# dSQL Functions

## Create Table

Create table dest\_tbl\_1

(empid number,

Empname varchar2(50) ,

date\_of\_hire date)

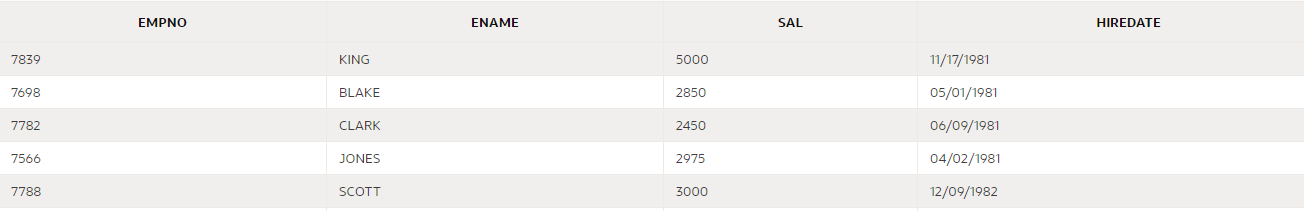
## Create Table from Another Existing table

CREATE TABLE EMPLOYEE AS

SELECT EMPNO, ENAME, SAL, HIREDATE

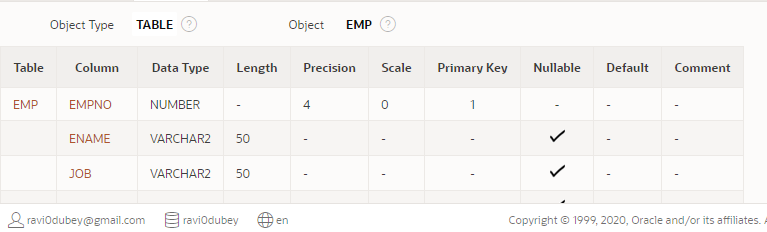
FROM EMP

SELECT \* FROM EMPLOYEE



## DESCRIBE

DESCRIBE EMP; DESC EMP



## ALTER

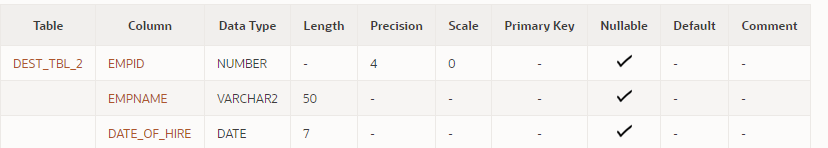
alter table dest\_tbl\_2

modify

(empid number(4),

Empname varchar2(50) ,

date\_of\_hire date)



To add Primary Key

alter table dest\_tbl\_3

modify

(empid number,

Empname varchar2(50) ,

date\_of\_hire date,

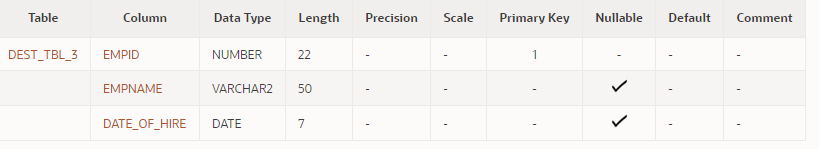
constraint pk\_key1 primary key(empid))

or

To add Primary Key column

ALTER TABLE stores

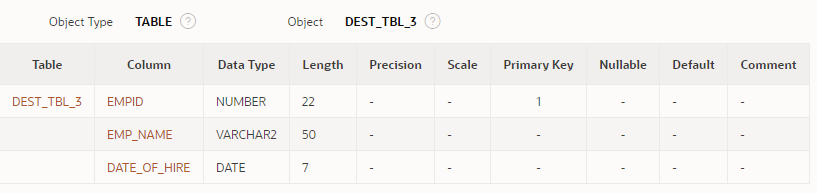
ADD CONSTRAINT store\_id\_pk PRIMARY KEY (store\_id)



To rename a field

ALTER TABLE DEST\_TBL\_3

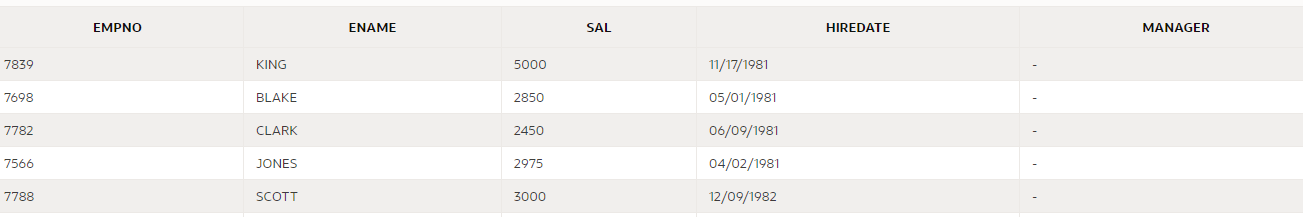
RENAME COLUMN EMPNAME to EMP\_NAME



To ADD a column

ALTER TABLE EMPLOYEE

ADD (MANAGER VARCHAR2 (10))



ATLER TABLE EMPLOYEE

DROP CONSTRAINT SYS\_C0001

To Drop a constraint

ALTER TABLE PRODUCTS

DROP CONSTRAINT fk\_supplier

To add a constraint

ALTER TABLE PRODUCTS

ADD CONSTRAINT fk\_supplier

FOREIGN KEY (supplier\_id)

REFERENCES supplier(supplier\_id)

ON DELETE CASCADE

To drop single column

ALTER TABLE SUPPLIER DROP (contact\_name)

To drop multiple column together

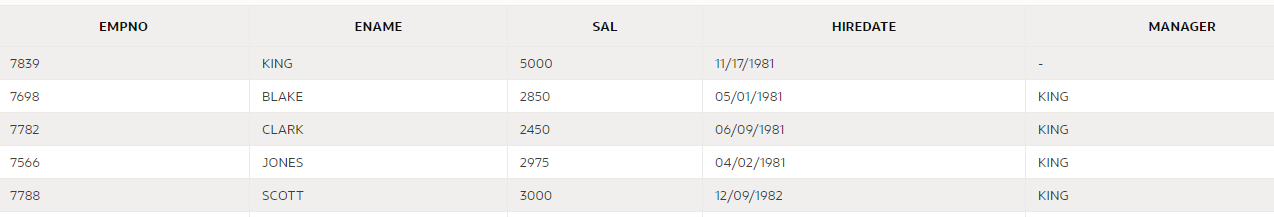
ALTER TABLE SUPPLIER DROP (contact\_name, supplier\_name)

## UPDATE

UPDATE EMPLOYEE

SET MANAGER = 'KING'

