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 2_CY_Updated

Attempt : 2
Total Mark : 50
Marks Obtained : 50

Section 1 : COD

1. Problem Statement

Ben is developing a health monitoring app to track user activities and vital signs. The app needs to handle three main types of data: USERS, ACTIVITIES, and VITALS

Write the Query statements to create the following tables to handle the data:

<span style="color: rgb(51, 51, 51);">symbol refers to the primary key&nbsp;</span>

NN refers to Not NULL

Sample record:

Tasks:

Create the USERS table to store user details, including UserID, UserName, Email, and RegistrationDate.Create the ACTIVITIES table to store user activities, including ActivityID, UserID, ActivityType, and ActivityDate.Create the VITALS table to store vital signs, including VitalID, UserID, HeartRate, BloodPressure, and RecordedDate.After creating the tables, Ben needs to:

Inser the Sample Records in the appropriate tables.Alter the ACTIVITIES table to add a new column Duration (in minutes) to track the duration of the activity.Update the UserName field in the USERS table for a specific user (UserID = 2), changing it to "Janet Smith".Updates the BloodPressure value to '125/80' in the VITALS table for the record where the UserID is 3

Note: The user must write only the query to create and alter the table. The query to display the description of the table is already given.

Answer

oracle.sql

```
CREATE TABLE USERS (
   UserID NUMBER,
   UserName VARCHAR2(100),
   Email VARCHAR2(100),
   RegistrationDate DATE
);

CREATE TABLE ACTIVITIES (
   ActivityID NUMBER,
   UserID NUMBER,
   ActivityType VARCHAR2(50),
   ActivityDate DATE
);

CREATE TABLE VITALS (
   VitalID NUMBER,
   UserID NUMBER,
   HeartRATE NUMBER,
   BloodPressure VARCHAR2(20),
    RecordedDate DATE
);

INSERT INTO USERS (UserID, UserName, Email, RegistrationDate)
VALUES (1, 'John Doe', 'johndoe@example.com', TO_DATE('2023-01-01', 'YYYY-MM-DD'));

INSERT INTO USERS (UserID, UserName, Email, RegistrationDate)
VALUES (2, 'Jane Smith', 'janesmith@example.com', TO_DATE('2023-01-05', 'YYYY-MM-DD'));

INSERT INTO USERS (UserID, UserName, Email, RegistrationDate)
VALUES (3, 'Alice Brown', 'alicebrown@example.com', TO_DATE('2023-02-10', 'YYYY-MM-DD'));

INSERT INTO ACTIVITIES (ActivityID, UserID, ActivityType, ActivityDate)
VALUES (1, 1, 'Running', TO_DATE('2023-03-01', 'YYYY-MM-DD'));

INSERT INTO ACTIVITIES (ActivityID, UserID, ActivityType, ActivityDate)
VALUES (2, 2, 'Swimming', TO_DATE('2023-03-05', 'YYYY-MM-DD'));

INSERT INTO ACTIVITIES (ActivityID, UserID, ActivityType, ActivityDate)
VALUES (3, 3, 'Cycling', TO_DATE('2023-03-10', 'YYYY-MM-DD'));

INSERT INTO VITALS (VitalID, UserID, HeartRate, BloodPressure, RecordedDate)
VALUES (1, 1, 75, '120/80', TO_DATE('2023-03-01', 'YYYY-MM-DD'));

INSERT INTO VITALS (VitalID, UserID, HeartRate, BloodPressure, RecordedDate)
VALUES (2, 2, 80, '110/70', TO_DATE('2023-03-05', 'YYYY-MM-DD'));

INSERT INTO VITALS (VitalID, UserID, HeartRate, BloodPressure, RecordedDate)
VALUES (3, 3, 85, '130/85', TO_DATE('2023-03-10', 'YYYY-MM-DD'));

UPDATE USERS
SET UserName = 'Janet Smith'
WHERE UserID = 2;

ALTER TABLE ACTIVITIES
ADD Duration NUMBER;

UPDATE VITALS
SET BloodPressure = '125/80'
```

```
WHERE UserID = 3;
```

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

2. Problem Statement

Alexander is developing a ride-sharing application to track drivers, vehicles, and trips. The system needs to handle three main types of data: DRIVERS, VEHICLES, and TRIPS.

