

# LMS Database Practical

♦ Create below tables and also add required primary key, foreign key and constraints in it.

- User,
- Product,
- Order (with User ID, Order status and expected delivery date)
- Order detail (order id, product ids)

## Queries:

- **Creating Database**

```
CREATE DATABASE user_details;
```

```
USE user_details;
```

- **Creating Tables**

### 1. Creating **Users** Table

```
CREATE TABLE Users
```

```
(user_id INTEGER NOT NULL PRIMARY KEY,
```

```
first_name VARCHAR(50),
```

```
last_name VARCHAR(50),
```

```
email_id VARCHAR(100),
```

```
contact_no BIGINT,
```

```
gender VARCHAR(10));
```

## 2. Creating **Products** Table

```
CREATE TABLE Products
(product_id INTEGER NOT NULL PRIMARY KEY,
product_name VARCHAR(50),
price INTEGER);
```

## 3. Creating **Orders** Table

```
CREATE TABLE Orders
(order_id INTEGER NOT NULL,
user_id INTEGER,
product_id INTEGER,
order_status VARCHAR(50),
order_placed_date DATE,
expected_delivery_date DATE,
FOREIGN KEY(user_id) REFERENCES Users(user_id) ON DELETE
SET NULL,
FOREIGN KEY(product_id) REFERENCES Products(product_id)
ON DELETE SET NULL);
```

## 4. Creating **Order\_details** Table

```
CREATE TABLE Order_details
(order_id INTEGER NOT NULL ,
product_id INTEGER NOT NULL,
PRIMARY KEY(order_id, product_id));
```

- **Inserting Values**

### **1. Inserting Values into `Users` Table**

```
INSERT INTO Users VALUES
```

```
(1, 'Juhi', 'Mehta', 'juhim@gmail.com', 5678435624, 'Female');
```

```
INSERT INTO Users VALUES
```

```
(2, 'Janvi', 'Sureja', 'janvis@gmail.com', 3546782345, 'Female');
```

```
INSERT INTO Users VALUES
```

```
(3, 'Dhruvi', 'Patel', 'dhruvip@gmail.com', 4523894676, 'Female');
```

```
INSERT INTO Users VALUES
```

```
(4, 'Meet', 'Radadia', 'meettr@gmail.com', 5438765126, 'Male');
```

```
INSERT INTO Users VALUES
```

```
(5, 'Tirth', 'Bhatt', 'tirthb@gmail.com', 6723945871, 'Male');
```

```
INSERT INTO Users VALUES
```

```
(6, 'Dev', 'Kapadia', 'devk@gmail.com', 2365975678, 'Male');
```

```
INSERT INTO Users VALUES
```

```
(7, 'Pakshal', 'Ghiya', 'pakshalg@gmail.com', 6598357612, 'Male');
```

```
INSERT INTO Users VALUES
```

```
(8, 'Khushali', 'Shah', 'khushalis@gmail.com', 6237834509, 'Female');
```

```
INSERT INTO Users VALUES
```

```
(9, 'Hinal', 'Panchal', 'hinalp@gmail.com', 3487561245, 'Female');
```

## 2. Inserting Values into **Products** Table

```
INSERT INTO Products VALUES
```

```
(101, 'Apple Iphone 14', 69000);
```

```
INSERT INTO Products VALUES
```

```
(102, 'Lenovo Thinkpad', 73000);
```

```
INSERT INTO Products VALUES
```

```
(103, 'OnePlus Z Buds', 3499);
```

```
INSERT INTO Products VALUES
```

```
(104, 'Asus VivoBook', 89999);
```

```
INSERT INTO Products VALUES
```

```
(105, 'Phillips Hair Straightening Brush', 2500);
```

```
INSERT INTO Products VALUES
```

```
(106, 'Vella Hair Dryer', 1800);
```

```
INSERT INTO Products VALUES
```

```
(107, 'Loreal Paris Shampoo', 1149);
```

```
INSERT INTO Products VALUES
```

```
(108, 'Apple MacBook', 119000);
```

```
INSERT INTO Products VALUES
```

```
(109, 'Apple AirPods', 18000);
```

```
INSERT INTO Products VALUES
```

```
(110, 'Samsung Galaxy Watch 3', 21199);
```

### 3. Inserting Values into **Orders** Table

```
INSERT INTO Orders VALUES
(1, 3, 107, 'Shipped', '2023-03-13', '2023-03-17');
INSERT INTO Orders VALUES
(2, 1, 108, 'Delivered', '2023-03-10', NULL);
INSERT INTO Orders VALUES
(3, 5, 102, 'Not Shipped', '2023-03-14', '2023-03-19');
INSERT INTO Orders VALUES
(4, 8, 105, 'Shipped', '2023-03-14', '2023-03-16');
INSERT INTO Orders VALUES
(5, 2, 107, 'Shipped', '2023-03-14', '2023-03-18');
INSERT INTO Orders VALUES
(6, 7, 110, 'Not Shipped', '2023-03-14', '2023-03-20');
INSERT INTO Orders VALUES
(7, 4, 101, 'Shipped', '2023-03-12', '2023-03-16');
INSERT INTO Orders VALUES
(8, 9, 106, 'Delivered', '2023-03-04', NULL);
INSERT INTO Orders VALUES
(9, 1, 104, 'Not Shipped', '2023-03-15', '2023-03-21');
INSERT INTO Orders VALUES
(10, 8, 109, 'Delivered', '2023-03-01', NULL);
INSERT INTO Orders VALUES
(11, 4, 110, 'Shipped', '2023-03-13', '2023-03-16');
INSERT INTO Orders VALUES
(12, 8, 106, 'Shipped', '2023-03-14', '2023-03-16');
INSERT INTO Orders VALUES
(13, 1, 105, 'Shipped', '2023-03-14', '2023-03-16');
INSERT INTO Orders VALUES
(14, 5, 103, 'Not Shipped', '2023-03-14', '2023-03-18');
```

