DataSense Analytics SQL Master Class

Seafoods Inventory Management System

Project Overview:

Project Title: Seafoods Inventory Management System Database Architecture

Project Description:

The Seafood Inventory Management System, developed using Microsoft SQL Server, acts as a comprehensive repository for organizing and overseeing our company's operational data. This system is specifically designed to facilitate seamless tracking of supplier details, employee information, transportation logistics, supplier payment records, and quality control measures. It streamlines the company's operations, ensuring efficient management of essential facets within the business.

Project Phases and Tasks:

Phase 1: Database Design (DDL)

Task 1: Define the Database Schema

Description: Create the database schema for the Seafood Inventory Management System, defining the tables and their relationships.

```
--CREATING THE SUPPLIERS TABLE
CREATE TABLE suppliers (
    SupplierID INT PRIMARY KEY,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    SupplierAddress VARCHAR(255),
    PhoneNumber VARCHAR(100),
    Email VARCHAR(100),
    DateStarted DATE
);

--CREATING THE EMPLOYEES TABLE
CREATE TABLE employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    EmpAddress VARCHAR(255),
    PhoneNumber VARCHAR(100),
    Email VARCHAR(100),
    Position VARCHAR(100),
    Department VARCHAR(100),
    Salary DECIMAL(10,2),
    DateStarted DATE,
    DateEnded DATE
);
```

```sql
--CREATING THE CUSTOMER TABLE
CREATE TABLE customers (
    CustomerID INT PRIMARY KEY,
    CompanyName VARCHAR(255),
    PhoneNumber VARCHAR(100),
    Email VARCHAR(100),
    CompanyAddress VARCHAR(255)
);

--CREATING THE SEAFOOD INVENTORY TABLE
CREATE TABLE seafoodInventory (
    StockID INT PRIMARY KEY,
    Quantity DECIMAL(10,2),
    DateReceived DATE,
    Date_Delivered DATE
);
--CREATING THE SEAFOOD TYPES TABLE
CREATE TABLE seafoodTypes (
    SeafoodID INT PRIMARY KEY,
    TypeName VARCHAR(150),
    Category VARCHAR(100),
    UOM VARCHAR(100),
    PPU DECIMAL(5,2)
);

--CREATING THE PAYMENTS TO SUPPLIERS TABLE
CREATE TABLE SupplyPayment (
    SupplyPaymentID INT PRIMARY KEY,
    InvoiceNumber INT,
    PaymentDate DATE,
    Amount DECIMAL(10,2)
);

--CREATING THE PAYMENTS FROM CUSTOMERS/COMPANIES TABLE
CREATE TABLE seafoodSales (
    seafoodSalesID INT PRIMARY KEY,
    SalesInvoiceNumber INT,
    SalesDate DATE,
    Amount DECIMAL(10,2)
);

--CREATING THE TRANSPORTATION TABLE
CREATE TABLE Transpo (
    TransportID INT PRIMARY KEY,
    TransportMode VARCHAR(100),
    TransportDate DATE,
    DepartureLoc VARCHAR(255),
    ArrivalLoc VARCHAR(255)
);
```

