + 2500 where sal between 10000 and 20000 and comm is null;
select empno, ename, sal, job, deptno from emp where job in ('MANAGER', 'ANALYST')
and sal > 15000 and deptno = 30;
delete from emp where job = 'CLERK' or sal < 3000;
```

Arthematic Expressions in Select :

```
select empno, ename, job, sal ,
sal * .25 ,
sal * .35 ,
sal * .15 ,
sal + sal * .25 + sal * .35 - sal * .15
from emp where sal > 20000;
```

Alias Names :

Used to provide temporary headings for columns

or expressions in select statement.

They are valid in that statement only.

They are only for display purpose .

```
select empno, ename , job, sal Basic,
sal * .25 Da,
sal * .35 Hra,
sal * .15 Pf,
sal + sal * .25 + sal * .35 - sal * .15  "Gross Pay" from emp where sal > 20000;

select dname as "Dept Name" , loc as "Location" from dept;
select dname,loc from dept; -- alias names r not repeated.
select ename, job, sal * 2 Bonus from emp;
select ename "Emp Name", job "Job", sal * 2 "Bonus" from emp;
select ename, job, sal * 12 "Annual Pay" from emp;
```

Built_in Functions in SQL :

Automatically provided by oracle

4 Types

1. Column Functions

Applied on each and every column value.

ex: Arthematic , Character & Date functions

2. Group Functions or Aggregate Functions

Applied on Group of rows

ex: sum ,count, avg ... etc

3. General Functions

Applied on any type of data

ex: Least , Greatest , vsize ... etc

4. Analytical Functions [9i]

Used to retrieve data analysis reports

ex: Lag , Lead , Rank ... etc

Arthematic Functions :

-- Applied on Numeric information only

* **abs(n)** -- gives absolute value of n

if n is -ve it converts to +ve

* **sqrt(n)** -- gives square root of given number

* **power(m,n)** -- gives m^n result

* **mod(m,n)** -- gives remainder after m/n operation

* **sin(n), cos(n) ,tan(n)** -- Trignometric functions gives results in Radians .

* **Ln(n)** -- Gives Natural Logarithm value for n

* **Log(m,n)** -- Gives Logarithm value for n to the base of m.

* **exp(n)** -- Gives e^n result

* **sign(n)** -- if n is +ve gives 1

if n is -ve gives -1

if n is 0 gives 0

* **round(m,n)** -- Gives nearest whole number for m.

* **trunc(m,n)** -- Gives only whole number and eleminates decimal value.

'n' indicates no.of digits after decimal (optional)

***ceil(n) [8.0]** --Gives value "more" than or equal to n similar to Round.

***floor(n) [8.0]** -- Gives value "less" than or equal to n similar to Trunc .

Examples:

select empno,ename, abs(comm - sal) from emp;

comm sal

80000 - 30000 = 50000

10000 - 30000 = 20000

null - 70000 = null

select abs(9999 - 1000000) from dual;

select abs(-990), abs(7766 - 8899) ,abs(55555 - 99999) from dual;

Dual : system table

Used to retrieve the general information thru select statement . single column , single row table . used to fulfil Select stmt syntax .

> desc dual

dummy varchar2(1) -- 'x'

select 4 * 5 from dual;

select 'iLOGIC @ 410' from dual;

```
select sqrt(625) ,power(4,2) , mod(15,2) from dual;

select sin(30) , cos(60) ,tan(90) from dual;

select Ln(35), Log(2,35) , Log(10,35) from dual;

select exp(1.17) from dual;

select ename, job, sal, power(sal,2) bonus from emp;

select sign(600 * 80), sign(800 - 9 * 600), sign(900 - 2 * 450) from dual;  ---  1  -1  0

select ename, sign(comm - sal) from emp;

* list the employees whose salary is less than 30000  using > symbol.

 select ename, sal, comm, deptno from emp where 30000 > sal;

 -- salary below 30000 [ where sal < 30000 ]

* list the employees whose salary is more than 30000  using < symbol.

 select ename, sal, comm, deptno from emp where 30000 < sal;

  -- salary above 30000 [ where sal > 30000 ]

 select round(19.7686), round(19.7686,2),round(20.2324), round(20.2324,2) from dual;

   20       19.77      20      20.23

 select trunc(19.7686), trunc(19.7686,2), trunc(20.2324), trunc(20.2324,2) from dual;

  19          19.76            20           20.23

  round(19.5)  -- 20   trunc(19.5)  --  19

 select ename, round(sal * .15) pf from emp;

select round(76867686.88667,-1) from dual; -- 76867690

select round(76867686.88667,-2) from dual; -- 76867700

select round(76867686.88667,-3) from dual; -- 76868000

select round(76867686.88667,-4) from dual; -- 76870000

select round(76867686.88667,-5) from dual; -- 76900000

select round(29.121) , ceil(29.121) from dual;

  29          30     ceil(89.000001)  -- 90
```

ceil(40) -- 40 round(40) -- 40

floor(40) -- 40 trunc(40) -- 40

select trunc(45.8688), floor(45.8688), trunc(45.8688,3) from dual;

45 45 45.868

Character Functions :

-- Applied on character information only

* **upper(s)** -- converts the given string to capital letters

* **lower(s)** -- converts the given string to lower case letters

* **initcap(s)** -- converts the starting letter of every word in given string to capital letters

* **Length(s)** -- gives no.of characters in given string

* **Reverse(s) [8.0]** -- gives the reversed string

select ename, upper(ename) , lower(ename), initcap(ename) , length(ename), reverse(ename) from emp;

select initcap('ilogic technologies') from dual;

select ename, sal, job from emp where length(ename) = 5;

* Employ name is a Palindrome (MADAM, NITIN)

select ename, sal, job from emp where reverse(ename) = ename;

* **Ascii(c)** -- American Standard Code for Information Interchange.

converts the given character to ascii value.

* **chr(n)** -- converts the given number to character (ascii to character)

select ascii('A'), ascii('B') , ascii('a'), ascii('b') from dual; --- 65 66 97 98

select chr(65),chr(66), chr(97), chr(99) from dual; A B a c

chr(0) .... chr(255) --- 256 ascii values

* **concat(s1,s2)** -- used to join the strings s1 & s2

|| -- concatenation operator

select concat('oracle', '11g') from dual; -- oracle11g

select 'iLOGIC'||' '||'Technologies' from dual; iLOGIC Technologies

select concat(concat(ename,' is a '),job) from emp;

KRISHNA is a MANAGER

select ename || ' is a '|| job from emp;

* **Lpad (s,n,c)** -- Left padding

* **Rpad (s,n,c)** -- Right padding

-- Fills the given string 's' upto 'n' characters with character 'c' on left or right side .

select ename , lpad(ename,10,'*') left, rpad(ename,10,'*') right from emp;

ename	left	right
ram	*******ram	ram*******
sridhar	***sridhar	sridhar***
anil kumar	anil kumar	anil kumar

Ex: 5000 ---> ***5000***

select lpad(rpad('5000',7,'*'),10,'*') from dual; -- ***5000***

select ename, lpad(rpad(ename,length(ename) + 3,'*'),

length(ename) + 6, '*') "Emp Name" from emp;

rpad(ename,length(ename) + 3,'*') -- ram***

lpad('ram***',9,'*') -- ***ram***

ram	***ram***
sridhar	***sridhar***
anil kumar	***anil kumar***

*** Ltrim(s) , Rtrim(s) , Trim(s) [8.0]**

eliminates the repeating characters on left or right or on both sides of given string.

By default it eliminates spaces.

```
select Ltrim('sssram','s'), Rtrim('ramsss','s'),
trim('s' from 'sssramsss') from dual;
  -- ram    ram    ram
select length('     srinivas     ') from dual;
select length(trim('     srinivas     ')) from dual;
length(Rtrim(Ltrim('     srinivas     ')))
```

*** replace(s,c1,c2)**

It replaces character c1 with c2.

It is an value by value replacement function.

```
select replace('Jack and Jue' , 'J', 'Bl') from dual;
      --  Black and Blue
select ename , replace (ename ,'I','EE') from emp;
update emp set ename = replace (ename ,'I','EE');
```

*** translate(S,c1,c2)** -- It is similar to replace but c1 and c2 must have equal no.of characters. It is a char by char replacement function.

```
select ename, translate(ename,'AEIOU','aeiou') from emp;
```

RAM RaM

SITA SiTa

UDAY uDaY

* **Soundex(s)** -- It gives encrypted sound value of given string.

used to compare strings with sound. used to search for character data very easily.

```
select soundex('oracle'), soundex('ilogic'), soundex('sridhar') from dual;
```

select ename, soundex(ename) from emp;

select * from emp where ename = 'smythy'; -- no rows

select * from emp where soundex(ename) like soundex('smythy');

7369 SMITH .....

*** substr(s,m,n)** -- used to extract a particular portion of string.

s -- string **m** -- starting position

n -- no.of characters to be extracted (optional)

select substr('oracle11g',7), substr('iLOGIC',2), substr('iLOGIC',2,3) from dual;

-- 11g LOGIC LOG

* employ names start and end with same letter

(ex: srinivas, nitin, david, asma, pradeep, naveen)

select ename, sal, job from emp where substr(ename,1,1) =

substr(ename,length(ename),1);

* employees having last characters as 'RAO'

select ename, sal, job from emp

where substr(ename,length(ename) - 2) = 'RAO';

[where ename like '%RAO']

RAMA RAO SRINIVAS RAO

select ename, sal, job from emp where substr(ename,length(ename) - 4) = 'REDDY';

-- last five characters

*** Instr(s,c,m,n)** -- Returns Number

Used to find the position of character in given string.

s -- string **c** -- character

m -- starting position **n** -- occurance no.

'ANAND' -- position of 'A' for 1st time -- 1

position of 'A' for 2nd time -- 3

position of 'A' for 3rd time -- 0

select instr('ANAND', 'A',1,1), instr('ANAND', 'A',1,2) from dual; -- 1 3

instr('ANAND', 'A',1,3) -- 0

instr('ANAND', 'A',2,1) -- 3

* position will be given according to compleate string

Ex: S -- 500 chars

find position of 'x' in last 50 chars

select instr(S,'x',451,1) from dual;

S -- 'ram oracle 1000' -- find position of space

select instr(S,' ',1,1) , instr(S,' ',1,2) from dual;

Date Functions:

Applied on Date information only

*** sysdate** -- Pseudo column

It gives the system date from server. Standard format of date is dd-mon-yy .

* **Pseudo column** -- It is an system key word where

value will be provided by oracle automatically.

* **Add_months(d, +N)** -- Returns Date

It adds N no.of months to the given date and returns date .

select sysdate from dual;

select sysdate , add_months(sysdate,12), add_months(sysdate,-12) from dual;

select add_months(sysdate,1200) from dual;

select ename, hiredate ,add_months(hiredate,6) "incr day" from emp;

*** Next_day(d,'day')** -- Returns date

It gives the date of next coming week day .

* **Last_day(d)** -- Returns Date

It gives the last day of the month in given date

```
select sysdate, next_day(sysdate,'sunday'), next_day(sysdate,'monday') from dual;
select last_day(sysdate), last_day('10-mar-15'),last_day('20-feb-15') from dual;
```

* no.of days in a month of given date

```
select substr(last_day(sysdate),1,2) days from dual;
select substr(last_day('&d'),1,2) days from dual;
```

* employees joined between 1-5 in every month

```
select empno,ename,hiredate from emp where substr(hiredate,1,2) between 1 and 5;
```

* **Months_between(d1,d2)** -- Returns number

It gives the difference between 2 dates in terms of months .

* Calculating Experiance of Employ in months

```
select ename, hiredate, months_between(sysdate , hiredate) "exp in months" from emp;
```

* Calculating Experiance of Employ in years

```
select ename, hiredate,
round(round(months_between(sysdate , hiredate))/12 ) "exp in years" from emp ;
```

* Experiance more than 5 years

```
where round(round(months_between(sysdate ,hiredate))/12 ) > 5
```

set linesize 130 -- no.of characters per line

set pagesize 40 -- no.of lines per page

```
select ename, birthday,
round(round(months_between(sysdate , birthday))/12 ) "Age in years" from emp ;
```

Date Arthematic :

Supports arthematic operations on dates. only Addition and Minus are supported .

Results will be calculated in terms of days .