WHERE ENAME IN ('BLAKE','CLARK','JONES','SCOTT')



## CONCAT

select concat(concat(ename,upper(' is the Name')), concat(' and their job is: ', JOb)) as Name from emp

OR

SELECT EMPNO AS "Employee Number”, Ename ||' Makes ' ||Sal as "Salary"

FROM EMP

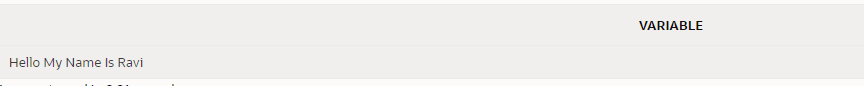
WHERE job = 'SALESMAN' AND (COMM = 300 OR COMM > 1000)

## INITCAP -> It Upper Case first Letter of each word

SELECT INITCAP ('hello my name is Ravi') as Variable

FROM DUAL

**Output**



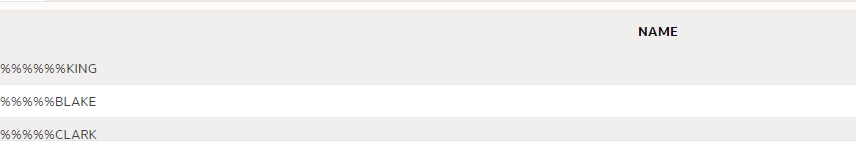
## LENGTH-> It will give length of data in each field

SELECT Ename, length(Ename) as Length from Emp

## LPAD -> it will do padding of extra spaces with value specified

SELECT LPAD (ENAME, 10, '%') AS NAME

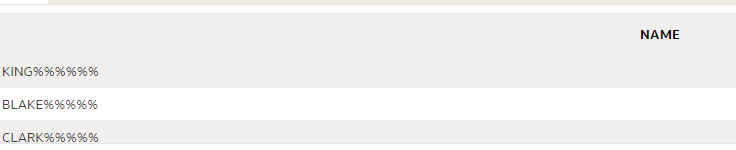
FROM EMP



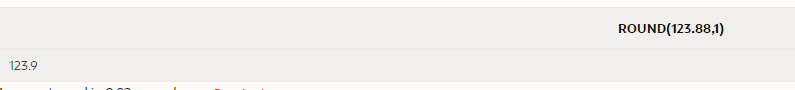
## RPAD

select RPAD(ENAME, 10, '%') AS NAME

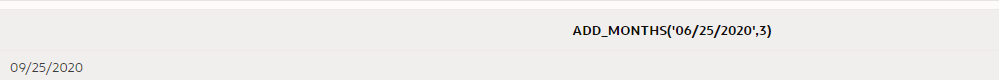
FROM EMP

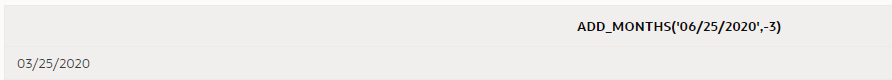


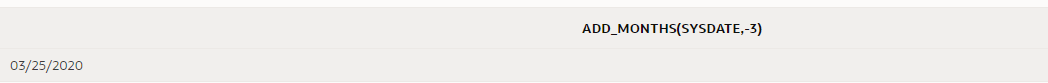
## ROUND

SELECT ROUND (123.88, 1) FROM DUAL;

## Add months

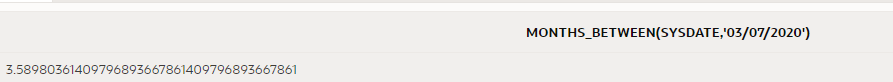
select add\_months('06/25/2020', 3) from dual;

select add\_months('06/25/2020', -3) from dual;

select add\_months(sysdate, -3) from dual;

## Months\_Between

select months\_between(sysdate,'03/07/2020') from dual;



## Trunc

select hiredate, trunc(hiredate,'month'), trunc(hiredate,'Year') from emp

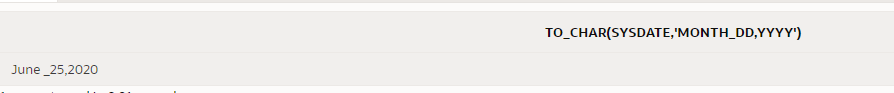
Trunc with month will give first date of the month

Trunc with year will give first date of year

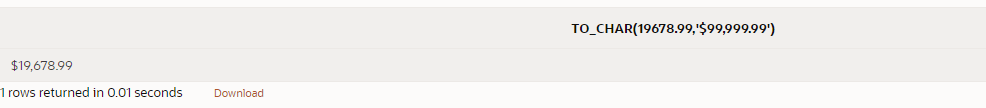


## TO\_CHAR

SELECT TO\_CHAR (SYSDATE, 'Month day year') from dual;

SELECT TO\_CHAR (SYSDATE, 'Month\_DD,YYYY') from dual;

SELECT SAL, TO\_CHAR(SAL, '$9999.00') AS Salary from EMP;

SELECT TO\_CHAR(19678.99,'$99,999.99') FROM DUAL

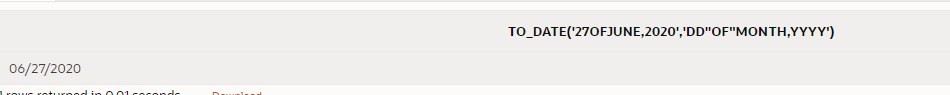
## ToDate

SELECT TO\_DATE('2020-06-27','YYYY-MM-DD') FROM DUAL

OR

SELECT TO\_DATE('27 OF JUNE, 2020','DD "OF" MONTH, YYYY')

FROM DUAL



## NVL-> Null Value used for Integer values

select comm, nvl(comm, '0')

from emp;

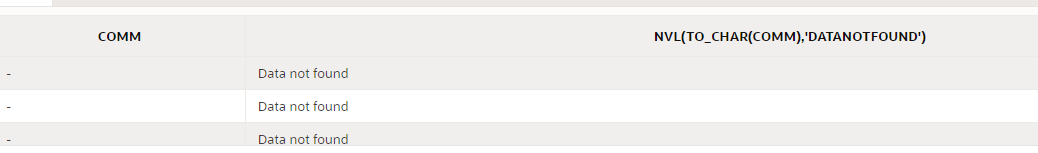
It replaces null value with provided value in this case 0



select comm, nvl(to\_Char(comm),'Data not found')

from emp;

It replaces null value with String “Data not Found “



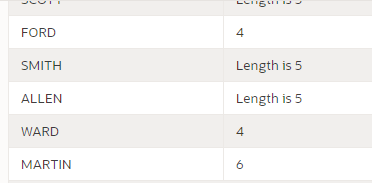
## NullIF

select eName, nvl(to\_Char(nullif(length(eName),5)),'Length is 5')

from emp;

