PRACTICE ON SQL

**ALTER TABLE AP\_for**

**adding constraint, modifying column sizes, setting NULL/NOT NULL, DEFAULT values**

* Adding CHECK constraints to enforce business rules

alter table AP\_EMP add constraint C\_EMP\_EMPNO check (EMPNO between 1000 and 9999);

alter table AP\_PROJECT add constraint C\_PROJECT\_PROJID check (PROJID between 100 and 999);

alter table AP\_DEPT add constraint C\_DEPT\_DEPTNO check (DEPTNO between 10 and 99);

alter table AP\_EMP add constraint C\_EMP\_ENAME check (ENAME=upper(ENAME));

* Test constraint enforcements

insert into AP\_EMP values (10001, 'Robert', 'TEMP',7369,sysdate-40,1000,null,10);

insert into AP\_EMP values (9999, 'Robert', 'TEMP',7369,sysdate-40,1000,null,10);

* Modify column to add DEFAULT value

alter table AP\_PROEMP modify ( HOURS default 0 );

* Modify column to allow for NULL

alter table AP\_EMP modify job NULL;

* Modify column to change size and set NOT NULL

alter table AP\_EMP modify job varchar2(20) NOT NULL;

* Adding foreign key constrain

alter table AP\_EMP add constraint FK\_EMP\_MGR foreign key (MGR) references AP\_EMP(EMPNO)

**ALTER TABLE AP\_to add/remove column**

alter table AP\_EMP add BIRTHDATE date;

alter table AP\_EMP add ADDRESS varchar2(50);

alter table AP\_EMP drop ( BIRTHDATE, ADDRESS);

**Creating table from other table**

CREATE TABLE AP\_EMPTEST AS SELECT \* FROM AP\_EMP -- CREATE A TABLE FROM EXISITNG TABLE

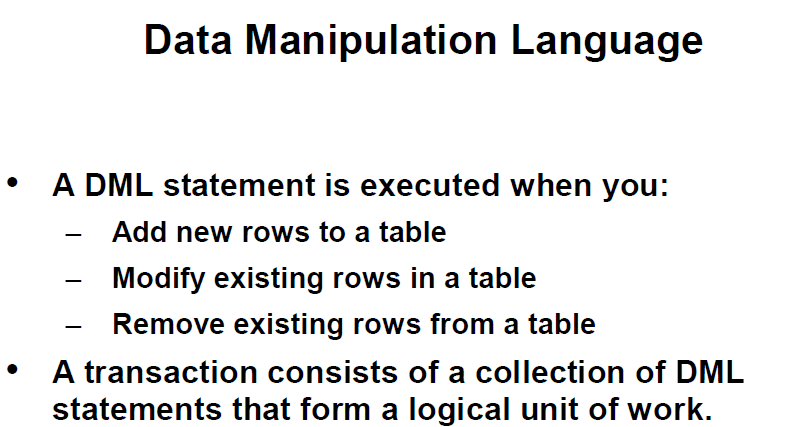
SELECT \* FROM AP\_EMPTEST -- RETRIVING/SELECTING ALL RECORDS OF A TABLE

TRUNCATE TABLE AP\_EMPTEST -- REMOVING/DELETING ALL RECORDS OF A TABLE

SELECT \* FROM AP\_EMPTEST-- RETRIVING/SELECTING ALL RECORDS OF A TABLE

DROP TABLE AP\_EMPTEST -- DROPPING A TABLE

SELECT \* FROM AP\_EMPTEST -- RETRIVING/SELECTING ALL RECORDS OF A TABLE



– CREATING A EMPTY TABLE FROM ANOTHER TABLE;

CREATE TABLE AP\_DEPTTEST AS SELECT \* FROM AP\_DEPT WHERE 1=2;

CREATE TABLE AP\_EMPTEST AS SELECT \* FROM AP\_EMP WHERE 1=2;

--POPULATING TABLE FROM ANOTHER TABLE;

INSERT INTO AP\_DEPTTEST SELECT \* FROM AP\_DEPT;

INSERT INTO AP\_EMPTEST SELECT \* FROM AP\_EMP;

COMMIT;

INSERT INTO AP\_DEPTTEST (deptno,dname,loc) VALUES ('TRAIING','AUSTIN');

INSERT INTO AP\_DEPTTEST (deptno,dname,loc) VALUES (NULL,'TRAINING','AUSTIN');

INSERT INTO AP\_DEPTTEST (deptno,dname,loc) VALUES (400,'TRAINING','AUSTIN');

INSERT INTO AP\_DEPTTEST VALUES (50,'TRAINING','AUSTIN');

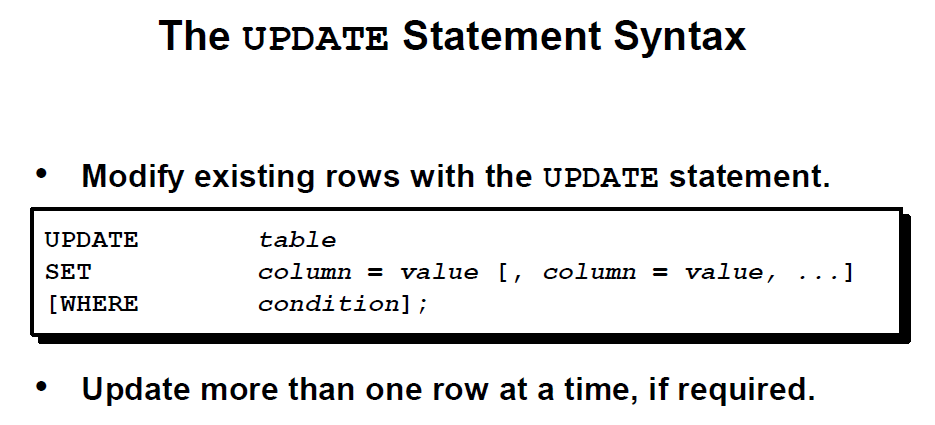
SELECT \* FROM AP\_DEPTTEST;

COMMIT;

INSERT INTO AP\_DEPTTEST VALUES (60,'LEGAL',NULL);

SELECT \* FROM AP\_DEPTTEST;

SELECT \* FROM AP\_DEPTTEST;



SELECT deptno

FROM ap\_emp

WHERE empno=7369;

UPDATE ap\_emptest

SET deptno=30

WHERE empno=7369;

SELECT empno,deptno

FROM ap\_emptest;

UPDATE ap\_emptest

SET sal=sal+100

WHERE job='CLERK';