INSERT INTO Orders VALUES

(15, 2, 107, 'Not Shipped', '2023-03-15', '2023-03-19');

- **Displaying Values**

**1. SELECT \* FROM Users;**

user_id	first_name	last_name	email_id	contact_no	gender
1	Juhi	Mehta	juhim@gmail.com	5678435624	Female
2	Janvi	Sureja	janvis@gmail.com	3546782345	Female
3	Dhruvi	Patel	dhruvip@gmail.com	4523894676	Female
4	Meet	Radadia	meet@gmail.com	5438765126	Male
5	Tirth	Bhatt	tirthb@gmail.com	6723945871	Male
6	Dev	Kapadia	devk@gmail.com	2365975678	Male
7	Pakshal	Ghiya	pakshalg@gmail.com	6598357612	Male
8	Khushali	Shah	khushalis@gmail.com	6237834509	Female
9	Hinal	Panchal	Hinalp@gmail.com	3487561245	Female
NULL	NULL	NULL	NULL	NULL	NULL

**2. SELECT \* FROM Products;**

product_id	product_name	price
101	Apple Iphone 14	69000
102	Lenovo Thinkpad	73000
103	OnePlus Z Buds	3499
104	Asus VivoBook	89999
105	Phillips Hair Straightening Brush	2500
106	Vella Hair Dryer	1800
107	Loreal Paris Shampoo	1149
108	Apple MacBook	119000
109	Apple AirPods	18000
110	Samsung Galaxy Watch 3	21199
NULL	NULL	NULL

### 3. SELECT \* FROM Orders;

order_id	user_id	product_id	order_status	order_placed_date	expected_delivery_date
1	3	107	Shipped	2023-03-13	2023-03-17
2	1	108	Delivered	2023-03-10	NULL
3	5	102	Not Shipped	2023-03-14	2023-03-19
4	8	105	Shipped	2023-03-14	2023-03-16
5	2	107	Shipped	2023-03-14	2023-03-18
6	7	110	Not Shipped	2023-03-14	2023-03-20
7	4	101	Shipped	2023-03-12	2023-03-16
8	9	106	Delivered	2023-03-04	NULL
9	1	104	Not Shipped	2023-03-15	2023-03-21
10	8	109	Delivered	2023-03-01	NULL
11	4	110	Shipped	2023-03-13	2023-03-16
12	8	106	Shipped	2023-03-14	2023-03-16
13	1	105	Shipped	2023-03-14	2023-03-16
14	5	103	Not Shipped	2023-03-14	2023-03-18
15	2	107	Not Shipped	2023-03-15	2023-03-19
NULL	NULL	NULL	NULL	NULL	NULL

1) Fetch all the User order list and include at least following details in that.

- Customer name
- Product names
- Order Date
- Expected delivery date (in days, i.e. within X days)

#### Query:

```
SELECT concat(u.first_name, ' ', u.last_name) AS customer_name,  
p.product_name,  
o.order_placed_date, o.order_status,  
coalesce(datediff(o.expected_delivery_date, o.order_placed_date),0) AS  
expected_delivery_in_days  
FROM Orders o INNER JOIN Products p ON p.product_id = o.product_id  
INNER JOIN Users u ON u.user_id = o.user_id  
ORDER BY expected_delivery_in_days DESC;
```

## Output:

customer_name	product_name	order_placed_date	order_status	expected_delivery_in_days
Juhi Mehta	Asus VivoBook	2023-03-15	Not Shipped	6
Pakshal Ghiya	Samsung Galaxy Watch 3	2023-03-14	Not Shipped	6
Tirth Bhatt	Lenovo Thinkpad	2023-03-14	Not Shipped	5
Meet Radadia	Apple Iphone 14	2023-03-12	Shipped	4
Tirth Bhatt	OnePlus Z Buds	2023-03-14	Not Shipped	4
Dhruvi Patel	Loreal Paris Shampoo	2023-03-13	Shipped	4
Janvi Sureja	Loreal Paris Shampoo	2023-03-14	Shipped	4
Janvi Sureja	Loreal Paris Shampoo	2023-03-15	Not Shipped	4
Meet Radadia	Samsung Galaxy Watch 3	2023-03-13	Shipped	3
Khushali Shah	Phillips Hair Straightening Brush	2023-03-14	Shipped	2
Juhi Mehta	Phillips Hair Straightening Brush	2023-03-14	Shipped	2
Khushali Shah	Vella Hair Dryer	2023-03-14	Shipped	2
Hinal Panchal	Vella Hair Dryer	2023-03-04	Delivered	0
Juhi Mehta	Apple MacBook	2023-03-10	Delivered	0
Khushali Shah	Apple AirPods	2023-03-01	Delivered	0

- 1) Create summary report which provide information about
  - i. All undelivered Orders
  - ii. 5 Most recent Orders
  - iii. Top 5 active users (Users having max number of orders)
  - iv. Inactive users (Users who hasn't done any order)
  - v. Top 5 Most purchased products
  - vi. Most expensive and cheapest orders.