If length is 5 it replaces with output to Character string “Length is 5”





## Manager with Maximum Salary

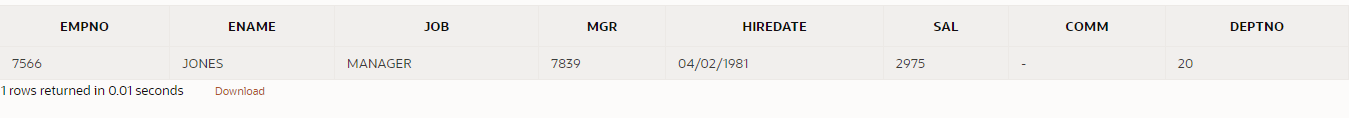
SELECT \* FROM EMP

WHERE SAL =

(SELECT MAX (SAL)

FROM EMP

WHERE JOB = 'MANAGER');



## Managers having Salary greater than Average Salary drawn by all Managers

SELECT \* FROM EMP

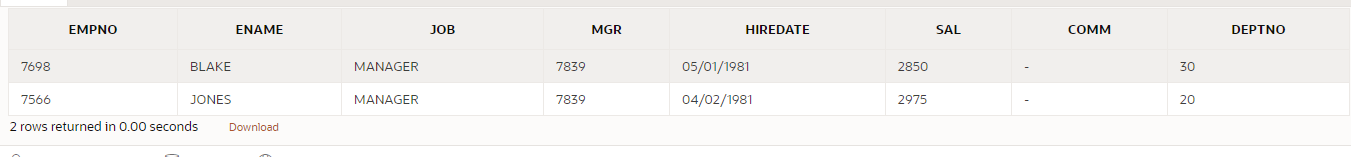
WHERE job = 'MANAGER'

AND SAL >

(SELECT AVG (SAL)

FROM EMP

WHERE JOB = 'MANAGER');

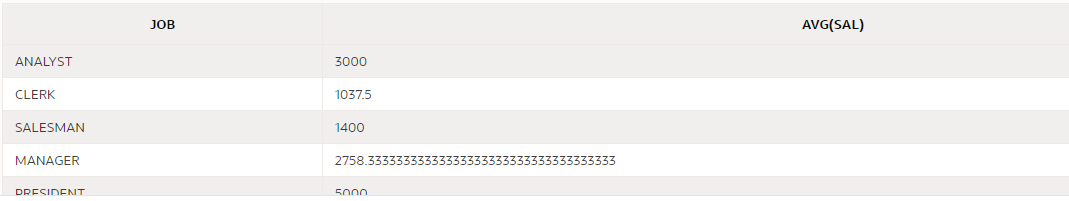


## Average Salary of each Job

SELECT JOB, AVG (SAL)

FROM EMP

GROUP BY JOB

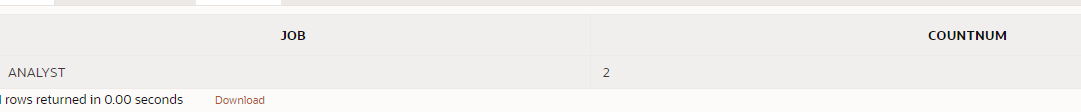


## Job where Count of employee is 2.

select job, count(job) as countnum from emp

group by job

having count(job)= 2

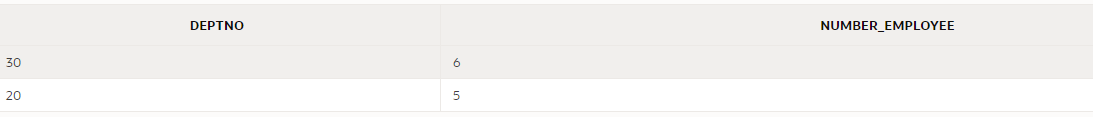


## Department where Count of employee is greater than 3 in that Department.

select deptno, count(\*) as Number\_Employee from emp

group by deptno

having count(\*)> 3



## Employees who work in Dallas.

SELECT A.\*

FROM EMP A JOIN DEPT B

ON A.DEPTNO = B.DEPTNO

WHERE UPPER (B.LOC) = 'DALLAS'

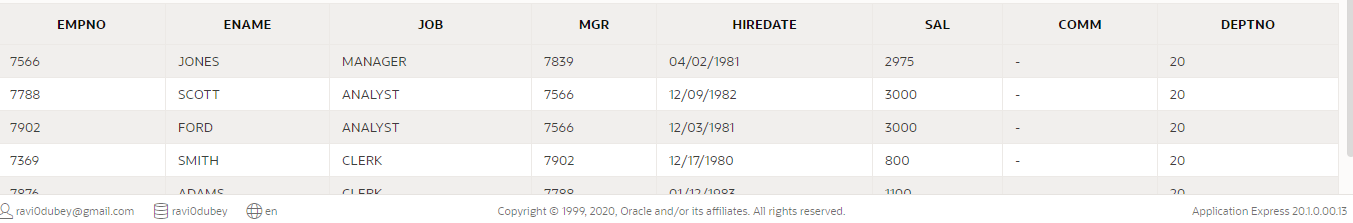
OR

SELECT A.\* FROM

(SELECT \* FROM EMP WHERE JOB in ('MANAGER', 'CLERK')) A,

(SELECT \* FROM DEPT WHERE UPPER (LOC) = 'DALLAS') B

WHERE A.DEPTNO = B.DEPTNO



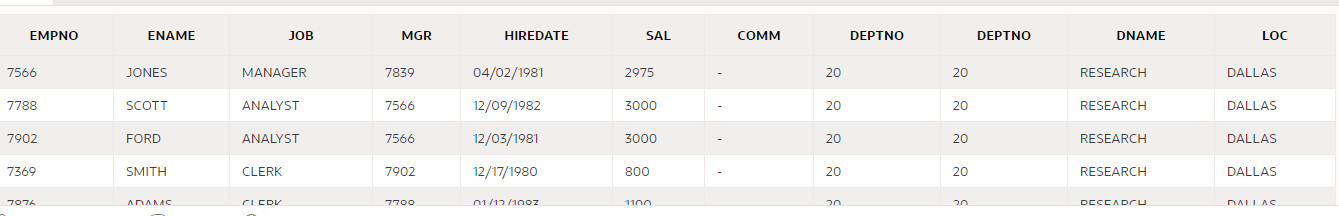
## Left Join.