UPDATE ap\_emptest

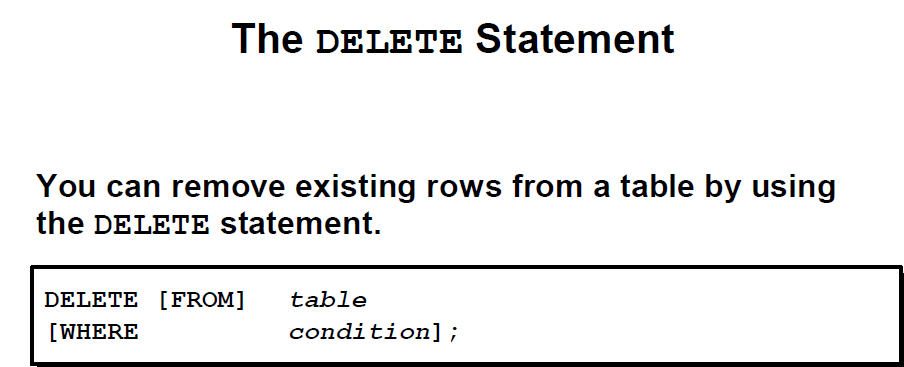
SET comm=com +100;

UPDATE ap\_emptest

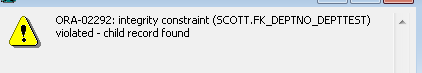
SET comm=nvl(comm,0)+100 , ename=INITCAP(ename);

SELECT \* FROM ap\_emptest;

commit;



DELETE FROM depttest;

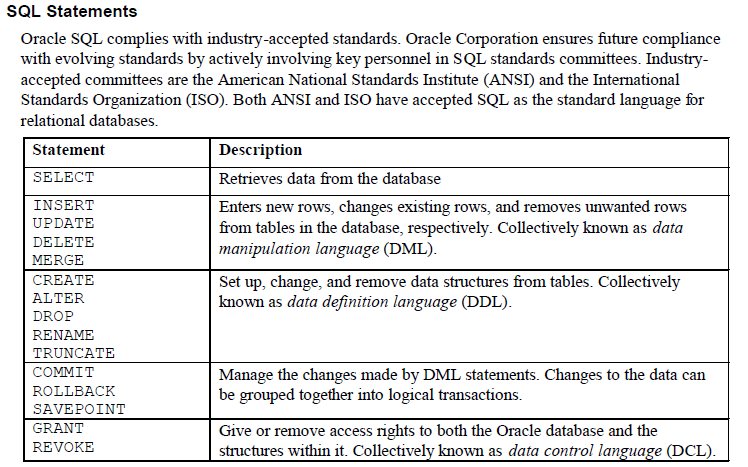


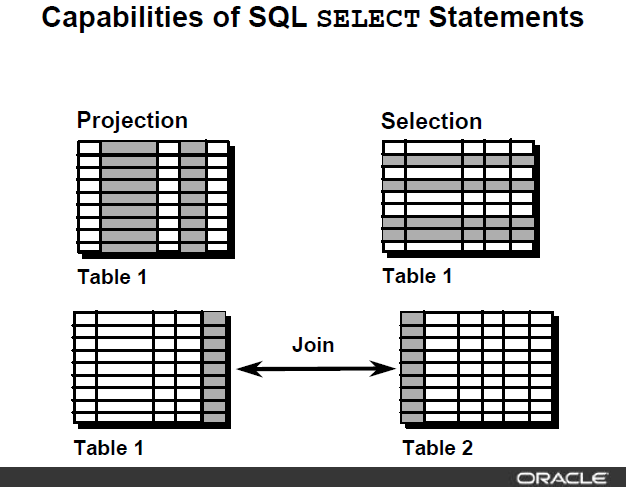
DELETE FROM ap\_emptest

WHERE deptno=30;

DELETE FROM ap\_emptest;

SELECT COUNT(\*) FROM ap\_emptest;





**Basic SELECT structure**

**SELECT < column list, expressions, literals>**

**FROM <table list>**

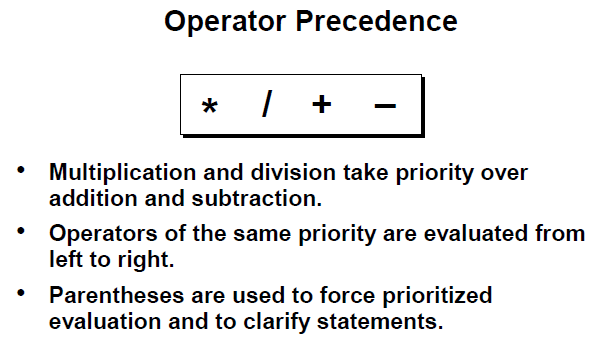
**WHERE <filter conditions with AND/OR/NOT logical operators)**

**GROUP BY <column list for aggregate functions COUNT/SUM/MIN/MAX/AVG etc.>**

**HAVING <filter conditions for grouping results>**

**ORDER BY <column list for sorting result set>**

**(SELECT and FORM clause are mandatory, all other clauses are optional and to be used as required by query result)**



SELECT \*

FROM ap\_emp;

SELECT empno, ename, job, sal

FROM ap\_emp;

SELECT empno "Employee Number", ename as name , job, sal "Monthly Salary USD"

FROM ap\_emp;

SELECT ename||' was hired on '||hiredate||' and current monthly salary is USD'||sal

FROM ap\_emp;

SELECT A.ename, A.sal, A.sal+100, A.comm, A.comm+10

from ap\_emp A;

SELECT A.ename, A.sal, A.sal+100, A.comm, nvl(A.comm,0)+10

FROM ap\_emp A;

SELECT DISTINCT job

FROM ap\_emp;

SELECT DISTINCT job,deptno

FROM ap\_emp;

SELECT DISTINCT job,deptno

FROM ap\_emp

ORDER BY job, deptno;

SELECT DISTINCT job,deptno

FROM ap\_emp

ORDER BY deptno, job;

SELECT DISTINCT job,deptno

FROM ap\_emp

ORDER BY deptno, job desc;

SELECT ename, sal

FROM ap\_emp

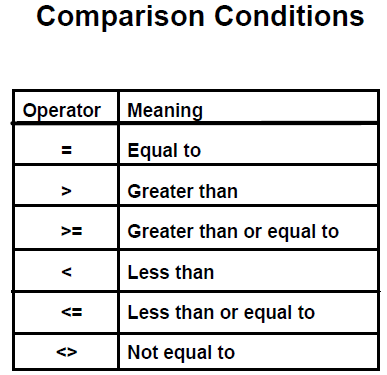
ORDER BY 2;

SELECT ename, sal

FROM ap\_emp

ORDER BY 2 desc, 1;

**WHERE clause (SELECTION of results - Limiting result sets based upon the conditions)**



SELECT \*

FROM ap\_emp

WHERE sal>9000;

SELECT \*

FROM ap\_emp

WHERE sal<9000;