Task 2: Specify Keys and Constraints
Description: Define primary keys, foreign keys, constraints, and relationships between tables.

```sql
-- DEFINE FOREIGN KEY FOR SEAFOOD INVENTORY TABLE
ALTER TABLE seafoodInventory
ADD SeafoodID INT REFERENCES seafoodTypes(SeafoodID);

ALTER TABLE seafoodInventory
ADD CONSTRAINT FK_seafoodInventory_SeafoodTypes
FOREIGN KEY (SeafoodID) REFERENCES seafoodTypes(SeafoodID);

ALTER TABLE seafoodInventory
ADD SupplierID INT REFERENCES suppliers(SupplierID);

ALTER TABLE seafoodInventory
ADD CONSTRAINT FK_seafoodInventory_suppliers
FOREIGN KEY (SupplierID) REFERENCES suppliers(SupplierID);

-- DEFINE FOREIGN KEY FOR PAYMENTS TO SUPPLIERS TABLE
--CREATE A COLUMN FOR THE FOREIGN KEY
ALTER TABLE SupplyPayment
ADD SupplierID INT REFERENCES suppliers(SupplierID);

ALTER TABLE SupplyPayment
ADD CONSTRAINT FK_SupplyPayment_suppliers
FOREIGN KEY (SupplierID) REFERENCES suppliers(SupplierID);

-- DEFINE FOREIGN KEY FOR PAYMENTS FROM CUSTOMERS/COMPANIES TABLE
--CREATE A COLUMN FOR THE FOREIGN KEY
ALTER TABLE seafoodSales
ADD CustomerID INT REFERENCES customers(CustomerID);

ALTER TABLE seafoodSales
ADD CONSTRAINT FK_seafoodSales_customers
FOREIGN KEY (CustomerID) REFERENCES customers(CustomerID);

-- DEFINE FOREIGN KEY FOR TRANSPORTATION TABLE
--CREATE A COLUMN FOR THE FOREIGN KEY
ALTER TABLE Transpo
ADD CustomerID INT REFERENCES customers(CustomerID);

ALTER TABLE Transpo
ADD CONSTRAINT FK_Transpo_customers
FOREIGN KEY (CustomerID) REFERENCES customers(CustomerID);
```

DataSense Analytics SQL Master Class

Phase 2: Data Population (DML)

