```
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;