SELECT ename, sal AS salary

FROM ap\_emp

WHERE sal>=9000

ORDER BY salary;

SELECT ename, deptno

FROM ap\_emp

WHERE deptno=10

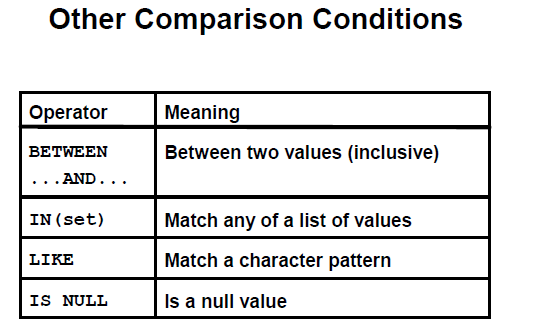
ORDER BY 1;

SELECT ename, deptno

FROM ap\_emp

WHERE deptno<>10

ORDER BY 2, 1;



SELECT ename, sal

FROM ap\_emp

WHERE sal BETWEEN 4500 AND 9000

ORDER BY sal;

SELECT ename, deptno

FROM ap\_emp

WHERE deptno IN (10,30)

ORDER BY 2;

SELECT ename, deptno

FROM ap\_emp

WHERE ename like 'A%'

ORDER BY 1;

SELECT ename, deptno

FROM ap\_emp

WHERE ename like '%N'

ORDER BY 1;

SELECT ename, deptno

FROM ap\_emp

WHERE ename like '%A%'

ORDER BY 1;

SELECT ename, sal, comm

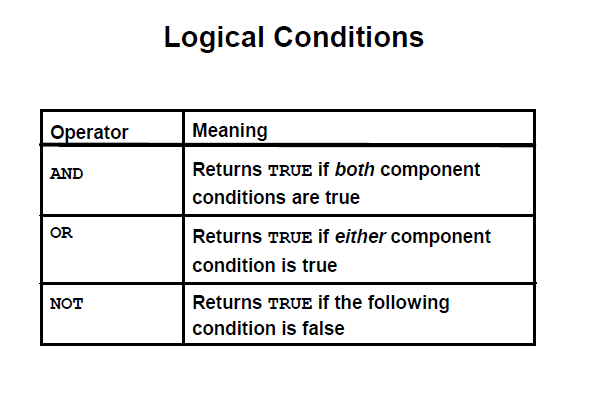
FROM ap\_emp

WHERE comm IS NULL;

SELECT ename, sal, comm

FROM ap\_emp

WHERE comm IS NOT NULL;



SELECT ename, sal, deptno

FROM ap\_emp

WHERE sal>5000 AND deptno=10;

SELECT ename,sal, deptno

FROM ap\_emp

WHERE sal>5000 OR deptno=10;

SELECT ename,sal, deptno, job

FROM ap\_emp

WHERE sal<5000 OR deptno=10 AND job='ANALYST';

SELECT ename,sal, deptno, job

FROM ap\_emp

WHERE (sal<5000 OR deptno=10) AND job='ANALYST';

SELECT ename,sal, deptno, job

FROM ap\_emp

WHERE sal<5000 OR deptno=10 OR job='ANALYST';

SELECT ename, sal

FROM ap\_emp

WHERE sal NOT BETWEEN 4500 AND 9000

ORDER BY sal;

SELECT ename,deptno

FROM ap\_emp

WHERE deptno NOT IN (10,30)

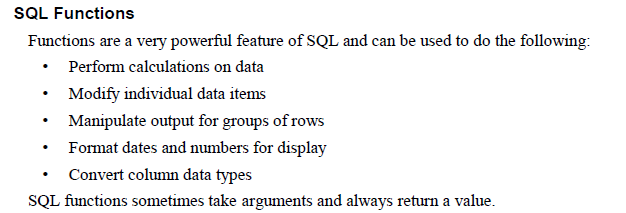
ORDER BY 2;

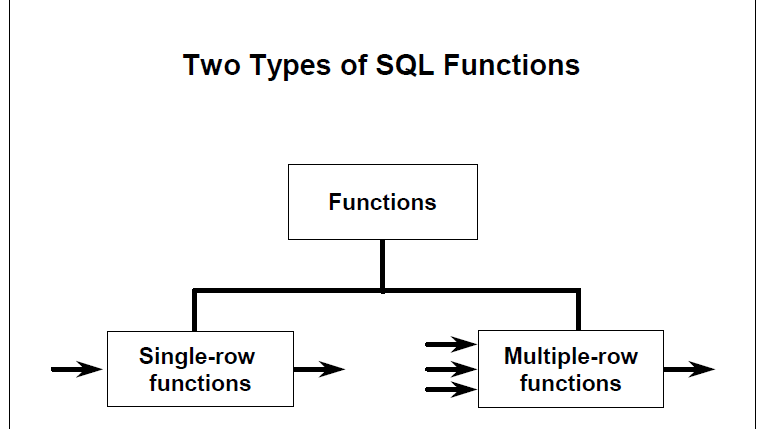
SELECT ename, deptno

FROM ap\_emp

WHERE ename NOT like 'A%'

ORDER BY 1;





**ORACLE SQL FUCTIONS**

[**https://docs.oracle.com/database/121/SQLRF/functions.htm#SQLRF006**](https://docs.oracle.com/database/121/SQLRF/functions.htm#SQLRF006)

SELECT ename, lower(ename), initcap(ename), substr(ename, 1,2), substr(ename, -3,2), substr(ename, -2), lower(substr( ename, 1,2))

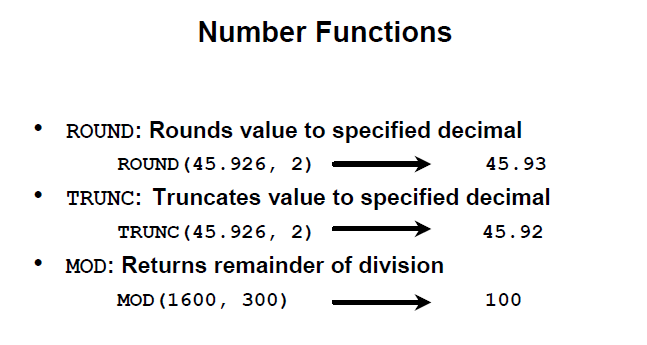
FROM ap\_emp

WHERE UPPER(JOB)='CLERK';

SELECT ename, LENGTH(ename), sal, LPAD(sal, 10,0)

FROM ap\_emp

WHERE UPPER(JOB)='CLERK';

