**CSE2007 DBMS LAB**

**SLOT: L39+L40**

**NAME - AMAN SAHU**

**REG. NO – 22BCE7224**

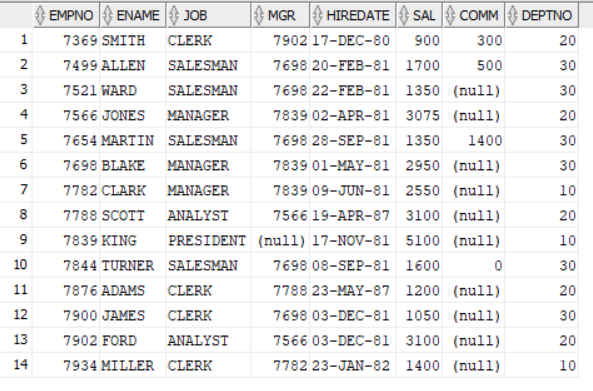
**EXPERIMENT NO.-6**

For 1 to 14 queries: Apply suitable revert statement,wherever applicable,to change the state of table asconsistent.

1. Update the salaries of employees by 100.

Update emp SET SAL = Sal +100;

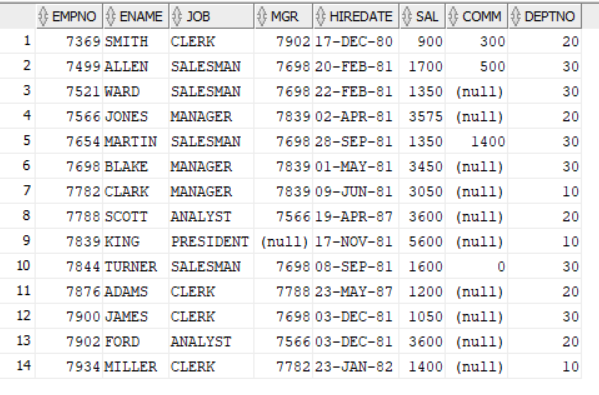
select \* from emp;



2. Update the salaries of employees by 500 whose having salary greater than 2000.

Update emp SET SAL = SAL + 500 WHERE SAL>2000;

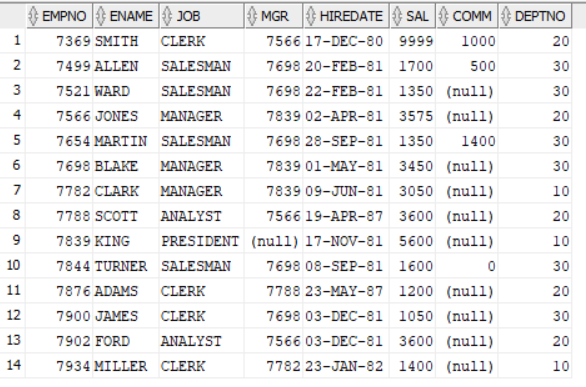
Select \* from emp;



3. Update employee data with salary=9999, commission = 1000, managerid = 7566 for the employee whose id = 7369.

Update emp SET SAL =9999,COMM=1000,MGR=7566 WHERE EMPNO=7369;

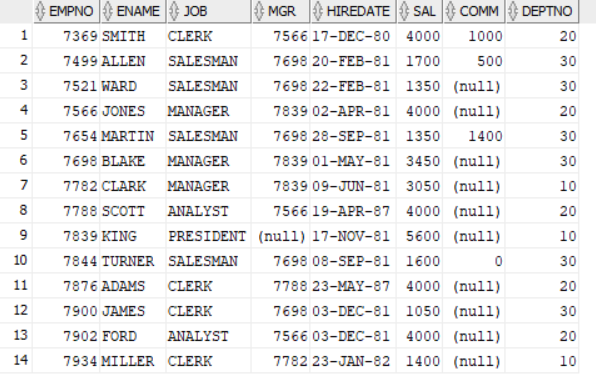
Select \* from emp;



4. Update the salaries of employees of department ‘20’to 4000.

Update emp SET SAL =4000 WHERE DEPTNO=20;