Write the Qeury statements to create the following tables:

symbol refers to the primary key&nbsp;

NN refers to Not NULL

Sample Records:

Alexander Task is to:

Create the DRIVERS table to store driver details, including DriverID, DriverName, Phone, and LicenseNumber.Create the VEHICLES table to store vehicle details, including VehicleID, DriverID, Make, and Model.Create the TRIPS table to store trip details, including TripID, VehicleID, StartLocation, EndLocation, and TripDate.Insert Sample Records in to the appropriate Tables.Remove the DriverID column from the VEHICLES table (since no foreign key is used).Note: The user must write only the query to create and alter the table. The query to display the description data of the table is already given.

Answer

oracle.sql

```
CREATE TABLE DRIVERS (
   DriverID NUMBER,
   DriverName VARCHAR2(100),
   Phone VARCHAR2(15),
   LicenseNumber VARCHAR2(20)
);

CREATE TABLE VEHICLES (
   VehicleID NUMBER,
   DriverID number,
   Make VARCHAR2(50),
   Model VARCHAR2(50)
);

CREATE TABLE TRIPS (
   TripID NUMBER,
   VehicleID NUMBER,
   StartLocation VARCHAR2(100),
   EndLocation VARCHAR2(100),
   TripDate DATE
);

INSERT INTO DRIVERS (DriverID, DriverName, Phone, LicenseNumber)
VALUES (1, 'John Doe', '123-456-7890', 'ABC12345');

INSERT INTO DRIVERS (DriverID, DriverName, Phone, LicenseNumber)
VALUES (2, 'Jane Smith', '987-654-3210', 'XYZ67890');

INSERT INTO DRIVERS (DriverID, DriverName, Phone, LicenseNumber)
VALUES (3, 'Alice Brown', '555-555-5555', 'LMN11223');

INSERT INTO VEHICLES (VehicleID, DriverID, Make, Model)
VALUES (1, 1, 'Toyota', 'Corolla');

INSERT INTO VEHICLES (VehicleID, DriverID, Make, Model)
VALUES (2, 2, 'Honda', 'Civic');

INSERT INTO VEHICLES (VehicleID, DriverID, Make, Model)
VALUES (3, 3, 'Ford', 'Focus');

INSERT INTO TRIPS (TripID, VehicleID, StartLocation, EndLocation, TripDate)
VALUES (1, 1, 'New York', 'Boston', TO_DATE('2023-03-01', 'YYYY-MM-DD'));

INSERT INTO TRIPS (TripID, VehicleID, StartLocation, EndLocation, TripDate)
VALUES (2, 2, 'Los Angeles', 'San Francisco', TO_DATE('2023-03-05', 'YYYY-MM-
DD'));
```

```
INSERT INTO TRIPS (TripID, VehicleID, StartLocation, EndLocation, TripDate)
VALUES (3, 3, 'Chicago', 'Detroit', TO_DATE('2023-03-10', 'YYYY-MM-DD'));

ALTER TABLE VEHICLES
DROP COLUMN DriverID;
```

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

3. Problem Statement

Samantha is a data analyst at a fitness center, and she has been given the task of managing exercise and user data. The database admin has already created two tables, Exercises and Users. Samantha needs to perform several operations on these tables based on specific conditions. The structure of the tables is shown below:

Sample input records:

Tasks for Samantha

Insert the above-given records into the Exercise and Users Table.Update the CaloriesBurned for exercises where ExerciseDate is between '2024-07-08' and '2024-07-15', CaloriesBurned is less than 300, and ExerciseType is not 'Yoga' by adding 100 to the CaloriesBurned value.Delete exercises where ExerciseDate is older than '2023-07-01' AND CaloriesBurned is greater than 500 OR ExerciseType is 'Cardio'.Select exercises where ExerciseDate is within the last 30 days (from '2024-07-15') AND CaloriesBurned is more than 200 AND ExerciseType is not 'Strength Training'.Select users with UserID 1101 OR 1102.