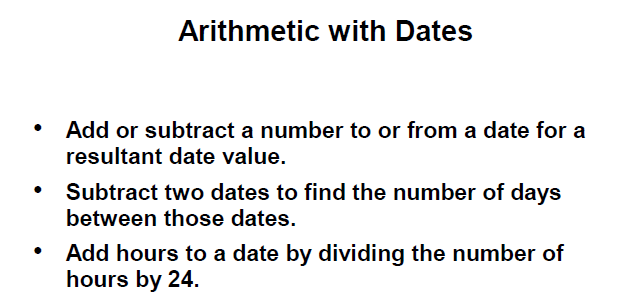
0

SELECT ROUND(45.926,2), ROUND( 45.926) FROM dual;

SELECT TRUNC(45.926,2), TRUNC( 45.926) FROM dual;

SELECT ename, sal, MOD(sal, 1000)

FROM ap\_emp;



SELECT sysdate

FROM DUAL;

SELECT sysdate+14

FROM DUAL;

SELECT SYSDATE-30

FROM DUAL;

SELECT ename, hiredate, sysdate-hiredate "No of days at work"

FROM ap\_emp;

SELECT ename, hiredate, (sysdate-hiredate)/365 "No of years at work"

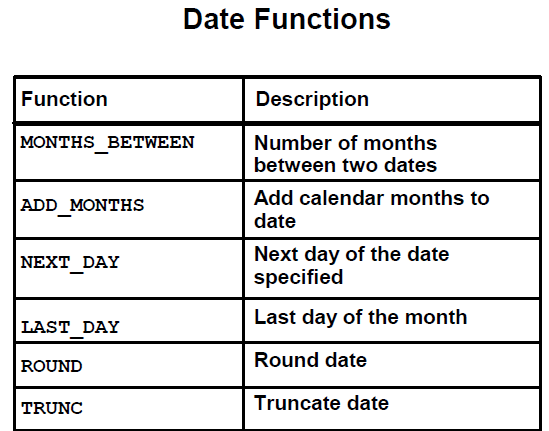
FROM ap\_emp;

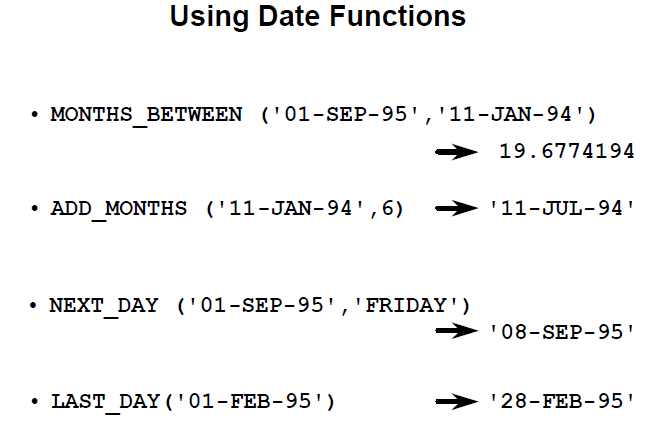
SELECT ename, hiredate, trunc( (sysdate-hiredate)/365 ) "No of years at work"

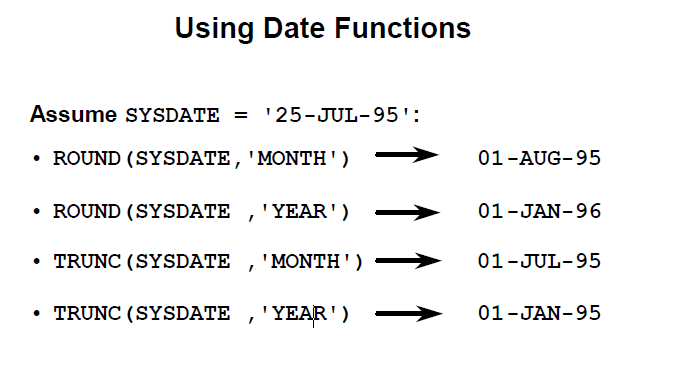
FROM ap\_emp;

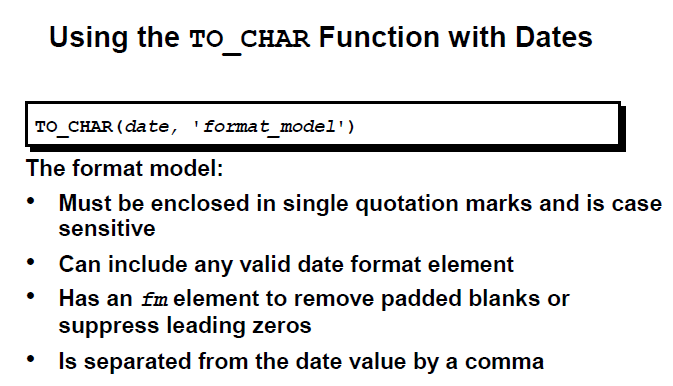
SELECT ename, hiredate, round( (sysdate-hiredate)/7) "No of weeks at work"

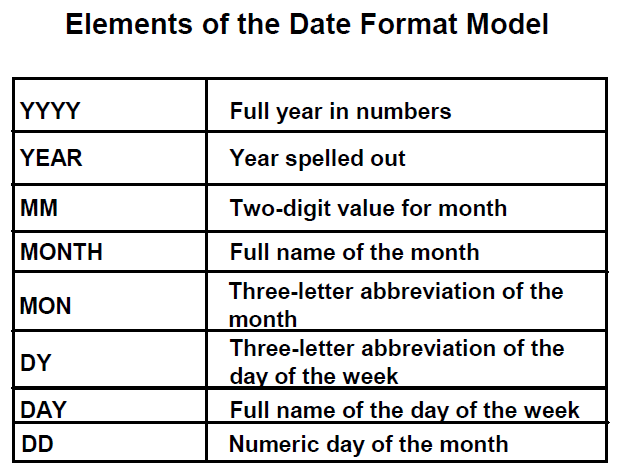
FROM ap\_emp;

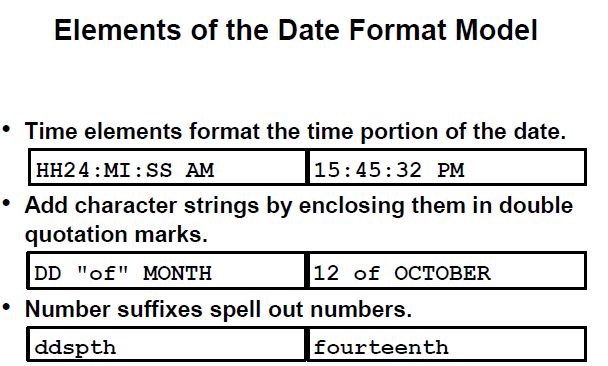












SELECT to\_char(sysdate, 'dd-mon-yy hh24:mi:ss ')

FROM dual;