SELECT \* FROM

EMP A LEFT JOIN DEPT B

ON A.DEPTNO = B.DEPTNO

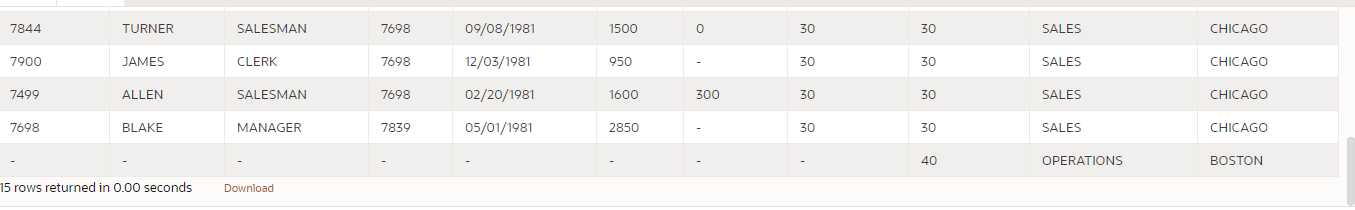
WHERE UPPER (B.LOC) = 'DALLAS'



## RIGHT Join.

SELECT \* FROM

EMP A RIGHT JOIN DEPT B

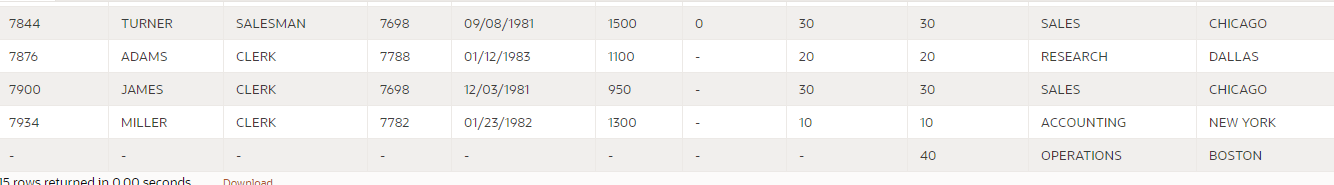
ON A.DEPTNO = B.DEPTNO

## FULL OUTER JOIN

SELECT \* FROM

EMP A FULL OUTER JOIN DEPT B

ON A.DEPTNO = B.DEPTNO



## LEFT OUTER JOIN

SELECT A.\*, B.\* FROM

DEPT B LEFT OUTER JOIN (SELECT \* FROM EMP WHERE JOB = ‘SALESMAN’) A

ON A.DEPTNO = B.DEPTNO

## SELFJOIN List of all Employees with their Respective Manager

SELECT E.EMPNO EMP\_ID, E.ENAME EMP\_NAME, E.MGR MGR\_ID, M.ENAME MGR\_NAME

FROM EMP E, EMP M

WHERE E.MGR = M.EMPNO

ORDER BY E.EMPNO

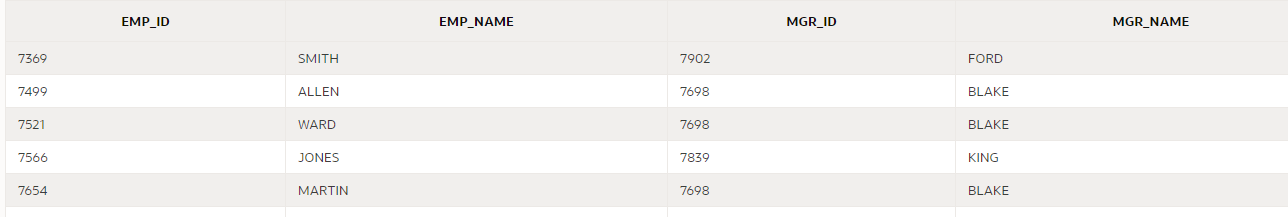
Or

SELECT E.EMPNO EMP\_ID, E.ENAME EMP\_NAME, E.MGR MGR\_ID, M.ENAME MGR\_NAME

FROM EMP E JOIN EMP M

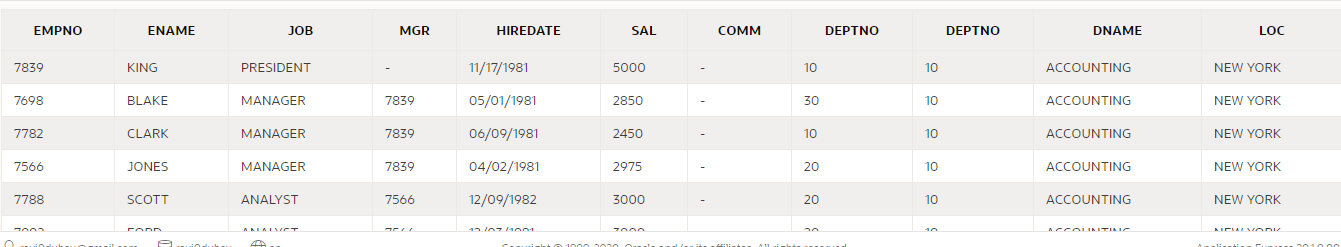
ON E.MGR = M.EMPNO

ORDER BY E.EMPNO



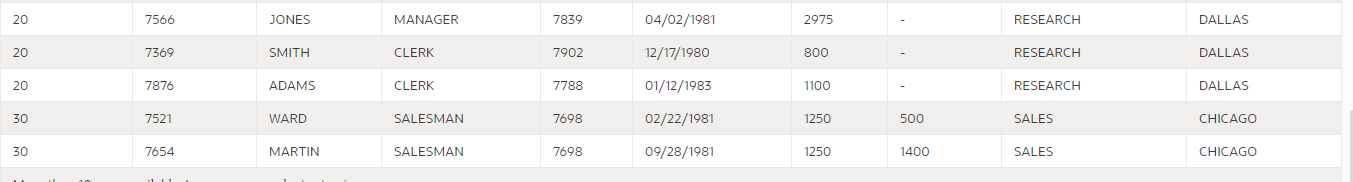
## NO Join, Cartesian product

SELECT \* FROM EMP, DEPT -> It will give Cartesian product of Each row of EMP multiplied with each row of DEPT



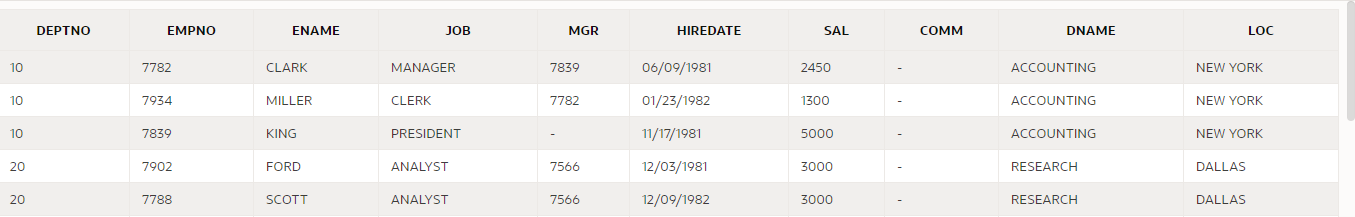
## Natural Join: it is inner join without explicitly mentioning field on which join is done, provided field name remains same in both the files

