Name – SOUMYA BAIRAGYA

Queries on Library Database

Insert:

```
INSERT INTO BOOK VALUES (25, 'World of Life', '');
UPDATE BOOK SET NAME='WORLD OF LIFE' WHERE NAME='World of Life';

INSERT INTO BOOK_LOANS VALUES (25, 673, 132, '05-12-2021', '06-03-2021');

INSERT INTO BOOK VALUES (45, 'WINGS OF FIRE', 'NANCY');
INSERT INTO BOOK_COPIES VALUES (45, 673, 1);

INSERT INTO BOOK_AUTHORS VALUES (25, 'BOSE');
INSERT INTO BOOK_AUTHORS VALUES (45, 'KALAM');

INSERT INTO BOOK_COPIES VALUES (25, 774, 3);

INSERT INTO BOOK_LOANS VALUES (1, 334, 132, '02-12-2016', '');
INSERT INTO BOOK_LOANS VALUES (3, 673, 119, '05-01-2004', '');
INSERT INTO BOOK_LOANS VALUES (25, 774, 119, '07-27-2021', '');
INSERT INTO BOOK_LOANS VALUES (25, 334, 132, '02-12-2016', sysdate);
INSERT INTO BOOK_LOANS VALUES (25, 673, 119, '05-01-2004', '07-25-2004');
INSERT INTO BOOK_LOANS VALUES (3, 774, 119, '07-27-2021', sysdate);
INSERT INTO BOOK_LOANS VALUES (1, 673, 152, '09-12-2021', sysdate);


INSERT INTO BOOK_AUTHORS VALUES (1, 'ROWLING');
INSERT INTO BOOK_AUTHORS VALUES (1, 'ERIK');

INSERT INTO BOOK VALUES (82, 'ART OF HELIA', 'HARRY');

INSERT INTO BOOK_COPIES VALUES (82, 673, 5);
INSERT INTO BOOK_COPIES VALUES (82, 334, 5);
```

Statements:

1. **Get the total number of books written by author POTTER.**
   SELECT COUNT(BOOK_ID) FROM BOOK_AUTHORS WHERE AUTHOR_NAME = 'POTTER';

2. **List book names which are not yet issued by anyone till date.**
   SELECT NAME FROM BOOK WHERE PUBLISHER_NAME IS NULL;

3. **How many times the book with id 3 is issued?**
   SELECT NUMBER_OF_COPIES FROM BOOK_COPIES WHERE BOOK_ID=3;

4. **List author names who have written less number of books.**
   SELECT AUTHOR_NAME FROM BOOK_AUTHORS WHERE BOOK_ID = (SELECT BOOK_ID FROM BOOK_COPIES WHERE NUMBER_OF_COPIES = (SELECT MIN(NUMBER_OF_COPIES) FROM BOOK_COPIES));

5. **Get the average number of books in library branch id 774.**
   SELECT AVG(NUMBER_OF_COPIES) FROM BOOK_COPIES WHERE BRANCH_ID = 774;

6. **List the publisher names with less than 2 book publications.**
   SELECT B.PUBLISHER_NAME FROM BOOK B INNER JOIN BOOK_COPIES C ON B.BOOK_ID=C.BOOK_ID WHERE (C.NUMBER_OF_COPIES < 2);

7. **Get borrower details who have issued 'ABC' book in the month of May 2021**
   select name from borrower where card_no = (select card_no from book_loans where date_out between '05-01-2021' and '05-31-2021');

8. **Which library branch (name) has only one copy of book 'Wings of Fire'**
   i. select branch_name from library_branch where branch_id = (select branch_id from book_copies where number_of_copies = 1);
   ii. select L.branch_name from library_branch L inner join book_copies B on B.book_id=(select book_id from book where name='WINGS OF FIRE') where L.branch_id = B.branch_id and B.number_of_copies = 1;

9. **Retrieve the names of all borrowers who do not have any books checked out.**
   SELECT NAME FROM BORROWER WHERE CARD_NO IN (SELECT CARD_NO FROM BOOK_LOANS WHERE DUE_DATE IS NULL);

10. **How many books are written by each author?**
    select AUTHOR_NAME, NUMBER_OF_COPIES from BOOK_AUTHORS Natural join Book_copies;

11. **Get the book titles issued by Nancy from Penguin branch library.**
    select b.name from book b inner join book_copies c on b.book_id=c.book_id where b.publisher_name='NANCY' and c.branch_id = (select branch_id from library_branch where branch_name='PENGUIN');

12. **For each book that is loaned out from the Swargate branch and whose DueDate is today, retrieve the book title, the borrower's name and the borrower's address.**