SELECT to\_char(sysdate, 'dd-mon-yy hh:mi:ss AM')

FROM dual;

SELECT to\_char(sysdate, 'dd-MON-yyyy hh:mi:ss AM')

FROM dual;

SELECT to\_char(sysdate, 'dd-Month,yyyy hh:mi:ss AM')

FROM dual;

SELECT to\_char(sysdate, 'ddspth -Month,yyyy hh:mi:ss AM')

FROM dual;

SELECT to\_char(sysdate, 'Ddspth -Month,yyyy hh:mi:ss AM')

FROM dual;

SELECT to\_char(sysdate, 'Ddspth -Month "of" Year hh:mi:ss AM')

FROM dual;

SELECT to\_char (sysdate, 'DDD')

FROM dual;

SELECT to\_char (sysdate, 'DD')

FROM dual;

SELECT to\_char (sysdate, 'D')

FROM dual;

SELECT to\_char (sysdate, 'DY')

FROM dual;

SELECT to\_char (sysdate, 'Day')

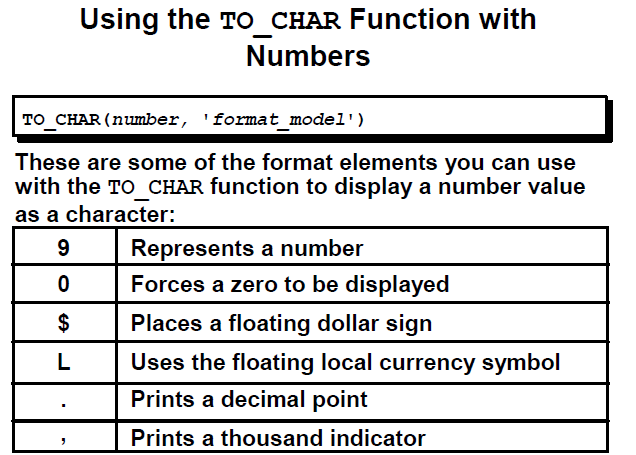
FROM dual;

SELECT to\_char (sysdate, 'WW')

FROM dual;

SELECT to\_char (sysdate, 'W')

FROM dual;



SELECT ename, sal, to\_char(sal, '99999'),

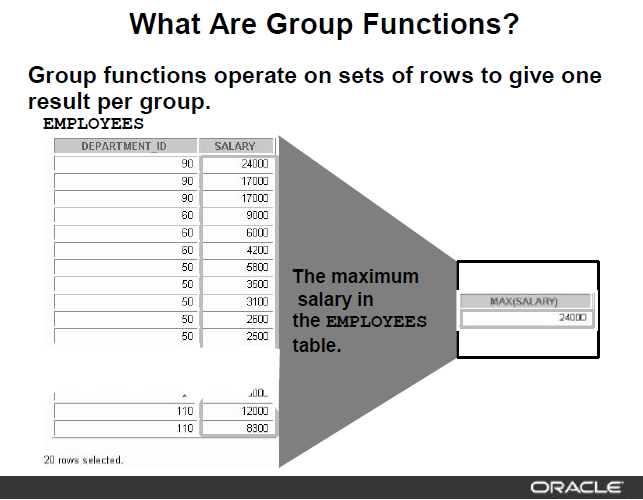
to\_char(sal, '09999.99'),

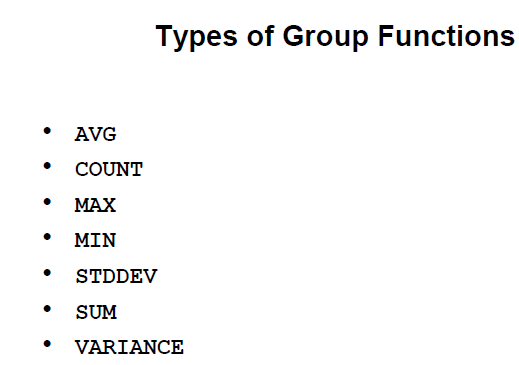
to\_char(sal, '$99999'),

to\_char(sal, '$99,999'),

to\_char(sal, 'L99999')

FROM ap\_emp;





SELECT COUNT(\*), MIN(sal), MAX(sal), AVG(sal), SUM(sal)

FROM ap\_emp

WHERE deptno=10;

SELECT MIN(hiredate), MAX(hiredate)

FROM ap\_emp

WHERE deptno=20;

SELECT COUNT( DISTINCT job)

FROM ap\_emp

WHERE deptno=20;

SELECT AVG(comm)

FROM ap\_emp;

SELECT AVG(NVL(comm,0))

from ap\_emp;

SELECT deptno, SUM(sal)

FROM ap\_emp

GROUP BY deptno

ORDER BY 2;

SELECT deptno, job, AVG(sal)

FROM ap\_emp

GROUP BY deptno, job

ORDER BY 1,2;

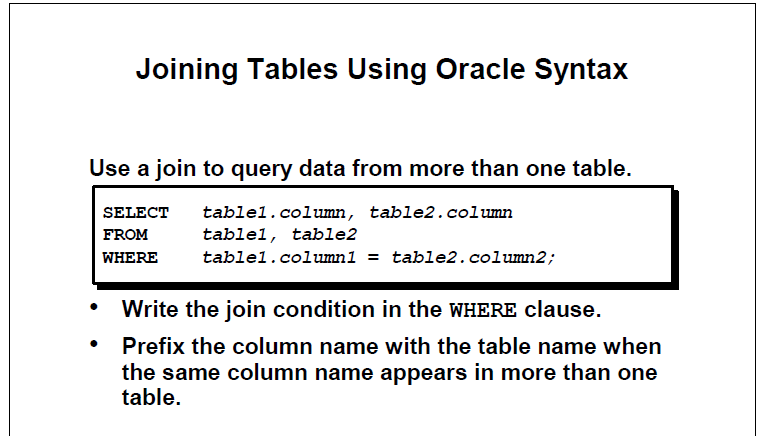
SELECT deptno, job, AVG(sal)