SELECT \* FROM EMP NATURAL JOIN DEPT



## Using in Join:

SELECT \* FROM EMP JOIN DEPT USING (DEPTNO)



## EQUI Join: it uses equal to in select statement

SELECT E.EMPNO EMP\_ID, E.ENAME EMP\_NAME, E.MGR MGR\_ID, M.ENAME MGR\_NAME

FROM EMP E JOIN EMP M

ON E.MGR = M.EMPNO

ORDER BY E.EMPNO

## NON EQUI Join: when there is no common field between 2 tables

SELECT E.NAME, E.SAL, J.GRADE\_LEVEL

FROM EMP E JOIN JOB\_GRADE J

ON E.SAL BETWEEN J.LOWESTSAL AND J.HIGHEST\_SAL

## Case Select

SELECT ENAME, SAL,

(Case WHEN SAL > 1500 THEN 'MANAGER11'

ELSE 'NO COMMENT'

END)

AS "COMMENT\_PEAK"

FROM EMP

## Correlated Sub query to showcase number of people working in same DEPTNO

SELECT E.\*,

(SELECT COUNT (\*)

FROM EMP

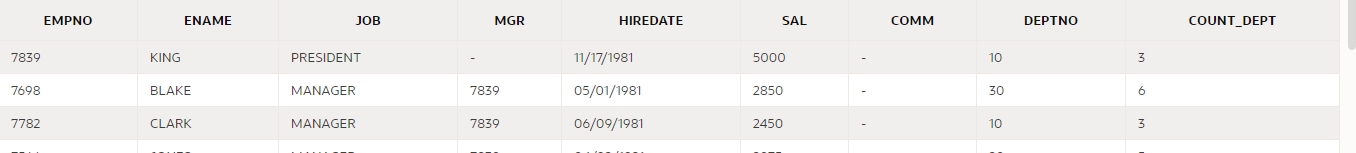
WHERE DEPTNO = E.DEPTNO) AS COUNT\_DEPT

FROM EMP E

OR

SELECT E.\*, COUNT (\*) OVER (PARTITION BY DEPTNO) AS COUNT\_DEPT

FROM EMP E



## Correlated Sub query to showcase Total SALARY of people working in same DEPTNO

SELECT E.\*,

(SELECT COUNT (\*)

FROM EMP

WHERE DEPTNO = E.DEPTNO) AS COUNT\_DEPT,

(SELECT SUM (SAL) FROM EMP

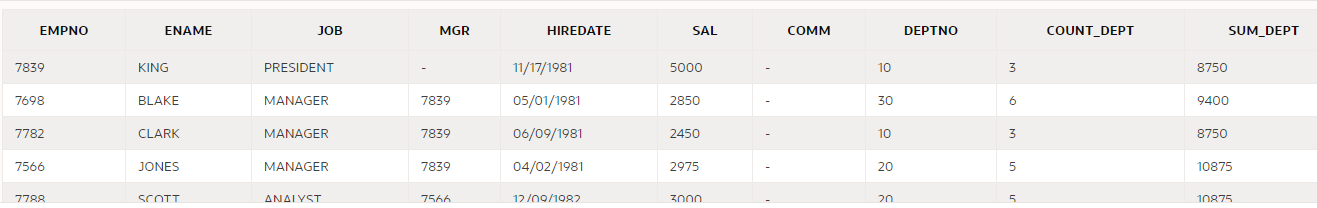
WHERE DEPTNO = E.DEPTNO) AS SUM\_DEPT

FROM EMP E

OR

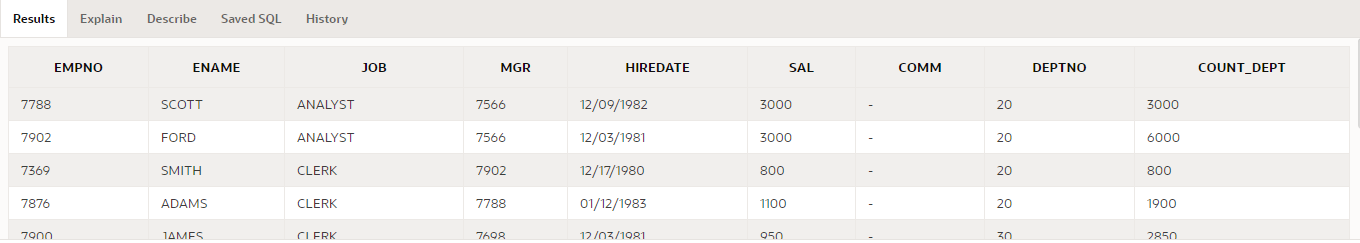
SELECT E.\*, SUM (SAL) OVER (PARTITION BY DEPTNO) AS COUNT\_DEPT

FROM EMP E



SELECT E.\*, SUM (SAL) OVER (PARTITION BY JOB ORDER BY EMPNO) AS COUNT\_DEPT

FROM EMP E



## INSERT ALL – It will insert into different tables, values based on condition given below

INSERT ALL

WHEN SAL <= 1000 THEN

INTO DEST\_TBL\_1 VALUES (EMPNO, ENAME, HIREDATE)

WHEN SAL BETWEEN 1001 AND 3000 THEN

INTO DEST\_TBL\_2 VALUES (EMPNO, ENAME, HIREDATE)

ELSE

--WHEN SAL > 3000 THEN

INTO DEST\_TBL\_3 VALUES (EMPNO, ENAME, HIREDATE)

SELECT EMPNO, ENAME, HIREDATE, SAL

FROM EMP

alter table dest\_tbl\_3

modify

(empid number,

Empname varchar2(50) ,

date\_of\_hire date,

constraint pk\_key1 primary key(empid)

)

## MERGE

MERGE INTO EMPLOYEE E1

USING EMP E2

ON (E1.EMPNO = E2.EMPNO)

WHEN MATCHED THEN

UPDATE SET

E1.EMPNO = E2.EMPNO,

E1.ENAME = E2.ENAME,

E1.SAL = E2.SAL,

E1.MANAGER = E2.MGR

WHEN NOT\_MATCHED THEN

INSERT (E1.EMPNO, E1.ENAME, E1.SAL, E1.MANAGER)

VALUES (E2.EMPNO, E2.ENAME, E2.SAL, E2.MGR);