## Queries:

### 1. All undelivered Orders

```
SELECT concat(u.first_name, ' ', u.last_name) AS customer_name,
p. product_name ,
o.order_status
FROM Orders o INNER JOIN Products p ON p.product_id = o.product_id
INNER JOIN Users u ON u.user_id = o.user_id
WHERE NOT o.order_status = 'delivered'
ORDER BY o.order_status DESC;
```



### Output:

customer_name	product_name	order_status
Dhruvi Patel	Loreal Paris Shampoo	Shipped
Khushali Shah	Phillips Hair Straightening Brush	Shipped
Janvi Sureja	Loreal Paris Shampoo	Shipped
Meet Radadia	Apple Iphone 14	Shipped
Meet Radadia	Samsung Galaxy Watch 3	Shipped
Khushali Shah	Vella Hair Dryer	Shipped
Juhi Mehta	Phillips Hair Straightening Brush	Shipped
Tirth Bhatt	Lenovo Thinkpad	Not Shipped
Pakshal Ghiya	Samsung Galaxy Watch 3	Not Shipped
Juhi Mehta	Asus VivoBook	Not Shipped
Tirth Bhatt	OnePlus Z Buds	Not Shipped
Janvi Sureja	Loreal Paris Shampoo	Not Shipped

## 2. 5 Most recent Orders

```
SELECT concat(u.first_name, ' ', u.last_name) AS customer_name,  
p.product_name,  
o.order_placed_date  
FROM Orders o INNER JOIN Products p ON p.product_id = o.product_id  
INNER JOIN Users u ON u.user_id = o.user_id  
ORDER BY o.order_placed_date DESC  
LIMIT 5;
```

### Output:

customer_name	product_name	order_placed_date
Juhi Mehta	Asus VivoBook	2023-03-15
Janvi Sureja	Loreal Paris Shampoo	2023-03-15
Tirth Bhatt	Lenovo Thinkpad	2023-03-14
Khushali Shah	Phillips Hair Straightening Brush	2023-03-14
Janvi Sureja	Loreal Paris Shampoo	2023-03-14

### 3. Top 5 active users (Users having maximum number of orders)

```
SELECT concat(u.first_name, ' ', u.last_name) AS customer_name,  
COUNT(o.user_id) AS Total_orders  
FROM Orders o INNER JOIN Users u ON u.user_id = o.user_id  
GROUP BY o.user_id  
ORDER BY Total_orders DESC  
LIMIT 5;
```

#### Output:

customer_name	Total_orders
Juhi Mehta	3
Khushali Shah	3
Janvi Sureja	2
Meet Radadia	2
Tirth Bhatt	2

### 4. Inactive users (Users who hasn't done any order)

```
SELECT concat(u.first_name, ' ', u.last_name) AS customer_name,  
IF(COUNT(o.user_id) = 0, 'Inactive User', 'Active User') AS User_status  
FROM Orders o RIGHT JOIN Users u ON o.user_id = u.user_id  
GROUP BY u.user_id  
HAVING COUNT(o.user_id) = 0;
```

#### Output:

customer_name	User_status
Dev Kapadia	Inactive User

## 5. Top 5 Most purchased products

```
SELECT p.product_id, p.product_name,  
COUNT(o.product_id) AS Total_purchase  
FROM Orders o INNER JOIN Products p ON p.product_id = o.product_id  
GROUP BY o.product_id  
ORDER BY Total_purchase DESC, o.product_id ASC  
LIMIT 5;
```

### Output:

product_id	product_name	Total_purchase
107	Loreal Paris Shampoo	3
105	Phillips Hair Straightening Brush	2
106	Vella Hair Dryer	2
110	Samsung Galaxy Watch 3	2
101	Apple Iphone 14	1

## 6. Most expensive and cheapest orders

```
SELECT *  
FROM (SELECT p.product_id, p.product_name, p.price  
FROM Orders o INNER JOIN Products p ON p.product_id = o.product_id  
ORDER BY p.price DESC  
LIMIT 1) AS Most_expensive_product  
UNION  
SELECT *  
FROM (SELECT p.product_id, p.product_name, p.price  
FROM Orders o INNER JOIN Products p ON p.product_id = o.product_id  
ORDER BY p.price ASC  
LIMIT 1) AS Cheapest_product;
```

**Output:**

product_id	product_name	price	
108	Apple MacBook	119000	
107	Loreal Paris Shampoo	1149	