Create Command: Table Walmart

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> show user
USER is "SYS"
SQL> create table eshwar(id number(10),firstname varchar2(20),pay number(10));

Table created.

SQL> desc
Usage: DESCRIBE [schema.]object[@db_link]
SQL> desc eshwar
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER(10)
 FIRSTNAME                                          VARCHAR2(20)
 PAY                                                NUMBER(10)
```

Insert Values into Table and Verify:

```
SQL> insert into eshwar values(1,'eshu',10000);

1 row created.

SQL> insert into eshwar values(2,'mani',11000);

1 row created.

SQL> insert into eshwar values(3,'lalitha',12000);

1 row created.

SQL> insert into eshwar values(4,'candy',13000);

1 row created.
```

Alter Command:

```
SQL> alter table eshwar ADD Gender varchar2(6);

Table altered.

SQL> select * from eshwar;

        ID FIRSTNAME                 PAY GENDER
---------- -------------------- ---------- ------
         1 eshu                      10000
         2 mani                      11000
         3 lalitha                   12000
         4 candy                     13000
```

Update Command:

```
SQL> update eshwar set Gender='male' where id=1;

1 row updated.

SQL> select * from eshwar;

        ID FIRSTNAME                  PAY GENDER
---------- -------------------- ---------- ------
         1 eshu                      10000 male
         2 mani                      11000
         3 lalitha                   12000
         4 candy                     13000

SQL> upda eshwar set Gender='male' where' id=2;
SP2-0734: unknown command beginning "upda eshwa..." - rest of line ignored.
SQL> update eshwar set Gender='male' where id=2;

1 row updated.

SQL> select * from eshwar;

        ID FIRSTNAME                  PAY GENDER
---------- -------------------- ---------- ------
         1 eshu                      10000 male
         2 mani                      11000 male
         3 lalitha                   12000
         4 candy                     13000
```

Truncat Command:

```
SQL> truncate table eshwar;

Table truncated.

SQL> desc eshwar;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER(10)
 FIRSTNAME                                          VARCHAR2(20)
 PAY                                                NUMBER(10)

SQL> select * from eshwar;

no rows selected

SQL> rollback;

Rollback complete.

SQL> select * from eshwar;

no rows selected
```

Delete & Rollback Command:

```
SQL> delete from eshwar;

4 rows deleted.

SQL> rollback;

Rollback complete.

SQL> select * from eshwar;

        ID FIRSTNAME                  PAY GENDER
---------- -------------------- ---------- ------
         1 eshu                      10000
         2 mani                      11000
         3 lalitha                   12000
         4 candy                     13000
```

Drop Table Command:

```
SQL> select * from mobile;

        ID LIST       PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         18000
         4 apple      30000

SQL> drop table mobile;

Table dropped.

SQL> select * from mobile;
select * from mobile
              *
```

Delete & commit command:

```
SQL> select * from mobile;

        ID LIST       PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL> delete from mobile;

4 rows deleted.

SQL> commit;

Commit complete.

SQL> desc mobile;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER(5)
 LIST                                               VARCHAR2(5)
 PRICE                                              NUMBER(5)

SQL>
```

Retrieve Backup/Restore Data from Database:

```
SQL> select * from mobile;

        ID LIST       PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL>  create table phones as select * from mobile;

Table created.

SQL> select * from mobile;

        ID LIST       PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000
```

Alter/ Rename Table:

```
SQL> select * from mobile;

        ID LIST       PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL> alter table mobile rename column list to models;

Table altered.

SQL> select * from mobile;

        ID MODEL      PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL>
```

DQL: DATA QUERY LANGUAGE

Arithmetic Operators: +, -, x./ :

```
SQL> select * from mobile;

        ID MODEL      PRICE
---------- ----- ----------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL> select id,price+150 from mobile;

        ID  PRICE+150
---------- ----------
         1      15150
         2      18150
         3      19150
         4      30150

SQL> select id,price-150 from mobile;

        ID  PRICE-150
---------- ----------
         1      14850
         2      17850
         3      18850
         4      29850

SQL> select id,price*150 from mobile;

        ID  PRICE*150
---------- ----------
         1    2250000
         2    2700000
         3    2850000
         4    4500000

SQL> select id,price/150 from mobile;

        ID  PRICE/150
---------- ----------
         1        100
         2        120
         3 126.666667
         4        200
```

Relational Operators: <,>, =,==

```
Command Prompt - sqlplus /nolog
SQL> select * from office;

        ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti    9000       manager
         2 mani       muthayla   9000       supervisor
         3 zeeshan    sir        5000       astmanager
         4 samara     sunny      15000      director
         5 shamu      shambhavi  7000       cashier