Note:

Write Oracle statements to accomplish each of the above tasks.

Ensure that your INSERT, UPDATE, DELETE, and SELECT queries are correctly formatted.

Test your queries to verify that they produce the expected results.

Answer

oracle.sql

```
INSERT INTO Exercises (ExerciseID, ExerciseType, CaloriesBurned,
ExerciseDate)
VALUES (4, 'Cardio', 600, DATE '2023-06-30');

INSERT INTO Exercises (ExerciseID, ExerciseType, CaloriesBurned,
ExerciseDate)
VALUES (5, 'Strength Training', 350, DATE '2024-07-14');

INSERT INTO Exercises (ExerciseID, ExerciseType, CaloriesBurned,
ExerciseDate)
VALUES (6, 'Walking', 250, DATE '2024-07-10');

INSERT INTO Exercises (ExerciseID, ExerciseType, CaloriesBurned,
ExerciseDate)
VALUES (7, 'Swimming', 280, DATE '2024-07-12');

INSERT INTO Exercises (ExerciseID, ExerciseType, CaloriesBurned,
ExerciseDate)
VALUES (8, 'Cycling', 290, DATE '2024-07-14');

INSERT INTO Users (UserID, UserName, Email) VALUES (1101, 'Alice',
'alice@example.com');
INSERT INTO Users (UserID, UserName, Email) VALUES (1102, 'Bob',
'bob@example.com');
INSERT INTO Users (UserID, UserName, Email) VALUES (1103, 'Charlie',
'charlie@example.com');
INSERT INTO Users (UserID, UserName, Email) VALUES (1104, 'David',
'david@example.com');
INSERT INTO Users (UserID, UserName, Email) VALUES (1105, 'Eve',
'eve@example.com');

UPDATE Exercises
SET CaloriesBurned = CaloriesBurned + 100
WHERE ExerciseDate BETWEEN DATE '2024-07-08' AND DATE '2024-07-15'
  AND CaloriesBurned < 300
```

```
   AND ExerciseType <> 'Yoga';

DELETE FROM Exercises
WHERE (ExerciseDate < DATE '2023-07-01' AND CaloriesBurned > 500)
   OR ExerciseType = 'Cardio';

SELECT * FROM Exercises
WHERE ExerciseDate BETWEEN DATE '2024-06-15' AND DATE '2024-07-15'
   AND CaloriesBurned > 200
   AND ExerciseType <> 'Strength Training';

SELECT * FROM Users
WHERE UserID = 1101
  OR UserID = 1102;
```

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

4. Problem Statement

Taylor is responsible for managing the product and supplier details for an online retail platform. The database includes two crucial tables: Products and Suppliers. Taylor needs to perform several operations on these tables according to specific requirements. The structure of the table is given below:

Sample Input Records:

Tasks for Taylor

Insert Records: Insert the above-mentioned records into the Products and Suppliers tables.Update: Update the Price of products by 10%, if the Category is 'Electronics' AND Price is greater than $500 AND StockQuantity is less than 10.Delete: Remove products where StockQuantity is less than

5 AND Category is not 'Home Appliances' OR ProductID is 1001 OR 1002.Select: Retrieve products where Category is 'Furniture' AND Price is between 100 dollars AND 500 dollars AND StockQuantity is at least 20.Select: Identify suppliers with SupplierID 1301, 1302, 1303, or 1304.Note:

Write Oracle statements to accomplish each of the above tasks.

Ensure that your INSERT, UPDATE, DELETE, and SELECT queries are correctly formatted.

Test your queries to verify that they produce the expected results.

