**Lab-1 Creating database & table:**

**Database 1:**

CREATE DATABASE OnlinePasal;

**Table 1:**

CREATE TABLE Customers

(

customerID int NOT NULL,

customerName varchar(50),

contactName varchar(50),

Address varchar(50),

City varchar(50),

postalCode varchar(50),

Country varchar(50),

PRIMARY KEY(customerID)

);

**Table 2:**

CREATE TABLE Categories

(

catagoryID int NOT NULL,

catagoryName varchar(50),

Description varchar(50),

PRIMARY KEY(catagoryID)

);

**Table 3:**

CREATE TABLE Emmployees

(

employeeID int NOT NULL,

lastName varchar(50),

firstName varchar(50),

birthDate date,

Notes varchar(250),

PRIMARY KEY(employeeID)

);

**Table 4:**

CREATE TABLE Shippers

(

shipperID int NOT NULL,

shipperName varchar(50),

Phone varchar(50),

PRIMARY KEY(shipperID)

);

**Table 5:**

CREATE TABLE Suppliers

(

supplierID int NOT NULL,

supplierName varchar(50),

contactName varchar(50),

Address varchar(50),

City varchar(50),

postalCode varchar(50),

Country varchar(50),

Phone varchar(50),

PRIMARY KEY(shipperID)

);

**Table 6:**

CREATE TABLE Products

(

productID int NOT NULL,

productName varchar(50),

supplierID int,

catagoryID int,

Unit int,

Price int,

PRIMARY KEY(productID),

FOREIGN KEY(supplierID) REFERENCES suppliers(supplierID)

ON DELETE CASCADE,

FOREIGN KEY(catagoryID) REFERENCES categories(catagoryID)

ON DELETE CASCADE

);

**Table 7:**

CREATE TABLE orders

(

orderID int NOT NULL,

customerID int,

employeeID int,

orderDate date,

shipperID int,

PRIMARY KEY(orderID),

FOREIGN KEY(customerID) REFERENCES customers(customerID)

ON DELETE CASCADE,

FOREIGN KEY(employeeID) REFERENCES emmployees(employeeID)

ON DELETE CASCADE,

FOREIGN KEY(shipperID) REFERENCES shippers(shipperID)

ON DELETE CASCADE

);

**Table 8:**

CREATE TABLE orderDetails

(

orderDetailID int NOT NULL,

orderID int,

productID int,

Quantity int,

PRIMARY KEY(orderDetailID),

FOREIGN KEY(orderID) REFERENCES orders(orderID)

ON DELETE CASCADE,

FOREIGN KEY(productID) REFERENCES products(productID)

ON DELETE CASCADE

);

**Lab-2 Inserting Data:**

**Inserting in categories:**

INSERT INTO `categories` (`catagoryID`, `catagoryName`, `Description`) VALUES

(1, 'Beverages', 'Soft Drinks, coffees, teas, beers, and ales'),

(2, 'Condiments', 'Sweet and savory sauces, relishes, spreads, and seasoings'),

(3, 'Confections', 'Deserts, candies, and sweet breads'),

(4, 'Dairy Products', 'Cheeses');

**Inserting in customer:**

INSERT INTO `customers` (`customerID`, `customerName`, `contactName`, `Address`, `City`, `postalCode`, `Country`) VALUES

(1, 'Alfreds Futterkiste', 'Alfred Schmidt', 'Obere Str. 57', 'Frankfurt', '12209', 'Germany'),

(2, 'Ana Trujilo Emparedados yhelados', 'Ana Trujilo', 'Avda. de la Constitucion 2222', 'Mexico D.F.', '05021', 'Mexico'),

(3, 'Antonio Moreno Taqueria', 'Antonio Moreno', 'Mataderos 2312', 'Mexico D.F.', '05023', 'Mexico'),

(4, 'Around the Horn', 'Thomas Hardy', '120 Hanover Sq.', 'London', 'WA1 1DP', 'UK'),

(5, 'Berglunds snabbkop', 'Christina Berglund', 'berguvsvagen 8', 'Lulea', 'S-958 22', 'Sweden'),

(6, 'Biauer See Delikatessen', 'Hanna Moos', 'Forsterstr, 57', 'Mannheim', '68306', 'Germany');

**Inserting in emmployees:**

INSERT INTO `emmployees` (`employeeID`, `lastName`, `fistName`, `birthDate`, `Notes`) VALUES

(1, 'Davolio', 'Nancy', '1968-12-08', 'Education includes a BA in psychology from Colorado State University. She also completed (The Art of the Cold Cal). Nancy is a member of \'toastmasters International.\''),

(2, 'Fuller', 'Andrew', '1952-02-19', 'Andrew received jis BTS commercial and Ph.D. in international marketing from the University of Dallas. He is fluent in French and Italian and reads German. He joined the company as a sales representative'),

(3, 'Leverling', 'Janet', '1963-08-30', 'Janet has a BS degree in chemistry from Boston college. She has also completed a certificate program in food retailing management. Janet was hired as a sales associate and was promoted to sales representative');

**Inserting in orderdetails:**

INSERT INTO `orderdetails` (`orderDetailID`, `orderID`, `productID`, `Quantity`) VALUES

(1, 10248, 11, 12),

(2, 10248, 42, 10),

(3, 10248, 72, 5),

(4, 10249, 14, 9),

(5, 10249, 51, 40),

(6, 10250, 41, 10);