SQL> select FIRSTNAME from office where LASTNAME='sir';

FIRSTNAME
----------
zeeshan

SQL> select FIRSTNAME from office where ID='3';

FIRSTNAME
----------
zeeshan

SQL> select ID,FIRSTNAME,ROLL from office where SALARY='9000';

        ID FIRSTNAME  ROLL
---------- ---------- ----------
         1 eshwar     manager
         2 mani       supervisor

SQL> select ID,FIRSTNAME,ROLL from office where SALARY<'9000';

        ID FIRSTNAME  ROLL
---------- ---------- ----------
         3 zeeshan    astmanager
         4 samara     director
         5 shamu      cashier

SQL> select ID,FIRSTNAME,ROLL from office where SALARY>'9000';

no rows selected
```

Logical operators: and or not

(If both conditions are true only then value will get printed or if any one value is true, output will be printed)

```
SQL>
SQL> select * from mobile;

        ID MODEL      PRICE PLACE
---------- ----- ---------- --------------------
         1 vivo       15000 HYD
         2 oppo       18000 UPPAL
         3 mi         19000 AP
         4 apple      30000 DELHI

SQL> select PRICE from mobile where id='3'and PLACE='AP';

     PRICE
----------
     19000

SQL> select PRICE from mobile where id='2'or PLACE='delhi';

     PRICE
----------
     18000

SQL> select PRICE from mobile where id='2'or PLACE='DELHI';

     PRICE
----------
     18000
     30000
```

Special Operators: in, not in, between, like, not like

```
Command Prompt - sqlplus /nolog

SQL> select * from office;

       ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
        1 eshwar     emkanti    9000       manager
        2 mani       muthayla   9000       supervisor
        3 zeeshan    sir        5000       astmanager
        4 samara     sunny      15000      director
        5 shamu      shambhavi  7000       cashier

SQL> select * from office where FIRSTNAME in ('office','mani');

       ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
        2 mani       muthayla   9000       supervisor

SQL> select * from office where FIRSTNAME not in ('office','mani');

       ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
        1 eshwar     emkanti    9000       manager
        3 zeeshan    sir        5000       astmanager
        4 samara     sunny      15000      director
        5 shamu      shambhavi  7000       cashier
```

```
SQL> select ID,FIRSTNAME,LASTNAME,SALARY from office where ROLL like'%a%';

       ID FIRSTNAME  LASTNAME   SALARY
---------- ---------- ---------- ----------
        1 eshwar     emkanti    9000
        3 zeeshan    sir        5000
        5 shamu      shambhavi  7000

SQL> select ID,FIRSTNAME,LASTNAME,SALARY from office where ROLL not like'%a%';

       ID FIRSTNAME  LASTNAME   SALARY
---------- ---------- ---------- ----------
        2 mani       muthayla   9000
        4 samara     sunny      15000
```

```
SQL> select FIRSTNAME,SALARY from office where SALARY  between 9000 and 12000;

FIRSTNAME  SALARY
---------- ----------
eshwar     9000
mani       9000
```

```
Command Prompt - sqlplus /nolog

SQL> select FIRSTNAME,SALARY from office where SALARY not between 9000 and 12000;

FIRSTNAME  SALARY
---------- ----------
zeeshan    5000
samara     15000
shamu      7000
```

Oracle version and tables:

```
SQL>
SQL> select * from v$version;

BANNER
--------------------------------------------------------------------------------
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
PL/SQL Release 11.2.0.2.0 - Production
CORE    11.2.0.2.0      Production
TNS for 64-bit Windows: Version 11.2.0.2.0 - Production
NLSRTL Version 11.2.0.2.0 - Production

SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
ACCESS$                        TABLE
ALERT_QT                       TABLE
ALL_ALL_TABLES                 VIEW
ALL_APPLY                      VIEW
ALL_APPLY_CHANGE_HANDLERS      VIEW
ALL_APPLY_CONFLICT_COLUMNS     VIEW
ALL_APPLY_DML_CONF_HANDLERS    VIEW
ALL_APPLY_DML_HANDLERS         VIEW
ALL_APPLY_ENQUEUE              VIEW
ALL_APPLY_ERROR                VIEW
ALL_APPLY_ERROR_MESSAGES       VIEW

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
ALL_APPLY_EXECUTE              VIEW
ALL_APPLY_KEY_COLUMNS          VIEW
ALL_APPLY_PARAMETERS           VIEW
ALL_APPLY_PROGRESS             VIEW
ALL_APPLY_TABLE_COLUMNS        VIEW
ALL_ARGUMENTS                  VIEW
ALL_ASSEMBLIES                 VIEW
ALL_ASSOCIATIONS               VIEW
ALL_ATTRIBUTE_TRANSFORMATIONS  VIEW
ALL_AUDIT_POLICIES             VIEW
ALL_AUDIT_POLICY_COLUMNS       VIEW