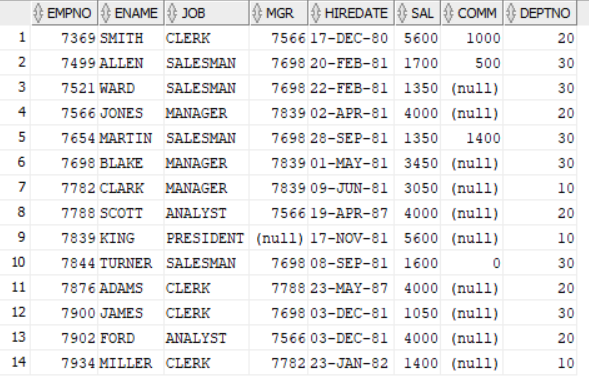
Select \* from emp;



5. Update the employee(7369) salary with max of all salaries.

Update emp SET SAL =(SELECT MAX(SAL) FROM emp) WHERE empno=7369;

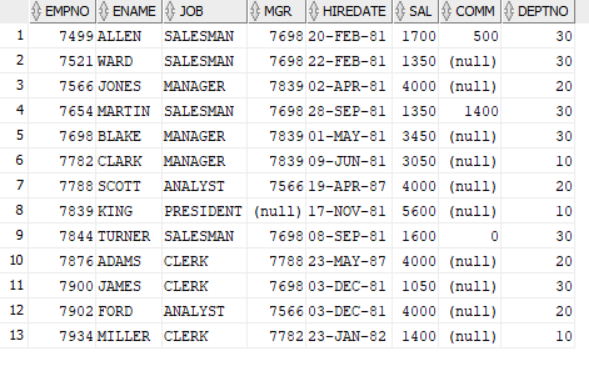
Select \* from emp;



6. Delete the employee data whose employee id is 7369.

DELETE FROM emp WHERE EMPNO=7369;

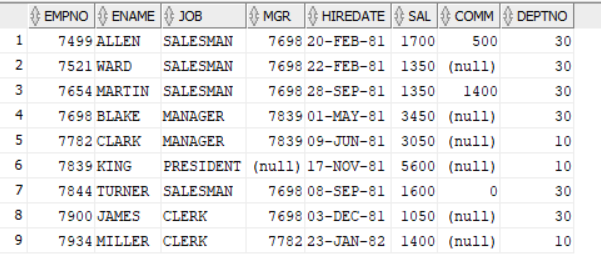
Select \* from emp;



7. Delete all employees data belongs to dept 20.

DELETE FROM emp WHERE DEPTNO=20;

Select \* from emp;

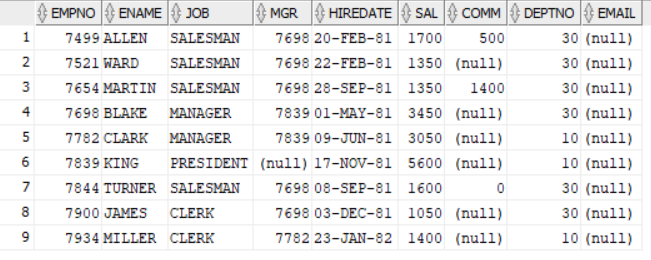


8. Apply truncate to delete the data and identify the difference between truncate and delete.

9. Modify the employee table to add email attribute of size 30.

ALTER TABLE EMP ADD email varchar2(30);

Select \* from emp;



10. Update few employees by adding emails to their entities.

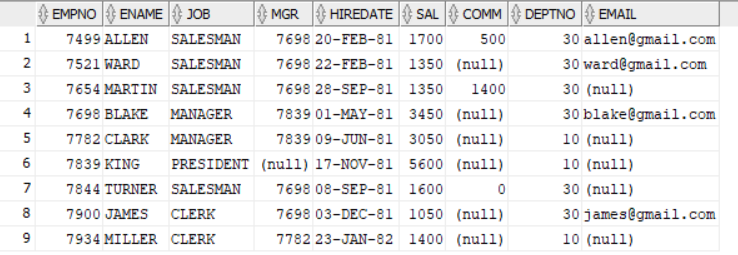
update EMP SET email='allen@gmail.com' WHERE EMPNO IN 7499;

update EMP SET email='ward@gmail.com' WHERE EMPNO IN 7521;

update EMP SET email='blake@gmail.com' WHERE EMPNO IN 7698;

update EMP SET email='james@gmail.com' WHERE EMPNO IN 7900;