```
select sysdate , sysdate + 10 , sysdate – 10 from dual;
select ename, hiredate, sysdate - hiredate  " exp in days "  from emp;
select ename, hiredate, round((sysdate - hiredate) / 365)  " exp in years " from emp;
```

date + number = date

date - number = date

date - date = number (difference in days)

date + date --- error (invalid operation)

Round & Trunc on dates:

Used to adjust the date to the beginning or end of week day (sunday) , month or year .

*** day , month , year** ---- key words

```
select sysdate, round(sysdate ,'day'), round(sysdate ,'month'),
round(sysdate ,'year') from dual;
select sysdate, trunc(sysdate ,'day'),   trunc(sysdate ,'month'),
trunc(sysdate ,'year') from dual;
```

Date Conversion Formats :

dd

ddd [day of year] 1 - 365

dy

day

mm

mon

month

yy

yyyy

year

hh

hh12

hh24

mi

ss

cc [Century]

Q [Quarter] 1 - 4

ww [week of year] 1-52

w [week of month]

sp -- spell out

ddsp

am / pm

* **to_char(d,'o/p format')** -- Returns Character

Used to convert the given date into character in specified format. (Date ----> Character)

Format indicates into which it has to be converted. Used for reporting.

```
select sysdate, to_char(sysdate,' dd/mm/yyyy hh:mi:ss am') from dual ;

select to_char(sysdate ,' dd  day,month,year hh24') from dual;

select to_char(sysdate , 'ddd') from dual;

select ename, hiredate , to_char(hiredate,'day') "week day" from emp;
```

* employees joined on SUNDAY only.

```
select ename, hiredate , to_char(hiredate,'day')
from emp where trim(upper(to_char(hiredate,'day'))) = 'SUNDAY';
```

select ename, hiredate , to_char(hiredate,'day')

from emp where upper(to_char(hiredate,'dy')) in ('SAT','SUN');

select to_char(sysdate , 'cc'), to_char(sysdate , 'Q'), to_char(sysdate , 'ww'),

to_char(sysdate , 'w'), to_char(sysdate , 'ddsp') from dual;

* Week Day of 1st of Next month

select to_char(last_day(sysdate) + 1,'day') from dual;

select last_day(sysdate) + 1 from dual;

*** to_date(c,'i/p format')** -- Returns Date in standard format of oracle. (dd-mon-yy)

Used to convert the given character into date.

Format indicates how the character value(input) is provided.

* to_char(d,'o/p format') -- Returns Character

* to_date(c,'i/p format') -- Returns Date(dd-mon-yy)

select to_date('12/25/2015', 'mm/dd/yyyy') from dual ; -- 25-dec-15

select to_char(sysdate,'mm/dd/yyyy hh24') from dual;

select to_date('21/apr/15 10:45:56 pm' , 'dd/mon/yy hh:mi:ss pm') from dual; **21-apr-15**

select months_between('20/05/20','10/16/05') from dual; -- error

select months_between(to_date('20/05/20','dd/mm/yy'), to_date('10/16/05','mm/dd/yy')) "mdiff" from dual;

months_between('20/05/20','10/16/05')

* months_between('20-may-20','16-oct-05')

Ex : Using to_date while inserting date values

create table person(pcode number(3), pname varchar2(20), birthday date);

insert into person values(101,'Ram','16-oct-87');

insert into person values(102,'Hari','12-nov-99');

insert into person values(103,'Giri','08-dec-08');

insert into person values(104,'Pavan','12-apr-11');

insert into person values(105,'Raj','05/11/1756');

**** Note: Default Notation**

If year <= 49 it considers current century (ex: 23 - 2023)

If year > 49 it considers previous century (ex: 65 - 1965)

Inserting exact date values with to_date :

insert into person values(101,'Ram',to_date('16-oct-1987','dd-mon-yyyy'));

insert into person values(102,'Hari',to_date('12-nov-1899','dd-mon-yyyy'));

insert into person values(103,'Giri',to_date('08-dec-2008','dd-mon-yyyy'));

insert into person values(104,'Pavan',to_date('12-apr-1911','dd-mon-yyyy'));

insert into person values(105,'Raj',to_date('05/11/1756','dd/mm/yyyy'));

select pname, round(months_between(sysdate,birthday))/12 "Age" from person ;

select pname, birthday, to_char(birthday,'dd/mm/yyyy') bday from person;

*** to_date converts the given input into standard date format i.e. dd-mon-yy**

insert into person values (106,'srinivas',to_date('20141229','yyyymmdd'));

Invalid options with to_date:

to_date('monday','day')

to_date('21','cc')

to_date('32','ww')

to_date('3','w')

to_date('2','Q')

to_date('Ten ,Nov,2010','ddsp,mon,yyyy')

to_date(' 25, dec,Two thousand ten','dd,mon,year')

Ex: Using to_date while searching date values

create table temp (c1 number(3) , c2 date);

select * from temp;

select * from temp where c2 = sysdate;

select c1, to_char(c2,'dd-mon-yy hh:mi:ss') from temp;

select * from temp where to_date(c2) = to_date(sysdate); -- 4 rows

*** to_date -- eliminates the time value and compares only date information .**

* while inserting date value time also recorded automatically.It is not visible in select output

Ex: select * from transaction where amt >= 100000 and tran_date = sysdate;

* Retrieving bank transactions done today

select * from transaction where amt >= 100000

and to_date(tran_date) = to_date(sysdate);

Transaction --- accno, tran_type, amt, tran_date

* Retrieve the students whose birthday is Today

select * from student where to_char(birthday,'dd-mon') = to_char(sysdate,'dd-mon');

* Employees joined in April Month

select empno, ename, sal, job, hiredate from emp where to_char(hiredate,'mon') = 'apr';

Inserting Time :

insert into student(admno,name,birthday) values(101,'MAHESH',sysdate);

*** sysdate automatically stores the system time along with date value .**

select sysdate from dual;

select ename, hiredate from emp;

NLS_date_format ---> dd-mon-yy

* changing system date format to accept time info

* valid only per session

alter session set NLS_date_format = ' dd/mm/yyyy hh24:mi:ss ';

NLS -- National Language set

select ename, hiredate from emp;

select sysdate from dual;

Group or Aggregate Functions :

Applied on group of rows.

They consider entire column content as one input source. They ignore null values.

* **Count** : Gives no.of rows in given column. Supports with all types of data.

* **Sum** : Gives total of given column . Supports with Numbers only

* **Avg** :(sum / count) Gives average of given column . Supports with Numbers only .

* **Min** : Gives smallest value of given column. Supports with Numbers and Date values.

* **Max** : Gives largest value of given column. Supports with Numbers and Date values.

* **Stddev** : Gives Standard deviation of given column. Supports with Numbers only [8.0].

* **Variance** : Gives Variance of given column. Supports with Numbers only [8.0].

```
select count(*) from emp;
select sum(sal) from emp;
select count(*) , sum(fee) from student  where course = 'oracle';
select sum(sal) from emp where job = 'MANAGER';
select count(*), sum(sal), avg(sal), min(sal), max(sal) from emp where deptno = 30;
select min(hiredate) , max(hiredate) from emp;
select stddev(sal) , variance(sal) from emp;
select count(comm), sum(comm) ,avg(comm) from emp;  -- nulls are ignored

select count(*) from emp; -- 28  ( nulls r not  ignored )
```

select count(empno) from emp; -- 25 (nulls r ignored)

General Functions:

Applied on any type of data , gives general info

* **User** -- gives user name.

* **Uid** -- gives user identification no.

* **Error / Err** -- gives errors occured in pl/sql block.

* **vsize(column)** -- gives the memory occupied by column value in bytes.

* **Least**(list of values / columns)

gives smallest value from given list.

* **Greatest**(list of values / columns)

gives largest value from given list.

> select user from dual; -- scott

> select uid from dual; -- 57

> show user -- scott

> show error

> show err

> select ename , vsize(ename), sal , vsize(sal) , hiredate , vsize(hiredate) from emp;

* Memory occupied by each row in DEPT table.

select deptno, dname, loc ,vsize(deptno) + vsize(dname) + vsize(loc) totmem from dept;

* Memory occupied by entire table content.

select sum(vsize(deptno) + vsize(dname) + vsize(loc)) from dept;

select least(1024, 21, 8899, 90, 3) from dual;

select greatest(1024, 5676, 8899, 90) from dual;

select name , least(m1,m2,m3,m4,m5,m6),

greatest(m1,m2,m3,m4,m5,m6) from stu_marks;

RAM 67 99

stu_marks -- roll , name , class , m1 , m2 , m3, m4, m5 ,m6

101 RAM 10 78 67 88 90 77 99 ---> 67 / 99

102 ANIL 10 89 88 77 66 55 58 [Row wise]

103 SIVA 10 58 87 78 98 86 75 student wise

min(m1) -- 58 max(m1) -- 89 [Column wise]

subject wise

least, greatest -- applied on rows

min, max -- applied on columns

```
select least('a','x','A','D') from dual; -- A
select greatest('a','x','A','D') from dual; -- x
select least('anil','anand','sunil','sham') from dual;
select greatest('anil','anand','sunil','sham') from dual;
select least('31-dec-15', '26-jan-26', '15-aug-20') from dual;  -- 15-aug-20
select greatest('31-dec-15', '26-jan-26', '15-aug-20') from dual; -- 31-dec-15
select greatest( to_date('31-dec-15'), to_date('26-jan-26'), to_date('15-aug-20'))
from dual; -- 26-jan-26
select least(to_date('31-dec-15'), to_date('26-jan-26'), to_date('15-aug-20'))
from dual;  -- 31-dec-15
```

*** to_date confirms the given input as date value**

*** NVL (column,value) :**

It assigns the value to the column if it is null.

It is used to perform arthematic operations with null values . Any arthematic operation with NULL returns

```
select ename, sal, comm, sal + comm net from emp;

select ename, sal, comm, sal + comm net ,

nvl(sal,0) + nvl(comm,0) "net pay" from emp;

select empno, ename, job, comm * 2 "Bonus" from emp;

select empno, ename ,job, nvl(comm,5000) * 2 "Bonus" from emp;

update emp set comm = nvl(comm,3000);

update emp set comm = 3000 where comm is null;
```

*** Decode :**

Used to check for mutiple conditions while retrieving or manipulating data.

It implements " IF " construct logic.

It will check for "Equality" condition only.

```
select ename, sal, deptno ,

decode(deptno , 10 , sal * 1.25 ,

                    20 , sal * 1.35 ,

                    30 , sal * 1.45 ,

                    sal * 1.5 ) bonus  from emp;

select empno ,ename , job , sal ,

decode( job , 'CLERK' , sal * .2 ,

              'SALESMAN' , sal * .3 ,

              'MANAGER' , sal * .4 ,

              'ANALYST' , sal * .6 ,

               sal * .11) " Incr amt "  from emp;
```

* Adding increment to employ salary

```
 update emp set sal = sal +

 decode( job , 'CLERK' , sal * .2 ,
```

```
'SALESMAN' , sal * .3 ,
'MANAGER' , sal * .4 ,
'ANALYST' , sal * .6 ,
sal * .11);
```

*** to_number(c) :**

Used to convert given character to number.

```
select to_number('1010') + 10 from dual;
select '1010' + 10 from dual;
```

*** to_char , to_date , to_number -- data convertion functions**

to_char --- date to character / number to character

to_date --- character to date

to_number --- character to number

*** Distinct - clause**

Used to eliminate duplicate values in select stmt

output . It is only for display purpose.

Applied on columns with duplicate values.