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
ALL_AWS                        VIEW
ALL_AW_PS                      VIEW
ALL_BASE_TABLE_MVIEWS          VIEW
ALL_CAPTURE                    VIEW
ALL_CAPTURE_EXTRA_ATTRIBUTES   VIEW
ALL_CAPTURE_PARAMETERS         VIEW
ALL_CAPTURE_PREPARED_DATABASE  VIEW
ALL_CAPTURE_PREPARED_SCHEMAS   VIEW
ALL_CAPTURE_PREPARED_TABLES    VIEW
ALL_CATALOG                    VIEW
ALL_CHANGE_PROPAGATIONS        VIEW
```

Savepoint, Delete, Rollback Command:

```
SQL> select * from mobile;

        ID MODEL      PRICE PLACE
---------- ----- ---------- --------------------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL> savepoint A;

Savepoint created.

SQL> delete from mobile;

4 rows deleted.

SQL> select * from mobile;

no rows selected

SQL> rollback to A;

Rollback complete.

SQL> select * from mobile;

        ID MODEL      PRICE PLACE
---------- ----- ---------- --------------------
         1 vivo       15000
         2 oppo       18000
         3 mi         19000
         4 apple      30000

SQL>
```

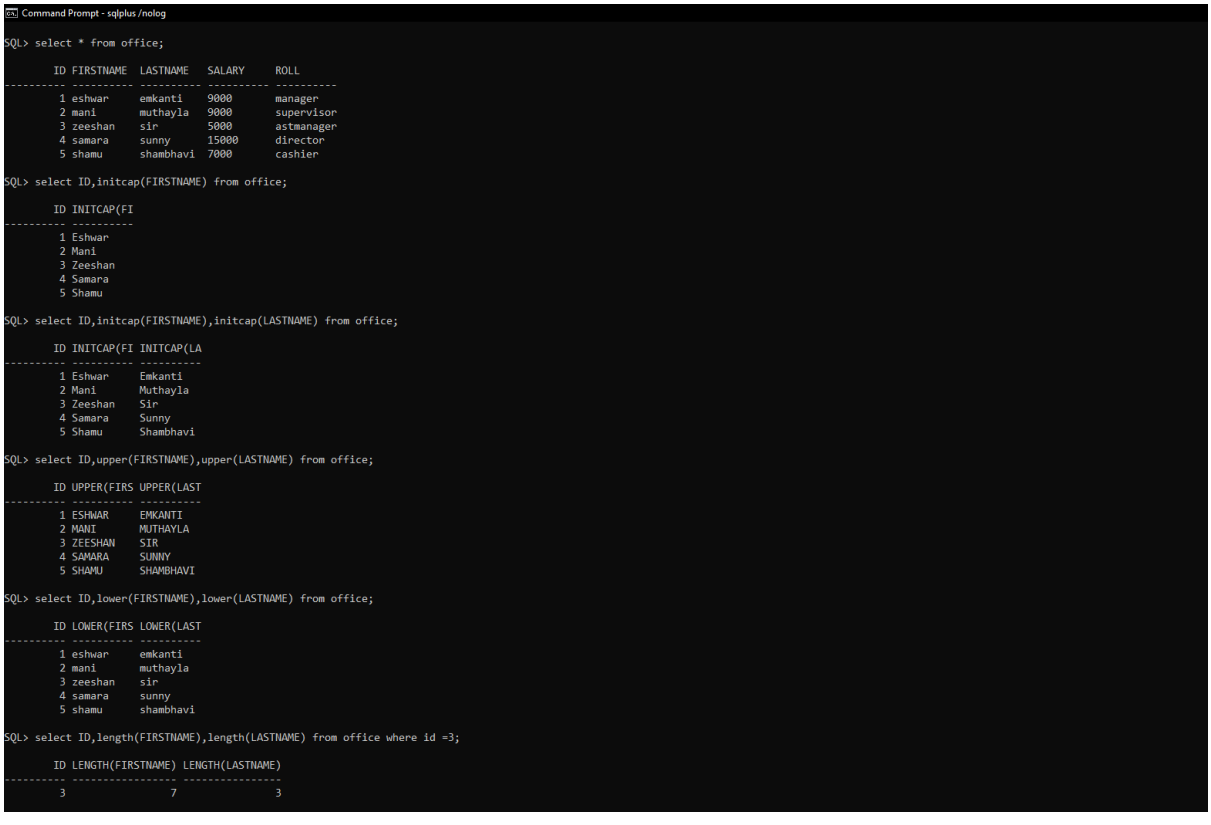
TCL: TRANSACT CONTROL LANGUAGE

Function will take one input value, perform calculation and provide results with different output.

Single Row function: String function, Mathematical function, Date function, Special function.

1)String function:

a) Initcap () first letter capital rest small
b) Upper () All letters capital
c) Lower () All letters small
d) Length () Number of characters
e) Substr() to subtract number of character



Multi row functions:

a) Min() min salary from the table
b) Max() max salary from the table
c) Avg() avg salary from the table
d) Count() count salaries or id's from the table
e) Round() to round ID's from the table (will not work with other group functions like; min,max etc.)