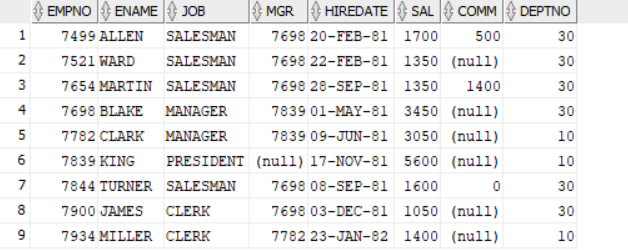
Select \* from emp;



11. Remove the added email attribute from the employee table.

ALTER TABLE emp drop Column email;

Select \* from emp;



12. Modify the employee name field size to support 10 more characters.

ALTER TABLE emp MODIFY ename varchar2(20);

Select \* from emp;

13. Add the following columns years, months, days to refer the experience in the company. Calculate the experience of each employee, and update these three fields for each employee.Display the employee data.

ALTER TABLE emp ADD (YEARS INT, MONTHS INT, DAYS INT);

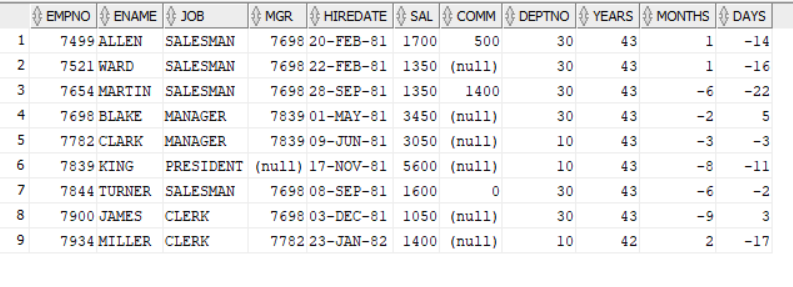
UPDATE emp SET

YEARS = EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM HIREDATE),

MONTHS = EXTRACT(MONTH FROM SYSDATE) - EXTRACT(MONTH FROM HIREDATE),

DAYS = EXTRACT(DAY FROM SYSDATE) - EXTRACT(DAY FROM HIREDATE);

SELECT \* FROM emp;



Help:



14. Drop the above three columns from the employees table.Consider the following relations(Data is attached as files in the assignment. No Duplicate records in the output):

ALTER TABLE employees

DROP COLUMN years,

DROP COLUMN months,

DROP COLUMN days;

15. Find the names of all Juniors (level = JR) who are enrolled in a class taught by I. Teach.

SELECT s.sname

FROM student s

JOIN enrolled e ON s.snum = e.snum

JOIN class c ON e.cname = c.name

JOIN faculty f ON c.fid = f.fid

WHERE s.standing = 'JR' AND f.fname = 'I. Teach';

16. Find the age of the oldest student who is either a History major or enrolled in a coursetaught by I. Teach.

SELECT MAX(age)

FROM student s

WHERE s.major = 'History' OR

s.snum IN (

SELECT e.snum

FROM enrolled e

JOIN class c ON e.cname = c.name

JOIN faculty f ON c.fid = f.fid

WHERE f.fname = 'I. Teach'

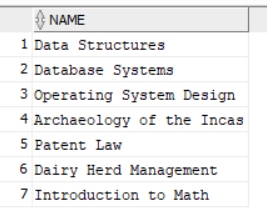
);



17. Find the names of all classes that either meet in room R128 or have five or more students enrolled.

SELECT c.name FROM class c WHERE c.room = 'R128' OR

(SELECT COUNT(\*) FROM enrolled e WHERE e.cname = c.name) >= 5;