```
select course from student;
select job from emp;
select deptno from emp;
select distinct course from student;
select distinct job from emp;
select distinct deptno from emp;
select count(distinct job) from emp;
```

*** distinct clause consider NULL values also**

*** count ignore Null values**

*** distinct clause arrange results in Ascending order**

*** distinct clause applied on duplicate rows also**

*** stu_info**

roll name fee

101 ram 2500

101 ram 2500

101 ram 2500

102 ravi 1500

102 ravi 1500

102 ravi 1500

select distinct roll,name,fee from stu_info;

101 ram 2500

102 ravi 1500

Select - clauses:

Syntax:

select <column list> from <table name>

[where <condition>

group by <columns>

having <condition>

order by <columns>] ;

Group by -- Clause

Used to group the rows based on specified column.

Applied on column with duplicate values.

* When ever an ordinary column retrieved along with aggregate functions then all the ordinary columns must be specified after Group by clause.

select deptno from emp;

select sum(sal) from emp;

select deptno, sum(sal) from emp;

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

select course, count(*) , sum(fee) from student group by course;

select job, count(*), sum(sal) from emp group by job;

select hiredate, count(*), sum(sal) from emp group by hiredate;

select to_date(hiredate), count(*), sum(sal) from emp group by to_date(hiredate);

*** to_date ignores Time while grouping rows by date column. dd-mon-yy is considered.**

select mgr, count(*), sum(sal) from emp group by mgr;

*** Group by consider Null values as 1 option.**

select job,count(*),sum(sal),avg(sal),min(sal), max(sal) from emp group by job;

Having -- Clause

Used to apply the conditions on grouped results.

Generally used after Group by clause. Supported without Group by clause also.

Where clause will not support to apply conditions on Group functions (sum,count,...).

select deptno, sum(sal) from emp group by deptno having sum(sal) > 200000;

**** Having with out "Group by" clause**

select count(*), sum(sal), avg(sal) from emp where job = 'ANALYST' having count(*) > 5;

**** Having with "Group by" clause**

select job, count(*), sum(sal), avg(sal) from emp where job = 'ANALYST'

group by job having count(*) > 5;

Where - clause used to check conditions on normal columns of tables.

used with select , update ,delete stmts.

Having - clause used to check conditions on aggregate values retrieved from tables.

used with select stmt only.

Order by -- Clause

Used to arrange the select stmt output in Ascending or Descending order.

Supports with all types of data. It is only for display purpose .

It must be the Last clause in Select.

Asc -- Ascending order [default order]

Desc -- Descending order

Examples:

```
select empno, ename, job, sal from emp order by ename;

select empno, ename, job, sal from emp order by sal ;

select empno, ename, job, sal from emp order by sal desc;

select empno, ename, job, hiredate from emp order by 4 desc;

select empno, ename, job, sal basic, sal * .25 da,
sal * .35 hra, sal * .15 pf, round( sal + sal * .25 + sal * .35 - sal * .15) gross
from emp order by 8 desc ;

select deptno , job, ename, sal from emp order by deptno , job desc ;

select sname, qualification, score from student order by qualification , score desc;
```

student -- sname ,qualification, score

select ename, sal, job from emp order by ename , sal ;

select job, count(*) ecount , sum(sal) totpay, avg(sal) avgpay , min(sal) lopay , max(sal) hipay from emp where deptno = 10 group by job having count(*) > 1 or avg(sal) > 20000 order by 2 desc;

Emp -- where -- group by -- aggr -- having -- ordby

1 2 3 4 5

*** Tab / Cat -- system Tables**

Gives list of tables available in user Login

desc tab

select * from tab;

select * from cat;

desc cat

System Tables are for Read only operations(Select) . DML are not allowed.

Joins:

Used to retrieve the data from more than 1 table in a single query.

supports to query multiple tables.

5 types [7.x]

1. Equi Join
2. Cartesian Join
3. Non Equi Join
4. Outer Join (+)
5. Self Join

8 Types [9i]

1. Natural join 2. Join with using

3. Join with On 4. Inner Join

5. Right outer Join 6. Left outer Join

7. Full Outer Join 8. Cross Join

Equi Join: (one - many or many - one)

Used to retrieve the data from more than 1 table

based on "equality" condition. Tables must have common column between them

to apply this join. If N tables are joined N - 1 conditions are applied.

Ex: 1

* List the employ details along with their deparment details .

```
select empno, ename, job, sal, emp.deptno, dname, loc from dept , emp
where emp.deptno = dept.deptno order by 5;
```

```
Select e.*, d.* from emp e , dept d
where e.deptno = d.deptno order by d.deptno;
```

Using Table Alias names

```
select e.empno, e.ename, e.job, e.sal, e.deptno, d.dname, d.loc from dept d , emp e
where e.deptno = d.deptno order by e.deptno;
```

e,d --- Table Alias Names

* They improve the performance while retrieving or manipulating data.

Ex: 2

* List the student details along with compleate course information and fee dues.

```
select s.roll, s.sname, s.phone, s.cid, c.cname,
c.room, c.timing, c.faculty, c.fee - s.fee_paid "Due"
from student s , course c where s.cid = c.cid;
```

s,c -- Table alias names

```
select m.accno,m.name,m.atype ,m.bal,t.ttype,
t.amt from bankmaster m , transaction t
where m.accno = t.accno and m.name = t.name;
```

Cartesian Join :

Used to retrieve the data from multiple tables with out any condition.

* Used to retrieve data analysis reports.

No need to have common column between tables to apply this join.

It will join every row in Table1 with every row in Table2.

Ex:

```
select empno, ename, sal, job, emp.deptno, dname, loc from emp ,dept ;
select s.roll, s.sname, s.phone, s.fee_paid, s.cid, c.cname , c.timing ,c.faculty , c.room
from student s , course c;
select fname, cname from faculty, course;
```

Non equi Join:

Used to retrieve the data from multiple tables based on any other condition but not equal to

* List the employees along with their Grade

```
select empno, ename, sal, job, deptno, grade
from emp, salgrade where sal between losal and hisal;
```

Ex: 3 **Equi Join with 3 tables**

create table course(cid char(6),ctitle varchar2(20), fee number(5), start_date date, end_date date, timings varchar2(10), faculty varchar2(20));

insert into course values

('ora10', 'oracle 11g', 3000 , '21-Feb-15', '30-Apr-15', '10 am', 'Sridhar');

create table student(roll number(3), sname varchar2(20), cid char(6),

phone number(10), email varchar2(50));

insert into student values(101, 'ram', 'ora10' ,9987667856, 'ram@gmail.com');

* course.cid = student.cid

create table fee_instalments

(roll number(3), amt number(5), dop date);

insert into fee_instalments values

(101, 1000 , '23-feb-15');

insert into fee_instalments values

(101 ,1000 , '10-mar-15');

insert into fee_instalments values

(101 ,500 , '20-mar-15');

* student.roll = fee_instalments.roll

Course ----> Student ----> Fee_instalments

cid roll

** Write a query to display student details along with complete course details and total fee paid, fee dues.*

Select s.roll, s.sname, s.cid, c.ctitle, c.start_date, c.timings, c.faculty, c.fee, sum(f.amt) "fee paid", c.fee - sum(f.amt) due from student s , course c , fee_instalments f where c.ctitle = 'oracle 11g' and s.cid = c.cid and s.roll = f.rollgroup by s.roll, s.sname, s.cid, c.ctitle, c.start_date, c.timings, c.faculty, c.fee order by s.cid;

s,c,f -- table alias names

Outer Join: (+)

Used to retrieve all the rows from Table1 even if

join condition is not matching but only matching rows from Table2 .

Table1 -- gives all rows

Table2 -- gives only matching rows

2 Types : **1. Left outer join**

2. Right outer join

** Retrieve ALL the employ details along with their department details . (Left outer join)*

```
select empno, ename, sal, emp.deptno, dname, loc from emp ,dept
where emp.deptno = dept.deptno order by emp.deptno;
```

** Retrieve ALL the dept details along with employees working in it. (Right outer join)*

```
select empno, ename, sal, dept.deptno, dname, loc from emp , dept
where emp.deptno(+) = dept.deptno  order by dept.deptno;
```

**** In every outer join equi join is in_built**

Outer join = Equi join + Extra rows from 1 Table

** List all the courses with or with out students*

```
select c.cid, c.cname, c.room, c.start_date, c.timing, s.roll, s.sname, s.fee_paid, s.phone
from course c , student s where c.cid = s.cid(+);
```

** List all the students with or with out course details.*

```
select s.cid, c.cname, c.room, c.start_date, c.timing, s.roll, s.sname, s.fee_paid, s.phone
from course c , student s where c.cid(+) = s.cid;
```

5. Self Join

Joining the table to itself.

Table must have similar column repeated in itself to apply this join .

select empno, ename, mgr "BOSS ID" from emp;

mgr -- manager no

* *List the employee names along with their superior names.*

select worker.ename "subordinate" ,

manager.ename "superior"

from emp worker , emp manager

where worker.mgr = manager.empno;

* worker , manager are Table alias names of EMP

Set Operators :

Used to join the outputs of select statements based on the operator specified.

Select statements must have equal no.of columns and similar data type columns.

Maximum 32 queries can be joined with set operators.

4 operators

1. **union all** --- o/p of Q1 + o/p of Q2
2. **union** --- o/p of Q1 + o/p of Q2 - duplicate rows
3. **minus** --- o/p of Q1 - o/p of Q2 (Rows unique to the Q1 will be retrieved)
4. **Intersect** --- o/p common to Q1 & Q2

Examples:

select distinct job from emp where deptno = 10

union all

select distinct job from emp where deptno = 20;

```
select distinct job from emp where deptno = 10
union
select distinct job from emp where deptno = 20;

select distinct job from emp where deptno = 10
minus
select distinct job from emp where deptno = 20;

select distinct job from emp where deptno = 20
minus
select distinct job from emp where deptno = 10;

select distinct job from emp where deptno = 10
intersect
select distinct job from emp where deptno = 20;

select distinct job from emp where deptno = 10
union
select distinct job from emp where deptno = 20
union
select distinct job from emp where deptno = 30;

select distinct job from emp where deptno = 10
union
select distinct job from emp where deptno in (20,30);
```

Ex :2

Student - Table

name	**course**		
a	oracle	a	unix
b	oracle	p	unix
c	oracle	x	oracle
p	java	a	java
q	java	x	java
r	java	p	oracle
x	unix	b	java
y	unix	q	oracle
z	unix	y	oracle

** list the students who joined for all the courses.*

```
select name from student where course = 'oracle'
intersect
select name from student where course = 'java'
intersect
select name from student where course = 'unix' ;
```

o/p: a p x

** list the students who joined for any course (Atleast one course)*

```
select name from student where course = 'oracle'
union
select name from student where course = 'JAVA'
union
select name from student where course = 'unix' ;
```

o/p: a b c p q r x y z

* list the students who joined only for 'oracle'

select name from student where course = 'oracle'

minus

select name from student where course <> 'oracle'; o/p: c

** list the students who joined for 'oracle' & 'java'*

select name from student where course = 'oracle'

intersect

select name from student where course = 'java' ;

o/p: a b p q x

** list the students who joined for 'oracle' or 'java'*

select name from student where course = 'oracle'

union

select name from student where course = 'java' ;

o/p: a b c p q r x y

select name from student where course in ('oracle','java');

** List all the employees & all the dept details .*

Left outer join UNION Right outer join = full outer join

select empno, ename, sal, job, dept.deptno, dname, loc from emp ,dept

where emp.deptno(+) = dept.deptno

union

select empno, ename, sal, job, emp.deptno, dname, loc from emp ,dept

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

order by 5;
```

Control Break Report :

```
select deptno,ename,sal from emp order by 1;

select job,ename,sal from emp order by 1;


break on deptno

compute sum of sal on deptno

select deptno,ename,sal from emp order by 1;


break on job

compute sum of sal on job

select job,ename,sal from emp order by 1;
```

clear breaks --- *removes break option*

```
break on course

compute sum of fee on course

select course,sname,fee from student order by 1;
```

Sub Queries / Nested Queries :

* Query with in a Query

* Select statement provided in a conditional clause (Where / Having) is known as sub query . select stmt provided in place of condition value.

* In a sub query first Inner query will be executed and based on the output of Inner query Outer query will be executed.

Inner Query ----> Outer Query

** List the employees who belongs to RAM's dept .*

Q1:

select deptno from emp where ename = 'RAM';

10

Q2:

select * from emp where deptno = 10;

Using Subquery:

select * from emp where deptno = (select deptno from emp where ename = 'RAM');

Ex: 2

RAM 10

RAM 20

RAM 30

select * from emp where deptno IN (select deptno from emp where ename = 'RAM');

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

** List the employees who belongs to SITA's Husband dept .*

Couple -- Table

Wife	**Husband**
SITA	RAM
UMA	MAHESH

select * from emp where deptno = (select deptno from emp

where ename = (select husband from couple where wife = 'SITA'));

Advantage: They improve performance while retrieving or manipulating data.

Maximum 32(7.x) / 256(8.0) queries can be Nested .