```
Command Prompt - sqlplus /nolog

SQL> select * from office
  2  ;

        ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti    9000       manager
         2 mani       muthayla   9000       supervisor
         3 zeeshan    sir        5000       astmanager
         4 samara     sunny      15000      director
         5 shamu      shambhavi  7000       cashier

SQL> select max (SALARY) from office;

MAX(SALARY
----------
9000

SQL> select COUNT (ID) from office;

 COUNT(ID)
----------
         5
```

Group by: sumofsalary, totalnoofemp, min,max(salary) group by

```
Command Prompt - sqlplus /nolog

SQL> select * from office;

       ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
        1 eshwar     emkanti    9000       manager
        2 mani       muthayla   9000       supervisor
        3 zeeshan    sir        5000       astmanager
        4 samara     sunny      15000      director
        5 shamu      shambhavi  7000       cashier

SQL> select ROLL,sum(SALARY) as sumfachjob from office group by Roll;

ROLL       SUMFACHJOB
---------- ----------
cashier          7000
director        15000
manager          9000
supervisor       9000
astmanager       5000

SQL> select ROLL,sum(SALARY) as totalnoofemp from office group by Roll;

ROLL       TOTALNOOFEMP
---------- ------------
cashier             7000
director           15000
manager             9000
supervisor          9000
astmanager          5000

SQL> select ROLL,count(SALARY) as totalnoofemp from office group by Roll;

ROLL       TOTALNOOFEMP
---------- ------------
cashier               1
director              1
manager               1
supervisor            1
astmanager            1

SQL> select ROLL,avg(SALARY) as totalnoofemp from office group by Roll;

ROLL       TOTALNOOFEMP
---------- ------------
cashier             7000
director           15000
manager             9000
supervisor          9000
astmanager          5000
```

Order by clause: ID count order by

```
Command Prompt - sqlplus /nolog

SQL> select * from office;

        ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti    9000       manager
         2 mani       muthayla   9000       supervisor
         3 zeeshan    sir        5000       astmanager
         4 samara     sunny      15000      director
         5 shamu      shambhavi  7000       cashier

SQL> select ROLL,min(SALARY) from office group by Roll;

ROLL       MIN(SALARY
---------- ----------
cashier    7000
director   15000
manager    9000
supervisor 9000
astmanager 5000
```

```
SQL> select id,firstname,lastname from office order by id asc;

        ID FIRSTNAME  LASTNAME
---------- ---------- ----------
         1 eshwar     emkanti
         2 mani       muthayla
         3 zeeshan    sir
         4 samara     sunny
         5 shamu      shambhavi

SQL> select id,firstname,lastname from office order by id desc;

        ID FIRSTNAME  LASTNAME
---------- ---------- ----------
         5 shamu      shambhavi
         4 samara     sunny
         3 zeeshan    sir
         2 mani       muthayla
         1 eshwar     emkanti

SQL>
```

Combine group by, order by:

```
SQL> select * from office;

        ID FIRSTNAME  LASTNAME   SALARY     ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti    9000       manager
         2 mani       muthayla   9000       supervisor
         3 zeeshan    sir        5000       astmanager
         4 samara     sunny      15000      director
         5 shamu      shambhavi  7000       cashier

SQL> select ROLL,id,sum(SALARY) as sumfachjob from office group by Roll,id order by id asc ;

ROLL              ID SUMFACHJOB
---------- ---------- ----------
manager            1       9000
supervisor         2       9000
astmanager         3       5000
director           4      15000
cashier            5       7000

SQL> select ROLL,id,sum(SALARY) as sumfachjob from office group by Roll,id order by id desc ;

ROLL              ID SUMFACHJOB
---------- ---------- ----------
cashier            5       7000
director           4      15000
astmanager         3       5000
supervisor         2       9000
manager            1       9000
```

Analytical functions: rank (), dense_rank (), partition by clause

