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 6_CY_Updated

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : COD

1. 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. Sophia needs to use the GROUP BY and HAVING clauses to aggregate and filter the data for detailed insights.

Table Details:

Sample Input Records:

Table Name: SUBSCRIPTIONS

Table Name: VIEWING_LOGS

Tasks to perform

Calculate the total view duration with the alias name total_view_duration for each subscription. Display the SUBSCRIPTION_ID and total_view_duration for subscriptions with a total view duration greater than 100 from the VIEWING_LOGS table and group by SUBSCRIPTION_ID.Calculate the average view duration with the alias name avg_view_duration for each subscription. Display the SUBSCRIPTION_ID and avg_view_duration for subscriptions with an average view duration greater than 60 from the VIEWING_LOGS table and group by SUBSCRIPTION_ID.Count the number of view logs with the alias name num_view_logs for each subscription. Display the SUBSCRIPTION_ID and num_view_logs for subscriptions with more than 2 view logs from the VIEWING_LOGS table and group by SUBSCRIPTION_ID.Count the number of subscriptions with the alias name num_subscriptions for each tier. Display the TIER and num_subscriptions for tiers with more than 1 subscription from the SUBSCRIPTIONS table and group by TIER.Find the maximum end date with the alias name max_end_date for each tier. Display the TIER and max_end_date for tiers with a maximum end date greater than 2024-06-01 from the SUBSCRIPTIONS table and group by TIER.

Note:

The user must write only the queries to select the required data using the GROUP BY and HAVING clauses.

Answer

oracle.sql

```
SELECT SUBSCRIPTION_ID,
   SUM(VIEW_DURATION) AS TOTAL_VIEW_DURATION FROM VIEWING_LOGS
GROUP BY SUBSCRIPTION_ID
HAVING SUM(VIEW_DURATION) > 100;

SELECT SUBSCRIPTION_ID,
   AVG(VIEW_DURATION) AS AVG_VIEW_DURATION FROM VIEWING_LOGS
```

```
GROUP BY SUBSCRIPTION_ID
HAVING AVG(VIEW_DURATION) > 60;

SELECT SUBSCRIPTION_ID,
   COUNT(*) AS NUM_VIEW_LOGS FROM VIEWING_LOGS
GROUP BY SUBSCRIPTION_ID
HAVING COUNT(*) > 2;

SELECT TIER,
COUNT(*) AS NUM_SUBSCRIPTIONS FROM SUBSCRIPTIONS
GROUP BY TIER HAVING COUNT(*) > 1;

SELECT TIER,
MAX(END_DATE) AS MAX_END_DATE FROM SUBSCRIPTIONS
GROUP BY TIER HAVING MAX(END_DATE) > DATE '2024-06-01';
```

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

2. Problem Statement:

Olivia is a data analyst working for a travel agency. She has been asked to analyze the performance of various travel packages and identify trends in customer preferences. Olivia needs to use the GROUP BY and HAVING clauses to extract specific insights from the TRAVEL_PACKAGES, BOOKINGS, and CUSTOMERS tables.

Table Details:

Sample input records:

Table Name: TRAVEL_PACKAGES

Table Name: BOOKINGS

Table Name: CUSTOMERS

Tasks to Perform:

Calculate the average price for each destination from the TRAVEL_PACKAGES table and display destinations where the average price is greater than 1500. Use the alias avg_price for the average price.Calculate the total number of tickets for each package from the BOOKINGS table and display packages where the total number of tickets is greater than 5. Use the alias total_tickets for the total number of tickets.Calculate the average age of customers in each city from the CUSTOMERS table and display cities where the number of customers is greater than 1. Use the alias avg_age for the average age.Calculate the total sales amount for each package from the BOOKINGS table and display packages where the total sales amount is greater than 2000. Use the alias total_sales for the total sales amount.Calculate the average number of tickets for each customer from the BOOKINGS table and display customers where the total number of tickets is greater than 3. Use the alias avg_tickets for the average number of tickets.

Note: The user must write only the queries to select the required data using the GROUP BY and HAVING clauses.

Answer

oracle.sql

```
SELECT DESTINATION, AVG(PRICE) AS AVG_PRICE FROM TRAVEL_PACKAGES
GROUP BY DESTINATION HAVING AVG(PRICE) > 1500;

SELECT PACKAGE_ID, SUM(TICKET_COUNT) AS TOTAL_TICKETS FROM
BOOKINGS
GROUP BY PACKAGE_ID HAVING SUM(TICKET_COUNT) > 5;

SELECT CITY,
   AVG(AGE) AS AVG_AGE FROM CUSTOMERS
   GROUP BY CITY HAVING COUNT(CUSTOMER_ID) > 1;

SELECT PACKAGE_ID, SUM(SALES_AMOUNT) AS TOTAL_SALES FROM
BOOKINGS
```

```
GROUP BY PACKAGE_ID HAVING SUM(SALES_AMOUNT) > 2000;

SELECT CUSTOMER_ID,
   AVG(TICKET_COUNT) AS AVG_TICKET FROM BOOKINGS
   GROUP BY CUSTOMER_ID HAVING SUM(TICKET_COUNT) > 3;
```

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

3. Problem Statement

Michael needs to perform aggregate functions to evaluate the work hours recorded for each employee. He needs to do the following tasks from the given table details

Table Details:

Sample Input Records

Table: EMPLOYEE_WORK_HOURS

Tasks to be Performed:

Calculate the average hours worked from the EMPLOYEE_WORK_HOURS table with the alias name AVG_HOURS_WORKED.Calculate the minimum hours worked from the EMPLOYEE_WORK_HOURS table with the alias name MIN_HOURS_WORKED.Calculate the maximum hours worked from the EMPLOYEE_WORK_HOURS table with the alias name MAX_HOURS_WORKED.Calculate the total hours worked from the EMPLOYEE_WORK_HOURS table with the alias name TOTAL_HOURS_WORKED.Calculate the standard deviation of hours worked from the EMPLOYEE_WORK_HOURS table with the alias name STDDEV_HOURS_WORKED.Calculate the variance of hours worked from the EMPLOYEE_WORK_HOURS table with the alias name VARIANCE_HOURS_WORKEDCalculate the rank of hours worked for each employee from the EMPLOYEE_WORK_HOURS table with the alias name HOURS_RANK.Calculate the median hours worked for each employee from

the EMPLOYEE_WORK_HOURS table with the alias name MEDIAN_HOURS_WORKED, group by EMP_ID.Calculate the count of work records from the EMPLOYEE_WORK_HOURS table with the alias name WORK_RECORD_COUNT.

Note:

While using the RANK function use the OVER keyword

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

Answer

oracle.sql

```
SELECT AVG(HOURS_WORKED) AS AVG_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS;

SELECT MIN(HOURS_WORKED) AS MIN_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS;
SELECT MAX(HOURS_WORKED) AS MAX_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS;
SELECT SUM(HOURS_WORKED) AS TOTAL_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS;
SELECT STDDEV(HOURS_WORKED) AS STDDEV_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS;
SELECT VARIANCE(HOURS_WORKED) AS VARIANCE_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS;

SELECT EMP_ID, HOURS_WORKED, RANK() OVER (ORDER BY HOURS_WORKED
DESC) AS HOURS_RANK FROM EMPLOYEE_WORK_HOURS;

SELECT EMP_ID,
   MEDIAN(HOURS_WORKED) AS MEDIAN_HOURS_WORKED FROM
EMPLOYEE_WORK_HOURS GROUP BY EMP_ID;

SELECT COUNT(*) AS WORK_RECORD_COUNT FROM
EMPLOYEE_WORK_HOURS;
```

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