** List the employees who got highest pay.*

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

** List the employees whose pay is MORE than Average pay.*

```
select empno,ename,job,sal from emp where sal >
( select avg(sal) from emp ) order by sal;
```

** List the employees whose pay is LESS than Average pay.*

```
select empno,ename,job,sal from emp where sal <
( select avg(sal) from emp ) order by sal;
```

** Raise the pay by 20% for employees whose pay is LESS than Average pay.*

```
Update emp set sal = sal + round(sal *.2)
 where sal < (select avg(sal) from emp);
```

** List the employees who r having sub-ordinates*

```
  select * from emp where empno in
  ( select distinct mgr from emp ); -- 101  102
```

* EMP

empno	ename	mgr
101	A	
102	B	101
103	C	102
104	D	101
105	E	102

** List the employees who r NOT having sub-ordinates.*

```
select * from emp where empno not in
(select distinct mgr from emp);  --  no rows
select * from emp where empno not in
(select distinct mgr from emp where
mgr is not null);  --  103 104 105
```

*** Distinct - clause picks the null values**

*** Not in operator will not support with null value**

```
 select distinct mgr from emp;  -- 3 rows
 null  101  102
select distinct mgr from emp where mgr is not null;
101  102   -- 2 rows
```

** List the employees who got increments*

```
select * from emp where empno in
( select distinct empno from incr );
```

** List the employees who didnt got increments*

```
select * from emp where empno not in
( select distinct empno from incr );
```

*** incr**

empno	***amt***
101	1000
103	3000
105	4000
103	3000
105	5000

create table incr (empno number(4), amt number(10));

110 3000

101 2000

102 3000

108 6000

* emp.empno = incr.empno

* *Remove the employees who didnt got increments.*

```
delete from emp where empno not in
( select distinct empno from incr );
```

* *Double the comm if they got increment*

```
update emp set comm = comm * 2
where empno in
( select distinct empno from incr );
```

* *List the employees who r working in New york city*

```
select empno, ename, sal, job,deptno from emp
where deptno = (select deptno from dept
where loc = 'NEW YORK');

select empno, ename, sal, job,deptno from emp
where deptno IN (select deptno from dept
where loc = '&city')  order by 5;
```

* *List the employee details along with dept details and no.of increments, total increment amount they recieved .*

```
select incr.empno, ename, sal, job, emp.deptno,
dname, loc , count(incr.empno) icount, sum(amt) total
from incr ,emp , dept where incr.empno = emp.empno and
emp.deptno = dept.deptno group by incr.empno, ename, sal, job, emp.deptno,
dname, loc ;
 101 ram   ..... 10 ,,,,, ,,,,,,    3  5000
```

** employees who got more than 1 increment or total increment amount is more than 3000.*

```
 [ same as above query ]
having count(incr.empno) > 1 or sum(amt) > 3000;
```

** employees who got maximum increments*

```
[ same as above query ]
having count(incr.empno) =  (select max(count(*)) from incr group by empno);
```

Special operators in sub queries :

1. **Exists** : Boolean operator

   It gives the status of inner query.

   If inner query is sucess it returns True otherwise it returns False .

2. **Any or Some** : It takes the smallest value from result of inner query.

3. **All** : It takes the Largest value from result of inner query.

```
select empno,ename,sal from emp where sal > ANY
( 30000,26000,18000,45000 ) order by 3;
select empno,ename,sal from emp where sal > ALL
( 30000,26000,18000,45000 ) order by 3;
```

* *List the employees whose salary is more than 10 th dept "Lowest" pay.*

```
select empno,ename,sal from emp where sal >
Any ( select sal from emp where deptno = 10 ) order by sal;
select * from emp where sal > ( select min(sal) from emp where deptno = 10 );
```

* *List the employees whose salary is more than 10 th dept "Highest" pay.*

```
select empno,ename,sal from emp where sal > all (select sal from emp
where deptno = 10) order by sal;

select empno,ename,sal from emp where sal > ( select max(sal) from emp
where deptno = 10 ) order by sal
```

Any -- min(sal) All -- max(sal)

* ALL will not support Null values
* < is not supported with any & all.

* *List the employees of dept 30 if there are more than or = 3 Analysts in same dept.*

```
select empno, ename, sal, job, deptno from emp where deptno = 30
and Exists (select count(*) from emp where deptno = 30 and job = 'ANALYST'
having count(*) >= 3);
```

** Fill the start_date based on No.of enquiries .*

(if enquiries are above 50 we start batch in 3 days)

```
update course set start_date = sysdate + 3
where cid = 'ora600' and Exists ( select count(*)   from enquiry where cid = 'ora600'
having count(*) >= 50 );
```

*** Course**

cid -- ora600

cname -- Oracle 11g

faculty -- Sridhar

room -- 414

start_date -- null

timing -- 6:00 pm

fee -- 2500

*** Enquiry**

eid -- 101

name -- srinivas

cid -- ora600

phone --

email --

** Remove the courses having below 3 enquiries and fee below 3000. (J10 - Java)*

```
delete from course
where cid = 'J10' and Exists ( select count(*) from enquiry where cid = 'J10'
having count(*) < 3 ) and fee <= 3000;
```

*** Correlated Sub Query** :

In a correlated subquery first outer query will be executed and based on the output of outer query inner query will be executed.

Outer --> Inner --> Outer -- correlated sub query

Inner --> Outer -- sub query

If outer query returns N rows inner query will be executed for N times.

A column from outer query will be substituted in the inner query condition using a Table alias name is known as correlated column.

There will be a continuous relationship between outer and inner query like a LOOP .

When ever the condition value has to be provided dynamically then correlated subquery can be used. It is an time consuming process.

* *List the employees whose salary is "More" than their dept average salary*.

```
select empno, ename, sal, job, deptno from emp E
where sal  > ( select avg(sal) from emp  where deptno = E.deptno )  order by deptno;

select deptno, round(avg(sal)) avgpay from emp group by deptno;
```

* *List the employees whose salary is "Less" than their dept average salary.*

```
select empno, ename, sal, job, deptno from emp E
where sal < ( select avg(sal) from emp
where deptno = E.deptno )  order by deptno;
```

* *Raise the salary by 25% if salary is Less than their dept average salary.*

```
Update emp E set sal = sal + round(sal * .25)
where sal < ( select avg(sal) from emp where  deptno = E.deptno );
```

* *List the employees whose salary is Less than their Job average salary.*

```
select empno, ename, sal, job, deptno from emp E
where sal < ( select avg(sal) from emp where job = E.job) order by job;

select job, round(avg(sal)) avgpay from emp group by job;
```

* *List the employees who got more than 1 increment*

```
select empno, ename, sal, job from emp E  where 1 < ( select count(*) from incr
 where empno = E.empno);
```

E -- Table alias name

empno -- It is an correlated column

** list the employees whose salary is more than his Boss salary.*

```
select empno, ename, sal, mgr from emp E  where sal > ( select sal from emp
 where empno = E.mgr ); -- 102
```

empno	**sal**	**mgr**
102	22000	101
101	21000	103
103	44000	

Scalar Query : [8.0]

Select statement provided in place of column name is known as Scalar query.

It is an Independent Query.

Advantage : Used to retrieve Data Analysis reports

** List the department details along with no.of Employees and Total Salary .*

```
select deptno, dname, loc ,
( select count(*) from emp where deptno =  d.deptno ) ecount,
( select sum(sal) from emp where deptno =  d.deptno ) totsal from dept d order by 1;
```

o/p: 10 ACCOUNTING NEW YORK 14 388525

```
select deptno, dname, loc ,
( select count(*) from emp where deptno =  d.deptno ) ecount ,
( select sum(sal) from emp where deptno =  d.deptno ) totsal from dept d
where ( select count(*) from emp where deptno =  d.deptno ) >= 1 order by 1;
```

Using 2 Queries :

```
select job, ename, sal from emp order by job;
select job ,min(sal) lopay , max(sal) hipay from   emp group by job order by job ;
```

Using Scalar Query :

```
select job, ename, sal,
( select min(sal) from emp where job = e.job )  lopay ,
( select max(sal) from emp where job = e.job )  hipay
from emp e order by job ;
```

```
Select deptno, ename, sal, ( select min(sal) from emp
where deptno = e.deptno) lopay ,
( select max(sal) from emp where deptno = e.deptno ) hipay
from emp e order by deptno;
```

*** Retrieve Mgr no and Name**

```
select distinct Mgr mgr_id ,(select ename from    emp where empno = E.mgr) name from
emp E;
```

*** employees with their manager names**

```
select empno, ename, sal, mgr,
(select ename from emp where empno = E.mgr)   "Manager" from emp E order by 4;
```

SELECT :

SELECT - clauses

(where , group by, having, order by)

Joins

Set operators

Sub Queries

Co-related sub query

Scalar query , Inline View

Arithematic Expressions

Built-in functions

*** Grouping Sets: (8.0)**

Applies Group by clause on multiple columns at a time. Columns are considered independently.

```
Select hiredate, mgr, job, count(*) from emp
group by grouping sets(hiredate,mgr,job);
```

```
select hiredate ,count(*) from emp group by hiredate;
select mgr ,count(*) from emp group by mgr;
select job ,count(*) from emp group by job;
```

```
Select deptno,job, count(*) ,sum(sal),avg(sal)
from emp group by grouping sets(deptno,job);
```

```
select deptno, count(*) ,sum(sal),avg(sal)
from emp group by deptno;
```

```
select job, count(*) ,sum(sal),avg(sal)
from emp group by job;
```

*** Matrix Query**

** Job & Dept wise Total Salary*

```
Select job, sum(decode(deptno,10,sal)) dept10,
sum(decode(deptno,20,sal)) dept20,
sum(decode(deptno,30,sal)) dept30,
sum(decode(deptno,40,sal)) dept40 from emp group by job;
```

** Job & Dept wise Employ count*

```
Select job, count(decode(deptno,10,empno)) dept10,
count(decode(deptno,20,empno)) dept20,
count(decode(deptno,30,empno)) dept30,
count(decode(deptno,40,empno)) dept40 from emp  group by job;
```

*** Data Integrity Rules { Constraints }:**

A set of pre-defined rules applied on table columns while creating Tables or after creation.

They are automatically activated when ever "DML" operations are performed on tables.

They are used to impose restrictions on Table Columns.

They are also activated when Tables are manipulated by other users or by other Application s/w Tools. They provide High security on Tables.

3 Types

1. Entity Integrity Rules

Used to restrict duplicate values into table columns.

ex: Unique , Primary Key

2. Domain Integrity Rules

Used to provide conditional restrictions on Table columns.

ex: Check , Not null , default

3. Referential Integrity Rule

Used to establish relationship between 2 tables.

ex: References (Foreign Key)

Oracle Constraints :

*** Not null :** Used to restrict null values . It is a Column Level Constraint .

*** Unique** : Used to restrict duplicate values but any no.of null values are allowed.

(2 null values are not equal)

*** Check :** Used to provide conditional restrictions on table columns.

*** Default :** Used to define initail value for a column (Column Level property).

If column is not assigned with a value then default value will be accepted.

*** Primary Key : Not Null + Unique + Index**

Used to define the Key column of a table.

It can be used only once in Table definition.

It will not allow Null values and Duplicate values into Key column.

It is supported with an Index automatically.

Index :

It is a pointer locates the physical address of data.

It will improve performance of oracle while Retrieving or Manipulating data using Key column. It is automatically activated whenever key column is used in "Where" clause.

*** References (Foreign Key) :**

Used to define relationship between 2 Tables. It allows Null and duplicate values .

It can be related to either Primary key or unique constraint column of other Table.

PK / UNQ <-----> FK

dept (parent)	**emp (child)**
deptno (pk)	deptno (fk)
	10 101
	10 102
10	10 103
	10 104
	10 105

Dept -- Master(Parent) // EMP -- Detail(Child)

dept.deptno = emp.deptno

* one - many or many - one Relation established between PK and FK

Note:

Constraints are defined in 2 Methods:

1. Column Constraint Syntax

2. Table Constraint Syntax

1. Column Constraint Syntax

Constraints are defined at the end of column definition.

All constraints are supported.

Used to define constraints only while creating Tables not for existing Tables.

Using Column constraint Syntax:

```
create table dept
(deptno number(2) primary key,
 dname varchar2(20) not null unique ,
 loc varchar2(20) default 'Hyderabad');
```

Activating default :

Default : keyword (8.0)

used to replace with default value defined.

```
insert into dept values(40,'TRAINING',default);

insert into dept(deptno,dname) values(50,'TESTING');  -- skip column

insert into dept values(60,'RESEARCH',null);

select * from dept;

40  TRAINING   Hyderabad

50  TESTING     Hyderabad

60  RESEARCH

select * from dept where deptno = 10;

update dept set loc = 'VIZAG' where deptno = 30;
```

-- Index is activated automatically while Retrieving or Manipulating data thru key column in WHERE clause.

```
Create table emp

 (empno number(4) primary key,

  ename varchar2(20) not null,

  sex char(1) check (sex in ('M','F')),

  sal number(12,2) check (sal >= 5000),

  hiredate date default sysdate,

  mail_id varchar2(100) unique,

 deptno number(2) references dept on delete cascade);
```

* *emp.deptno = dept.deptno -- Join condition*

dept - Master Table (parent) -- Independent Table

emp - Detail Table (child) -- Dependent Table

insert into emp values(1001,'RAM','M',20000,'10-JUl-15','ram@gmail.com',10);

update emp set sal = 4000 where empno = 1001;

* dept - deptno (10,20,30,40)

insert into emp values(101,.....,10);

insert into emp values(102,.....,20);

insert into emp values(103,.....,30);

insert into emp values(104,.....,10);

insert into emp values(105,.....,10);

insert into emp values(108,.....,null);

insert into emp values(110,.....,90); -- error

err: Parent key not found (-2291)

delete from dept where deptno = 10; -- error

err: Depending child rows exists - cannot remove parent (-2292)

Soln:

delete from emp where deptno = 10; -- remove child

delete from dept where deptno = 10; -- remove parent

create table incr(empno number(4) not null

references emp on delete cascade, amt number(10,2) not null);

incr.empno = emp.empno -- Join Condition

insert into incr values(999,10000); -- error

err: Parent key not found

Oracle Constraint Error Numbers:

Not null ---> -1400

Unique ---> -1

Check ---> -2290

References ---> -2291 -- Parent key not found

-2292 -- Depending child rows exists

insert		**delete**
1	Dept	3
	|	
2	Emp	2
	|	
3	Incr	1

If " On delete cascade " is not defined :

3. delete from dept where deptno = 10;

err - depending child rows exists

2. delete from emp where deptno = 10;

err - depending child rows exists

1. delete from incr where empno in (select empno from emp where deptno = 10);

On delete cascade - Clause

Automatically removes the child rows whenever parent record is removed.