13. **Get the month name along with the total number of books loaned out in that month.**

```
select count(book_id), to_char(date_out,'month') from book_loans group by date_out;
```

14. **Get the list of co-authors with the author ‘POTTER’ for book ‘TEST’**

```
SELECT AUTHOR_NAME FROM BOOK_AUTHORS WHERE BOOK_ID=1;
SELECT AUTHOR_NAME FROM BOOK_AUTHORS WHERE BOOK_ID=1 AND
AUTHOR_NAME <> 'POTTER';

SELECT AUTHOR_NAME FROM BOOK_AUTHORS A LEFT JOIN BOOK B ON
A.BOOK_ID=B.BOOK_ID WHERE NAME='SORCERORS';
SELECT AUTHOR_NAME FROM BOOK_AUTHORS A LEFT JOIN BOOK B ON
A.BOOK_ID=B.BOOK_ID WHERE NAME='SORCERORS' AND AUTHOR_NAME <>
'POTTER';
```

15. **List all publisher names with or without the book title.**

```
SELECT DISTINCT PUBLISHER_NAME FROM BOOK WHERE NAME IS NULL OR
NAME IS NOT NULL;
SELECT DISTINCT PUBLISHER_NAME FROM BOOK WHERE NAME IS NULL OR
NAME IS NOT NULL AND PUBLISHER_NAME IS NOT NULL;
```

16. **Get all book details starting and ending with letter A and published by publishers from PRIVETDRIVE city.**

```
SELECT * FROM BOOK B LEFT JOIN PUBLISHER P ON
B.PUBLISHER_NAME=P.NAME WHERE P.ADDRESS='PRIVETDRIVE' AND B.NAME
LIKE 'A%A';
```

17. **Get a pair of library branch name having same number of copies of “ART OF HELIA” book.**

```
SELECT DISTINCT BRANCH_NAME
FROM (SELECT BC1.BRANCH_ID
FROM BOOK_COPIES BC1
JOIN BOOK_COPIES BC2
ON BC1.BRANCH_ID != BC2.BRANCH_ID AND BC1.NUMBER_OF_COPIES =
BC2.NUMBER_OF_COPIES
WHERE BC1.BOOK_ID = (SELECT BOOK_ID
FROM BOOK
```

WHERE NAME = 'ART OF HELIA')) NATURAL JOIN LIBRARY_BRANCH;

18. Get the library branch name having all books published by “HARRY” publication.

```
SELECT BRANCH_NAME
FROM LIBRARY_BRANCH
WHERE BRANCH_ID IN (
   SELECT BRANCH_ID
   FROM BOOK_COPIES
   WHERE BOOK_ID IN (
         SELECT BOOK_ID
         FROM BOOK
         WHERE PUBLISHER_NAME = 'HARRY'
     )
   AND BRANCH_ID NOT IN (
         SELECT BRANCH_ID
         FROM BOOK_COPIES
         GROUP BY BRANCH_ID
         HAVING COUNT(BOOK_ID) > (
            SELECT COUNT(BOOK_ID)
            FROM BOOK
            WHERE PUBLISHER_NAME = 'HARRY'
         )
     )
   GROUP BY BRANCH_ID
);
```

19. **How many copies of each book are available in all library branches?**

```
select book_id, NUMBER_OF_COPIES, branch_name from book_copies natural join
library_branch;
```

20. Which are the books in demand?

```
SELECT NAME
FROM BOOK
WHERE BOOK_ID IN (
   SELECT BOOK_ID
   FROM BOOK_LOANS
```

```
GROUP BY BOOK_ID
HAVING COUNT(BOOK_ID) = (
   SELECT MAX(COUNT(BOOK_ID))
   FROM BOOK_LOANS
   GROUP BY BOOK_ID));
```

21. **Find out the publisher name whose books are not in demand.**

```
SELECT PUBLISHER_NAME
FROM BOOK
NATURAL JOIN BOOK_LOANS
GROUP BY PUBLISHER_NAME
HAVING COUNT(BOOK_ID) IN (
   SELECT MIN(COUNT(BOOK_ID))
   FROM BOOK_LOANS
   GROUP BY BOOK_ID);
```

22. **Get a library branch id that has more books than branch id 334.**

```
select branch_id from book_copies where NUMBER_OF_COPIES > (select
NUMBER_OF_COPIES from book_copies where branch_id=334);
```

23. **Get the total no of book copies where publisher name is Harry.**

```
SELECT SUM(NUMBER_OF_COPIES) FROM BOOK B NATURAL JOIN
BOOK_COPIES C WHERE B.PUBLISHER_NAME IN ('HARRY');
```