Task 3: Populate the Database with Sample Data

Description: Insert sample data into the database, including suppliers, customers, seafood types, seafood inventory, payments, and transportation.

```sql
--Phase 2: Data Population (DML)

--Inserting data to suppliers
INSERT INTO suppliers (SupplierID, FirstName, LastName, SupplierAddress, PhoneNumber, Email, DateStarted)
VALUES
(1001, 'John', 'Doe', '123 Main St, Lucena City, Quezon', '+1234567890', 'john@example.com', '2020-05-10'),
(1002, 'Jane', 'Smith', '456 Oak Ave, Calamba City, Laguna', '+1987654321', 'jane@example.com', '2011-11-20'),
(1003, 'Alice', 'Johnson', '789 Elm St, Santa Rosa, Laguna', '+1112223333', 'alice@example.com', '2013-01-15'),
(1004, 'Bob', 'Williams', '321 Pine St, Batangas City, Batangas', '+4445556666', 'bob@example.com', '2012-08-28'),
(1005, 'Eva', 'Brown', '555 Cedar St, Batangas City, Batangas', '+7778889999', 'eva@example.com', '2021-06-05');


--INSERTING DATA TO EMPLOYEES
INSERT INTO employees (EmployeeID, FirstName, LastName, EmpAddress, PhoneNumber, Email, Position, Department, Salary, DateStarted, DateEnded)
VALUES
(1, 'Mark', 'Enriquez', 'Lucena City, Quezon', '+1234567890', 'mark@example.com', 'Manager', 'Operations', 60000.00, '2011-11-15', NULL),
(2, 'Emily', 'Garcia', 'Tayabas City, Quezon', '+1987654321', 'emily@example.com', 'Senior Developer', 'IT', 55000.00, '2022-03-20', NULL),
(3, 'Alex', 'Garcia', 'Lucena City, Quezon', '+1112223333', 'alex@example.com', 'Marketing Specialist', 'Marketing', 50000.00, '2012-01-10', NULL),
(4, 'Nora', 'Gonzalez', 'Lucban, Quezon', '+4445556666', 'nora@example.com', 'Analyst', 'Finance', 52000.00, '2012-01-28', NULL),
(5, 'Oscar', 'Padilla', 'Lucena City, Quezon', '+7778889999', 'oscar@example.com', 'HR Manager', 'Human Resources', 58000.00, '2012-06-05', NULL),
(6, 'Lucas', 'Martinez', 'Sariaya, Quezon', '+3334445555', 'lucas@example.com', 'Sales Associate', 'Sales', 48000.00, '2012-11-12', NULL),
(7, 'Ava', 'Rodriguez', 'Lucena City, Quezon', '+6667778888', 'ava@example.com', 'Customer Support', 'Support', 47000.00, '2013-01-03', NULL),
(8, 'Liam', 'Lopez', 'Lucena City, Quezon', '+5556667777', 'liam@example.com', 'Admin Assistant', 'Administration', 45000.00, '2012-02-18', NULL),
(10, 'Ethan', 'Taylor', 'Lucena City, Quezon', '+2223334444', 'ethan@example.com', 'Intern', 'Operations', 30000.00, '2022-08-30', '2022-12-30');


--Inserting data to customers
INSERT INTO customers (CustomerID, CompanyName, PhoneNumber, Email, CompanyAddress)
VALUES
(1, 'ABC Inc.', '(02)34567890', 'abc@example.com', '24 Masagana St, Valenzuela City'),
(2, 'XYZ Corporation', '(02)87654321', 'xyz@example.com', '89 Em St, Purok Malamig, Navotas City'),
(3, 'LMN Enterprises', '(02)12223333', 'lmn@example.com', '90 Pen St, Purok Malamig, Navotas City');

--inserting data to seafood types
INSERT INTO seafoodTypes (SeafoodID, TypeName, Category, UOM, PPU)
VALUES
(1, 'Salmon', 'Fish', 'Kilo', 420),
(2, 'Shrimp', 'Shellfish', 'Kilo', 410),
(3, 'Tuna', 'Fish', 'Kilo', 510),
(4, 'Lobster', 'Shellfish', 'Kilo', 450),
(5, 'Crab', 'Shellfish', 'Kilo', 400),
(6, 'Squid', 'Cephalopod', 'Kilo', 310);

--INSERTING DATA TO SEAFOOD INVENTORY
INSERT INTO seafoodInventory (StockID, Quantity, DateReceived, Date_Delivered, SeafoodID, SupplierID)
VALUES
(1000000001, 120.55, '2011-11-25', NULL, 1, 1002),
(1000000002, 357.20, '2011-11-27', NULL, 3, 1002),
(1000000003, 442.80, '2011-11-30', NULL, 6, 1002),
(1000000004, 28.90, '2011-12-04', NULL, 5, 1002),
(1000000005, 650.75, '2011-12-10', NULL, 4, 1002),
(1000000006, 93.10, '2011-12-15', NULL, 6, 1002),
(1000000007, 811.45, '2011-12-17', NULL, 4, 1002),
(1000000008, 214.30, '2011-12-20', NULL, 2, 1002),
(1000000009, 482.60, '2011-12-28', NULL, 2, 1002),
(1000000010, 753.80, '2012-01-05', NULL, 1, 1002),
(1000000011, 38.20, '2012-01-08', NULL, 5, 1002),
(1000000012, 950.15, '2012-01-15', NULL, 3, 1002),
(1000000013, 521.40, '2012-01-20', NULL, 6, 1002),
(1000000014, 189.70, '2012-01-23', NULL, 4, 1002),
(1000000015, 67.90, '2012-01-26', NULL, 5, 1002),
(1000000016, 372.25, '2012-01-30', NULL, 6, 1002),
(1000000017, 819.60, '2012-02-05', NULL, 2, 1002),
(1000000018, 292.80, '2012-02-07', NULL, 1, 1002),
(1000000019, 505.35, '2012-02-15', NULL, 2, 1002),
(1000000020, 927.45, '2012-02-20', NULL, 3, 1002),
(1000000021, 111.75, '2012-02-25', NULL, 4, 1002),
(1000000022, 680.90, '2012-02-28', NULL, 6, 1002),
(1000000023, 833.60, '2012-03-07', NULL, 2, 1002),
(1000000024, 250.50, '2012-03-25', NULL, 2, 1002),
(1000000025, 404.75, '2012-04-25', NULL, 1, 1002),
(1000000026, 18.30, '2012-05-25', NULL, 1, 1002),
(1000000027, 796.10, '2012-06-25', NULL, 6, 1002),
(1000000028, 138.40, '2012-07-25', NULL, 1, 1002),
(1000000029, 575.65, '2012-08-25', NULL, 1, 1004),
(1000000030, 961.20, '2012-09-25', NULL, 1, 1004);
```

```sql
--insert data to transpo table
INSERT INTO Transpo (TransportID, TransportMode, TransportDate, DepartureLoc, CustomerID)
VALUES
(1, 'Truck', '2011-11-30', 'Lucena City, Quezon', 1),
(2, 'Truck', '2011-12-20', 'Lucena City, Quezon', 2),
(3, 'Truck', '2011-12-28', 'Lucena City, Quezon', 1),
(4, 'Truck', '2012-01-15', 'Lucena City, Quezon', 1),
(5, 'Truck', '2012-01-30', 'Lucena City, Quezon', 2);


--insert data to supply payment table
INSERT INTO SupplyPayment (SupplyPaymentID, InvoiceNumber, PaymentDate, Amount, SupplierID)
VALUES
(1, 1000000001, '2011-11-25', 725.12, 1002),
(2, 1000000002, '2011-11-27', 493.76, 1002),
(3, 1000000003, '2011-11-29', 611.88, 1002),
(4, 1000000004, '2011-11-27', 771.24, 1002),
(5, 1000000005, '2011-11-30', 110.43, 1002),
(6, 1000000006, '2011-12-03', 223.65, 1002),
(7, 1000000007, '2011-12-05', 982.31, 1002),
(8, 1000000008, '2011-12-08', 355.88, 1002),
(9, 1000000009, '2011-12-10', 827.46, 1002),
(10, 1000000010, '2011-12-13', 451.79, 1002),
(11, 1000000011, '2011-12-15', 937.02, 1002),
(12, 1000000012, '2011-12-18', 288.11, 1002),
(13, 1000000013, '2011-12-20', 642.22, 1002),
(14, 1000000014, '2012-01-02', 112.87, 1002),
(15, 1000000015, '2012-01-05', 827.92, 1002),
(16, 1000000016, '2012-01-07', 399.65, 1002),
(17, 1000000017, '2012-01-10', 762.14, 1002),
(18, 1000000018, '2012-01-12', 543.77, 1002),
(19, 1000000019, '2012-01-15', 188.59, 1002),
(20, 1000000020, '2012-01-17', 411.09, 1002),
(21, 1000000021, '2012-01-20', 720.44, 1002),
(22, 1000000022, '2012-01-22', 355.22, 1002),
(23, 1000000023, '2012-01-25', 287.91, 1002),
(24, 1000000024, '2012-01-27', 611.78, 1002),
(25, 1000000025, '2012-01-30', 927.45, 1002),
(26, 1000000026, '2012-02-01', 505.35, 1002),
(27, 1000000027, '2012-02-04', 111.75, 1002),
(28, 1000000028, '2012-02-06', 250.50, 1002),
(29, 1000000029, '2012-08-25', 681.19, 1004),
(30, 1000000030, '2012-09-25', 833.60, 1004);
```