```
Command Prompt - sqlplus /nolog

SQL> select firstname, Salary,rank() over(order by salary desc)as rnk from office;

FIRSTNAME  SALARY            RNK
---------- ---------- ----------
eshwar     9000                1
mani       9000                1
shamu      7000                3
zeeshan    5000                4
samara     15000               5

SQL> select firstname, Salary,dense_rank() over(order by salary desc)as rnk from office;

FIRSTNAME  SALARY            RNK
---------- ---------- ----------
eshwar     9000                1
mani       9000                1
shamu      7000                2
zeeshan    5000                3
samara     15000               4

SQL> select firstname, Salary,dense_rank() over(order by salary asc)as rnk from office;

FIRSTNAME  SALARY            RNK
---------- ---------- ----------
samara     15000               1
zeeshan    5000                2
shamu      7000                3
eshwar     9000                4
mani       9000                4

SQL> select firstname, Salary,dense_rank() over(partition by Roll order by salary desc)as rnk from office;

FIRSTNAME  SALARY            RNK
---------- ---------- ----------
zeeshan    5000                1
shamu      7000                1
samara     15000               1
eshwar     9000                1
mani       9000                1
```

Lead (next value displayed) and lag (previous value displayed), having clause (select specific group, filter groups):

```
Command Prompt - sqlplus /nolog

SQL> select firstname, Salary,dense_rank() over(partition by Roll order by salary desc)as rnk from office;

FIRSTNAME  SALARY            RNK
---------- ---------- ----------
zeeshan    5000                1
shamu      7000                1
samara     15000               1
eshwar     9000                1
mani       9000                1

SQL> select firstname, Salary,lead(salary,1,0)over(order by salary desc) as next_salary from office;

FIRSTNAME  SALARY     NEXT_SALAR
---------- ---------- ----------
eshwar     9000       9000
mani       9000       7000
shamu      7000       5000
zeeshan    5000       15000
samara     15000      0

SQL> select firstname, Salary,lead(salary,1,0)over(order by salary desc) as previos_salary from office;

FIRSTNAME  SALARY     PREVIOS_SA
---------- ---------- ----------
eshwar     9000       9000
mani       9000       7000
shamu      7000       5000
zeeshan    5000       15000
samara     15000      0

SQL> select sum(salary) from office group by roll having sum(salary)>14000;

SUM(SALARY)
-----------
      15000

SQL> select roll, sum(salary) from office group by roll having sum(salary)>14000;

ROLL       SUM(SALARY)
---------- -----------
director         15000
```

Rollup and cube clause:

```
Command Prompt - sqlplus /nolog

SQL> select roll,id,sum(salary)as sumofsalary from office group by rollup(roll,id);

ROLL               ID SUMOFSALARY
---------- ---------- -----------
cashier             5        7000
cashier                      7000
manager             1        9000
manager                      9000
director            4       15000
director                    15000
astmanager          3        5000
astmanager                   5000
supervisor          2        9000
supervisor                   9000
                            45000

11 rows selected.

SQL> select roll,id,sum(salary)as sumofsalary from office group by cube(roll,id);

ROLL               ID SUMOFSALARY
---------- ---------- -----------
                            45000
                    1        9000
                    2        9000
                    3        5000
                    4       15000
                    5        7000
cashier                      7000
cashier             5        7000
manager                      9000
manager             1        9000
director                    15000

ROLL               ID SUMOFSALARY
---------- ---------- -----------
director            4       15000
astmanager                   5000
astmanager          3        5000
supervisor                   9000
supervisor          2        9000

16 rows selected.
```

Where clause (select specific rows, filter rows):

```
Command Prompt - sqlplus /nolog

SQL> select ROLL,sum(SALARY) from office where ROLL in ('manager','cashier') group by ROLL;

ROLL       SUM(SALARY)
---------- -----------
cashier           7000
manager           9000

SQL>
```

Integrity constraints: Types of integrity (Entity integrity, Domain integrity, referential integrity)

Not Null: No null values

```
Connected.
SQL> create table system(id number(10) not null,firstname varchar2(10));

Table created.
```

```
SQL> insert into system values('','eshu');
insert into system values('','eshu')
                          *

ERROR at line 1:
ORA-01400: cannot insert NULL into ("SYS"."SYSTEM"."ID")
```

check constraint: providing limit condition

```
SQL> create table office1(list number(2),sal number(5) check(sal<6000));

Table created.

SQL> insert into office1 values(1,'15000');
insert into office1 values(1,'15000')
*
ERROR at line 1:
ORA-02290: check constraint (SYS.SYS_C007011) violated
```

Unique constraint: duplicate not allowed

```
Command Prompt - sqlplus /nolog

SQL> create table nokia(id number(10) unique,fname varchar2(20));

Table created.

