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_PAH

Attempt : 2
Total Mark : 50
Marks Obtained : 50

Section 1 : COD

1. Problem Statement

Arira is a career consultant who works with various companies to help them find suitable candidates for their job openings. She uses an online job portal that manages job postings and applications. To implement this task the system needs to handle three main types of data: EMPLOYERS, JOBPOSTINGS, APPLICATIONS

Write the DDL statements to create the following tables

symbol refers to the primary key

NN refers to Not NULL

Arira needs to:

Create an EMPLOYERS table to store employer details.Create a JOBPOSTINGS table to store job posting details.Create an APPLICATIONS table to store application details.

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

Answer

oracle.sql

```
CREATE TABLE EMPLOYERS (
   EMPLOYERID NUMBER,
   COMPANYNAME VARCHAR2(100),
   EMAIL VARCHAR2(100)
);

CREATE TABLE JOBPOSTINGS (
   JOBID NUMBER,
   EMPLOYERID NUMBER,
   JOBTITLE VARCHAR2(100)
);

CREATE TABLE APPLICATIONS (
   APPLICATIONID NUMBER,
   JOBID NUMBER,
   APPLICANTNAME VARCHAR2(100)
);
```

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

2. Problem Statement

Emily is developing a digital content platform to manage creators, content, and subscriptions. The system needs to handle data across three main types: CREATORS, CONTENT, and SUBSCRIPTIONS.

Write the DDL statements to create the following tables:

Emily tasks involve:

Create a CREATORS table to store creator details.Create a CONTENT table to store content details.Create a SUBSCRIPTIONS table to store subscription details.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 CREATORS (
    CREATORID NUMBER,
    CHANNELNAME VARCHAR2(100),
    EMAIL VARCHAR2(100),
    DATEJOINED DATE
);

CREATE TABLE CONTENT (
    CONTENTID NUMBER,
    CREATORID NUMBER,
    TITLE VARCHAR2(255),
    DESCRIPTION VARCHAR2(500),
    DATEPUBLISHED DATE
);

CREATE TABLE SUBSCRIPTIONS (
    SUBSCRIPTIONID NUMBER,
    CREATORID NUMBER,
    SUBSCRIBERID NUMBER,
    SUBSCRIPTIONDATE DATE
);
```

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

3. Problem Statement

Mia is developing a food delivery application to manage restaurants, menus, and orders. The system needs to handle data across three main types: RESTAURANTS, MENUS, and ORDERS.

Write the DDL statements to create the following tables:

NN refers to Not NULL

Mia needs to:

Create a RESTAURANTS table to store restaurant details.Create a MENUS table to store menu itemsCreate an ORDERS table to store customer orders.Alter the RESTAURANTS table to add a new column EstablishedYear to store the year the restaurant was established in the datatype NUMBER(4)Alter the MENUS table to add a new column Category to store the type of dish (e.g., vegetarian, non-vegetarian, dessert) in the datatype VAECHAR(50).

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 RESTAURANTS (
   RestaurantID NUMBER,
   RestaurantNAME VARCHAR2(100),
   Location VARCHAR2(200),
   ContactNumber VARCHAR2(15),
   Email VARCHAR2(100)
);

CREATE TABLE MENUS (
   MenuID NUMBER,
   RestaurantID NUMBER,
   DishName VARCHAR2(100),
   Price NUMBER,
   Description VARCHAR2(500)
);

CREATE TABLE ORDERS (
   OrderID NUMBER,
   CustomerID NUMBER,
   RestaurantID NUMBER,
   OrderDate DATE,
```

```
    Quantity NUMBER,
    TotalAmount NUMBER
);

ALTER TABLE RESTAURANTS
ADD EstablishedYear NUMBER(4);

ALTER TABLE MENUS
ADD Category VARCHAR2(50);
```

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

4. Problem Statement

Diego is working as an intern at a vacation rental company. His task is to manage and analyze rental and booking data. The database admin has already created two tables, Rentals and Bookings. Diego needs to perform several operations on these tables based on specific conditions.

