

**Introduction to Database System (Lab)**

**Spring 2023**

# Final Term Lab

# VERSION II

**Time Allowed: 90 min**

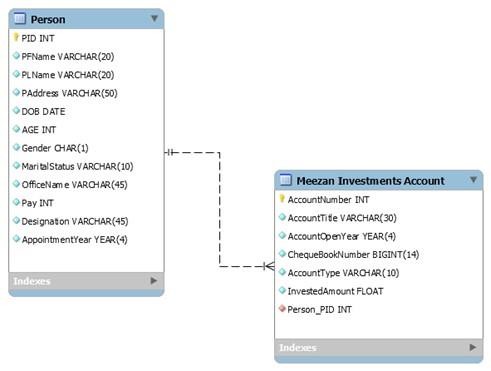
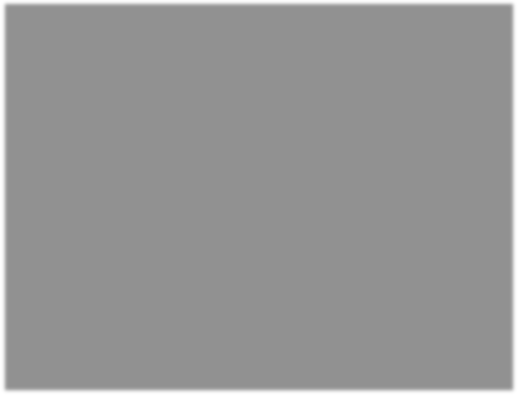
|  |  |
| --- | --- |
| Name: |  |
| Registration No |  |
| Serial Number (of attendance sheet) |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Q1 | Q2 | Total |
| Total Marks | 25 | 25 | 50 |
| Ob. Marks |  |  |  |

**Instructions:**

1. **This is a closed book, closed notes paper.**
2. **Understanding questions is part of the paper. Therefore, no queries will be entertained during examination.**
3. **NorthWind.sql file is uploaded on teams/portal. Use those files to attempt the DML section of paper.**
4. **Use proper indentation/formatting while writing queries. Not properly indenting will deduct 5% marks. 5) You need to make an MS Word with your name and registration should be mentioned on each page.**
5. **You need to write only queries with output in MS Word file, you need to write query (text form) + its output table (picture) if any.**

# Question 1: DDL Marks (25)



**Q1)** Implement the above schema. Create the tables and the relationship between them.

CREATE TABLE Person (

PersonID INT PRIMARY KEY,

FirstName VARCHAR(255),

LastName VARCHAR(255),

Age INT,

Pay INT,

BranchID INT,

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID)

);

CREATE TABLE MeezanInvestmentsAccount (

AccountNumber INT PRIMARY KEY,

AccountType VARCHAR(255),

PersonID INT,

BranchNumber VARCHAR(30),

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

**Q2)** Alter the table **Person** and change the data type of **Pay** from **INT** to **BIGINT.**

**ALTER TABLE Person**

**ALTER COLUMN Pay BIGINT;**

**Q3)** Remove the attribute **Marital Status** from the **Person** table.

ALTER TABLE Person

DROP COLUMN MaritalStatus;

**Q4)** Add a column **Branch Number** to the table **Meezan Investments Account** (Data type must be varchar(30))**.**

ALTER TABLE MeezanInvestmentsAccount

ADD BranchNumber VARCHAR(30);

**Q5)** Drop all the data from the table **Person** from the schema.

DELETE FROM Person;

# Question 2: DML Marks (25)

# Note: Import northwind schema and solve the below questions

**Question 1**

**Write a stored procedure that retrieves the names of customers whose orders have a shipped date in a specific year. The stored procedure should accept the year as a parameter.**

**Hint: [Use EXISTS operator]**

**CREATE PROCEDURE GetCustomersByShippedYear**

**@year INT**

**AS**

**BEGIN**

**SET NOCOUNT ON;**

**SELECT DISTINCT c.CustomerName**

**FROM Customers c**

**WHERE EXISTS (**

**SELECT \***

**FROM Orders o**

**WHERE c.CustomerID = o.CustomerID**

**AND YEAR(o.ShippedDate) = @year**

**);**

**END;**

**Question 2**

Create a view named “ProductOrderStats” that displays the name of every product along with the minimum, maximum, average, and total quantity ordered in all orders.

**Note: [Use Join]**

CREATE VIEW ProductOrderStats AS

SELECT p.ProductName, MIN(od.Quantity) AS MinQuantity, MAX(od.Quantity) AS MaxQuantity,

AVG(od.Quantity) AS AvgQuantity, SUM(od.Quantity) AS TotalQuantity

FROM Products p

JOIN OrderDetails od ON p.ProductID = od.ProductID

GROUP BY p.ProductName;

**Question 3**

Write a co-related query that display name of those customers whose at least one order has status id equals to zero.

**Hint: [Use EXISTS operator]**

**SELECT c.CustomerName**

**FROM Customers c**

**WHERE EXISTS (**

**SELECT \***

**FROM Orders o**

**WHERE c.CustomerID = o.CustomerID**

**AND o.StatusID = 0**

**);**

**Question 4**

Display Name of all products with total amount of all its purchased till now.

**[Use Joins. No cross product / Co-related sub query / nested subquery is allowed.]**

**Hint: You will use products and purchase\_order\_details table to solve this query.**

**SELECT p.ProductName, SUM(pod.Quantity \* pod.UnitPrice) AS TotalAmount**

**FROM Products p**

**JOIN PurchaseOrderDetails pod ON p.ProductID = pod.ProductID**

**GROUP BY p.ProductName;**

**Question 5**

Display name of all products categories with minimum, maximum, average and sum of list\_price of product belongs to it.

**Note: [You need to make one query that display all above values, not separate/individual query]**

SELECT c.CategoryName, MIN(p.ListPrice) AS MinListPrice, MAX(p.ListPrice) AS MaxListPrice,

AVG(p.ListPrice) AS AvgListPrice, SUM(p.ListPrice) AS SumListPrice

FROM Categories c

JOIN Products p ON c.CategoryID = p.CategoryID

GROUP BY c.CategoryName;