COMP1140 Assignment 3: Popular Pizza Project:

Physical Database Design

Due: 12 pm, Tuesday, October 24, 2017 | WORTH 15% of final assessment mark.

Contents

[Data Requirements 2](#_Toc532297113)

[Transaction Requirements 4](#_Toc532297114)

[Data Manipulation Operations 4](#_Toc532297115)

[Queries 4](#_Toc532297116)

[Business Rules 5](#_Toc532297117)

[EER Diagram 6](#_Toc532297118)

[Data Dictionary 8](#_Toc532297119)

[Entity Types 8](#_Toc532297120)

[Relationship Types 10](#_Toc532297121)

[Attributes 11](#_Toc532297122)

[Mapping of the EER model to the relational model 16](#_Toc532297123)

[Orders 16](#_Toc532297124)

[DiscountPrograms 16](#_Toc532297125)

[Staff 16](#_Toc532297126)

[Shifts 16](#_Toc532297127)

[PaymentHistory 16](#_Toc532297128)

[Customers 17](#_Toc532297129)

[ItemsOrdered 17](#_Toc532297130)

[MenuItems 17](#_Toc532297131)

[IngredientList 17](#_Toc532297132)

[Ingredients 17](#_Toc532297133)

[Suppliers 17](#_Toc532297134)

[IngredientOrders 17](#_Toc532297135)

[SQL Implementation 18](#_Toc532297136)

[SQL Database Script 18](#_Toc532297137)

[SQL Statements 21](#_Toc532297138)

# Data Requirements

1. **Order Processing**

Popular Pizza takes orders via phone and walk-in customers, and provides both delivery and pickup services. To take an order, the details of the order itself, as well as the customer’s details and the details of any relevant discount programs must be recorded.

**Order Details**

When an order is created, it is assigned a unique order number, and consists of: order number, order date, the customer’s details, details of the staff taking the order, type of order (phone or walk-in), items ordered (and their name, numbers and price), order description, order status, verification details if an phone order (time answered, time terminated), delivery driver (if a delivery order) and payment information - such as the method of payment, subtotal, any discounts applied, the amount that is discounted, amount of tax, a total amount and if applicable – a payment approval number.

**Customer Details**

When a customer’s details are recorded they are assigned a unique ID, and their first name, last name, address and phone number are recorded for future use

**Discount Program Details**

Popular Pizza offers several different discount programs, A discount program consists of a unique discount code, a description of the discount, the date it starts and ends, a percentage value of the discount, and finally the requirements that must be satisfied for the program to apply.

1. **Employees**

Employees at the store can be divided into two types: instore and delivery. Additionally, hours are not regular and a record is kept for each time an employee works. Finally, a record of all employee payments is maintained.

**Employee Details**

Every employee is assigned an employee number, which helps to manage their details which include: the employee number, their first and last names, postal address, contact number, tax file number, their bank details (including bank code, bank name and account number), their payment rate, description, status, type (in store or delivery driver) and their drivers licence number (for delivery drivers)

**Shift History**

A shift consists of a starting date, starting time, ending date, ending time, hours worked, employee number and for delivery drivers the number of orders a driver delivers during their shift.

**Payment History**

Employee payments are made for each shift to the employee’s bank account. Employee payment records consist of the details of the employee in question, the shift being paid, the total amount being paid, the payment date and time and payment confirmation number.

1. **Menu Items, Ingredients and Suppliers**

Popular Pizza maintains data on their menu items, ingredients and their suppliers for stock management purposes.

**Menu Items**

Each item on Popular Pizzas menu has a unique item number and is made up of the name, size, current selling price and a list of the quantity and type of ingredients making up the menu item.

**Ingredients**

Each Ingredient has a unique code, name, type (vegetable or meat etc), description, and maintains information on the stock level at current stocktake period, date last stocktake was taken, suggested current stock level, reorder level and a list of suppliers who supply the ingredients.

**Suppliers**

Information on suppliers includes an internal ‘supplier number’ to identify the supplier, the name of the supplier, their address and contact number, as well as the name of a contact at the supplier

**Ingredient Orders**

An ingredient order consists of an order number, an order date, a receival date, a total amount, the status of the order, a description of the order and the supplier and ingredients required are specified using their unique identifier codes.

# Transaction Requirements

## Data Manipulation Operations

* Insert/update/delete an order
* Insert/update/delete customer details
* Insert/update/delete a discount program
* Insert/update/delete an employee
* Insert/update/delete shift history
* Insert/update payment history
* Insert/update/delete menu items
* Insert/update/delete ingredients
* Insert/update/delete suppliers
* Insert/update/delete ingredient orders

## Queries

* Search for a customer based on phone number (keyword)
* List all currently active orders
* List all currently active discount programs
* List today’s shifts
* List stock level at current stocktake period vs suggested current stock level
* List Suppliers that can supply a specified ingredient
* List current week’s payment history
* List delivery drivers with shifts on the current day
* List Ingredient Order history
* List ingredients in a specified pizza