Table Name: Bookings

Table Name: Rentals

Sample Input Records:

Tasks for Diego

Alter the Rentals table to add a new column OwnerID to store the ID of the owner of the rental.Alter the Rentals table to change the data type of the PricePerNight column to DECIMAL(12, 2) to accommodate larger price values.Describe Bookings TableDescribe Rentals Table

Note:

Ensure that your CREATE and ALTER queries are correctly formatted and use appropriate data types for Oracle.

Answer

oracle.sql

```
ALTER TABLE RENTALS ADD OWNERID NUMBER;

ALTER TABLE RENTALS MODIFY PRICEPERNIGHT DECIMAL(12, 2);

DESCRIBE BOOKINGS;
DESCRIBE RENTALS;
```

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

5. Problem Statement

Alex is working on managing data for a restaurant management system. The database admin has already created Restaurants and Orders. Alex needs to perform a series of operations to maintain and analyze this data. The tasks involve inserting records, updating restaurant details, deleting outdated orders, and selecting relevant records based on specific conditions. The table structure is shown below:

Sample Input Records:

Tasks for Alex:

Insert Records: Insert the above-mentioned records into the Restaurants and Orders tables respectively.

Update Locations: Update the Location of restaurants is Rome where CuisineType is 'Italian' AND Location is not 'Rome'.

Delete Orders: Delete orders where OrderDate is before '2023-12-01' AND Amount is greater than $50 OR RestaurantID is 901.

Select Orders: Select all orders where OrderDate is between '2024-06-15'

and '2024-07-15' AND Amount is greater than $20 AND RestaurantID is not 1001.

Select Customers: Select distinct customers who have placed orders where the Amount is greater than $25 AND OrderDate is between '2024-06-01' and '2024-07-15' OR RestaurantID is 2.

Your task is to help Alex in implementing the same.

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 Restaurants (RestaurantID, RestaurantName, Location, CuisineType) VALUES (1, 'Pasta Paradise', 'New York', 'Italian');
INSERT INTO Restaurants (RestaurantID, RestaurantName, Location, CuisineType) VALUES (2, 'Sushi World', 'Tokyo', 'Japanese');
INSERT INTO Restaurants (RestaurantID, RestaurantName, Location, CuisineType) VALUES (3, 'Taco Havem', 'Los Angeles', 'Mexican');
INSERT INTO Restaurants (RestaurantID, RestaurantName, Location, CuisineType) VALUES (4, 'Curry Corner', 'London', 'Indian');
INSERT INTO Restaurants (RestaurantID, RestaurantName, Location, CuisineType) VALUES (5, 'Pasta Express', 'Chicago', 'Italian');

INSERT INTO Orders (OrderID, RestaurantID, CustomerID, OrderDate, Amount) VALUES (1, 1, 101, DATE '2024-07-15', 45.00);
INSERT INTO Orders (OrderID, RestaurantID, CustomerID, OrderDate, Amount) VALUES (2, 2, 102, DATE '2024-06-20', 55.00);
INSERT INTO Orders (OrderID, RestaurantID, CustomerID, OrderDate, Amount) VALUES (5, 5, 105, DATE '2024-06-30', 25.00);
INSERT INTO Orders (OrderID, RestaurantID, CustomerID, OrderDate, Amount) VALUES (6, 6, 106, DATE '2024-05-15', 60.00);
INSERT INTO Orders (OrderID, RestaurantID, CustomerID, OrderDate, Amount) VALUES (8, 901, 108, DATE '2024-07-01', 20.00);
```

```
UPDATE Restaurants
SET Location = 'Rome'
WHERE CuisineType = 'Italian'
AND Location <> 'Rome';

DELETE FROM Orders
WHERE (OrderDate < DATE '2023-12-01' AND Amount > 50)
OR RestaurantID = 901;

SELECT * FROM Orders
WHERE OrderDate >= DATE '2024-06-15'
AND OrderDate <= DATE '2024-07-15'
AND Amount > 20
AND RestaurantID <> 1001;

SELECT DISTINCT CustomerID
FROM Orders
WHERE (Amount > 25
AND OrderDate >= DATE '2024-06-01'
AND OrderDate <= DATE '2024-07-15')
OR RestaurantID = 2;
```

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