24. **Get the Card numbers of Borrowers where the Branch name is Vinewood.**

```
SELECT DISTINCT(CARD_NO) FROM BOOK_LOANS WHERE BRANCH_ID IN
(SELECT BRANCH_ID FROM LIBRARY_BRANCH WHERE
BRANCH_NAME='VINEWOOD');
```

25. **Get the Phone number of Publisher where the book title is “ART OF HELIA”.**

```
SELECT PHONE FROM PUBLISHER WHERE NAME IN (SELECT PUBLISHER_NAME
FROM BOOK WHERE NAME='ART OF HELIA');
```

26. **Get the Address of Library Branch having books of Publisher “NANCY”.**

```
SELECT DISTINCT(ADDRESS) FROM LIBRARY_BRANCH NATURAL JOIN
BOOK_COPIES NATURAL JOIN BOOK WHERE PUBLISHER_NAME IN ('NANCY');
```

SELECT ADDRESS FROM LIBRARY_BRANCH WHERE BRANCH_ID IN (SELECT BRANCH_ID FROM BOOK_LOANS WHERE BOOK_ID IN(SELECT BOOK_ID FROM BOOK WHERE PUBLISHER_NAME='NANCY'));

Section 2

Rank queries top N, second topper etc .

SELECT * FROM emp ORDER BY sal DESC;

Top N

SELECT * FROM (SELECT * FROM emp ORDER BY sal DESC) WHERE ROWNUM<=3 ORDER BY sal asc;

second highest

SELECT * FROM emp WHERE sal = (SELECT MAX(sal) FROM emp WHERE sal NOT IN (SELECT MAX(sal) FROM emp))

SELECT * FROM (SELECT empno, eNAME, RANK() OVER (ORDER BY sal DESC) RANKING FROM emp) WHERE RANKING=6;

CREATE TABLE STUDENTS(GRNO NUMBER(3),

NAME VARCHAR2(10),

CPI NUMBER(4,2));

INSERT INTO STUDENTS VALUES(101,'AMIT',5.67);

INSERT INTO STUDENTS VALUES(102,'RAJU',6.07);

INSERT INTO STUDENTS VALUES(103,'HARI',8.00);

INSERT INTO STUDENTS VALUES(104,'PUJA',7.75);

INSERT INTO STUDENTS VALUES(105,'RAVI',9.82);

INSERT INTO STUDENTS VALUES(106,'RANI',7.00);

INSERT INTO STUDENTS VALUES(107,'SUJA',8.04);

INSERT INTO STUDENTS VALUES(108,'ASHA',9.09);

INSERT INTO STUDENTS VALUES(109,'YASH',6.66);

INSERT INTO STUDENTS VALUES(110,'RAHUL',5.67);

INSERT INTO STUDENTS VALUES(111,'ALOK',9.09);

SELECT * FROM STUDENTS WHERE ROWNUM<=3 ORDER BY CPI DESC;

TOP 3 STUDENTS

SELECT * FROM (SELECT * FROM STUDENTS ORDER BY CPI DESC) WHERE ROWNUM<=2 ORDER BY CPI DESC;

SECOND HIGHEST

SELECT * FROM STUDENTS WHERE CPI = (SELECT MAX(CPI) FROM STUDENTS WHERE CPI NOT IN (SELECT MAX(CPI) FROM STUDENTS))

SELECT GRNO, NAME, cpi, DENSE RANK() OVER (ORDER BY CPI DESC) RANKING FROM STUDENTS WHERE RANKING=2

SELECT * FROM (SELECT GRNO, NAME, DENSE_RANK() OVER (ORDER BY CPI DESC) RANKING FROM STUDENTS) WHERE RANKING=2;

SELECT GRNO, NAME, RANK() OVER (ORDER BY CPI DESC) RANKING FROM STUDENTS

SELECT * FROM (SELECT GRNO, NAME, RANK() OVER (ORDER BY CPI DESC) RANKING FROM STUDENTS) WHERE RANKING=2;

SELECT GRNO, NAME, ROW_NUMBER() OVER (ORDER BY CPI DESC) RANKING FROM STUDENTS

SELECT * FROM (SELECT GRNO, NAME, ROW_NUMBER() OVER (ORDER BY CPI DESC) RANKING FROM STUDENTS) WHERE RANKING=3;

SELECT * FROM STUDENTS S1 WHERE 2 = (SELECT COUNT(DISTINCT(CPI)) FROM STUDENTS S2 WHERE S2.CPI>=S1.CPI)