It has to be specified with every child table along with references constraint.

It cannot be assigned seperately. It is activated by " Delete " stmt on parent table.

If On delete cascade is defined :

>delete from dept where deptno = 10;

-- Automatically removes dept 10 details from Emp, incr tables also.

>delete from emp where empno = 101;

-- removes 101 details from incr table

drop table dept;

error : depending child rows exists

drop table dept cascade constraints;

Cascade constraints - clause

Allows to remove the parent table even if child exists.

It will destroy the relationship between 2 tables.

Child records still exists even if parent table is dropped.

It is used with "drop" stmt on parent table.

On delete cascade	**Cascade constraints**
* removes child rows along with parent	* child rows still exists
* activated by delete	* used with drop

Ex:

create table course (cid char(6) primary key,

ctitle varchar2(20) not null, faculty varchar2(20) not null,

room number(3), start_date date);

ora630 oracle 11g Sridhar 414 26-jan-15

create table student (roll number(3) primary key,

sname varchar2(20) not null, sex char(1) check (sex in ('M','F')),

cid char(6) references course on delete cascade, email varchar2(30) unique);

101 RAM M ora630 ram@gmail.com

course.cid = student.cid --- join condition

create table fee_instalments (roll number(3) not null references student ,

amt number(5) not null , date_paid date default sysdate);

101 1000 28-jan-15

101 500 10-feb-15

101 1000 03-mar-15

student.roll = fee_instalments.roll -- join condition

*** Course --> Student --> Fee_instalments**

** List the Student details along with compleate course details and Total fee paid by student.*

** Give the Course details along with total fee collected from all students of that course.*

ora630 oracle 11g sridhar 120000

Table constraint syntax:

Constraints are defined at the end of table definition.

Supports to define Composite Primary key(CPK) and Composite Foreign key (CFK).

Not null and Default are not allowed. Used to define constraints on existing tables.

Max 32 columns can be defined in CPK or CFK.

Ex: Create table Reservation

(train_no number(4) , coach_id varchar2(5) , seat_no number(3) ,

doj date , pname varchar2(20), age number(3), sex char(1),

to_stn varchar2(20), from_stn varchar2(20), fare number(5),

constraint pk_rail primary key (train_no, coach_id, seat_no, doj));

Ex: create table bankmaster

```
( accno number(4), acc_type char(1),
  name varchar2(20) , curr_bal number(12,2),
  pan_no varchar2(10),
 constraint pk_bank primary key(accno,name),
 constraint chk_atype check(acc_type in('S','C','R')),
 constraint chk_bal check(curr_bal >= 5000),
 constraint unq_pan unique(pan_no));
```

Ex: create table Transaction

```
 ( tran_id number(6) primary key,
 accno number(4) NOT NULL,
 name varchar2(20) NOT NULL,
 tran_type char(1), tran_date date default sysdate,
 amt number(12,2) not null,
 constraint fk_bank foreign key (accno,name)  references bankmaster on delete cascade,
 constraint chk_ttype check (tran_type in ('W','D')),
 constraint chk_amt check (amt >= 100));
```

Join condition :

```
 bankmaster.accno = transaction.accno and
 bankmaster.name = transaction.name
insert into bankmaster values(1001,'S','SRIDHAR',30000,'AOSPK453E');
insert into transaction values(454232,1001,'SRIDHAR','D',sysdate,20000);
delete from bankmaster where accno = 1001;
-- removes child rows from transaction table
```

ATM_Master	**ATM_Trans**
accno	accno
card_no	card_no
pin_no	pin_no
name	amt
bank	tran_date
branch	
validity_date	
amt_limit	

Adding Constraints to existing Tables:

Alter - add constraint :

```
Alter table dept add constraint pk_dept
primary key(deptno);
Alter table dept add constraint unq_dname unique(dname);
Alter table emp add constraint pk_emp primary key(empno);

Alter table emp add constraint chk_sal  check(sal >= 5000);

Alter table emp drop constraint chk_sal ;
Alter table emp add constraint chk_sal  check(sal >= 15000);

Alter table emp add constraint fk_dept  foreign key(deptno)
references dept(deptno) on delete cascade;
```

Note:

* While adding constraints to existing tables with information, existing data must satisfy constraint rule

Adding Not null & Default : (Properties)

Alter - Modify

Alter table emp modify ename varchar2(20) not null;

Alter table emp modify hiredate date default sysdate;

Removing Not null & Default : (Properties)

Alter table emp modify ename varchar2(20) null;

Alter table emp modify hiredate date default null;

Alter table <table name> drop constraint <cons name>;

Alter table emp disable constraint chk_sal;

Alter table emp enable constraint chk_sal;

Alter table emp drop constraint chk_sal;

Scott: (Sharing Constraints)

Grant References on dept to user1;

User1:

Create table employ (empno number(4) constraint pk_emp primary key,

ename varchar2(20), sal number(12,2), mgr number(4) References employ,

deptno number(2) references scott.dept);

insert into employ values(101,.....,null,10);

insert into employ values(102,.....,101,10);

insert into employ values(103,.....,101,10);

Join Conditions :

employ.empno = employ.mgr (Self Join)

user1.employ.deptno = scott.dept.deptno

*** Self Reference Key: (mgr)**

Table Referencing to itself.

Same table acts as a Parent to itself .

Table must have similar column to apply this relation.

create table emp (empno number(4) primary key, ...);

Removing Primary Key:

* Find the constraint name from system table

* remove constraint

> Alter table emp drop constraint *sys_c002345* ;

sys_c002345 -- constraint name

System Tables:

*** User_constraints** --- Holds the compleate details of constraints defined on table columns.

*** User_cons_columns** --- Holds the brief information about the constraints applied on table columns.

desc user_constraints

select * from user_constraints where table_name = 'EMP';

select constraint_name, constraint_type from user_constraints where table_name = 'EMP';

sys_c002345 P (PK)

sys_c002346 C (Chk/NN)

sys_c002347 U (Unq)

sys_c002348 R (Ref)

select constraint_name, constraint_type, SEARCH_CONDITION from user_constraints where constraint_name = 'CHK_SAL';

desc user_cons_columns

select * from user_cons_columns where table_name = 'EMP';

* *Display common column name between 2 tables*

select column_name from user_tab_columns where table_name = 'DEPT'

intersect

select column_name from user_tab_columns where table_name = 'EMP' ;

deptno

Database Objects in Oracle (SQL):

Tables :

Used to store information & allows to manipulate , retrieve & share information.

(user_tables, user_tab_columns ,TAB, CAT , user_constraints, user_cons_columns)

Views & Synonyms :

Used to manipulate , retrieve & share information. They will not hold data.

(user_views , user_synonyms , CAT ,TAB)

Sequences : Used to generate the numbers automatically . (user_sequences)

Index & Cluster : Used to improve performance of oracle while retrieving or manipulating data. (user_indexes , user_clusters)

Views :

It is an stored select statement.

It is an virtual component .

It allows Desc, DML, Select on it.

It will not hold data.

DML on view are reflected in Table and DML on Table are reflected in view .

It is stored permanantly in "User_views" system table.

It can be shared with other users.

* It is used to share "Selected Rows and Columns" with other users.

* It is used for Reporting Purpose .

* It will Improve performance while manipulating or retrieving data thru Views.

Syntax:

create view <view name> as <select stmt>;

Ex: Simple View

create view v1 as select * from emp;

v1

select * from emp

desc v1 = desc emp

select * from v1;

insert into v1 values(.....);

update v1 set sal = sal + 3000 where empno = 7788;

delete from v1 where empno = 7902;

select * from emp;

Sharing View : Scott

grant all on v1 to user1;

user1:

select * from scott.v1;

insert into scott.v1 values (.....);

commit;

Checking for existing Views:

describe user_views

select * from user_views;

select VIEW_NAME ,TEXT from user_views where VIEW_NAME = 'V1';

select * from tab;

select * from cat;

drop table emp; -- view become invalid object but still exists in database.

desc v1 -- error

select * from v1; -- error

Removing View : scott

drop view <view name>;

drop view v1; -- Table is not effected

*** Sharing selected rows and columns thru views :**

Ex: Complex View

create view v10 as select * from emp where deptno = 10 ;

desc v10 = desc emp

select * from v10;

grant all on v10 to user10; **v10**

user10: select * from emp

where deptno = 10

desc scott.v10

select * from scott.v10;

update scott.v10 set sal = sal + 2000 where sal < 10000;

delete from scott.v10 where job = 'CLERK';

insert into scott.v10 values(.....,10);

insert into scott.v10 values(.....,20); -- error

With check option : Clause

Used to check for condition while inserting rows into view.

Where clause in view definition will not activate while inserting rows thru view.

v10 --- user10

v20 --- user20 emp -- Scott

v30 --- user30

v40 --- user40

Ex: Sharing selected columns with rows

Student -- Table

roll, sname, course, fee, phone, mail_id

create view stu_oracle as select roll, sname, course, fee from student

where course = 'ORACLE' with check option;

select * from stu_oracle;

desc stu_oracle

insert into stu_oracle values(101,'RAM','ORACLE',2100);

insert into stu_oracle values(102,'RAVI','JAVA',4000); -- error

grant all on stu_oracle to user1;

user1:

desc scott.stu_oracle

select * from scott.stu_oracle;

View based on view:

create view v11 as select * from emp;

create view v12 as select empno, ename, job, deptno from v11;

desc v12

select * from v12;

update v12 set job = 'EXECUTIVE' where empno = 7900;

Read only views :

*** View based on Arithematic Expressions**

create view pay_info as select empno ecode, sal basic, round(sal * .25) da, round(sal * .35) hra , round(sal * .15) pf, round(sal + sal * .25 + sal * .35 - sal * .15) gross from emp where sal >= 5000;

desc pay_info

select * from pay_info;

grant select on pay_info to user1;

user1: select * from scott.pay_info;

*** View based on Aggregate Functions**

create view dept_analysis as select deptno,

count(*) ecount, sum(sal) totpay, avg(sal) avgpay, min(sal) lopay, max(sal) hipay

from emp group by deptno;

desc dept_analysis

select * from dept_analysis;

Advantage: used for reporting purpose

Improves performance while retrieving data thru these views.

* Views will support Constraints automatically

* Constraints will not be applied on views

create view v101 as select ename, hiredate, job, deptno from emp; -- Read only view

desc v101

select * from v101;

insert into v101 values('ram',sysdate,'CLERK',20); --- error

Emp

empno -- Pk

ename -- nn

sal -- chk

job

hiredate

deptno --- Fk

Ex:

```
create view v1 as select * from emp;

create table temp as select * from emp;

select * from v1;

select * from temp;

* drop table emp;

select * from v1; -- invalid stmt

select * from temp;


create table emp as select * from temp;

select * from v1;  -- retrieve rows from emp
```

Note :

New EMP table must have same structure as Old EMP table . Then view v1 will be active .

```
create view v5 as select * from emp;

select * from v5;

alter table emp add(dob date, email varchar2(30));

desc v5 -- error

select * from v5; -- view retrieve the rows

desc v5     --  view got compiled and active.
```

* alter table emp drop column comm;

select * from v5; -- error (Invalid object)

Note:

* After providing DDL stmts on Table , Views related to Table will become Invalid .

DDL on Table are not reflected in view.

* But without disturbing existing structure if we modify the table(adding new columns) view will be active .

* Alter on Table will not reflect on views . (only DML are reflected)

* View Structure cannot be changed . (Alter not allowed on views)

*** Force - option**

It allows to create a view with out Table.

create FORCE view eview as select * from Etab; -- view created with errors

desc Etab Error : object not exists

create table Etab as select * from emp;

desc eview -- error

select * from eview; -- view will be compiled

desc eview

Adv : Used to Register the component name in DB.

