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 7_CY

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : COD

1. Problem Statement

Sophia, a data analyst at a music streaming service, has been tasked with providing insights on user interactions and song performance. She needs to perform various analyses by joining data from two tables: `USER_LISTENING` and `SONG_PLAY_HISTORY`.

Table details:

Sample Input Records:

Table Name: USER_LISTENING

Table Name: SONG_PLAY_HISTORY

Tasks to perform:

Write a query to retrieve the LISTENING_ID and USER_ID from the USER_LISTENING table, along with the PLAY_COUNT from the SONG_PLAY_HISTORY table. The query should join these tables on the SONG_ID field to get the play count for each user's listening record.Write a query to calculate the total duration of listening for each song. Retrieve the SONG_ID from the USER_LISTENING table and the sum of DURATION for each song, using the alias TOTAL_DURATION. The query should join the USER_LISTENING and SONG_PLAY_HISTORY tables on the SONG_ID field and group the results by SONG_ID.Write a query to calculate the total play count for each song. Retrieve the SONG_ID from the USER_LISTENING table and the sum of PLAY_COUNT from the SONG_PLAY_HISTORY table, using the alias TOTAL_PLAY_COUNT. The query should join these tables on the SONG_ID field and group the results by SONG_ID.

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

Answer

oracle.sql

```
SELECT UL.LISTENING_ID, UL.USER_ID, SPH.PLAY_COUNT FROM
USER_LISTENING UL
INNER JOIN SONG_PLAY_HISTORY SPH ON UL.SONG_ID = SPH.SONG_ID;

SELECT UL.SONG_ID, SUM(UL.DURATION) AS TOTAL_DURATION FROM
USER_LISTENING UL
INNER JOIN SONG_PLAY_HISTORY SPH ON UL.SONG_ID = SPH.SONG_ID
GROUP BY UL.SONG_ID;

SELECT UL.SONG_ID, SUM(SPH.PLAY_COUNT) AS TOTAL_PLAY_COUNT FROM
USER_LISTENING UL
INNER JOIN SONG_PLAY_HISTORY SPH ON UL.SONG_ID = SPH.SONG_ID
GROUP BY UL.SONG_ID;
```

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

2. Problem Statement

Bob is tasked with analyzing data related to artists and their songs for a music company. To gain insights into trends involving artists and their songs, Bob needs to use inner joins along with aggregate functions, particularly the COUNT function. He must join the Artists and Songs tables to extract and analyze information about the number of songs and their performance metrics.

Table Details:

Sample Input Records:

Table: ARTISTS

Table: SONGS

Tasks to be Performed:

Write a query to retrieve the title of songs and the artist_name from the Songs and Artists tables, respectively. The query should join these tables on the artist_id field to match each song with its artist.Write a query to retrieve the artist_name from the Artists table and the play_count from the Songs table. The query should join these tables on the artist_id field to associate each song's play count with the respective artist.Write a query to retrieve the title of songs and the genre of artists from the Songs and Artists tables, respectively. The query should join these tables on the artist_id field to get the genre of the artist who performed each song.

Note: The user must write only the queries to select the required data with the relevant aggregate functions.

Answer

oracle.sql

```
SELECT Songs.title, Artists.artist_name FROM Songs
INNER JOIN Artists ON Songs.artist_id = Artists.artist_id;

SELECT Artists.artist_name, Songs.play_count FROM Songs
INNER JOIN Artists ON Songs.artist_id = Artists.artist_id;
```

```
SELECT Songs.title, Artists.genre FROM Songs
INNER JOIN Artists ON Songs.artist_id = Artists.artist_id;
```

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

3. Problem Statement

Sophia is a business analyst working for a subscription-based streaming service. She needs to analyze subscriber data to understand viewing patterns and subscription status across different subscription tiers. Use inner joins to combine the SUBSRIPTIONS and VIEWING_LOGS tables.

Table Details:

Sample Input Records:

Table Name: SUBSCRIPTIONS

Table Name: VIEWING_LOGS

Tasks to perform:

Write a query to retrieve the SUBSCRIPTION_ID and SUBSCRIBER_ID from the SUBSCRIPTIONS table, along with the VIEW_DATE from the VIEWING_LOGS table. The query should join these tables on the SUBSCRIPTION_ID field to get the viewing dates for each subscription record.Write a query to calculate the total view duration for each subscriber. Retrieve the SUBSCRIBER_ID from the SUBSCRIPTIONS table and the sum of VIEW_DURATION from the VIEWING_LOGS table, using the alias TOTAL_DURATION. The query should join the SUBSCRIPTIONS and VIEWING_LOGS tables on the SUBSCRIPTION_ID field and group the results by SUBSCRIBER_ID.Write a query to retrieve the VIEW_DATE from the VIEWING_LOGS table for subscribers with a 'Premium' subscription tier.

The query should join the SUBSCRIPTIONS and VIEWING_LOGS tables on the SUBSCRIPTION_ID field and filter the results where the subscription tier is 'Premium'.

