1.select b.book\_id, b.title, b.publisher\_name, a.author\_name, c.no\_of\_copies, l.branch\_id from book b, book\_authors a, book\_copies c, library\_branch l where b.book\_id=a.book\_id and b.book\_id=c.book\_id and l.branch\_id=c.branch\_id;

2.select card\_no from book\_lending where date\_out between '01-jan-2017' and '01-jun-2017' group by card\_no having count(\*)>3;

3.select \* from book; select \* from book\_copies; delete from book where book\_id='03';

4.create table bookpart (partition by range (pub\_year) partition p1 values less than(2018), partition p2 values less than (maxvalue) as select \* from book); select table\_name,partition\_name from user\_tab\_partitions; select \* from bookpart partition (p1); select \* from bookpart partition (p2);

5. create view bc as select b.book\_id,c.title,b.branch\_id, (b.no\_of\_copies-(select count(\*) from book\_lending where b.book\_id=book\_id and b.branch\_id=branch\_id)) as no\_copy from book\_copies b,book c where b.book\_id=c.book\_id;

1.select grade, count (distinct customer\_id) as no\_of\_customer from customer group by grade having grade > (select avg(grade) from customer where city='bangalore');

2.select salesman\_id, name from salesman s where ((select count (\*) from customer where salesman\_id=s.salesman\_id)>1);

3.select s.salesman\_id, s.city from salesman s where exists (select city from customer where s.city=city and s.salesman\_id=salesman\_id) union select salesman\_id,'no match of cities' from salesman s where not exists (select city from customer where s.city=city and s.salesman\_id=salesman\_id);

4.select distinct s.salesman\_id,s.ord\_date from orders s where (select sum(purchase\_amt) from orders where salesman\_id=s.salesman\_id and ord\_date=s.ord\_date and s.customer\_id=customer\_id) =(select max(sum(purchase\_amt)) from orders s1 where s1.ord\_date=s.ord\_date group by s1.ord\_date,s1.salesman\_id,s1.customer\_id);

5.delete from salesman where salesman\_id=1000; select \* from salesman; select \* from customer; select \* from orders;

1.SELECT M.MOV\_TITLE FROM MOVIES M,DIRECTOR D WHERE M.DIR\_ID=D.DIR\_ID AND D.DIR\_NAME = 'HITCHCOCK';

2.SELECT MOV\_TITLE FROM MOVIES M, MOVIE\_CAST MV WHERE M.MOV\_ID=MV.MOV\_ID AND ACT\_ID IN (SELECT ACT\_ID FROM MOVIE\_CAST GROUP BY ACT\_ID HAVING COUNT (ACT\_ID)>=1) GROUP BY MOV\_TITLE HAVING COUNT (\*)>1;

3.SELECT A.ACT\_NAME FROM ACTOR A JOIN MOVIE\_CAST C ON A.ACT\_ID=C.ACT\_ID JOIN MOVIES M ON C.MOV\_ID=M.MOV\_ID WHERE M.MOV\_YEAR NOT BETWEEN 2000 AND 2015;

4.SELECT MOV\_TITLE, MAX(REV\_STARS) INNER JOIN RATING USING (MOV\_ID) GROUP BY MOV\_TITLE HAVING MAX (REV\_STARS)>0 ORDER BY MOV\_TITLE;

5.UPDATE RATING SET REV\_STARS=5 WHERE MOV\_ID IN (SELECT M.MOV\_ID FROM MOVIES M,DIRECTOR D WHERE M.DIR\_ID=D.DIR\_ID AND D.DIR\_NAME = 'STEVEN SPIELBERG'); BEFORE UPDATING AFTER UPDATING

SQL> SELECT \* FROM RATING;

1.SELECT S.\*, SS.SEM, SS.SEC FROM STUDENT S, SEMSEC SS, CLASS C WHERE S.USN = C.USN AND SS.SSID = C.SSID AND SS.SEM = 4 AND SS.SEC='C';

2.SELECT SS.SEM, SS.SEC, S.GENDER, COUNT (S.GENDER) AS COUNT FROM STUDENT S, SEMSEC SS, CLASS C WHERE S.USN = C.USN AND SS.SSID = C.SSID GROUP BY SS.SEM, SS.SEC, S.GENDER ORDER BY SEM;

3.CREATE VIEW STU\_TEST1\_MARKS\_VIEW AS SELECT TEST1, SUBCODE FROM IAMARKS WHERE USN = '1BI15CS101';

4.UPDATE IAMARKS SET FINALIA=((TEST1+TEST2+TEST3)- LEAST(TEST1,TEST2,TEST3))/2;

5.SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER, IA.SUBCODE, (CASE WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING' WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE' ELSE 'WEAK' END) AS CAT FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB WHERE S.USN = IA.USN AND SS.SSID = IA.SSID AND SUB.SUBCODE = IA.SUBCODE AND SUB.SEM = 8 ;

1.SELECT DISTINCT PNo FROM PROJECT WHERE PNo IN( (SELECT P.PNo FROM PROJECT P,DEPARTMENT D ,EMPLOYEE E WHERE P.DNo=D.DNo AND D.MgrSSN=E.SSN AND E.Name='Scott') UNION (SELECT W.PNo FROM WORKS\_ON W, EMPLOYEE E WHERE E.SSN=W.SSN AND E.Name='Scott'));

2.SELECT E.Name,1.1\* E.Salary AS Increased\_salary FROM EMPLOYEE E, WORKS\_ON W, PROJECT P WHERE E.SSN=W.SSN AND W.PNo=P.PNo AND P.PName='IoT';

3.SELECT SUM (E. Salary) AS TOTAL\_SALARY,MAX(E. Salary) AS MAX\_SALARY,MIN(E. Salary) AS MIN\_SALARY,AVG(E. Salary) AS AVG\_SALARY FROM EMPLOYEE E, DEPARTMENT D WHERE E. DNo= D. DNo AND D.DName='Accounts';

4.SELECT E.Name FROM EMPLOYEE E WHERE NOT EXISTS((SELECT PNo FROM PROJECT WHERE DNo='D5') MINUS (SELECT W.PNo FROM WORKS\_ON W WHERE E.SSN=W.SSN));

5.SELECT D.DNo,COUNT(\*) FROM EMPLOYEE E, DEPARTMENT D WHERE E.DNo= D.DNo AND E.Salary>600000 GROUP BY D.DNo HAVING COUNT(\*)>=5;