View based on other user Table :

scott:

grant select on student to user1;

user1:

select * from scott.student;

create view sinfo as select * from scott.student;

select * from sinfo;

desc sinfo

insert into sinfo values(.....); -- error

Join Views:

-- View based on multiple tables

create view edept as select empno, ename, sal, job, dept.deptno, dname, loc

from emp, dept where emp.deptno = dept.deptno;

desc edept

select * from edept;

Note:

7.x - Join views are read only views

8.0 - Only 1 table of view can be manipulated thru view i.e. Key preserved table

8i - Both the tables of view are allowed for manipulation thru view using "Instead of Triggers" in pl/sql.

*** 8.0 - Key preserved table**

Table whose key column is not duplicated in " view result " is known as key preserved table

Dept --- deptno (pk)

Emp --- empno (pk)

select * from edept;

empno	*deptno*
7900	10
7788	10
7499	10
7369	20
7844	20
7902	30
7566	30

In "view output" Primary key columns of table must not have duplicate values i.e. Key Preserved Table . Here **Emp** is key preserved table . DML allowed on Emp columns thru view. **Dept** is Non key preserved table . DML not allowed on Dept columns thru view.

update edept set sal = sal + 3000 where empno = 7900;

delete from edept where empno = 7902;

insert into edept(empno, ename, sal, job) values(.....);

update edept set deptno = 30 where empno = 7369; -- error

update edept set dname = 'PHARMA' where deptno = 30; -- error

insert into edept(dept.deptno, dname, loc) values(.....); -- error

* DML not allowed on non key preserved table

Materialized view : [8i]

It is a static view. It holds data in it.

It will not support DML on it. (Read only)

DML on Table will not be reflected in view.

To create it " create materialized view " permission is required.

It is used to maintain Historic data.

It is used for Data analysis & Reporting purpose.

System / Sys : (DBA)

Grant create materialized view to scott;

scott :

create materialized view mv1 as select * from emp;

desc mv1

select * from mv1;

drop table emp; - view still exists and valid

select * from mv1;

drop materialized view mv1;

select * from ALL_MVIEWS;

desc ALL_MVIEWS

* ALL_MVIEWS : It holds the list of materialized views defined by user.

Now : 1 st Year Later : 10 th Year

emp -- 100 rows emp -- 1000 rows

101 RAM 8000 101 RAM 38000

....

Mv1 -- 100 rows Mv1 -- 100 rows

101 RAM 8000 101 RAM 8000

... ...

Synonym :

It is used to hide original name and owner of the Table.

* It provides security by hiding identity of the component.

* Desc , DML and Select are allowed.

* DML on Table are reflected in synonym and DML on synonym are reflected in Table.

* It will not hold data.

* It is stored permanantly in "user_synonyms" system table.

* It can be shared with other users.

2 types

1. Private synonym : created by user

2. Public synonym : created by DBA only

Syntax : private synonym

create synonym <synonym name> for <db object name>;

Ex: create synonym Esyn for emp;

desc esyn

select * from esyn;

insert into esyn values (.....);

update esyn set sal = sal + 3000 where deptno = 30;

delete from esyn where sal > 30000;

select * from emp;

grant select,insert on esyn to user1;

user1 :

desc scott.esyn

select * from scott.esyn;

insert into scott.esyn values (.....);

Scott :

desc user_synonyms

select * from user_synonyms;

select * from tab/cat ;

drop table emp; -- Synonym is invalid

desc esyn -- error

drop synonym esyn;

Public Synonym : (DBA) sys/sysdba

create public synonym stu_info for student;

grant all on stu_info to public;

sys -- student -- stu_info

scott : select * from stu_info;

select * from tab;

user1: select * from stu_info;

insert into stu_info values(.....);

user2: select * from stu_info;

revoke all on stu_info from public;

* *constraints are supported on synonyms like views.*

*** View Vs Synonym :**

Views will support to share selected rows and columns with other users.

Views support to retrieve arthematic expressions , built_in functions and multiple table contents . Used to share complex query results.

Synonym supports to share entire object with other user. Synonym can be applied on views.

Sequence :

Used to generate the numbers automatically.

It is not related to any table.

It is stored permenantly in database in " user_sequences " system table.

It uses 2 pseudo columns

1**. nextval** - It gives the next value generated by the sequence.

2. **currval** - It gives the present value provided by sequence.

create sequence s1 increment by 1;

select s1.nextval from dual; 1

select s1.nextval from dual; 2

select s1.nextval from dual; 3

select s1.currval from dual; 3

insert into student values(s1.nextval,.....); 4

insert into student values(s1.nextval,.....); 5

insert into student values(s1.nextval,.....); 6

insert into emp values(s1.nextval,.....); 7

insert into emp values(s1.nextval,.....); 8

insert into emp values(s1.nextval,.....); 9

create sequence s2 increment by 1

start with 101 maxvalue 999;

insert into student1 values(s2.nextval,.....); 101

insert into student1 values(s2.nextval,.....); 102

……

insert into student1 values(s2.nextval,.....); 999

select s2.nextval from dual; -- error

create sequence s3 increment by 1

start with 101

minvalue 101

cycle

cache 5

maxvalue 200;

insert into stu_info1 values(s3.nextval,.....); 101

insert into stu_info1 values(s3.nextval,.....); 102

insert into stu_info1 values(s3.nextval,.....); 103

.....

insert into stu_info1 values(s3.nextval,.....); 200

insert into stu_info2 values(s3.nextval,.....); 101

insert into stu_info2 values(s3.nextval,.....); 102

insert into stu_info2 values(s3.nextval,.....); 103

.....

insert into stu_info2 values(s3.nextval,.....); 200

* **minvalue** - Indicates the sequence start up value when sequence is reused

* **cycle** - It allows to repeat the sequence after maximum limit .

* **cache** - It allocates the buffer memory to hold sequence of numbers generated for first time. **cache 5** -- allocates 5 blocks of memory .

alter sequence s3 maxvalue 999;

desc user_sequences

*select * from user_sequences*;

drop sequence s1;

create sequence s5 increment by 3;

select s5.nextval from dual; -- 1

select s5.nextval from dual; -- 4

Index :

It is a pointer locates the physical address of data.

It will improve performance of oracle while retrieving or manipulating data from Table.

It is automatically activated when indexed column is used in " Where " clause.

It is stored in "User_indexes" system Table.

Normal Index:

create index idx1 on emp(job);

select * from emp where job = 'CLERK';

update emp set sal = sal + sal * .15 where job = 'CLERK';

select * from emp where lower(job) = 'analyst';

-- idx1 will not support to retrieve data fastly

When Built_in functions applied on column, index applied on it will be deactivated .

Performance reduced.

Composite Index: (max 32 columns allowed)

create index idx2 on student(course,timing);

select * from student where course = 'oracle' and timing = '11.30 am' ;

delete from student where course = 'oracle' and timing = '6.30 pm' ;

Unique Index:

create unique index idx3 on dept(dname);

```
select * from dept where dname = 'SALES';

insert into dept values(50,'SALES','Chennai'); -- error


drop index idx3; -- removes index

desc user_indexes

select * from user_indexes where index_name = 'IDX1';
```

Note:

constraint name will be the index name for primary key column [pk_emp].

```
 select index_name from user_indexes where table_name = 'EMP';
```

** Gives the column name applied with index*

```
select table_name, column_name, index_name from ALL_IND_COLUMNS

 where table_name = 'EMP';
```

desc ALL_IND_COLUMNS

ALL_IND_COLUMNS : system table holds the details of the columns holding index.

Clusters :

* It holds the common column shared by 2 tables.

* It will improve the performance while retrieving or manipulating data from Master - Detail tables.

* It is stored in "user_clusters" system table.

* It has to be defined before creating tables.

* It cannot be applied to existing tables.

1. create cluster c1(deptno number(2));

2. create table dept(deptno number(2) primary key, dname varchar2(20), loc varchar2(20)) cluster c1(deptno);

3. create table emp(empno number(4)

```
    primary key,  .....................

   deptno number(2) references dept)  cluster c1(deptno);
```

4. create index cidx on cluster c1;

-- Perform DML on Dept and Emp Tables.

desc user_clusters

select * from user_clusters;

drop cluster c1;

drop cluster c1 including tables;

*** Dict - Data Dictionary Table**

-- Holds the list of system tables in database.

desc Dict

select * from Dict; - 508 rows

Pseudo columns:

Automatically filled by oracle.

Ex: sysdate, nextval, currval, rowid, rownum, level, sqlcode, sqlerrm, new, old .

Rowid:

It is automatically assigned with every row inserted into table. It is an unique value.

It is stored permanantly in database.

It comprises of Object id, Data file id, Block id &

Record id. It is Re usable.

select rowid,dname,loc from dept;

select rowid,empno,ename,sal from emp;

** *Removing Duplicate records :*

> Delete from emp where rowid not in

(select min(rowid) from emp group by empno);

....AAA 101 RAM 21000

....AAB 101 RAM 21000 ....AAA

....AAC 101 RAM 21000

....AAD 102 HARI 22000

....AAE 102 HARI 22000 ....AAD

....AAF 102 HARI 22000

....AAG 103 SHAM 24000

....AAH 103 SHAM 24000 ....AAG

....AAI 103 SHAM 24000

* *Retrieving Duplicate rows from Table*

select distinct roll, sname, fee from stud_info s where exists (select count(*) from stud_info where roll = s.roll having count(*) > 1);

* *Retrieve the rows which r not duplicated .*

select distinct roll, sname, fee from stud_info s where exists (select count(*) from stud_info where roll = s.roll having count(*) = 1);

* *Removing duplicate rows*

Delete from stud_info where rowid not in

(select min(rowid) from stud_info group by roll);

* *Modify all rows except First row in Table.*

>Update emp set sal = sal + 2500 where rowid not in (select min(rowid) from emp);

* *Retrieving Cumulative Salary:*

select ename, sal, (select sum(sal) from emp where rowid <= e.rowid) "cumsal" from emp e;

Rownum :

It is an dynamic value automatically retrievedalong with Select statement output.

It is an unique value.

It is only for display purpose. It is not stored in database.

```
select rownum, ename, sal from emp;
select rownum, dname, loc from dept;
```

* *Retrieving Top 5 Highly paid Employees*

```
> select rownum, empno, ename, job, sal from
( select rownum, empno, ename, job, sal from  emp order by sal desc )
  where rownum <= 5;
```

* *Retrieve the rows between 10 and 15.*

```
select * from
( select rownum rno, empno, ename, job, sal from       emp )
where rno between 10 and 15   order by rno;
```

* *Retrieving Nth maximum salaried employ details*

(2 max, 5 max) (Top N analysis query)

```
select rownum, empno, ename, job, sal from
( select rownum, empno, ename, job, sal from  emp order by sal desc )
 group by rownum, empno, ename, job, sal having rownum = &N;
```

* *It will not give accurate results with duplicate rows. (same salary repeated)*

Level gives accurate result.

* *Retrieving Alternate rows*

```
select rownum,empno,ename,job,sal from emp
group by rownum, empno, ename, job, sal having mod(rownum,2) = 0; --- EVEN Rows
[ having mod(rownum,2) != 0; ] --- ODD Rows
```

*** Inline view** : Select statement provided in place of table name is known as Inline view.

It will be created dynamically while executing query. It will save resources .

select < column list> from < table / view / syno >;

Level :

It will arrange the select statement output in Inverted tree structure (Hierarichal Tree) and gives the position of row in Tree. (Returns Number)

It contains duplicate values.

** Retrieving the Hierarichy of employees based on their Superior.*

```
Select Level,empno,ename,sal,job,mgr from emp

connect by prior empno = mgr start with mgr is null order by Level;
```

empno	mgr	Level
101	null	1
102	101	2
103	101	2
104	102	3
105	102	3
106	103	3
107	103	3
108	104	4
109	106	4

```
                    101
          102                 103
    104       105       106       107
108                 109
```

** Retrieving Nth maximum salary using Level. (* Duplicates are eliminated)*

```
Select Level, max(Sal) from emp where Level = &N connect by prior sal > sal group by
Level;
```

Locks :

scott :

emp ---> 7900 12000

grant all on emp to user1;

update emp set sal = sal + 3000 where empno = 7900;

user1 :

update scott.emp set sal = sal + 8000 where empno = 7900;

It leads to Deadlock. None of the users can update.

Locks are Used to preserve the rows for manipulation purpose to prevent the other users to access the same data at the same time.

They prevent Dead Locks . They improve Data concurrency .

(Supports to share with multiple users).

2 Types **1. Implicit Locks 2. Explicit Locks**

Implicit Locks : Automatically imposed by oracle whenever " DML" operations are performed by user.

Explicit Locks : They are imposed by user before manipulating data.