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

Answer

oracle.sql

```
SELECT s.SUBSCRIPTION_ID, s.SUBSCRIBER_ID, v.VIEW_DATE FROM
SUBSCRIPTIONS s
INNER JOIN VIEWING_LOGS v ON s.SUBSCRIPTION_ID = v.SUBSCRIPTION_ID;

SELECT s.SUBSCRIBER_ID, SUM(v.VIEW_DURATION) AS TOTAL_DURATION
FROM SUBSCRIPTIONS s
INNER JOIN VIEWING_LOGS v ON s.SUBSCRIPTION_ID = v.SUBSCRIPTION_ID
GROUP BY s.SUBSCRIBER_ID;

SELECT v.VIEW_DATE FROM SUBSCRIPTIONS s
INNER JOIN VIEWING_LOGS v ON s.SUBSCRIPTION_ID = v.SUBSCRIPTION_ID
WHERE s.TIER = 'Premium';
```

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

4. Problem Statement

David, an academic advisor at a university, has been assigned the task of analyzing student performance across various subjects to provide insights for improving academic programs. He needs to calculate the ranks of students based on their scores in different subjects by joining data from the STUDENT_SCORES and SUBJECT_DETAILS tables.

Table Details:

Sample input records:

Table Name: STUDENTS

Table Name: EXAM_SCORES

Tasks to Perform:

Write a query to retrieve the STUDENT_NAME and SCORE for students who took the subject 'Math'. Join the STUDENTS table with the EXAM_SCORES table on the STUDENT_ID field to get the relevant scores for the specified subject.Write a query to retrieve the STUDENT_NAME of students who scored greater than 90 in the subject 'English'. Join the STUDENTS table with the EXAM_SCORES table on the STUDENT_ID field and filter the results where the subject is 'English' and the score is greater than 90.Write a query to retrieve the distinct STUDENT_NAME of students who scored 85 or more in any subject. Join the STUDENTS table with the EXAM_SCORES table on the STUDENT_ID field and filter the results to include only those students with a score of 85 or higher.

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

Answer

oracle.sql

```
SELECT s.STUDENT_NAME, e.SCORE FROM STUDENTS s
INNER JOIN EXAM_SCORES e ON s.STUDENT_ID = e.STUDENT_ID WHERE
e.SUBJECT = 'Math';

SELECT s.STUDENT_NAME FROM STUDENTS S
INNER JOIN EXAM_SCORES e ON s.STUDENT_ID = e.STUDENT_ID
WHERE e.SUBJECT = 'English' AND e.SCORE > 90;

SELECT DISTINCT s.STUDENT_NAME FROM STUDENTS s
INNER JOIN EXAM_SCORES e ON s.STUDENT_ID = e.STUDENT_ID WHERE
e.SCORE >= 85;
```

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

5. Problem Statement

Alex, a data analyst at an electronics retailer, has been assigned the task of analyzing product return data to help the company better understand customer return patterns. To perform this analysis, Alex needs to join data from two tables: PRODUCTS and RETURNS.

Sample Table Records:

Table Name: PRODUCTS

Table Name: RETURNS

Tasks to be done

Write a query to retrieve the PRODUCT_ID, PRODUCT_NAME, the maximum RETURN_AMOUNT with the alias name MAX_RETURN_AMOUNT, and RETURN_DATE for each product. Join the PRODUCTS table with the RETURNS table on PRODUCT_ID using a LEFT OUTER JOIN. Group the results by PRODUCT_ID, PRODUCT_NAME, and RETURN_DATE.Write a query to retrieve the RETURN_ID, PRODUCT_ID, RETURN_AMOUNT, RETURN_DATE, PRODUCT_NAME, and CATEGORY for returns where the RETURN_AMOUNT is greater than the average RETURN_AMOUNT across all returns. Join the RETURNS table with the PRODUCTS table on PRODUCT_ID using a LEFT OUTER JOIN.Write a query to retrieve the PRODUCT_ID, PRODUCT_NAME, and RETURN_DATE for each product. Join the PRODUCTS table with the RETURNS table on PRODUCT_ID using a LEFT OUTER JOIN.

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

Answer

oracle.sql

```
SELECT p.PRODUCT_ID, p.PRODUCT_NAME,
   MAX(r.RETURN_AMOUNT) AS MAX_RETURN_AMOUNT,
   r.RETURN_DATE
FROM PRODUCTS p
LEFT OUTER JOIN RETURNS r ON p.PRODUCT_ID = r.PRODUCT_ID
GROUP BY p.PRODUCT_ID, p.PRODUCT_NAME, r.RETURN_DATE;

SELECT r.RETURN_ID, r.PRODUCT_ID, r.RETURN_AMOUNT, r.RETURN_DATE,
p.PRODUCT_NAME, p.CATEGORY
FROM RETURNS r
LEFT OUTER JOIN PRODUCTS p ON r.PRODUCT_ID = p.PRODUCT_ID
WHERE r.RETURN_AMOUNT > (
                    SELECT AVG(RETURN_AMOUNT) FROM RETURNS
                );

SELECT p.PRODUCT_ID, p.PRODUCT_NAME, r.RETURN_DATE
FROM PRODUCTS p LEFT OUTER JOIN RETURNS r ON p.PRODUCT_ID =
r.PRODUCT_ID;
```

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