Task 4: Implement Data Modification
Description: Create SQL scripts for updating, deleting, and modifying data as needed.

```sql
--Data Modification
--DELETE THE RECORD FROM SUPPLIER TABLE WHERE ID IS 1005
DELETE FROM suppliers
WHERE SupplierID = 1005;

--UPDATE THE RECORD OF THE SUPPLIER WHERE ID IS 1001
UPDATE suppliers
SET DateStarted = '2013-01-10'
WHERE SupplierID = 1001;
```

Phase 3: Data Retrieval (DQL)

Task 5: Create SQL Queries for Common Tasks
Description: Develop SQL queries to perform common library-related tasks, such as searching for books by title, author, or genre.

```sql
SELECT * FROM suppliers
```

	SupplierID	FirstName	LastName	SupplierAddress	PhoneNumber	Email	DateStarted
1	1001	John	Doe	123 Main St, Lucena City, Quezon	+1234567890	john@example.com	2013-01-10
2	1002	Jane	Smith	456 Oak Ave, Calamba City, Laguna	+1987654321	jane@example.com	2011-11-20
3	1003	Alice	Johnson	789 Elm St, Santa Rosa, Laguna	+1112223333	alice@example.com	2013-01-15
4	1004	Bob	Williams	321 Pine St, Batangas City, Batangas	+4445556666	bob@example.com	2012-08-28

```sql
SELECT * FROM seafoodTypes WHERE TypeName LIKE 'Squid';
```