SQL> insert into nokia values(1,'eshu');

1 row created.

SQL> insert into nokia values(1,'eshwar');
insert into nokia values(1,'eshwar')
*
ERROR at line 1:
ORA-00001: unique constraint (SYS.SYS_C007022) violated
```

Primary key: Null and duplicated not allowed

Foreign key: parent table and child to have similar values

Parent table= Nokia

Child table= oppo

```
Command Prompt - sqlplus /nolog

SQL> create table mi(id number(10) primary key,fname varchar2(20));

Table created.

SQL> insert into mi values('','eshwar');
insert into mi values('','eshwar')
                       *
ERROR at line 1:
ORA-01400: cannot insert NULL into ("SYS"."MI"."ID")


SQL> insert into mi values(1,'eshwar');

1 row created.

SQL> insert into mi values(1,'eshwar');
insert into mi values(1,'eshwar')
*
ERROR at line 1:
ORA-00001: unique constraint (SYS.SYS_C007023) violated
```

```
Command Prompt - sqlplus /nolog
SQL> create table vivo(sno number primary key,Mname varchar2(10));

Table created.

SQL> insert into vivo values(1,'y17');

1 row created.

SQL> insert into vivo values(2,'y21');

1 row created.

SQL> select * from vivo;

       SNO MNAME
---------- ----------
         1 y17
         2 y21

SQL> create table oppo(Mno number(3),Modelname varchar2(15),amt number(5),sno number(10) references
  2  vivo(sno));

Table created.
```

```
SQL> insert into oppo values(10,'nokia',10000,1);

1 row created.

SQL> insert into oppo values(10,'mi',10000,4);
insert into oppo values(10,'mi',10000,4)
*
ERROR at line 1:
ORA-02291: integrity constraint (SYS.SYS_C007025) violated - parent key not
found
```

SQL JOINs = combines columns of tables. There are 4 types of joins

- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

(INNER) JOIN:

```
SQL> select e.ROLL,d.work from emp e join dept d on e.ROLL=d.work;

ROLL       WORK
---------- ----------
casher     casher
tl         tl
manager    manager
```

LEFT (OUTER) JOIN:

```
SQL> select e.NAME,d.WORK from emp e left outer join dept d on e.ID=d.ID;

NAME  WORK
----- ----------
A1    casher
B1    tl
C1    manager
D1    cleaner
E1
```

RIGHT (OUTER) JOIN:

```
SQL> select e.NAME,d.WORK from emp e RIGHT outer join dept d on e.ID=d.ID;

NAME  WORK
----- ----------
A1    casher
B1    tl
C1    manager
D1    cleaner
```

FULL (OUTER) JOIN:

```
SQL> select e. NAME,d.WORK from emp e join dept d on e.id=d.id(+)
  2  UNION
  3  select e.NAME,d.WORK from emp e,dept d where e.id(+)=d.id;

NAME  WORK
----- ----------
A1    casher
B1    tl
C1    manager
D1    cleaner
E1
```

Sub queries/ complex queries:

1) Single row sub queries: aka complex query. for instance, select columns from <table>where column “operator” <select>. (<>!==)
2) Multi row sub queries: select columns from <table> where column “operator” <select>. In not in
3) Co-related sub queries
4) Inline views
5) Scalar sub queries

Single row sub queries:

```
SQL> SELECT FNAME,LNAME FROM STUDENT WHERE SAL<(SELECT SAL FROM STUDENT WHERE ID=7);

FNAME      LNAME
---------- ----------
eshwar     emkanti
kiran      kumar
sai        manoj
krishan    roa
samara     summy

SQL> SELECT FNAME,LNAME FROM STUDENT WHERE SAL>(SELECT SAL FROM STUDENT WHERE ID=1);

FNAME      LNAME
---------- ----------
kiran      kumar
chandu     babu
samara     summy
naveen     yadav
zeeshan    sir
```

```
SQL> select max(sal) from student where sal<(select max(sal) from student);

  MAX(SAL)
----------
      9000
```

Rownum:

```
SQL> select rownum, fname,lname from student;

    ROWNUM FNAME      LNAME
---------- ---------- ----------
         1 eshwar     emkanti
         2 kiran      kumar
         3 chandu     babu
         4 sai        manoj
         5 krishan    roa
         6 samara     summy
         7 naveen     yadav
         8 zeeshan    sir

8 rows selected.
```

Multi row sub queries:

```
SQL> select * from student where id IN(SELECT CASE ROWNUM
  2  WHEN 3 THEN ID
  3  WHEN 5 THEN ID
  4  END AS ID FROM STUDENT);