Answer

oracle.sql

```
INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)
VALUES (1001, 'SmartPhone', 'Electronics', 699.99, 5);
INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)
VALUES (1002, 'Laptop', 'Electronics', 1200.00, 8);
INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)
VALUES (1003, 'Sofa', 'Furniture', 450.00, 20);
INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)
VALUES (1004, 'Coffee Maker', 'Home Appliances', 120.00, 12);
INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)
VALUES (1005, 'Table Lamp', 'Furniture', 85.00, 25);

INSERT INTO Suppliers (SupplierID, SupplierName, ContactEmail) VALUES (1301,
'ABC Electronics', 'contact@abcelectronics.com');
INSERT INTO Suppliers (SupplierID, SupplierName, ContactEmail) VALUES (1302,
'XYZ Furnishings', 'info@xyzfurnishings.com');
INSERT INTO Suppliers (SupplierID, SupplierName, ContactEmail) VALUES (1303,
'Home Goods Inc', 'support@homegoods.com');
INSERT INTO Suppliers (SupplierID, SupplierName, ContactEmail) VALUES (1304,
'Tech World', 'service@techworld.com');
INSERT INTO Suppliers (SupplierID, SupplierName, ContactEmail) VALUES (1305,
'Luxury Living', 'contact@luxuryliving.com');

UPDATE Products
SET Price = Price * 1.10
WHERE Category = 'Electronics'
  AND Price > 500
  AND StockQuantity < 10;
```

```
DELETE FROM Products
WHERE (StockQuantity < 5 AND Category <> 'Home Appliances')
   OR ProductID = 1001
   OR ProductID = 1002;

SELECT *
FROM Products
WHERE Category = 'Furniture'
   AND Price BETWEEN 100 AND 500
   AND StockQuantity >= 20;

SELECT *
FROM Suppliers
WHERE SupplierID = 1301
  OR SupplierID = 1302
  OR SupplierID = 1303
  OR SupplierID = 1304;
```

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

5. Problem Statement

A financial management application needs to manage accounts and their transactions. You are tasked with writing queries to create and manipulate these tables. Assume that current date is '2024-01-01'. The table structure is shown below:

Sample Input Records:

Tasks to do:

Insert the above-mentioned records into the Transactions and Accounts table respectively.Select all the records from the Transaction and Accounts tableDelete transactions where TransactionDate is older than '2023-01-01' AND Amount is greater than $200 OR TransactionType is 'Transfer'.Select transactions where the Amount is greater than $1000 AND TransactionDate is within the last 10 days AND AccountID is not 501.Update the Balance of accounts where AccountID is 103 by increasing it by $1000.

Complete the tasks using Oracle SQL queries

Answer

oracle.sql

```
INSERT INTO Accounts (AccountID, AccountHolder, Balance) VALUES (101, 'John
Doe', 1500);
INSERT INTO Accounts (AccountID, AccountHolder, Balance) VALUES (102, 'Jane
Smith', 2500);
INSERT INTO Accounts (AccountID, AccountHolder, Balance) VALUES (103, 'Jim
Brown', 750);
INSERT INTO Accounts (AccountID, AccountHolder, Balance) VALUES (104, 'Jake
Blues', 500);
INSERT INTO Accounts (AccountID, AccountHolder, Balance) VALUES (105, 'Jill
Green', 1000);

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (201, 101, TO_DATE('2023-07-01', 'YYYY-MM-DD'), 600, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (202, 102, TO_DATE('2023-06-15', 'YYYY-MM-DD'), 800, 'Withdrawal');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (203, 103, TO_DATE('2023-05-10', 'YYYY-MM-DD'), 1200, 'Transfer');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (204, 104, TO_DATE('2023-04-20', 'YYYY-MM-DD'), 150, 'Deposit');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (205, 105, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 500, 'Withdrawal');
```

```
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (206, 104, TO_DATE('2023-02-10', 'YYYY-MM-DD'), 450, 'Transfer');

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
VALUES (207, 105, TO_DATE('2023-01-25', 'YYYY-MM-DD'), 300, 'Deposit');

SELECT * FROM transactions;
SELECT * FROM Accounts;

DELETE FROM Transactions
WHERE TransactionDate < TO_DATE('2023-01-01', 'YYYY-MM-DD')
   AND Amount > 200
   OR TransactionType = 'Transfer';

SELECT *
FROM Transactions
WHERE Amount > 1000
   AND TransactionDate >= TO_DATE('2024-01-01', 'YYYY-MM-DD') - 10
   AND AccountID != 501;

UPDATE Accounts
SET Balance = Balance + 1000
WHERE AccountID = 103;
```

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