## SEQUENCE

## CREATE SEQUENCE EMPLOYEE\_SEQ

MINVALUE 1

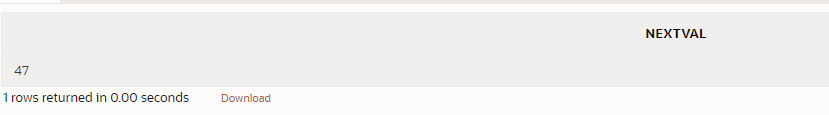
MAXVALUE 9999

START WITH 1

INCREMENT BY 2

CACHE 20;

SELECT EMPLOYEE\_SEQ.NEXTVAL FROM DUAL;



ALTER SEQUENCE EMPLOYEE\_SEQ

NOCACHE;

## TRUNCATE

TRUNCATE TABLE EMP

## INDEX

Create Index

CREATE INDEX EMP\_ENAME\_IDX

ON EMPLOYEE (ENAME);

Create Unique Index-> Field on which index is created has to be unique with no duplicate keys

CREATE UNIQUE INDEX EMP\_MANAGER\_IDX

ON EMPLOYEE (MANAGER);

Create Index with Compute Statistics

CREATE INDEX EMP\_MANAGER\_SAL\_IDX

ON EMPLOYEE (MANAGER, SAL)

COMPUTE STATISTICS;

Drop Index

DROP INDEX EMP\_MANAGER\_IDX;

DROP INDEX EMP\_ENAME\_IDX;

Alter Index

ALTER INDEX EMP\_ENAME\_IDX

REBUILD COMPUTE STATISTICS;

## Delete duplicate records from file.

DELETE FROM STORES

WHERE ROWID NOT IN

(SELECT MIN(ROWID)

FROM STORES

GROUP BY STOREID, CITY)

## FETCH.

LIMITED records from file

SELECT \* from employee

FETCH FIRST 2 ROWS ONLY

SELECT \* from employee

FETCH NEXT 2 ROWS ONLY

Fetch only 5 rows

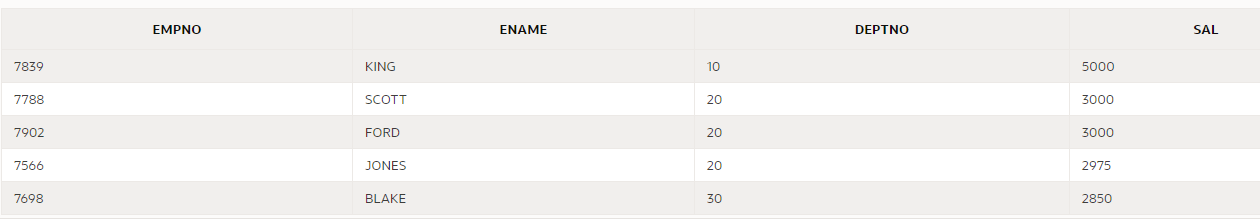
SELECT E.EMPNO, E.ENAME, E.DEPTNO, E.SAL

FROM EMP E INNER JOIN DEPT D

ON E.DEPTNO = D.DEPTNO

ORDER BY SAL DESC

FETCH NEXT 5 ROWS ONLY;



Skip first 5 rows

SELECT E.EMPNO, E.ENAME, E.DEPTNO, E.SAL

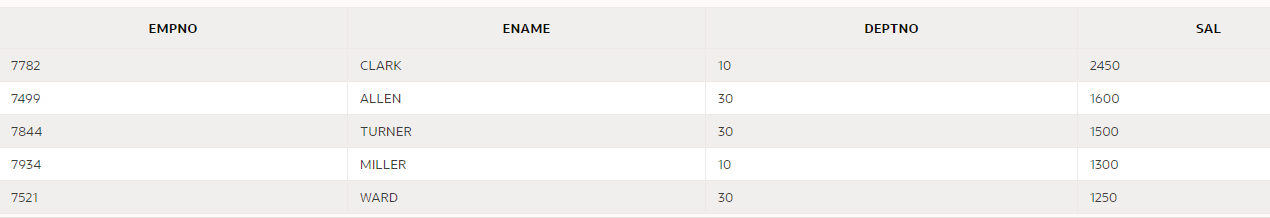
FROM EMP E INNER JOIN DEPT D

ON E.DEPTNO = D.DEPTNO

ORDER BY SAL DESC

OFFSET 5 ROWS (In case we give value of offset to negative, it means it will not skip any row)

FETCH NEXT 5 ROWS ONLY;



Show first 5 rows with ties will show same salary. Like in below 5th and 6th records had same salary so it show 6 records instead of 5

SELECT E.EMPNO, E.ENAME, E.DEPTNO, E.SAL

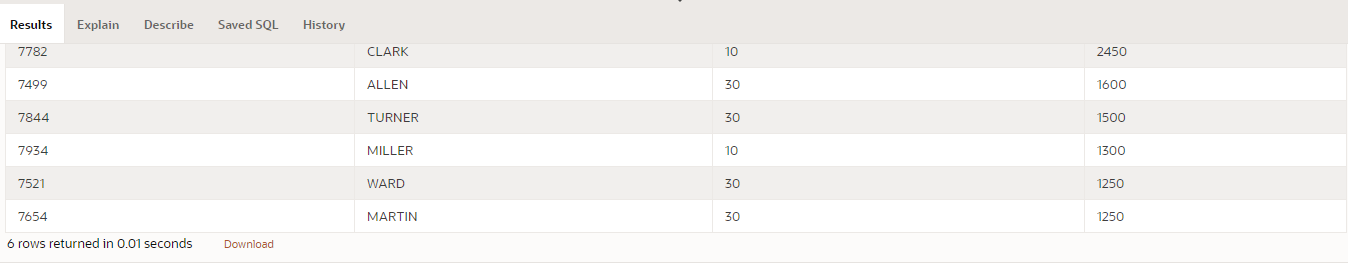
FROM EMP E INNER JOIN DEPT D

ON E.DEPTNO = D.DEPTNO

ORDER BY SAL DESC

OFFSET 5 ROWS

FETCH NEXT 5 ROWS WITH TIES;



PERCENT rows

SELECT E.EMPNO, E.ENAME, E.DEPTNO, E.SAL

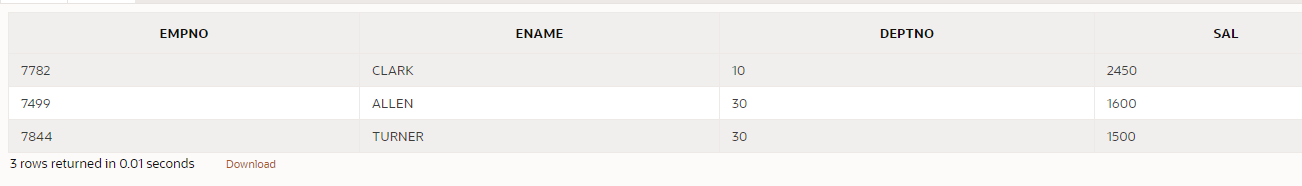
FROM EMP E INNER JOIN DEPT D

ON E.DEPTNO = D.DEPTNO

ORDER BY SAL DESC

OFFSET 5 ROWS

FETCH NEXT 5 PERCENT ROWS ONLY;

