

# 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\_COD**

Attempt : 2

Total Mark : 50

Marks Obtained : 50

#### **Section 1 : COD**

##### **1. Problem Statement**

Alice is developing a blockchain-based supply chain tracking system to manage the movement of products from manufacturers to customers. The system needs to handle three main types of data: Products, Shipments, and Transactions.

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

Table Structure:

Tasks for Alice:

Create the Products table with the following columns: ProductID,

ProductName, Category, Price, and StockQuantity. Create the Shipments table with the following columns: ShipmentID, ProductID, ShipmentDate, Destination, and Quantity. Create the Transactions table with the following columns: TransactionID, ShipmentID, TransactionDate, Amount, and PaymentStatus. Rename the Products table to SupplyChainProducts to better reflect its role in the system. Alter the Shipments table to add a new column ShipmentStatus to track the status of each shipment (e.g., Pending, Shipped, Delivered).

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

#### Answer

##### oracle.sql

```
CREATE TABLE PRODUCTS (
    PRODUCTID NUMBER,
    PRODUCTNAME VARCHAR2(255),
    CATEGORY VARCHAR2(100),
    PRICE DECIMAL(10, 2),
    STOCKQUANTITY NUMBER
);
```

```
CREATE TABLE SHIPMENTS (
    SHIPMENTID NUMBER,
    PRODUCTID NUMBER,
    SHIPMENTDATE DATE,
    DESTINATION VARCHAR2(255),
    QUANTITY NUMBER
);
```

```
CREATE TABLE TRANSACTIONS (
    TRANSACTIONID NUMBER,
    SHIPMENTID NUMBER,
    TRANSACTIONDATE DATE,
    AMOUNT DECIMAL(10, 2),
    PAYMENTSTATUS VARCHAR2(50)
);
```

```
ALTER TABLE PRODUCTS RENAME TO SUPPLYCHAINPRODUCTS;
```

```
ALTER TABLE SHIPMENTS ADD SHIPMENTSTATUS VARCHAR2(50);
```

Status : Correct

Marks : 10/10

## 2. Problem Statement

Lucas is developing a food delivery application and needs to store data about orders and restaurants. He has two tables: one for restaurants and one for the orders placed at these restaurants.

Table Name: Restaurants

The Restaurants table will store restaurant details like the restaurant name and cuisine type.

Table Name: Orders

The Orders table will store details about customer orders, including order ID, restaurant ID, and the total price of the order.

Sample Input Records:

Task for Lucas:

Create the Restaurants table to store restaurant details, including RestaurantID, RestaurantName, and CuisineType. Create the Orders table to store order details, including OrderID, RestaurantID, and TotalPrice. Insert at least 5 records into the Restaurants table with the following fields: RestaurantID, RestaurantName, and CuisineType. Insert at least 5 records into the Orders table with the following fields: OrderID, RestaurantID, and TotalPrice.

Note: The user must write only the query to create and insert the sample records in the tables. The Query to display the table is predefined.

**Answer**

oracle.sql

CREATE TABLE RESTAURANTS (

```
RESTAURANTID NUMBER,  
RESTAURANTNAME VARCHAR2(30),  
CUISINETYPE VARCHAR2(30)  
);
```

```
CREATE TABLE ORDERS (
    ORDERID NUMBER,
    RESTAURANTID NUMBER,
    TOTALPRICE NUMBER
);
```

```
INSERT INTO RESTAURANTS (RESTAURANTID, RESTAURANTNAME,  
CUISINETYPE)  
VALUES (1, 'PIZZA PALACE', 'ITALIAN');
```

```
INSERT INTO RESTAURANTS (RESTAURANTID, RESTAURANTNAME,  
CUISINETYPE)  
VALUES (2, 'SUSHI SPOT', 'JAPANESE');
```

```
INSERT INTO RESTAURANTS (RESTAURANTID, RESTAURANTNAME,  
CUISINETYPE)  
VALUES (3, 'BURGER KING', 'AMERICAN');
```

```
INSERT INTO ORDERS (ORDERID, RESTAURANTID, TOTALPRICE)  
VALUES (1, 1, 25);
```

```
INSERT INTO ORDERS (ORDERID, RESTAURANTID, TOTALPRICE)  
VALUES (2, 2, 30);
```

```
INSERT INTO ORDERS (ORDERID, RESTAURANTID, TOTALPRICE)  
VALUES (3, 3, 15);
```

