# Parul University

Name: Prerak Doshi
Email: prerak12102004@gmail.com
Roll no: 25UG033422
Phone: 8849921118
Branch: Parul University
Department: CSE10_Batch 1
Batch: 2028
Degree: B.Tech - CSE

Scan to verify results

## PIET_Oracle DBMS_Course

## PIEt_oracle DBMS_Session 8_PAH

Attempt : 1
Total Mark : 50
Marks Obtained : 50

## Section 1 : coding

1. Problem Statement

Priya manages a blood donation camp database. She wants to analyze donor contributions by checking total units donated per person, the average units donated, and identifying donors who have donated only once.

Table Details:

Sample Input Records:

Tasks to perform:

Show total units donated per donor.Calculate the average units donated

25UG033422

across all donations.List donors who have donated only once.

Note: The user must write only the join queries to select the required records.

***Answer***

*oracle.sql*

```
SELECT D.DONOR_NAME, SUM(N.UNITS) AS TOTAL_UNITS
FROM DONORS D
LEFT JOIN DONATIONS N ON D.DONOR_ID = N.DONOR_ID
GROUP BY D.DONOR_NAME;

SELECT ROUND(AVG(UNITS), 2) AS AVERAGE_UNITS
FROM DONATIONS;

SELECT D.DONOR_NAME
FROM DONORS D
JOIN DONATIONS N ON D.DONOR_ID = N.DONOR_ID
GROUP BY D.DONOR_NAME
HAVING COUNT(N.DONATION_ID) = 1;
```

***Status :*** Correct ***Marks : 10/10***

2. Problem Statement

Arjun manages a movie rating platform. He wants to analyze viewer ratings to calculate movie performance metrics, identify top-performing content, and find users who provide high-quality ratings.

Table Details:

Sample Input Records:

Tasks to perform:

Display the average rating for each movie, showing the movie title and its

average rating as AVG_RATING (rounded to 2 decimal places).Find the movie(s) with the highest average rating, displaying the movie title and its maximum average rating as AVG_RATING across all movies.List all distinct users (USER_ID) who gave a rating greater than 4.5.

Note: The user must write only the join queries to select the required records.

***Answer***

*oracle.sql*

```
SELECT M.TITLE, ROUND(AVG(R.RATING), 2) AS AVG_RATING
FROM MOVIES M
JOIN RATINGS R ON M.MOVIE_ID = R.MOVIE_ID
GROUP BY M.TITLE;

SELECT TITLE, AVG_RATING FROM (
   SELECT M.TITLE, ROUND(AVG(R.RATING), 2) AS AVG_RATING
   FROM MOVIES M
   JOIN RATINGS R ON M.MOVIE_ID = R.MOVIE_ID
   GROUP BY M.TITLE
)
WHERE AVG_RATING = (
   SELECT MAX(AVG_RATING) FROM (
   SELECT ROUND(AVG(R.RATING), 2) AS AVG_RATING
   FROM RATINGS R
   GROUP BY R.MOVIE_ID
   )
);

SELECT DISTINCT USER_ID
FROM RATINGS
WHERE RATING > 4.5;
```

***Status :*** Correct ***Marks : 10/10***

3. Problem Statement

Tanvi manages product view tracking on an e-commerce platform. She wants to find total views per product, average views per product, and products that have never been viewed..

Table Details:

Sample Input Records:

Tasks to perform:

Show total views per product.Calculate average views per product.List products never viewed.

Note: The user must write only the join queries to select the required records.

***Answer***

*oracle.sql*

```
SELECT P.PRODUCT_NAME, COUNT(V.VIEW_ID) AS TOTAL_VIEWS
FROM PRODUCTS P
LEFT JOIN VIEWS V ON P.PRODUCT_ID = V.PRODUCT_ID
GROUP BY P.PRODUCT_NAME;

SELECT ROUND(AVG(VIEW_COUNT), 2) AS AVERAGE_VIEWS
FROM (
  SELECT PRODUCT_ID, COUNT(*) AS VIEW_COUNT
  FROM VIEWS
  GROUP BY PRODUCT_ID
);

SELECT P.PRODUCT_NAME
FROM PRODUCTS P
LEFT JOIN VIEWS V ON P.PRODUCT_ID = V.PRODUCT_ID
WHERE V.PRODUCT_ID IS NULL;
```

***Status :*** Correct ***Marks : 10/10***

4. Problem Statement

Nisha manages a gym and wants to analyze member attendance. She

needs the attendance count per member, the maximum attendance recorded by any member, and to identify members who have never attended.

Table Details:

Sample Input Records:

Tasks to perform:

Count attendance per member.Find the maximum attendance count of a member.List members who have never attended.

Note: The user must write only the join queries to select the required records.

***Answer***

*oracle.sql*

```
SELECT M.MEMBER_NAME, COUNT(A.ATTENDANCE_ID) AS
TOTAL_ATTENDANCE
FROM MEMBERS M
LEFT JOIN ATTENDANCE A ON M.MEMBER_ID = A.MEMBER_ID
GROUP BY M.MEMBER_NAME;

SELECT MAX(ATTEND_COUNT) AS MAX_ATTENDANCE
FROM (
   SELECT MEMBER_ID, COUNT(*) AS ATTEND_COUNT
   FROM ATTENDANCE
   GROUP BY MEMBER_ID
);

SELECT M.MEMBER_NAME
FROM MEMBERS M
LEFT JOIN ATTENDANCE A ON M.MEMBER_ID = A.MEMBER_ID
WHERE A.MEMBER_ID IS NULL;
```

***Status :*** Correct ***Marks : 10/10***

5. Problem Statement

Dinesh maintains a library system. He needs to find out how many books each student has borrowed, the average borrow count, and which students have borrowed more than 5 books.

Table Details:

Sample Input Records:

Tasks to perform:

Display book count borrowed per student.Find average book borrow count.Display names of students who borrowed more than 5 books.

Note: The user must write only the join queries to select the required records.

***Answer***

*oracle.sql*

```
SELECT S.STUDENT_NAME, COUNT(B.BORROW_ID) AS
TOTAL_BOOKS_BORROWED
FROM STUDENTS S
LEFT JOIN BORROWINGS B ON S.STUDENT_ID = B.STUDENT_ID
GROUP BY S.STUDENT_NAME;

SELECT ROUND(AVG(BOOK_COUNT), 2) AS AVERAGE_BORROW_COUNT
FROM (
  SELECT STUDENT_ID, COUNT(*) AS BOOK_COUNT
  FROM BORROWINGS
  GROUP BY STUDENT_ID
);

SELECT S.STUDENT_NAME
FROM STUDENTS S
JOIN BORROWINGS B ON S.STUDENT_ID = B.STUDENT_ID
GROUP BY S.STUDENT_NAME
```

25UG033422

```
HAVING COUNT(B.BORROW_ID) > 5;
```

***Status :*** Correct **Marks : 10/10**

25UG033422