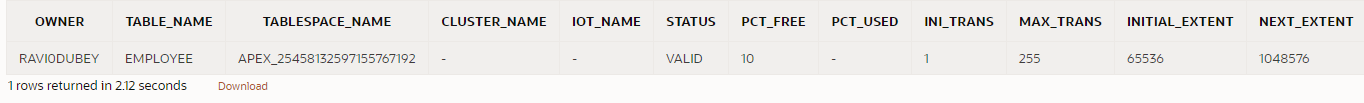


## ALL TABLES

SELECT \* FROM ALL\_TABLES

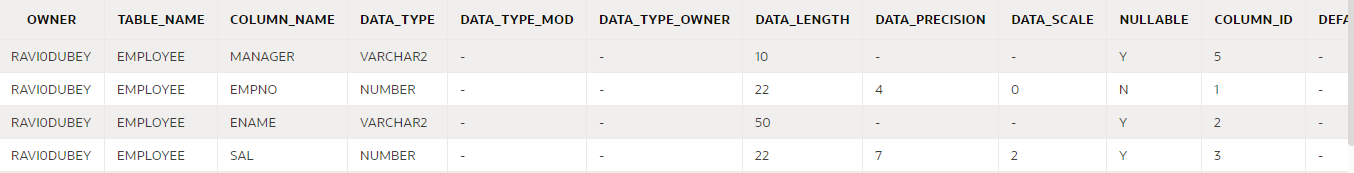
WHERE TABLE\_NAME = 'EMPLOYEE'

AND ROWNUM < 5



SELECT \* FROM ALL\_TAB\_COLUMNS

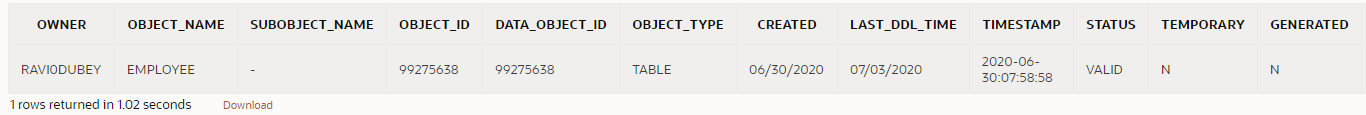
WHERE TABLE\_NAME = 'EMPLOYEE'



SELECT \* FROM ALL\_OBJECTS

WHERE OBJECT\_TYPE = 'TABLE'

AND OBJECT\_NAME = 'EMPLOYEE'



SELECT \* FROM ALL\_OBJECTS

WHERE OBJECT\_TYPE = 'INDEX'

AND OBJECT\_NAME = 'EMP\_MANAGER\_IDX'

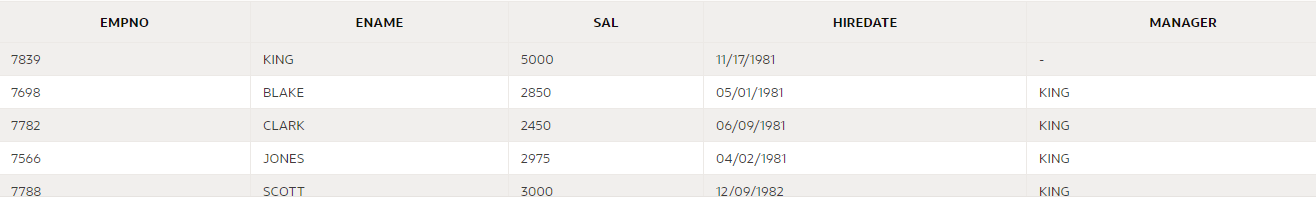


## SYNONYM

CREATE SYNONYM EMP\_TABLE

FOR EMPLOYEE

SELECT \* FROM EMP\_TABLE



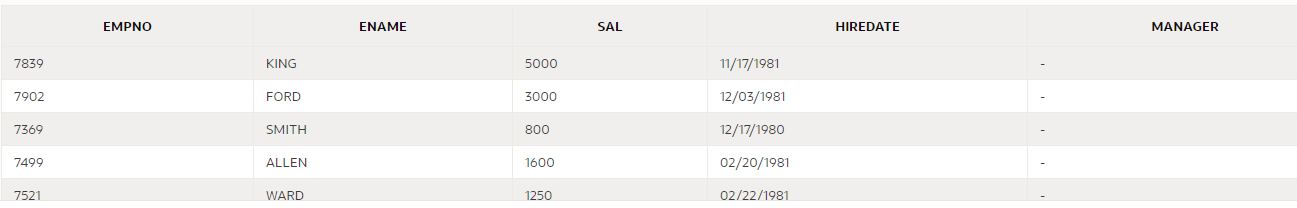
## VIEWS

CREATE VIEW MANAGER\_VIEW

AS SELECT \* FROM EMPLOYEE

WHERE MANAGER IS NULL

SELECT \* FROM MANAGER\_VIEW



DROP MANAGER\_VIEW

Update Existing Views by

CREATE OR REPLACE VIEW MANAGER\_VIEW

AS SELECT \* FROM EMPLOYEE

WHERE MANAGER IS NULL

## To Retrieve Employee Records with Maximum Salary in their Department

### Approach 1

Step1: Create View to fetch max Salary and Department number of each department

CREATE VIEW MAX\_RECORDS\_V

AS

SELECT DEPTNO, MAX (SAL) AS MAX\_SAL FROM EMP

GROUP BY DEPTNO

Step2: Join Table with View to get Records of Each Employee table from each department

SELECT A.\* FROM EMP A JOIN MAX\_RECORDS\_V M

ON A.DEPTNO = M.DEPTNO

AND A.SAL = M.MAX\_SAL

### Approach 2

SELECT E.\* FROM EMP E JOIN

(SELECT DEPTNO, MAX (SAL) AS MAX\_SAL FROM EMP

GROUP BY DEPTNO) E2

ON E.DEPTNO = E2.DEPTNO

AND E.SAL = E2.MAX\_SAL

## To Retrieve Employee Records with Maximum Salary in their Department

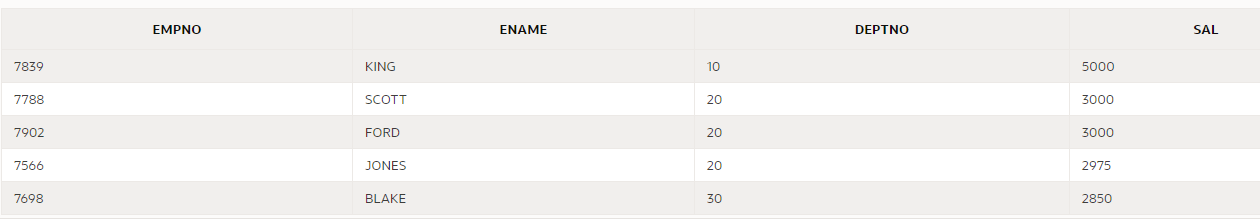
SELECT E.EMPNO, E.ENAME, E.DEPTNO, E.SAL