2 Types a). Row Level Locks b). Table Level Locks

i**> Row Level Locks :** used to lock the selected rows of table. It is imposed with " For Update " clause in select.

Ex:1

> select * from emp where deptno = 10 for update;

> update emp set sal = sal + 3000 where deptno = 10;

> commit; -- lock released

select * from emp where empno = 101 for update;

ii> **Table Level Locks :** Used to lock entire table.

> Lock table emp in Exclusive mode;

> update emp set sal = sal + sal * .25;

> commit; -- lock released

Note :

DML are NOT allowed by other users when table is locked by user.

Select is allowed by other users. Commit / rollback will release any type of Lock applied.

SQL Assignment I

1.Display the dept information from department table

2.Display the details of all employees

3.Display the name and job for all employees

4.Display name and salary for all employees

5.Display employee number and total salary for each employee

6.Display employee name and annual salary for all employees

7.Display the names of all employees who are working in department number 10

8.Display the names of all employees working as clerks and drawing a salary more than 3000

9.Display employee number and names for employees who earn commission

10.Display names of employees who do not earn any commission

11.Display the names of employees who are working as clerk , salesman or analyst and drawing a salary more than 3000

12.Display the names of employees who are working in the company for the past 5 years

13.Display the list of employees who have joined the company before 30th june 90 or after 31 st dec 90

14.Display current date

15.Display the list of users in your database (using log table)

16.Display the names of all tables from the current user

17.Display the name of the current user

18.Display the names of employees working in department number 10 or 20 or 40 or employees working as clerks , salesman or analyst

19.Display the names of employees whose name starts with alphabet S

20.Display employee name from employees whose name ends with alphabet S

21.Display the names of employees whose names have sencond alphabet A in their names

22.Display the names of employees whose name is exactly five characters in length

23.Display the names of employees who are not working as managers

24.Display the names of employees who are not working as SALESMAN or CLERK or ANALYST

25.Display all rows from emp table. The system should wait after every screen full of information

26.Display the total number of employees working in the company

27.Display the total salary and total commission to all employees

28.Display the maximum salary from emp table

29.Display the minimum salary from emp table

30.Display the average salary from emp table

31.Display the maximum salary being paid to CLERK

32.Display the maximum salary being paid in dept no 20

33.Display the minimum salary being paid to any SALESMAN

34.Display the average salary drawn by managers

35.Display the total salary drawn by analyst working in dept no 40

36.Display the names of employees in order of salary i.e. the name of the employee earning

37.Display the names of employees in descending order of salary

lowest salary shoud appear first

38.Display the details from emp table in order of emp name

39.Display empnno,ename,deptno and sal. Sort the output first based on name and within name by deptno and witdhin deptno by sal;

40) Display the name of employees along with their annual salary(sal*12).

the name of the employee earning highest annual salary should appear first?

41) Display name,salary,Hra,pf,da,TotalSalary for each employee.

The out put should be in the order of total salary ,hra 15% of salary ,DA 10% of salary .pf 5% salary Total Salary

will be (salary+hra+da)-pf?

42) Display Department numbers and total number of employees working in each Department?

43) Display the various jobs and total number of employees working in each job group?

44)Display department numbers and Total Salary for each Department?

45)Display department numbers and Maximum Salary from each Department?

46)Display various jobs and Total Salary for each job?

47)Display each job along with min of salary being paid in each job group?

48) Display the department Number with more than three employees in each department?

49) Display various jobs along with total salary for each of the job where total salary is greater than 40000?

50) Display the various jobs along with total number of employees in each job.The output should contain only those jobs with more than three employees?

51) Display the name of employees who earn Highest Salary?

52) Display the employee Number and name for employee working as clerk and earning highest salary among the clerks?

53) Display the names of salesman who earns a salary more than the Highest Salary of the clerk?

54) Display the names of clerks who earn a salary more than the lowest Salary of any salesman?

55) Display the names of employees who earn a salary more than that of jones or that of salary greater than that of scott?

56) Display the names of employees who earn Highest salary in their respective departments?

57) Display the names of employees who earn Highest salaries in their respective job Groups?

58) Display employee names who are working in Accounting department?

59) Display the employee names who are Working in Chicago?

60) Display the

job groups having Total Salary greater than the maximum salary for Managers?

61) Display the names of employees from department number 10 with salary greater than that of ANY employee working in other departments?

62) Display the names of employees from department number 10 with salary greater than that of ALL employee working in other departments?

63) Display the names of mployees in Upper Case?

64) Display the names of employees in Lower Case?

65) Display the names of employees in Proper case?

Q:66) Find the length of your name using Appropriate Function?

67) Display the length of all the employee names?

68) Display the name of employee Concatinate with Employee Number?

69) Use appropriate function and extract 3 characters starting from 2 characters from the following string 'Oracle' i.e., the out put should be ac?

70) Find the first occurance of character a from the following string Computer Maintenance Corporation?

71) Replace every occurance of alphabet A with B in the string .Alliens (Use Translate function)?

72) Display the information from the employee table . where ever job Manager is found it should be displayed as Boss?

73) Display empno,ename,deptno from tvsemp table. Instead of display department numbers

display the related department name(Use decode function)?

74) Display your Age in Days?

75) Display your Age in Months?

76) Display current date as 15th August Friday Nineteen Nienty Seven?

78) Scott has joined the company on 13th August ninteen ninety?

79) Find the nearest Saturday after Current date?

80) Display the current time?

81) Display the date three months before the Current date?

82) Display the common jobs from department number 10 and 20?

83) Display the jobs found in department 10 and 20 Eliminate duplicate jobs?

84) Display the jobs which are unique to department 10?

85) Display the details of those employees who do not have any person working under him?

86) Display the details of those employees who are in sales department and grade is 3?

87) Display thoes who are not managers?

88) Display those employees whose name contains not less than 4 characters?

89) Display those department whose name start with"S" while location name ends with "K"?

90) Display those employees whose manager name is Jones?

91) Display those employees whose salary is more than 3000 after giving 20% increment?

92) Display all employees with their department names?

93) Display ename who are working in sales department?

94) Display employee name,dept name,salary,and commission for those sal in between 2000

to 5000 while location is Chicago?

95) Display those employees whose salary is greater than his managers salary?

96) Display those employees who are working in the same dept where his manager is work?

97) Display those employees who are not working under any Manager?

98) Display the grade and employees name for the deptno 10 or 30 but grade is not 4 while joined the company before 31-DEC-82?

99) Update the salary of each employee by 10% increment who are not eligible for commission?

100) Delete those employees who joined the company before 31-Dec-82 while their department Location is New York or Chicago?

101) Display employee name ,job,deptname,loc for all who are working as manager?

102) Display those employees whose manager name is jones and also display their manager name?

103) Display name and salary of ford if his salary is equal to hisal of his grade?

104) Display employee name, job, deptname, his manager name ,his grade and make an under department wise?

105) List out all the employee names ,job,salary,grade and deptname for every one in a company except 'CLERK' . Sort on salary display the highest salary?

106) Display employee name,job and his manager .Display also employees who are with out managers?

107) Display Top 5 employee of a Company?

108) Display the names of those employees who are getting the highest salary?

109) Display those employees whose salary is equal to average of maximum and minimum?

110) Select count of employees in each department where count >3?

111) Display dname where atleast three are working and display only deptname?

112) Display name of those managers name whose salary is more than average salary of Company?

113) Display those managers name whose salary is more than average salary salary of his employees?

114) Display employee name, sal,comm and netpay for those employees whose netpay is greater than or equal to any other employee salary of the company?

115) Display those employees whose salary is less than his manager but more than salary of other managers?

116) Display all employees names with total sal of company with each employee name?

117) Find the last 5(least) employees of company?

118) Find out the number of employees whose salary is greater than their managers salary?

119) Display the manager who are not working under president but they are working under any other manager?

120) Delete those department where no employee working?

121) Delete those records from emp table whose deptno not available in dept table?

122) Display those enames whose salary is out of grade available in salgrade table?

123) Display employee name, sal, comm and whose netpay is greater than any othere in the company?

124) Display name of those employees who are going to retire 31-Dec-99 if maximum job period is 30 years?

125) Display those employees whose salary is odd value?

126) Display those employees whose salary contains atleast 3 digits?

127) Display those employees who joined in the company in the month of Dec?

128) Display those employees whose name contains A?

129) Display those employees whose deptno is available in salary?

130) Display those employees whose first 2 characters from hiredate - last 2 characters sal?

131) Display those employees whose 10% of salary is equal to the year joining?

132) Display those employees who are working in sales or research?

133) Display the grade of jones?

134) Display those employees who joined the company before 15th of the month?

135) Display those employees who has joined before 15th of the month?

136) Delete those records where no of employees in particular department is less than 3?

137A) Delete those employee who joined the company 10 years back from today?

137B) Display the deptname the number of characters of which is equal to no of employee

in any other department?

138) Display the deptname where no employee is working?

139) Display those employees who are working as manager?

140) Count the number of employees who are working as managers (Using set operators)?

141) Display the name of the dept those employees who joined the company on the same date?

142) Display those employees whose grade is equal to any number of sal but not equal to first number of sal?

143) Count the no of employees working as manager using set operation?

144) Display the name of employees who joined the company on the same date?

145) Display the manager who is having maximum number of employees working under him?

146) List out the employee name and salary increased by 15% and express as whole

number of Dollars?

147) Produce the output of the emptable "EMPLOYEE_AND JOB" for ename and job ?

148) List of employees with hiredate in the format of 'June 4 1988'?

149) print list of employees displaying 'Just salary' if more than 1500 if exactly 1500

display 'on taget' if less than 1500 display below 1500?

151) Given a string of the format 'nn/nn' . Verify that the first and last 2 characters are numbers .And that the middle character is '/' Print the expressions 'Yes' IF valid

'NO' of not valid . Use the following values to test your solution'12/54',01/1a,'99/98'?

152) Employes hire on OR Before 15th of any month are paid on the last friday of that month

those hired after 15th are paid the last friday of th following month .print a list of employees .their hiredate and first pay date sort those who se salary contains first

digit of their deptno?

153) Display those managers who are getting less than his employees salary?

154) Print the details of employees who are subordinates to BLAKE?

155.Display those who working as manager using co related sub query

156.Display those employees whose manager name is JONES and also with his manager name

157.Define variable representing the expressions used to calculate on employees total annual renumaration

158.Use the variable in a statement which finds all employees who can earn 30000 a year or more

159.Find out how many managers are there with out listing them

160.Find out the avg sal and avg total remuneration for each job type remember

salesman earn commission define emp_ann_sal=(sal+nvl(comm,0))*.12

161.Check whether all employees number are indeed unique

162.List out the lowest paid employees working for each manager, exclude any groups where minsal is less than

1000 sort the output by sal

163.List ename, job, annual sal, deptno, dname and grade who earn 30000 per year and who are not clerks

164.Find out the job that was falled in the first half of 1983 and the same job that was filled during the same period on 1984

165.Find out the all employees who joined the company before their manager

166.List out the all employees by name and number along with their manager's name and number also display 'NO MANAGER' who has no manager

167.Find out the employees who earned the highest sal in each job typed sort in descending sal order

168.Find out the employees who earned the min sal for their job in ascending order

169.Find out the most recently hired employees in each dept order by hire date

170.Display ename, sal and deptno for each employee who earn a sal greater than the avg of their department order by deptno

171.Display the department where there are no employees

172.Display the dept no with highest annual remuneration bill as compensation

173.In which year did most people join the company. Display the year and number of employees

174.Display avg sal figure for the dept

175.Write a query of display against the row of the most recently hierd employee.display ename hire date and column max date showing

176.Display employees who can earn more than lowest sal in dept no 30.

177.Find employees who can earn more than every employees in dept no 30. select dept name and deptno and sum of sal break on deptno on dname;

178.Find out avg sal and avg total remainders for each job type

179.Find all dept's which have more than 3 employees

/*If the pay day is next Friday after 15th and 30th of every month. What is the next pay day from their hire date for employee in emp table /*

/*If an employee is taken by you today in your organization and is a policy in your company to have a review after 9 months the joined date (and of 1st of next month after 9 months) how many days from today your employee has to wait for a review*/

180.Display the 10 th record of emp table (without using rowid)

181.Display the half of the enames in upper case and remaining lower case

182.Display the 10th record of emp table without using group by and rowid

183.Delete the 10th record of emp table

184.Create a copy of emp table

185.select ename if ename exists more than once

186.Display all enames in reverse order

187.Display those employee whose joining of month and grade is equal

188.Display those employee whose joining date is available in deptno

189.Display those employee name as follows A ALLEN, B BLAKE

190.List out the employees ename,sal,pf from emp

/*191.Display RSPS from emp without using updating, inserting */

192.Create table emp with only one column empno

193.Add this column to emp table ename varchar2(20)

194.OOPSI i forget to give the primary key constraint. Add it now

195.Now increase the length of ename column to 30 characters

196. Add salary column to emp table?

197.I want to give a validation saying that sal can not be greater 10000(note give a name to this column)

198.For the time being i have decided that i will not impose this validation. My boss has agreed to pay

more than 10000

199.My boss has changed his mind. Now he doesn't want to pay more than 10000 So revoke that salary constraint

200.Add column called as mgr to your emp table

201.Oh! This column should be related to empno, Give a command tdo add this constraint

202.Add dept no column to your emp table

203.This deptno column should be related to deptno column of dept table

204.Create table called as new emp. Using single command create this table as well as to get data into

this table (use create table as)

205.Create table called as newemp. This table should contain only empno, ename, dname ?