**Status :** Correct

Marks : 10/10

### 3. Problem Statement

Olivia is developing a simple ticket booking system for a local cinema. She needs to create two tables to store data about movies and tickets.

## Table Name: Movies

Table Name: Tickets

The Movies table will store movie details like the movie title and its release year.

Sample Input Record:

</strong>

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

**Answer**

oracle.sql

```
CREATE TABLE MOVIES (
    MOVIETITLE VARCHAR2(30),
    RELEASEYEAR NUMBER
);
```

```
CREATE TABLE TICKETS (
    TICKETID NUMBER,
    MOVIEID NUMBER,
    TICKETPRICE NUMBER
);
```

```
INSERT INTO MOVIES (MOVIETITLE, RELEASEYEAR) VALUES ('THE AVENGERS', 2012);
```

```
INSERT INTO MOVIES (MOVIETITLE, RELEASEYEAR) VALUES ('IRON MAN', 2008);
```

```
INSERT INTO MOVIES (MOVIETITLE, RELEASEYEAR) VALUES ('BLACK PANTHER', 2018);
```

```
INSERT INTO TICKETS (TICKETID, MOVIEID, TICKETPRICE) VALUES (1, 1, 10);
```

```
INSERT INTO TICKETS (TICKETID, MOVIEID, TICKETPRICE) VALUES (2, 2, 12);
```

INSERT INTO TICKETS (TICKETID, MOVIEID, TICKETPRICE) VALUES (3, 3, 15);

Status : Correct

Marks : 10/10

#### 4. Problem Statement

Syed is tasked with setting up the database schema for managing employee information. The platform requires the following table:

Table Name: Employee

The Employee table will store details such as employee name, street, and city. The following data must be inserted into the table:

Sample Input Records:

</strong>

Note: The user must write only the query to create and insert the sample records in the table.

The query to display the description of the table is already given.

*Answer*

oracle.sql

```
CREATE TABLE EMPLOYEE (
    EMP_NAME VARCHAR2(30),
    STREET VARCHAR2(30),
    CITY VARCHAR2(30)
);
```

INSERT INTO EMPLOYEE (EMP\_NAME, STREET, CITY) VALUES ('ADAM', 'SPRING', 'PITTSFIELD');

INSERT INTO EMPLOYEE (EMP\_NAME, STREET, CITY) VALUES ('BROOKS', 'SENATOR', 'BROOKLYN');

INSERT INTO EMPLOYEE (EMP\_NAME, STREET, CITY) VALUES ('CURRY', 'NORTH', 'RYE');

INSERT INTO EMPLOYEE (EMP\_NAME, STREET, CITY) VALUES ('DEMALO', 'SUNSHINE', 'SAN DEAGO');

Status : Correct

Marks : 10/10

## 5. Problem Statement

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

Table Name: Rentals

Table Name: Bookings

Sample Input Records:

Tasks for Diego

Insert Records: Insert at least 5 records into the Rentals and Bookings tables with the values mentioned.

Update Prices: Apply a 20% discount to rentals located in 'Forest' or 'Mountain' with a price greater than \$120.

Delete Bookings: Delete bookings made before '2024-02-01' where the customer ID is less than 203.

Select Rentals: Retrieve rentals that are available in July or August 2024 and priced under \$100.

Note:

Ensure that your INSERT, UPDATE, DELETE, and SELECT queries are correctly formatted and use appropriate date and number formats for Oracle SQL.

