1.

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-JUL-2017’
GROUP BY CARD_NO HAVING COUNT (*)>3;

3.DELETE FROM BOOK
WHERE BOOK_ID=3;

4.CREATE VIEW V_PUBLICATION AS
SELECT PUB_YEAR
FROM BOOK;

5.CREATE VIEW V_BOOKS AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH L
WHERE B.BOOK_ID=C.BOOK_ID
AND C.BRANCH_ID=L.BRANCH_ID;

2.

1.SELECT GRADE, COUNT (DISTINCT CUSTOMER_ID) FROM CUSTOMER1
GROUP BY GRADE
HAVING GRADE > (SELECT AVG(GRADE) FROM CUSTOMER1
WHERE CITY='BANGALORE');

2.SELECT SALESMAN_ID, NAME FROM SALESMAN A
WHERE 1 < (SELECT COUNT (*)
FROM CUSTOMER1
WHERE SALESMAN_ID=A.SALESMAN_ID);

3.SELECT SALESMAN.SALESMAN_ID, NAME, CUST_NAME, COMMISSION FROM SALESMAN, CUSTOMER1
WHERE SALESMAN.CITY = CUSTOMER1.CITY UNION
SELECT SALESMAN_ID, NAME, 'NO MATCH', COMMISSION FROM SALESMAN
WHERE NOT CITY = ANY (SELECT CITY
FROM CUSTOMER1) ORDER BY 2 DESC

4.CREATE VIEW ELITSALESMAN AS
SELECT B.ORD_DATE, A.SALESMAN_ID, A.NAME FROM SALESMAN A, ORDERS B
WHERE A.SALESMAN_ID = B.SALESMAN_ID
AND B.PURCHASE_AMT=(SELECT MAX (PURCHASE_AMT)
FROM ORDERS C
WHERE C.ORD_DATE = B.ORD_DATE);

5.DELETE FROM SALESMAN
WHERE SALESMAN_ID=1000;

3.

1.SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID

```
FROM DIRECTOR
WHERE 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 ACT_NAME, MOV_TITLE, MOV_YEAR
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;
```
OR
```
SELECT A.ACT_NAME, A.ACT_NAME, C.MOV_TITLE, C.MOV_YEAR FROM
ACTOR A, MOVIE_CAST B, MOVIES C WHERE A.ACT_ID=B.ACT_ID
AND B.MOV_ID=C.MOV_ID
AND C.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```
4.
```
SELECT MOV_TITLE, MAX (REV_STARS)
FROM MOVIES
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 MOV_ID FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = 'STEVEN
SPIELBERG'));
```
4.

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 WHERES.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 = '4BD13CS091';
```
4.
```
CREATE OR REPLACE PROCEDURE AVGMARKS IS
CURSOR C_IAMARKS IS
SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,
GREATEST(TEST3,TEST2) AS C
FROM IAMARKS
```

```
WHERE FINALIA IS NULL
FOR UPDATE;
C_A NUMBER;
C_B NUMBER;
C_C NUMBER;
C_SM NUMBER;
C_AV NUMBER;
BEGIN
OPEN C_IAMARKS;
LOOP
FETCH C_IAMARKS INTO C_A, C_B, C_C;
EXIT WHEN C_IAMARKS%NOTFOUND;
--DBMS_OUTPUT.PUT_LINE(C_A || ' ' || C_B || ' ' || C_C); IF (C_A != C_B) THEN
C_SM:=C_A+C_B;
ELSE
C_SM:=C_A+C_C;
END IF;
C_AV:=C_SM/2;
--DBMS_OUTPUT.PUT_LINE('SUM = '||C_SM);
--DBMS_OUTPUT.PUT_LINE('AVERAGE = '||C_AV);
UPDATE IAMARKS SET FINALIA=C_AV WHERE CURRENT OF C_IAMARKS;
END LOOP;
CLOSE C_IAMARKS;
END;
/
SQL> SELECT * FROM IAMARKS;
BEGIN
AVGMARKS;
END;
SQL> SELECT * FROM IAMARKS;
```

5.SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER, (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;

5.

1.(SELECT DISTINCT P.PNO
FROM PROJECT P, DEPARTMENT D, EMPLOYEE E WHERE E.DNO=D.DNO
AND D.MGRSSN=E.SSN
AND E.LNAME=’SCOTT’)
UNION
(SELECT DISTINCT P1.PNO
FROM PROJECT P1, WORKS_ON W, EMPLOYEE E1 WHERE P1.PNO=W.PNO
AND E1.SSN=W.SSN
AND E1.LNAME=’SCOTT’);

2.(SELECT DISTINCT P.PNO

FROM PROJECT P, DEPARTMENT D, EMPLOYEE E WHERE E.DNO=D.DNO
AND D.MGRSSN=E.SSN
AND E.LNAME='SCOTT')
UNION
(SELECT DISTINCT P1.PNO
FROM PROJECT P1, WORKS_ON W, EMPLOYEE E1 WHERE P1.PNO=W.PNO
AND E1.SSN=W.SSN
AND E1.LNAME='SCOTT');

3.SELECT SUM (E.SALARY), MAX (E.SALARY), MIN (E.SALARY), AVG (E.SALARY)
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.DNO=D.DNO
AND D.DNAME='ACCOUNTS';

4.SELECT E.FNAME, E.LNAME
FROM EMPLOYEE E
WHERE NOT EXISTS ((SELECT PNO
FROM PROJECT
WHERE DNO='5')
MINUS (SELECT PNO
FROM WORKS_ON
WHERE E.SSN=SSN));

5.SELECT D.DNO, COUNT (*)
FROM DEPARTMENT D, EMPLOYEE E
WHERE D.DNO=E.DNO
AND E.SALARY>600000
AND D.DNO IN (SELECT E1.DNO
FROM EMPLOYEE E1
GROUP BY E1.DNO
HAVING COUNT (*)>5)
GROUP BY D.DNO;