	SeafoodID	TypeName	Category	UOM	PPU
1	6	Squid	Cephalopod	Kilo	310.00

Task 6: Implement Advanced Queries Description: Construct more advanced SQL queries for generating reports.

```sql
--CALCULATE THE TOTAL SALES
SELECT SUM(sales.Amount) AS TotalSales, customer.CompanyName
FROM seafoodSales AS sales
JOIN customers AS customer
    ON customer.CustomerID = sales.CustomerID
GROUP BY customer.CompanyName
ORDER BY TotalSales;
```

	TotalSales	CompanyName
1	420.75	XYZ Corporation
2	430.90	ABC Inc.
3	487.70	LMN Enterprises

```sql
--COUNT HOW MANY TIMES THE SUPPLIERS WERE ABLE TO DELIVER
SELECT COUNT(inv.StockID) AS Number_of_Deliveries, suppliers.FirstName, suppliers.LastName
FROM seafoodInventory AS inv
JOIN suppliers AS suppliers
    ON suppliers.SupplierID = inv.SupplierID
GROUP BY suppliers.FirstName, suppliers.LastName;
```

	Number_of_Deliveries	FirstName	LastName
1	28	Jane	Smith
2	2	Bob	Williams

```sql
-- QUANTITY PER SEAFOOD TYPES
SELECT SUM(inv.Quantity) AS Quantity, cat.TypeName
FROM seafoodInventory AS inv
JOIN seafoodTypes AS cat
 ON cat.SeafoodID = inv.SeafoodID
GROUP BY cat.TypeName;
```

	Quantity	TypeName
1	135.00	Crab
2	1763.65	Lobster
3	3265.45	Salmon
4	3105.95	Shrimp
5	2906.55	Squid
6	2234.80	Tuna

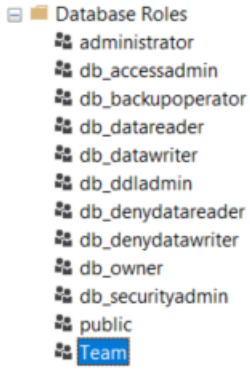
Phase 4: Access Control (DCL)

Task 7: Define User Roles and Permissions

Description: Define user roles and set appropriate permissions. Create roles for ‘Admin’ and ‘Team’.

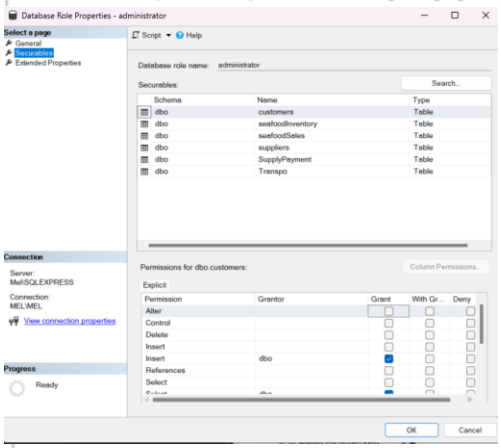
```sql
--create role for admin
CREATE ROLE administrator;
--create role for team
CREATE ROLE Team;
```

DataSense Analytics SQL Master Class



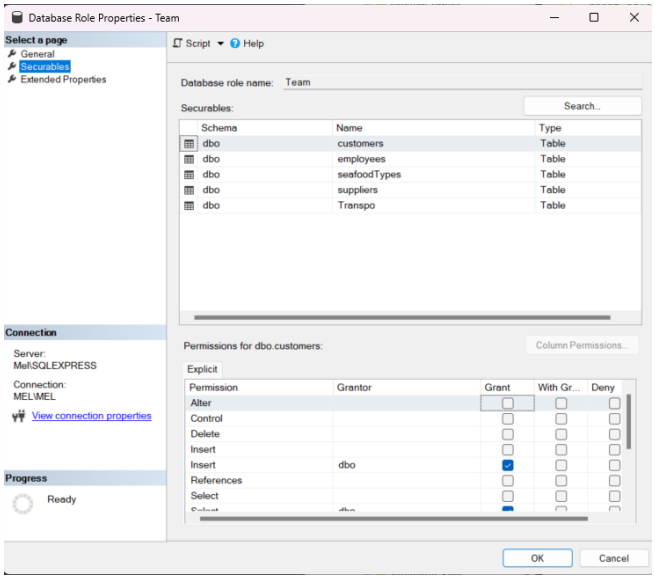
Task 8: Implement Access Control Statements
Description: Implement DCL statements to control access to the database based on user roles and permissions.

```
--grant select, update, insert, and delete to administration
GRANT SELECT, INSERT, UPDATE ON [dbo].[seafoodInventory] TO administrator;
GRANT SELECT, INSERT, UPDATE ON [dbo].[SupplyPayment] TO administrator;
GRANT SELECT, INSERT, UPDATE ON [dbo].[seafoodSales] TO administrator;
GRANT SELECT, INSERT, UPDATE ON [dbo].[suppliers] TO administrator;
GRANT SELECT, INSERT, UPDATE ON [dbo].[customers] TO administrator;
GRANT SELECT, INSERT, UPDATE ON [dbo].[Transpo] TO administrator;
```




```
--grant select to team
GRANT SELECT, INSERT ON [dbo].[employees] TO Team;
GRANT SELECT, INSERT ON [dbo].[customers] TO Team;
GRANT SELECT, INSERT ON [dbo].[Transpo] TO Team;
GRANT SELECT, INSERT ON [dbo].[suppliers] TO Team;
GRANT SELECT, INSERT ON [dbo].[seafoodTypes] TO Team;
```