FROM EMP E INNER JOIN DEPT D

ON E.DEPTNO = D.DEPTNO

ORDER BY SAL DESC

FETCH NEXT 5 ROWS ONLY;



## UNION It will give unique records

SELECT \* FROM EMP

UNION

SELECT \* FROM EMPLOYEES

## UNION ALL it will give all records

SELECT \* FROM EMP

UNION ALL

SELECT \* FROM EMPLOYEES

## MINUS it will give all records which are not in below table

SELECT \* FROM EMP

MINUS

SELECT E.\* FROM EMP E JOIN

(SELECT DEPTNO , MAX(SAL) AS MAX\_SAL FROM EMP

GROUP BY DEPTNO) E2

ON E.DEPTNO = E2.DEPTNO

AND E.SAL = E2.MAX\_SAL

## GRANT

GRANT SELECT ON PRODUCTS TO USER2-> It will grant select authority to user2 on PRODUCTS table of User1

Now User2 can search using SELECT USER1.PRODUCTS

## Revoke

REVOKE SELECT ON PRODUCTS TO USER2-> It will revoke select authority to user2 on PRODUCTS table of User1

## ROLE

CREATE ROLE R1

GRANT P1, P2, P3 TO R1

ALTER USER U1 DEFAULT ROLE R1

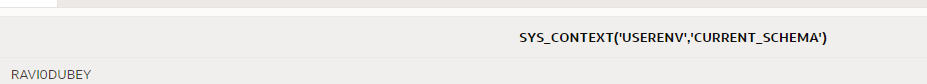
## To see user logged in

SHOW USER in SQL

In APEX

SELECT SYS\_CONTEXT ('USERENV', 'CURRENT\_SCHEMA')

FROM DUAL



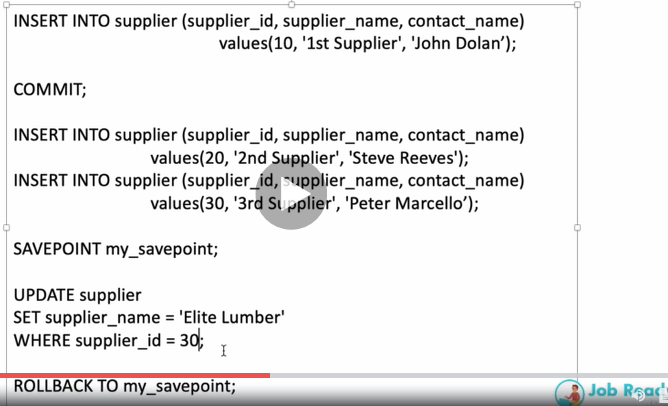
## ROLLBACK, COMMIT, SAVEPOINT are part of TCL (Transaction Control Language)

SAVEPOINT serve as an identifier where we can reach using ROLLBACK

ROLLBACK will undo everything to last COMMIT

COMMIT will commit everything.

Implicit COMMIT does happen when Grant, Revoke, Create table command gets executed.



## Cascade Constraints->

It will first remove Constraints and then it will drop the column

ALTER TABLE SUPPLIER DROP COLUMN supplier\_id CASCADE CONSTRAINTS

## Cascade Delete-

If record is delete from Parent table is deleted then corresponding record from child table.

Create table with syntax ON DELETE CASCADE

CREATE TABLE products

( product\_id numeric(10) not null,

supplier\_id numeric(10) not null,

CONSTRAINT fk\_supplier

FOREIGN KEY (supplier\_id)

REFERENCES supplier(supplier\_id)

ON DELETE CASCADE

);

## Column Unused

This command work much faster than Drop Column. If column is set to unused then we cannot query it nor can we retrieve or restore its data.

ALTER TABLE SUPPLIER

SET UNUSED COLUMN SUPPLIER\_ID;

OR

ALTER TABLE SUPPLIER

SET UNUSED (SUPPLIER\_ID, SUPPLIER\_NAME);

Once we wish to finally drop the column, we need to mention below command

ALTER TABLE SUPPLIER

DROP UNUSED COLUMNS;

## To see all Columns marked as Unused

## SELECT \* FROM USER\_UNUSED\_COL\_TABS;

## External Tables

## With External tables we can see it as window in oracle, Data is not loaded into Oracle.

When External table are created it create log file in Shared folder. In syntax we use Oracle Loader.We can use Skip records in Syntax to skip how many records we wish to skip while creating external tables

## We cannot

## Use Large Object data type like LOB, CLOB etc.

## Create a constraint

## Change column unused (Oracle will drop it)

## SQL Plus and Substitution Variables

1. Show all -> all system defined variables
2. DEFINE -> ALL user defined variable
3. SET LINESIZE 200 -> It changes line size to 200
4. SET DEFINE OFF -> It will not replace with substituted values
5. PROMPT Hello There -> It will how on terminal value “Hello There”
6. ACCPET VINPUT PROMPT “Give an input” -> it will ask user to enter value which will be stored in VINPUT variable

* Give an input 200
* Prompt Hey there value of VINPUT is : &VINPUT -> it will display line below as

Hey there the value of VINPUT is 200

1. Prompt Hey user give me your name :” value of VINPUT is : &VINPUT

* select \* from customer where fname = ‘&VINPUT’

1. select \* from customer where fname = &VINPUT

* Enter value for VINPUT: ‘RAVI’ ( It will substitute value entered in above query and execute)
* select \* from customer where fname = ‘RAVI’

1. Define vlastname = ‘RAVI’

* Prompt vlastname -> RAVI(it display)
* select \* from customer where fname = ‘&VLASTNAME’
* It gets substitute to select \* from customer where fname = ‘RAVI’

# Notes

## Oracle does not have LIMIT Clause

## We cannot drop column from a table where we have only one column in it

A table cannot have more than 1000 columns

LOB data type cannot be used as Primary Keys. So DISTINCT, GROUP BY or ORDER BY

|  |  |  |  |
| --- | --- | --- | --- |
| BLOB | Binary Large Object Data type | 4 GB | Image, Video Files |
| CLOB | Character Large Object Data Type | 4 GB | Books, String Sequences, Articles |
| NCLOB |  | 4 GB | Unicode |