18. Find the names of all students who are enrolled in two classes that meet at the same time.

SELECT DISTINCT s.sname

FROM student s

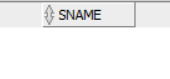
JOIN enrolled e1 ON s.snum = e1.snum

JOIN class c1 ON e1.cname = c1.name

JOIN enrolled e2 ON s.snum = e2.snum

JOIN class c2 ON e2.cname = c2.name

WHERE c1.meets\_at = c2.meets\_at AND c1.name <> c2.name;



19. Find the names of faculty members who teach in every room in which some class istaught.

SELECT f.fname

FROM faculty f

WHERE NOT EXISTS (

SELECT DISTINCT room

FROM class

WHERE room NOT IN (

SELECT c.room

FROM class c

WHERE c.fid = f.fid

)

);



20. Find the names of faculty members for whom the combined enrollment of the coursesthat they teach is less than five.

SELECT f.fname

FROM faculty f

WHERE (

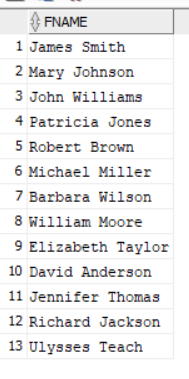
SELECT COUNT(\*)

FROM enrolled e

JOIN class c ON e.cname = c.name

WHERE c.fid = f.fid

) < 5;

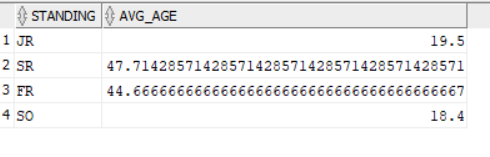


21. Print the level and the average age of students for that level, for each level.

SELECT standing, AVG(age) AS avg\_age

FROM student

GROUP BY standing;



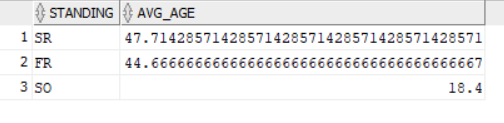
22. Print the level and the average age of students for that level, for all levels except JR.

SELECT standing, AVG(age) AS avg\_age

FROM student

WHERE standing != 'JR'

GROUP BY standing;



23. For each faculty member that has taught classes only in room R128, print the facultymember’s name and the total number of classes she or he has taught.

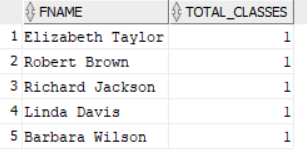
SELECT f.fname, COUNT(\*) AS total\_classes

FROM faculty f

JOIN class c ON f.fid = c.fid

WHERE c.room = 'R128'

GROUP BY f.fname



24.Find the names of students enrolled in the maximum number of classes.

SELECT s.sname

FROM student s

JOIN enrolled e ON s.snum = e.snum

GROUP BY s.sname

HAVING COUNT(DISTINCT e.cname) = (

SELECT MAX(classes\_count)

FROM (

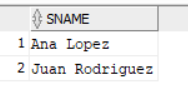
SELECT COUNT(DISTINCT e2.cname) AS classes\_count

FROM enrolled e2

GROUP BY e2.snum

)

);



25. Find the names of students not enrolled in any class.

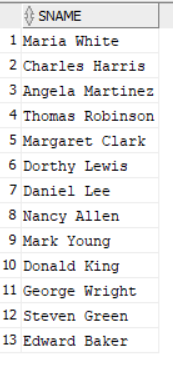
SELECT s.sname FROM student s WHERE NOT EXISTS (

SELECT 1

FROM enrolled e

WHERE e.snum = s.snum

);



26. For each age value that appears in Students, find the level value that appears most often.For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

SELECT age, standing

FROM (

SELECT age, standing, COUNT(\*) AS count,

ROW\_NUMBER() OVER (PARTITION BY age ORDER BY COUNT(\*) DESC) as rn

FROM student

GROUP BY age, standing

)

WHERE rn = 1;