DataSense Analytics SQL Master Class

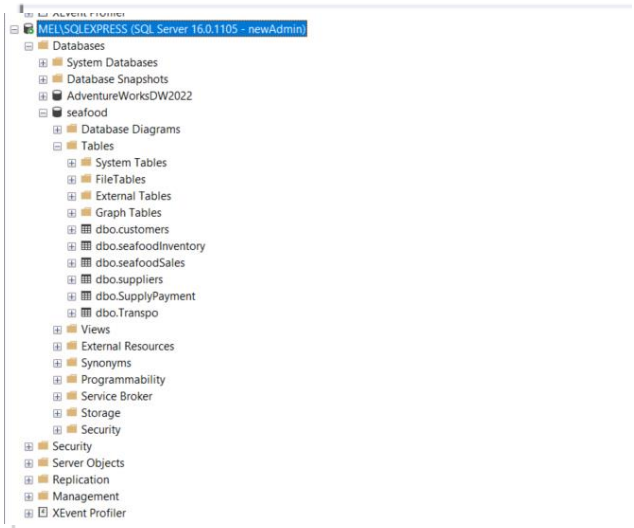


Task 9: Simulate User Interactions

Description: Simulate user interactions with the system, demonstrating how access control works based on user roles.

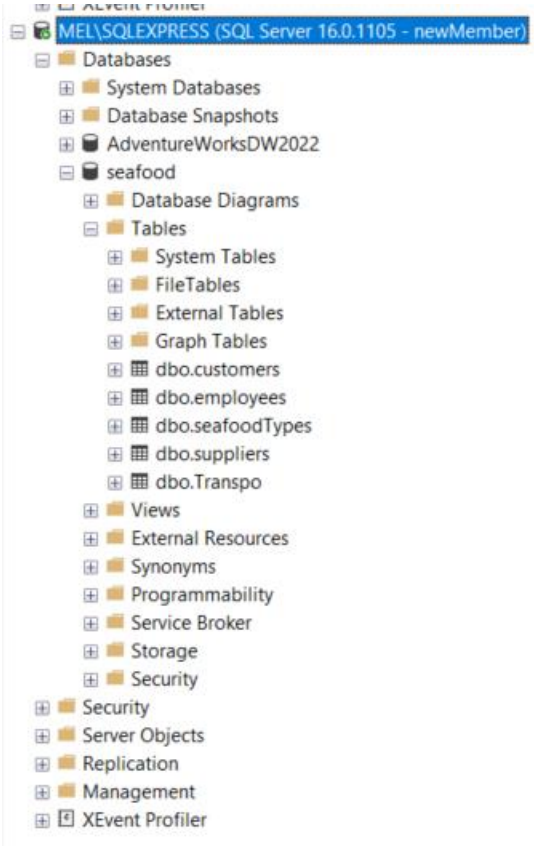
```
--CREATE AN ACCOUNT FOR ADMINISTRATOR
USE master;
CREATE LOGIN newAdmin WITH PASSWORD = 'Admin123'
USE seafood;
CREATE USER NewAdmin FOR LOGIN newAdmin;

EXEC sp_addrolemember 'administrator', 'NewAdmin';
```




```
--CREATE AN ACCOUNT FOR TEAM
USE master;
CREATE LOGIN newMember WITH PASSWORD = 'Team123'
USE seafood;
CREATE USER NewMember FOR LOGIN newMember;

EXEC sp_addrolemember 'Team', 'NewMember';
```