**Inserting in order:**

INSERT INTO `orders` (`orderID`, `customerID`, `employeeID`, `orderDate`, `shipperID`) VALUES

(10248, 1, 1, '1996-07-04', 3),

(10249, 2, 2, '1996-07-05', 1),

(10250, 3, 3, '1996-07-08', 2),

(10251, 4, 3, '1996-07-08', 1),

(10252, 5, 2, '1996-07-09', 2),

(10253, 6, 3, '1996-07-10', 2);

**Inserting in products:**

INSERT INTO `products` (`productID`, `productName`, `supplierID`, `catagoryID`, `Unit`, `Price`) VALUES

(1, 'Chais', 1, 1, 10, '18'),

(2, 'Chang', 1, 1, 24, '19'),

(3, 'Anissed Syrup', 1, 2, 12, '10'),

(4, 'Chef Anton\'s Cajun Seasoning', 2, 2, 48, '22'),

(5, 'Chef Anton\'s Gumbo mix', 2, 2, 36, '21'),

(6, 'Grandma\'s Boysenberry Spread', 3, 2, 12, '25'),

(7, 'Uncle bob\'s Organic Dried Pears', 3, 2, 12, '30');

**Inserting in shippers:**

INSERT INTO `shippers` (`shipperID`, `shipperName`, `Phone`) VALUES

(1, 'Speedy Express', '(503) 555-9831'),

(2, 'United Package', '(503) 555-3109'),

(3, 'Federal Shipping', '(503) 555-9931');

**Inserting in suppliers:**

INSERT INTO `suppliers` (`supplierID`, `supplierName`, `contactName`, `Address`, `City`, `postalCode`, `County`, `Phone`) VALUES

(1, 'Exotic Liquid', 'Charlotte Copper', '49 Gibert St.', 'Londona', 'EC1 4SD', 'UK', '(171) 555-2222'),

(2, 'New Orleans Cajun Delights', 'Shelley Burke', 'P.O. Box 78934', 'New Orleans', '7117', 'USA', '100 555-4822'),

(3, 'Grandma Kelly\'s Homestead', 'Regina Murphy', '707 Oxford Rd.', 'Ann Arbor', '48104', 'USA', '(313) 555-5735'),

(4, 'Tookyo Traders', 'Yoshi Nagase', '9-8 Sekimai Musashino-shi', 'Tokyo', '100', 'Japan', '(03) 3555-5011'),

(5, 'Cooperativa de Quesos', 'Antoni dei Valle Seavdra', 'Calle dek Rosal 4', 'Oviedo', '33007', 'Spain', '(98) 598 76 54');

**Lab-3 Searching in database:**

SELECT \* from customer where country ='germany';

SELECT customer\_ID, customerName, postalcode FROM customer;

**Search by firstname:**

SELECT \* FROM customer WHERE customerName LIKE "a%";

**Search between:**

SELECT \* FROM products where productID BETWEEN 3 AND 6;

SELECT \* FROM products where productID NOT BETWEEN 3 and 6;

**Updating in database:**

UPDATE orderdetails SET quantity = quantity \* 1.1;

UPDATE products SET price = price +5;

**Temporary Update:**

SELECT productId, productName, price\*1.20 FROM products;

**Temporary Update with name change:**

SELECT productId, productName, price\*1.20 AS IncreasedPrice FROM products;

**Deleting:**

DELETE FROM orderdetails WHERE quantity<7;

**Listing:**

SELECT \* FROM `suppliers` WHERE city IN('Londona', 'Tokyo');

**Alternatively:**

SELECT \* FROM `suppliers` WHERE city ='Londona' or city= 'tokyo';

**Aggregate function:**

**Average:**

SELECT AVG(Price) AS AveragePrice FROM products;

**Sum:**

SELECT SUM(Price) AS Sum FROM products;

**Maximum:**

SELECT Max(Price) AS Sum FROM products;

**Minimum:**

SELECT Min(Price) AS Mininum FROM products;

**Consider a table with different data, now print the maximum and minimum salary of different city.**

SELECT Address, MIN(Salary) AS minimumSalary, MAX(Salary) AS MaximumSalay FROM tbl\_emp GROUP BY Address;

**Display the maximum salary for each address where the maximum salary is greater than 1000.**

SELECT Address, MAX(Salary) AS MaximumSalary FROM tbl\_emp GROUP BY Address HAVING MAX(Salary)>10000;

**Display the average salary paid for all address except Butwal(btl).**

SELECT address, AVG(Salary) AS AverageSalary FROM tbl\_emp GROUP BY address HAVING address <> 'btl';

**Lab-4 Subquery/Nested query:**

**1. Display all the records of employee who have salary greater than that of Rita.**

SELECT \* FROM tbl\_emp WHERE salary > (SELECT salary from tbl\_emp WHERE name ='Rita');

**2. Display all records of employee whose salary is greater than average salary of all employees.**

SELECT \* FROM tbl\_emp WHERE salary >( SELECT AVG(salary) FROM tbl\_emp);

**3. Display address of employees whose address is that of Rita or Sita.**

SELECT Address FROM tbl\_emp WHERE salary >( SELECT address FROM tbl\_emp WHERE name ='Rita' or 'Sita');

**Display multiple columns from Multiple table:**SELECT orderdetails.OrderDetailID, orders.OrderDate, products.productName FROM orderdetails, orders, products WHERE orderdetails.OrderDetailID = orders.orderID AND orderdetails.ProductID = products.productId;