        ID FNAME      LNAME             SAL ROLL
---------- ---------- ---------- ---------- ----------
         3 chandu     babu             9000 sales
         5 krishan    roa              4500 linux
```

Rowid: Hexa mumber:

```
SQL> select rownum,rowid,fname from student;

    ROWNUM ROWID              FNAME
---------- ------------------ ----------
         1 AAAE+pAABAAALJ5AAA eshwar
         2 AAAE+pAABAAALJ5AAB kiran
         3 AAAE+pAABAAALJ5AAC chandu
         4 AAAE+pAABAAALJ5AAD sai
         5 AAAE+pAABAAALJ5AAE krishan
         6 AAAE+pAABAAALJ5AAF samara
         7 AAAE+pAABAAALJ5AAG naveen
         8 AAAE+pAABAAALJ5AAH zeeshan

8 rows selected.
```

Co-related sub queries:

```
SQL> select * from student;

        ID FNAME      LNAME             SAL ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti          5600 tester
         2 kiran      kumar            7000 java
         3 chandu     babu             9000 sales
         4 sai        manoj            4000 sql
         5 krishan    roa              4500 linux
         6 samara     summy            7500 desktop
         7 naveen     yadav            8500 csaher
         8 zeeshan    sir             10000 admin

8 rows selected.

SQL> select * from student where id in(select id from student where FNAME='eshwar');

        ID FNAME      LNAME             SAL ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti          5600 tester
```

```
SQL> select * from student where SAL< ALL(select SAL from student where LNAME='kumar');

        ID FNAME      LNAME             SAL ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti          5600 tester
         5 krishan    roa              4500 linux
         4 sai        manoj            4000 sql

SQL> select * from student where SAL> ALL(select SAL from student where LNAME='kumar');

        ID FNAME      LNAME             SAL ROLL
---------- ---------- ---------- ---------- ----------
         6 samara     summy            7500 desktop
         7 naveen     yadav            8500 csaher
         3 chandu     babu             9000 sales
         8 zeeshan    sir             10000 admin
```

```
SQL> select * from (select FNAME, SAL  as FINALSAL from student);

FNAME        FINALSAL
---------- ----------
eshwar           5600
kiran            7000
chandu           9000
sai              4000
krishan          4500
samara           7500
naveen           8500
zeeshan         10000

8 rows selected.

SQL> select * from (select FNAME, SAL/12  as FINALSAL from student);

FNAME        FINALSAL
---------- ----------
eshwar     466.666667
kiran      583.333333
chandu            750
sai        333.333333
krishan           375
samara            625
naveen     708.333333
zeeshan    833.333333

8 rows selected.

SQL> select * from (select FNAME, SAL*12  as FINALSAL from student);

FNAME        FINALSAL
---------- ----------
eshwar          67200
kiran           84000
chandu         108000
sai             48000
krishan         54000
samara          90000
naveen         102000
zeeshan        120000

8 rows selected.
```

Inline views sub queries (RNK):

```
SQL> select FNAME,LNAME,SAL,RANK()OVER(ORDER BY SAL DESC) as rnk from student;

FNAME      LNAME             SAL        RNK
---------- ---------- ---------- ----------
zeeshan    sir             10000          1
chandu     babu             9000          2
naveen     yadav            8500          3
samara     summy            7500          4
kiran      kumar            7000          5
eshwar     emkanti          5600          6
shambhavi  travels          5600          6
krishan    roa              4500          8
sai        manoj            4000          9

9 rows selected.

SQL> select FNAME,LNAME,SAL,DENSE_RANK()OVER(ORDER BY SAL DESC) as dense_rnk from student;

FNAME      LNAME             SAL  DENSE_RNK
---------- ---------- ---------- ----------
zeeshan    sir             10000          1
chandu     babu             9000          2
naveen     yadav            8500          3
samara     summy            7500          4
kiran      kumar            7000          5
eshwar     emkanti          5600          6
shambhavi  travels          5600          6
krishan    roa              4500          7
sai        manoj            4000          8

9 rows selected.
```

Create user in database:

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> create user bhai identified by system;

User created.

SQL> grant connect,resource to bhai;

Grant succeeded.

SQL> show user
USER is "SYS"
SQL> conn bhai
Enter password:
Connected.
SQL>
```

Grant access and privileges:

```
SQL> grant all on eshu to emkanti;

Grant succeeded.

SQL> conn emkanti
Enter password:
Connected.
SQL> select * from sys.eshu;

        ID FIRSTNAME               INCOME
---------- -------------------- ----------
         1 eshwar                     5000
         2 pavan                      8000
         3 naveen                     6500

SQL> show user
USER is "EMKANTI"
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> revoke all on eshu from emkanti;
```

Revoked access, privileges and verified:

```
SQL> revoke all on eshu from emkanti;

Revoke succeeded.
```

```
SQL> conn emkanti
Enter password:
Connected.
SQL> select * from sys.eshu;
select * from sys.eshu
                  *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Views: part of the table or subset of the table to view.

(Mainly used for database security).