Phase 6: Documentation and Presentation

Task 11: Prepare Documentation

Description: Document the entire project, including a project report, schema diagrams, SQL scripts, and explanations of DML, DDL, DQL, and DCL operations.

The data within this database comprises entirely of fabricated information. The names, locations, and numbers contained herein are entirely fictitious and have no real-world correlation.

Suppliers

Name	Data Type	Description/Attributes
SupplierID	Int	Primary key
FirstName	VARCHAR(100)	First name of the Supplier
LastName	VARCHAR(100)	Last name of the Supplier
SupplierAddress	VARCHAR(255)	Address of the supplier
PhoneNumber	VARCHAR(100)	Contact number of the supplier
Email	VARCHAR(100)	Email address of the supplier
DateStarted	date	The date where the supplier started supplying the company.

DataSense Analytics SQL Master Class

Employees

Name	Data Type	Description/Attributes
EmployeeID	Int	Primary key
FirstName	VARCHAR(100)	First name of the employee.
LastName	VARCHAR(100)	Last name of the employee
EmpAddress	VARCHAR(255)	Address of the employee
PhoneNumber	VARCHAR(100)	Contact number of the employee
Email	VARCHAR(100)	Email address of the employee
Position	VARCHAR(100)	Position of the employee
Department	VARCHAR(100)	Department where the employee is assigned
Salary	Decimal(10,2)	The monthly salary of the employee
DateStarted	Date	The date where the supplier started supplying the company.
DateEnded	Date	The date where the employee leave the company.

Customers

Name	Data Type	Description/Attributes
CustomerID	Int	Primary key
CompanyName	VARCHAR(255)	Name of the Company
PhoneNumber	VARCHAR(100)	Contact number of the Company
Email	VARCHAR(100)	Email address of the Company
CompanyAddress	VARCHAR(255)	Address of the Company

Seafood Inventory

Name	Data Type	Description/Attributes
StockID	Int	Primary key
Quantity	VARCHAR(255)	Total kilos of the product
DateReceived	VARCHAR(100)	Date when the products were received
Date_Delivered	VARCHAR(100)	Date when the products were delivered
SeafoodID	Int	Foreign Key, many-to-one relationship with seafoodTypes table.
SupplierID	Int	Foreign Key, many-to-one relationship with suppliers table.

Seafood Types

Name	Data Type	Description/Attributes
SeafoodID	Int	Primary key
TypeName	VARCHAR(150)	Name of the Product
Category	VARCHAR(100)	Product's category

UOM	VARCHAR(100)	Unit of measure use (eg. Kilo)
PPU	Decimal(5,2)	The price per unit

Supply Payment

Name	Data Type	Description/Attributes
SupplyPaymentID	Int	Primary key
InvoiceNumber	Int	The official invoice number
PaymentDate	Date	Product's category
Amount	Decimal(5,2)	The amount paid to supplier
SupplierID	Int	Foreign Key, many-to-one relationship with suppliers table.

Seafood Sales

Name	Data Type	Description/Attributes
seafoodSalesID	Int	Primary key
SalesInvoiceNumber	Int	The official invoice number
SalesDate	Date	Product's category
Amount	Decimal(5,2)	The amount paid to supplier
CustomerID	Int	Foreign Key, many-to-one relationship with customers table.

Transpo

Name	Data Type	Description/Attributes
TransportID	Int	Primary key
TransportMode	VARCHAR(100)	The mode of transportation used (eg. Truck)
TransportDate	Date	Date when the product was transport/delivered.
DepartureLoc	VARCHAR(255)	Location where the vehicle departed.
ArrivalLoc	VARCHAR(255)	Location of where the vehicle arrived.
CustomerID	Int	Foreign Key, many-to-one relationship with customers table.