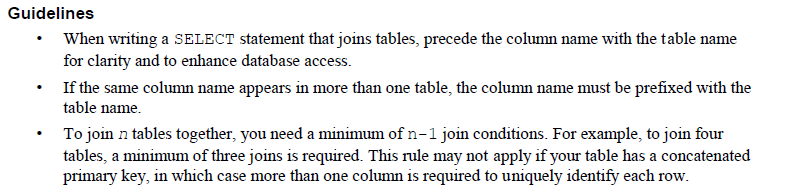
FROM ap\_emp

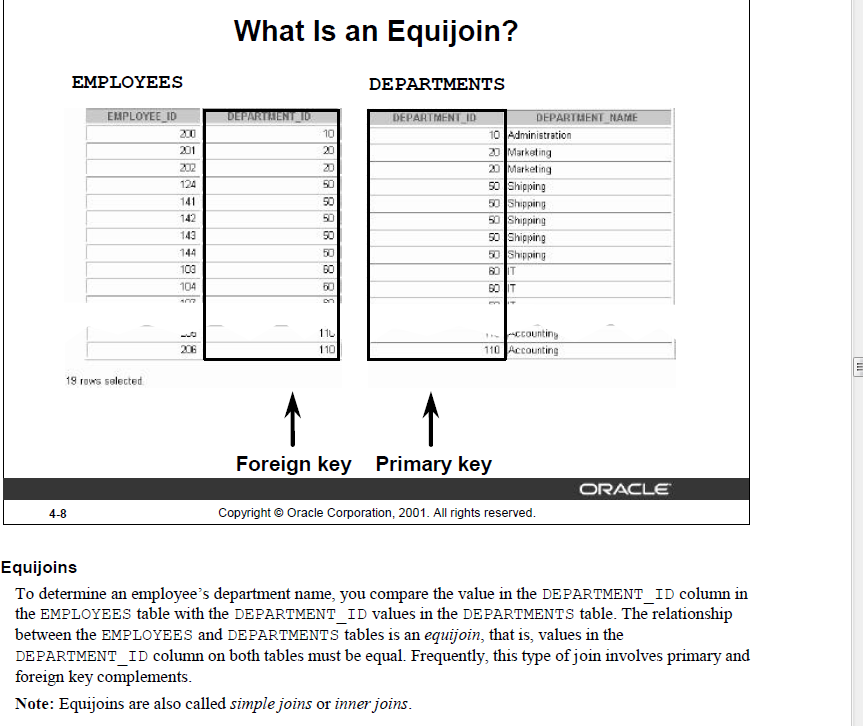
GROUP BY deptno, job

HAVING AVG(sal)>3000

ORDER BY 1,2;







SELECT a.empno, a.ename,a.deptno, b.dname

FROM ap\_emp a INNER JOIN ap\_dept b ON a.deptno=b.deptno

order by 3

;

SELECT a.empno, a.ename,a.deptno, b.dname

FROM ap\_emp a JOIN ap\_dept b ON a.deptno=b.deptno

order by 3

;

SELECT a.empno, a.projid,b.pname,a.hours

FROM ap\_proemp a JOIN ap\_project b ON a.projid=b.projid

WHERE a.hours>50

ORDER BY 1

;

SELECT a.empno, a.ename,a.deptno, b.dname,b.loc,c.hours, d.pname

FROM ap\_emp a JOIN ap\_dept b ON a.deptno=b.deptno JOIN ap\_proemp c ON a.empno=c.empno JOIN ap\_project d ON c.projid=d.projid

WHERE b.loc IN ('NEW YORK', 'DALLAS') AND c.hours>40

ORDER BY 1;

**NATURAL JOIN: when FK and PK column of same name, do not need to write join condition.**

SELECT empno, ename,deptno, dname – column alias on common column will return error.

FROM ap\_emp NATURAL JOIN ap\_dept;

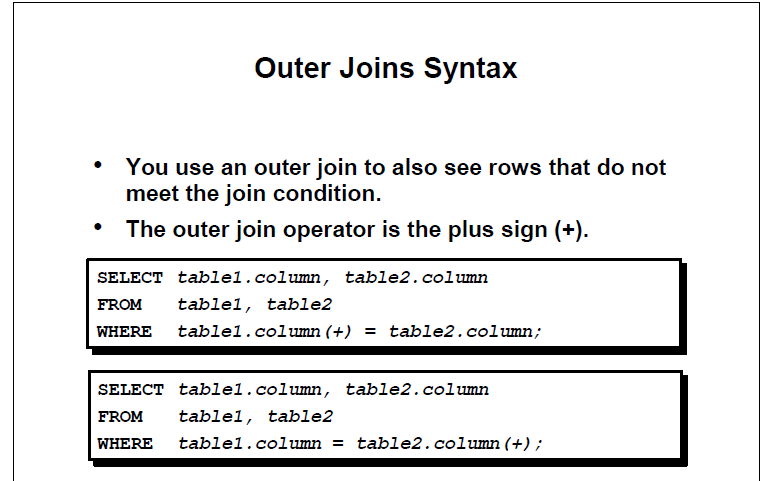
**NON-EQUIJOIN: Join based on other than equality operator**

SELECT a.empno, a.ename,a.sal,b.grade, b.losal, b.hisal

FROM ap\_emp a , ap\_salgrade b

WHERE a.sal between b.losal AND b.hisal

ORDER BY b.grade;



SELECT a.empno, a.ename,b.deptno,b.dname

FROM ap\_emp a RIGHT OUTER JOIN ap\_dept b ON a.deptno=b.deptno

ORDER BY a.deptno

;

SELECT b.empno, b.ename,a.deptno,a.dname

FROM ap\_dept a LEFT OUTER JOIN ap\_emp b ON a.deptno=b.deptno

ORDER BY b.deptno

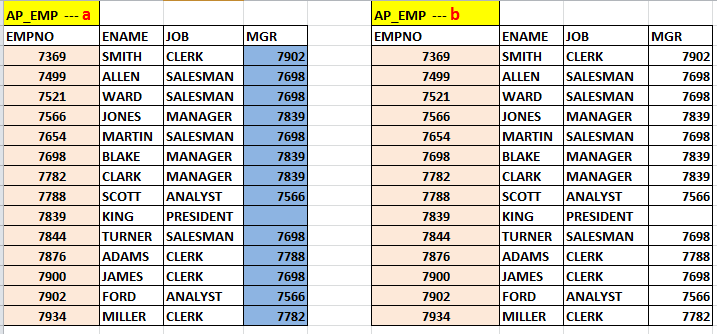
;

SELECT b.empno, b.ename,a.deptno,a.dname

FROM ap\_dept a FULL OUTER JOIN ap\_emp b ON a.deptno=b.deptno

ORDER BY b.deptno;