206.Delete the rows of employees who are working in the company for more than 2 years

207.Provides a commission to employees who are not earning any commission

208.If any employee has commission his commission should be incremented by 100% of his salary

209.Display employee name and department name for each employee

210.Display employee number,name and location of the department in which he is working

211.Display ename,dname even if there no employees working in a particular department(use outer join)

212.Display employee name and his manager name.

213.Display the department name along with total salary in each department

214.Display the department name and total number of employees in each department

215. Select child tables for a given parent table?

SQL Assignment II

1. See table structure of Designation_Master, Department_Master, Student_Master, Student_Marks, Staff_Master, Book_Master and Book_Transaction table.
2. Query Student_Code, Student_Name and Department_Code of every Student. Retrieve detail of every Student and Staff.
3. Display Student_Code, Subjects and Total_Marks for every student. Total_Marks will calculate as Subject1 + Subject2 + Subject3 from Student_Marks.
4. . Display Student_Code, Student_Year, Subjects and Percentage of all student and heading for the columns would be Student#, Join_Year, Subject1, Subject2, Subject3, Total, and Percentage from Student_Marks.
5. Create a query to display unique Department_Code from Student_Master table.
6. . Create a query which will display all students' data those who are passing (getting pass Subject1 + Subject2 + Subject3 > 180. And save the query
7. . Create a query which displays Student_Code, Join_Year and Subject 1 of all students who scored more than 75 in Subject1. And save the query
8. Modify the query to display student detail of those who getting marks in Subject1 between 50 and 75. And save the query .

9. Write a query which will display Student_Name, Departmnet_Code and DOB of all students who born between January 1, 1981 and March 31, 1983. Order the query in ascending order of DOB.

10. . Write a query which will display Student_Name, Department_Code of all students' studies in departments 10 and 30, in alphabetical order by name.

11. Select Student_Name and Department_Code of all students those who have born in 1982.

12. . Display Book_Details for all books which have not being returned.

13. . Display Book_Details for all books which have not being returned and expected returned was last Monday.

14. . Select all students who have completed 25 years of age and their name starts with 'S';

15. . Display Student_Code & Year, of all students who is getting more than 60 in each subject and aggregate of marks is greater then 190 for year 2006 or 2007.

16. . Display Student_Code, Subject1, Subject2, Subject3, Total_Marks, and Percentage as whole number for current year and percentage should be greater that 75.

17. Modify the query to add a column that will find the difference between Total_Marks and Maximum_Marks. Label the column as "Diff_Marks".

18. Write a query which displays Student_Name in upper case, Student_Name first letter capitalized and all other letters lower case and length of the Student_Name for all students whose name starts with 'S' or 'J'.

19. . Create a Query which displays the Student_Name and Department_Code. If student does not belong to any department, put "No Department". Label the column as "Department".

20. . Create a Query which display the Student_Name and Department_Code and Total_Marks and order it on Total_Marks descending and Student_Name ascending.

21. . List the Book details issued to Students for current month and last month.

22. . List all Table Name contains 'MASTER' in their names.

23. . Display the Staff_Name Staff_Salary and the Salary in X. Each X represents a 1000 in Salary.

Sample Output.

KING	10000	XXXXXXXXXX
FORD	12000	XXXXXXXXXXX

24. Display Student_Name and Date of birth where DOB must be displayed in the format similar to "January, 12 1981" for those who born on Saturday or Sunday.
25. Display Staff_Name, Hiredate and review date of all staff. Review date is first Monday after six months of Service. Label the Column REVIEW
26. Display each Staff_Name and number of months he worked for the organization. Label the column as 'MONTHS WORKED'. Order your result by number of months employed. Round the number of months to closest whole number.
27. Write a query that produces the following for each staff :
28. <Staff_Name> earns <Salary> monthly but wants <2 times salary. Label the column as DREAM SALARY. Format the dream salary as Rs. 99,999.00.
29. Create a query which will display Staff_Name, Salary of each staff. Format the salary to be 15 character long and left padded with '$'.
30. Display the Staff_Name, Hiredate and day of the week on which staff was hired. Label the column as DAY. Order the result by the day of the week staring with Monday.
31. Write a query that displays Staff_Name, Salary, and Grade of all staff. Grade depends on the following table.

Salary	Grade
Salary >=50000	A
Salary >= 25000 < 50000	B
Salary>=10000 < 25000	C
OTHERS	D

32. Write a query to find the pay date for the month. Pay date is the last Friday of the month. Display the date in the format "Twenty Eighth of January, 2002". Label the heading as PAY DATE.
33. Display the Highest, Lowest, Total & Average salary of all staff. Label the columns Maximum, Minimum, Total and Average respectively. Round the result to nearest whole number. Save the query as E2Q20.
34. Edit the above query and display the same for each Department Name.
35. Write a query to display number of people in each Department. Output should display Department_Code, Department_Name, and Number of People.

36. Determine the number of managers without listing them. Label the column as 'Total Number of Managers'.
37. Display Manager_Code, Manager_Name and salary of lowest paid staff in that manager's team. Exclude any group where minimum salary is less than 10000. Order you result on descending order of salary.
38. Display the Manager_Name and the total strength of his/her team.
39. Create a query which will give the following output ignore the lines.

Dept_Code	1980	1981	1982	1983	Total
10		2	1		3
20	1	2	1	1	5
30		6			6

40. Write a query which displays Staff_Name, Department_Name and Department_Code Salary for all staff who earns more than 20000.
41. Write a query to display Staff_Name, Department_Code, and Department_Name for all staff who do not works in Department code 10 and have 'A' in their name.
42. Display Staff_Code, Staff_Name Department_Name, and his manager's number and name. Label the columns Staff#, Staff_Name, Mgr#, Manager.
43. Create a query that will display Student_Code, Sudent_Name, Department_Name, Subjec1, Subject2, and Subject3 for all students who are getting more than 60 in each subject from department 10 and 20.
44. Create a query which will display Manager_Name, Staff_Name, Salary, Staff Hiredate, Manager Hiredate of all staff hired before their manager.
45. Create a query that will display Student_Code, Sudent_Name, Department_Name, Book_Code, Book_Name for all students whose expected book return date is today.
46. Create a query that will display Staff_Code, Staff_Name, Department_Name, Designation, Book_Code, Book_Name, Issue_Date. For only those staff who have taken any book in last 30 days.
47. Generate a report which contains the following information.

Staff Code Staff Name Designation Department Name Department Head

For all staff excluding HOD (List should not contain the details of Department head).

48. Generate a report which contains the following information

Student Code Student Name Department Name Total Marks HOD Name

Sort the output on Department Name and Total Marks.

49. Generate a report which contains the following information.

Staff Code, Staff Name, Designation, Department, Book Code, Book Name, Author, Fine

For the staff who have not return the book. Fine will be calculated as Rs. 5 per day.

Fine = 5 * (No. of days = Current Date – Expected return date).

50. List Staff_Code, Staff_Name, and Salary for those who are getting less than the average salary of organization.
51. List the Staff_Code, Staff_Name who are not Manager.
52. Display Author_Name, Book_Name for those authors who wrote more than one book.
53. Display Staff_Code, Staff_Name, Department Name for those who have taken more than one book.
54. Display top ten students for a specified department.

Details are:

Student_Code, Student_Name, Department_Name, Subject1, Subject2, Subject3, Total.

55. Display the Staff_Name, Department_Name, and Salary for those staff who are getting less then average salary in their own department
56. Create a query that will display the Staff_Name, Department_Name, and all the staff that work in the same department as a given staff. Give the column as appropriate label.
57. List the Student_Code, Student_Name for that student who got highest marks in all three subjects in Computer Science department for current year.
58. Display the Student_Code, Student_Name, and Department_Name for that department in which there are maximum number of student are studying.
59. Display Staff_Code, Staff_Name, Department_Name, and Designation for those who have joined most recently.
60.
61. List the total staff and the number of staff hired in each Department

Department No	No. of Staff
10	30
20	40

30	25
Total	95

62. Display the Manager_Name, Staff_Name, Salary (The highest and the lowest paid staff in his team).

Manager Name	Staff_Name	Salary
Allen	Amit	10000
Allen	Raju	20000
Ravi	Arvind	15000
Raj	Ajay	30000

63. Write a query which will display Staff_Name, and Department_Name, Designation of all staff. Also display the department name where there is no staff. Display"NO STAFF" where there is no staff in the department.
64. List the books that are returned on expected date, before expected date and after expected date. Calculate the fine for books returned after due date as Rs. 5 per day. Output should be in following format.

Book Code	**Book Name**	**Author**	**Status**	**Fine**
1000010	Java Complete Reference	Schield	Expected date	
1000020	C++ Complete Reference	Schield	Before date	
100001	Oracle Complete Reference	Scott	After date	15
Total	**15**			

65. Accept a specific book code from user and find the details of the students/staff who have borrowed that book and has not returned the same. The following details should be displayed

Student/Staff Code Student/Staff Name Issue Date Expected Return Date

66. Accept the student code and display only faculty details of that department. Display Details for only (HOD/Professor/Reader/Lecturer).Use Department Master & Designation Master.

Output should be in following format.

Faculty Code Faculty Name Designation Department Name

67. List the department Name which has borrowed maximum number of time (Any book).
68. .List the faculties of the department in which the student obtaining the maximum total marks for current year
69. For current year in which month maximum numbers of books are borrowed by which department?
70. Send a report to every department in the following format:

Dept Name, Total No: of students yet to return the book

71. List the details of the book which has not been borrowed so far.
72. List the student detail, who has borrowed maximum number of books.
73. Generate the report that displays details for all books.

Book Code Book Name Author Number of Times Issues

74. Generate a report which contains the following information.

Book Code Book Name Staff Code Staff Name Designation Department Name Number of Times Issued

For current year only.

75. Add a new staff with Staff_Code: 8888, Name: HARRIS, Mgr_Code: 7566, Hiredate: 08-APR-1985, Sal: 25000, Dept_Code: 20.
76. Change the Miller's department from 10 to 20.
77. Give all the staff in department 30 a salary hike of 10%.
78. Delete all staff from department 20.
79. Create a table Dept_Master with the following columns

    DeptNo – Primary Key

    DeptName -- It should not be null

80. Create a table Staff_Master_Dup which contains the following columns:

    Staff_Code -- Primary key should generate numbers automatically

    Staff_Name – Should not be null

    Designation –Should always be one of them HOD / Reader/ Professor/ Lecturer.

81. Salary – Should be always >7000
82. HireDate –Default should be current date
83. Dept_Code – Should refer to the DEPT_Code in Dept_Master
84. Insert the values
85. Create a view consisting of names, code, designation, salaries and departments of all staff in department 10.
86. From the above view list the average annual salary for all the people in department 10 who makes less than 5000.
87. Create a view FINANCE with data from the Staff_Master which gives their total salary and average salary.
88. Create a view Staff_VU from Staff_Master, Designation_Master & Dept_Master to make it possible to see where some one works.
89. Display the structure of the view Staff_VU.
90. Display the contents of the STAFF_VU.
91. Select view name and text from USER_VIEWS.
92. Create a view name DEPT20 that contains the staff code, staff name and department number for all staff in department 20. Label the view column Staff_Code, Staff_Name and Dept_Code. Do not allow a staff to be reassigned to another department through this view.
93. Display the structure of the DEPT20 view.
94. Assign SMITH to department 30. What happened?
95. Create a view called SALARY_VU based on the Staff_Name, Department_Name, Salary and salary grade for all staff. Label the columns staff, Department, Salary and Grade respectively.
96. Create an index on salary column of Staff_Master table.
97. Confirm the same from data dictionary that the index is created.
98. Create a table called TEST with one column as Primary key.
99. Confirm that an index is created implicitly.
100. Drop the table TEST and check that the corresponding index is also dropped.
101. Create an index on salary of Staff_Master table.
102. Create an index on Department_Code and Designation_Code of Staff_Master table.

Happy Learning !!!