Answer

oracle.sql

```
INSERT INTO RENTALS (RENTALID, LOCATION, PRICEPERNIGHT,  
AVAILABLEFROM, AVAILABLETO)  
VALUES (1, 'BEACH', 120.00, TO_DATE('2024-07-25', 'YYYY-MM-DD'),  
TO_DATE('2024-09-10', 'YYYY-MM-DD'));
```

```
INSERT INTO RENTALS (RENTALID, LOCATION, PRICEPERNIGHT,  
AVAILABLEFROM, AVAILABLETO)  
VALUES (2, 'MOUNTAIN', 250.00, TO_DATE('2024-08-01', 'YYYY-MM-DD'),  
TO_DATE('2024-10-01', 'YYYY-MM-DD'));
```

```
INSERT INTO RENTALS (RENTALID, LOCATION, PRICEPERNIGHT,  
AVAILABLEFROM, AVAILABLETO)  
VALUES (3, 'CITY', 90.00, TO_DATE('2024-07-20', 'YYYY-MM-DD'),  
TO_DATE('2024-08-30', 'YYYY-MM-DD'));
```

```
INSERT INTO RENTALS (RENTALID, LOCATION, PRICEPERNIGHT,  
AVAILABLEFROM, AVAILABLETO)  
VALUES (4, 'BEACH', 80.00, TO_DATE('2024-07-28', 'YYYY-MM-DD'),  
TO_DATE('2024-09-15', 'YYYY-MM-DD'));
```

```
INSERT INTO RENTALS (RENTALID, LOCATION, PRICEPERNIGHT,  
AVAILABLEFROM, AVAILABLETO)  
VALUES (5, 'FOREST', 150.00, TO_DATE('2024-07-22', 'YYYY-MM-DD'),  
TO_DATE('2024-09-20', 'YYYY-MM-DD'));
```

```
INSERT INTO BOOKINGS (BOOKINGID, RENTALID, CUSTOMERID, BOOKINGDATE)  
VALUES (1, 1, 201, TO_DATE('2024-07-01', 'YYYY-MM-DD'));
```

```
INSERT INTO BOOKINGS (BOOKINGID, RENTALID, CUSTOMERID, BOOKINGDATE)  
VALUES (2, 2, 202, TO_DATE('2023-12-15', 'YYYY-MM-DD'));
```

```
INSERT INTO BOOKINGS (BOOKINGID, RENTALID, CUSTOMERID, BOOKINGDATE)  
VALUES (3, 3, 203, TO_DATE('2024-01-10', 'YYYY-MM-DD'));
```

```
INSERT INTO BOOKINGS (BOOKINGID, RENTALID, CUSTOMERID, BOOKINGDATE)  
VALUES (4, 4, 204, TO_DATE('2023-11-20', 'YYYY-MM-DD'));
```

```
INSERT INTO BOOKINGS (BOOKINGID, RENTALID, CUSTOMERID, BOOKINGDATE)  
VALUES (5, 5, 205, TO_DATE('2024-07-15', 'YYYY-MM-DD'));
```

```
UPDATE RENTALS
```

```
SET PRICEPERNIGHT = PRICEPERNIGHT * 0.8  
WHERE LOCATION IN ('FOREST', 'MOUNTAIN') AND PRICEPERNIGHT > 120;
```

```
DELETE FROM BOOKINGS  
WHERE BOOKINGDATE < TO_DATE('2024-02-01', 'YYYY-MM-DD') AND  
CUSTOMERID < 203;
```

```
SELECT * FROM RENTALS  
WHERE PRICEPERNIGHT < 100 AND AVAILABLEFROM <= TO_DATE('2024-08-31',  
'YYYY-MM-DD') AND AVAILABLETO >= TO_DATE('2024-07-01', 'YYYY-MM-DD');
```

**Status :** Correct

**Marks :** 10/10