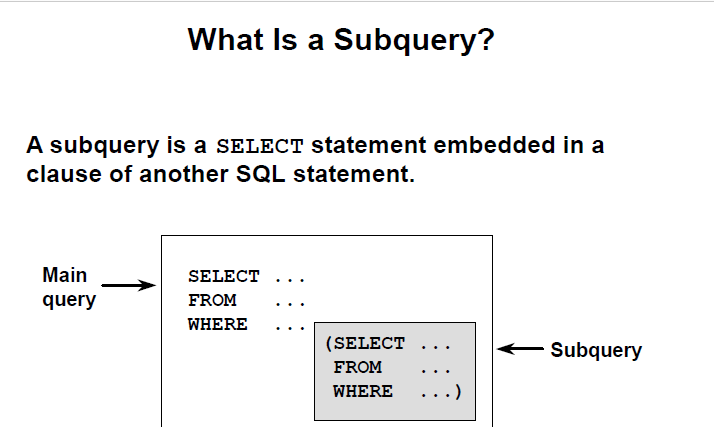
**SELF JOIN**

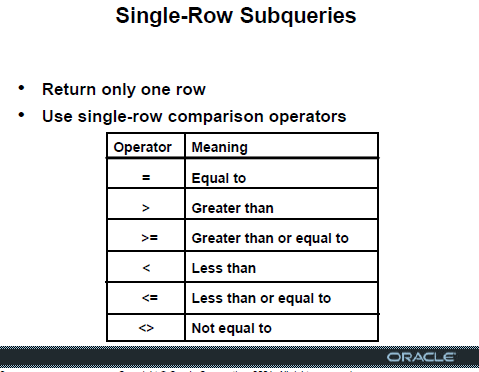


SELECT a.empno,a.ename,a.mgr, b.empno " Manager ID", b.ename "Manager Name"

FROM ap\_emp a JOIN ap\_emp b ON a.mgr=b.empno;

**USING SUBQUERY**





SELECT ename, sal

FROM ap\_emp

WHERE sal>( SELECT sal FROM ap\_emp WHERE ename='ALLEN');

SELECT ename, job

FROM ap\_emp

WHERE deptno=(SELECT deptno FROM ap\_dept WHERE loc='NEW YORK');

SELECT ename, job,sal

FROM ap\_emp

WHERE sal>(SELECT avg(sal) FROM ap\_emp);

SELECT ename, job,sal

FROM ap\_emp

WHERE sal>(SELECT avg(sal) FROM ap\_emp) AND

deptno= (SELECT deptno FROM ap\_dept where loc ='NEW YORK')

SELECT job, AVG(sal)

FROM ap\_emp

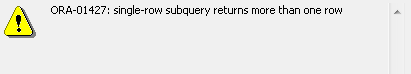
GROUP BY job

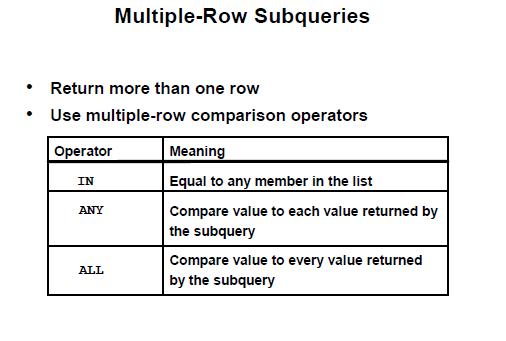
HAVING AVG(sal)=(SELECT MIN(AVG(sal)) FROM ap\_emp GROUP BY job);

SELECT ename, sal

FROM ap\_emp

WHERE sal > ( SELECT avg(sal) FROM ap\_emp GROUP BY deptno);





SELECT ename, sal

FROM ap\_emp

WHERE sal > ALL ( SELECT avg(sal) FROM ap\_emp GROUP BY deptno);

SELECT ename, sal

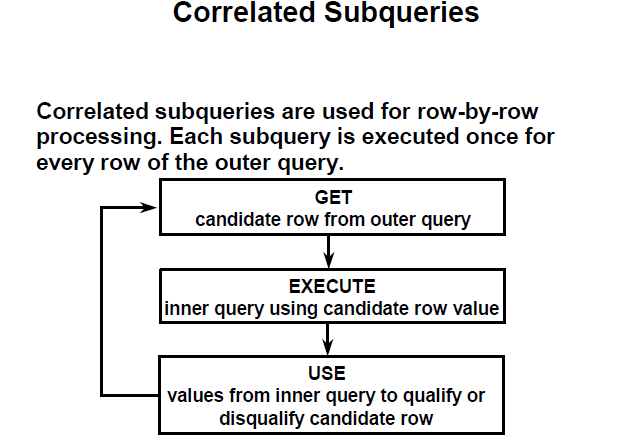
FROM ap\_emp

WHERE sal > ANY ( SELECT avg(sal) FROM ap\_emp GROUP BY deptno);

SELECT ename, sal

FROM ap\_emp

WHERE sal IN ( SELECT avg(sal) FROM ap\_emp GROUP BY deptno);

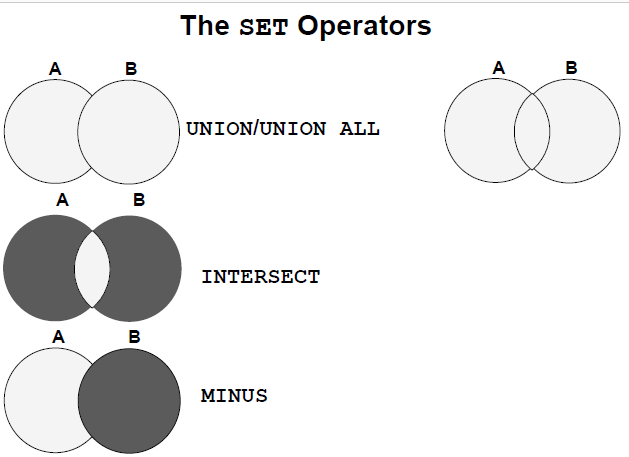


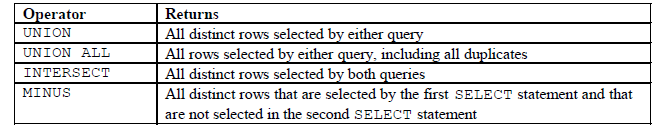
**List employees whose salary is higher than their respective department’s average salary**

SELECT a.empno,a.deptno, a.sal

FROM ap\_emp a

WHERE a.sal > (SELECT AVG(sal) deptavg FROM ap\_emp b WHERE a.deptno=b.deptno);





SELECT empno, ename, deptno,job

FROM ap\_emp

WHERE deptno in (20,30)

**UNION**

SELECT empno, ename,deptno, job

FROM ap\_emp

WHERE job='ANALYST'

ORDER BY deptno,job;

SELECT empno, ename, deptno,job

FROM ap\_emp

WHERE deptno in (20,30)

**UNION ALL**

SELECT empno, ename,deptno, job

FROM ap\_emp

WHERE job='ANALYST'

ORDER BY deptno,job;

SELECT empno, ename, deptno,job

FROM ap\_emp

WHERE deptno in (20,30)

**INTERSECT**

SELECT empno, ename,deptno, job

FROM ap\_emp

WHERE job='ANALYST'

ORDER BY deptno,job;

SELECT empno, ename, deptno,job

FROM ap\_emp

WHERE deptno in (20,30)