```
SQL> create view manager as select ID,FNAME,LNAME from students;

View created.

SQL> grant all on manager to emkanti;

Grant succeeded.

SQL> conn emkanti/system
Connected.
SQL> select * from sys manager;
select * from sys manager
              *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> select * from sys.manager;

        ID FNAME      LNAME
---------- ---------- ----------
         1 eshwar     emkanti
         2 kiran      kumar
         3 chandu     babu
         4 sai        manoj
         5 krishan    roa
         6 samara     summy
         7 naveen     yadav
         8 zeeshan    sir
```

Sequence:

Sequence= 5max → enter

Create sequence <name>→enter

Start with 1→enter

Max value→ 5 or 7 or 100.

Curval= current value

Nextval= next value

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> create sequence A1
  2  START WITH 1
  3  INCREMENT BY 1
  4  MAXVALUE 6;
create sequence A1
                *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> CREATE TABLE ASTUDENT(SNO NUMBER(5),SNAME VARCHAR2(10));

Table created.

SQL> INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME');
Enter value for sname: ESHU
old   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME')
new   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'ESHU')

1 row created.

SQL> /
Enter value for sname: ZEESHAN
old   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME')
new   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'ZEESHAN')

1 row created.

SQL> /
Enter value for sname: LADDU
old   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME')
new   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'LADDU')

1 row created.

SQL> /
Enter value for sname: SHAMU
old   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME')
new   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'SHAMU')

1 row created.

SQL> /
Enter value for sname: CANDY
old   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME')
new   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'CANDY')

1 row created.

SQL> /
Enter value for sname: KISHORE
old   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'&SNAME')
new   1: INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'KISHORE')
INSERT INTO ASTUDENT VALUES(E1.NEXTVAL,'KISHORE')
                            *
ERROR at line 1:
ORA-08004: sequence E1.NEXTVAL exceeds MAXVALUE and cannot be instantiated


SQL> SELECT * FROM ASTUDENT;

       SNO SNAME
---------- ----------
         1 ESHU
         2 ZEESHAN
         3 LADDU
         4 SHAMU
         5 CANDY

SQL>
```

Synonym:

```
SQL> CREATE SYNONYM E FOR STUDENT;
CREATE SYNONYM E FOR STUDENT
               *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> SELECT * FROM E;

        ID FIRSTNAME                 INCOME
---------- -------------------- ----------
         1 eshwar                     5000
         2 pavan                      8000
         3 naveen                     6500

SQL>
```

Data Directoty: To retrieve all users and tables from Database

List of users:

```
SQL> desc all_users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(30)
 USER_ID                                   NOT NULL NUMBER
 CREATED                                   NOT NULL DATE

SQL> select username from all_users;

USERNAME
------------------------------
XS$NULL
EMKANTI
BHAI
ROKEY
APEX_040000
APEX_PUBLIC_USER
FLOWS_FILES
HR
MDSYS
ANONYMOUS
XDB

USERNAME
------------------------------
CTXSYS
APPQOSSYS
DBSNMP
ORACLE_OCM
DIP
OUTLN
SYSTEM
SYS

19 rows selected.

SQL>
```

Indexes: Improves performance of query

```
Command Prompt - sqlplus /nolog
SQL> select * from student;

        ID FNAME      LNAME           SAL ROLL
---------- ---------- ---------- ---------- ----------
         1 eshwar     emkanti          5600 tester
         2 kiran      kumar            7000 java
         3 chandu     babu             9000 sales
         4 sai        manoj            4000 sql
         5 krishan    roa              4500 linux
         6 samara     summy            7500 desktop
         7 naveen     yadav            8500 csaher
         8 zeeshan    sir             10000 admin
         9 shambhavi  travels          5600 car

9 rows selected.
```

```
SQL> create index ESHU on student(fname);

Index created.

SQL> explain plan for select * from student where fname='eshwar';

Explained.

SQL> select * from table(dbms_xplan.display());

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
Plan hash value: 2356778634

--------------------------------------------------------------------------------
| Id  | Operation         | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |         |     1 |    25 |     2   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| STUDENT |     1 |    25 |     2   (0)| 00:00:01 |
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

   1 - filter("FNAME"='eshwar')

13 rows selected.

SQL>
```