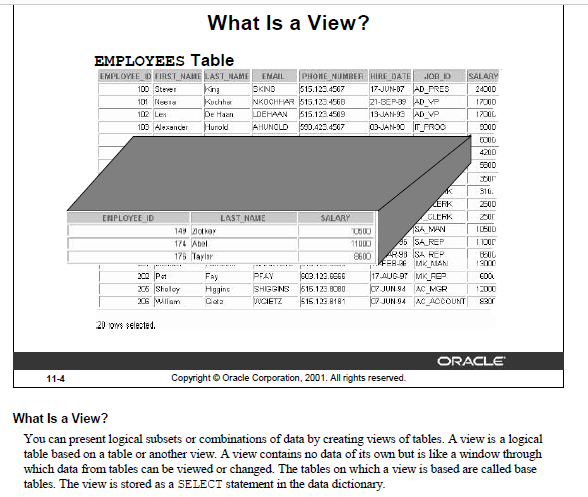
**MINUS**

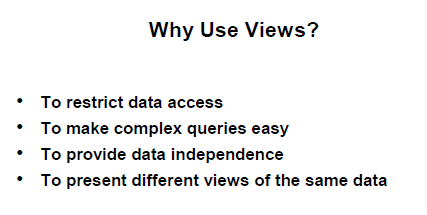
SELECT empno, ename,deptno, job

FROM ap\_emp

WHERE job='ANALYST'

ORDER BY deptno,job;





**== SIMPLE, SINGLE TABLE VIEW**

create or replace view testview

as

select \* from ap\_emp

where deptno=30

;

SELECT \* FROM testview;

update testview

set sal=sal+100

where empno=7499;

SELECT \* FROM testview;

**== READ ONLY VIEW**

create or replace view testview

as

select \* from ap\_emp

where deptno=30

with READ ONLY;

SELECT \* FROM testview;

update testview

set sal=sal+100

where empno=7499;

**== VIEW WITH CHECK OPTION**

create or replace view testview

as

select \* from ap\_emp

where deptno=30

with CHECK OPTION;

SELECT \* FROM testview;

update testview

set sal=sal+100

where empno=7499;

UPDATE TESTVIEW

SET DEPTNO=20

WHERE EMPNO=7499;

== COMPLEX, MULTI TABLES / AGGREGATE functions

CREATE OR REPLACE VIEW dept30\_V

AS

SELECT a.empno "Employee Number", a.hiredate "Hire Date", a. sal "Monthly Salary", b.dname "Department Name", b. loc "Location"

FROM ap\_emp a, ap\_dept b

WHERE a.deptno=b.deptno AND

a.deptno=30 ;

SELECT \* FROM dept30\_V;

SELECT \* FROM USER\_VIEWS

WHERE VIEW\_NAME='DEPT30\_V';

CREATE OR REPLACE VIEW emp\_proj\_detail\_V

AS

SELECT a.empno AS Employee, a.ename AS name, a.hiredate AS HireDate, a. sal AS Salary, b.dname AS Department, c.pname AS Project, d.hours AS Hours

FROM ap\_emp a, ap\_dept b, ap\_project c, ap\_proemp d

WHERE a.deptno=b.deptno AND

a.empno=d.empno AND

d.projid=c.projid

ORDER BY 4;

SELECT \* FROM emp\_proj\_detail\_V;

SELECT name,sum(hours)

FROM ap\_emp\_proj\_detail\_V

GROUP BY name

ORDER by 2;

CREATE OR REPLACE VIEW dept\_summary\_V

(name,location,minsal,maxsal,avgsal)

AS

SELECT dname , loc, MIN(sal), MAX(sal) , AVG(sal)

FROM ap\_emp a, ap\_dept b

WHERE a.deptno=b.deptno

GROUP BY dname,loc;

SELECT \* FROM dept\_summary\_V;

**==== SUBQUERY in FROM CLAUSE**

* **List employee number, name, department, salary along with their respective department’s total number of employee, total salary, average salary, minimum, and maximum salary**

**SELECT a.empno, a.ename, a.deptno,sal, b.deptempcnt , b.depttotsal , trunc(b.deptavgsal) deptavgsal ,**

**b.deptminsal, b.deptmaxsal**

**FROM ap\_emp a, ( select deptno, count(\*) deptempcnt, sum(sal) depttotsal, avg(sal) deptavgsal, min(sal)**

**deptminsal, max(sal) deptmaxsal from ap\_emp group by deptno ) b**

**WHERE a.deptno=b.deptno ;**

**==== WITH CLAUSE [just like a temporary table that can be used multiple times in a query as needed**

**WITH DEPTVAL AS**

**( select deptno, count(\*) deptempcnt, sum(sal) depttotsal, avg(sal) deptavgsal, min(sal) deptminsal, max(sal) deptmaxsal from ap\_emp group by deptno)**

**SELECT a.empno, a.ename, a.deptno,sal, b.deptempcnt , b.depttotsal , trunc(b.deptavgsal) deptavgsal, b.deptminsal, b.deptmaxsal**

**FROM ap\_emp a, DEPTVAL b**

**WHERE a.deptno=b.deptno ;**

**== CASE statement [Conditional processing]**

**Increase employee’s salary by $500 if their salary is more than $5000, increase by $300 if their salary is over $3000, otherwise increase by $100.**

**select sal, case**

**when sal>5000 then sal+500**

**when sal>3000 then sal+300**

**else**

**sal+100**

**end “Salary Raise”**

**from ap\_emp;**

**update ap\_emp**

**set sal = case**

**when sal>5000 then sal+500**

**when sal>3000 then sal+300**

**else**

**sal+100**

**end ;**

**== RANK function for TOP-N queries**

* **List employee number and name of the employee who worked on highest number of projects**

**WITH PROJCNTRANK**

**AS**

**( select empno,count(projid) cnt,**

**rank () over (order by count(projid) desc) as myrank**

**from ap\_proemp**

**group by empno )**

**SELECT A.EMPNO, A.ENAME, B.CNT, B.MYRANK**

**FROM AP\_EMP A JOIN PROJCNTRANK B ON A.EMPNO=B.EMPNO AND**

**MYRANK<2;**

**-- Subquery in SELECT clause**

**-- List of depatment name, and total number of employyees in those department**

**select dname name,**

**(select count(\*)**

**from ap\_emp B**

**where A.deptno= B.deptno)**

**as Num\_Of\_Employees**

**from ap\_dept A ;**

**-- Subquery in HAVING clause**

**-- Department wise Total Salary for those departments which has total salary more than that of department 30**

**select deptno,sum(sal) Total\_Salary**

**from ap\_emp**

**group by deptno**

**having sum(sal) >= (select sum(sal) from ap\_emp where deptno=30);**