# Business Rules

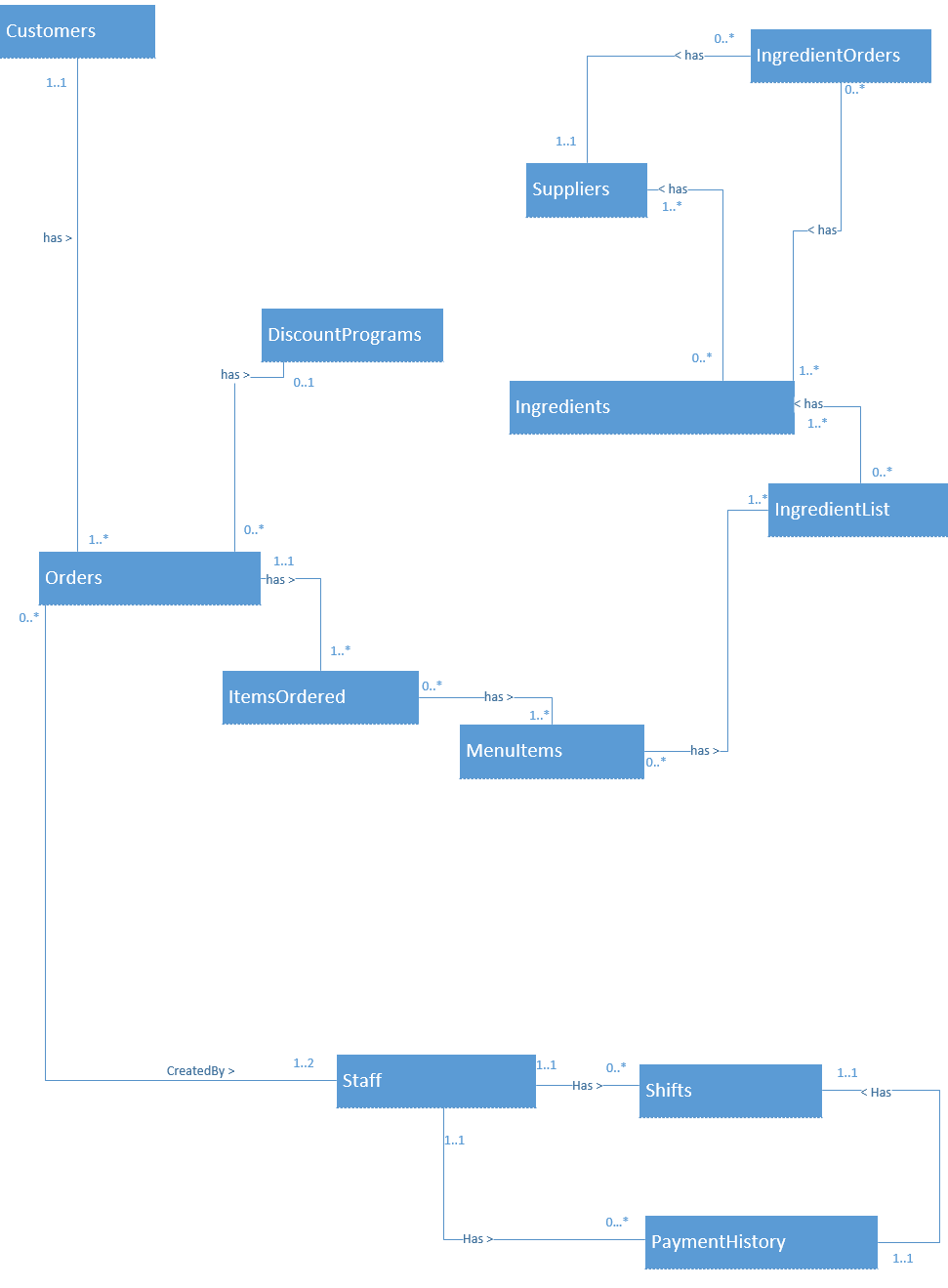
* The amount of each ingredient remaining must be updated every time some is used.
* The results of the weekly stocktake must be input into the database.
* When an ingredients stock level decreases below its reorder level an order for the ingredient must be placed.
* A new customer must be marked as un-verified until the verification process is successfully completed.
  + No customer with an un-verified status shall be able to make an order by phone.
  + If the name and address a customer gives do not match an existing record a new record must be created and the verification process must take place.
* Employees must record each shift they work in the database.
* An employee can only be either an in-store worker or a delivery driver.
* Employees cannot delete data from the database.
* An Employees’ status can only be either:
  + Full time
  + Part time
* Payments can only be added by accounting staff
* An orders payment method can only have one of the following values:
  + Credit card
  + Debit card
  + Cash
* An orders type can only be either:
  + Pick up
  + Delivery
* If an order is paid for using a card the approval number must be stored in the order’s payment approval number
* Only one discount program can be used on an order.
* Discount Programs can only be added/updated by an administrator

# EER Diagram

Entities and attributes - without relationships



Without attributes



# Data Dictionary

## Entity Types

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Description | Aliases | Occurrence |
| Orders | An order is a customer’s request for pizza, it contains details on what specifically the customer would like as well as how they intend to receive their order | N/A | The business has many orders per day |
| DiscountPrograms | Discounts can be applied to orders, giving bonuses or often lower prices to customers | N/A | Discount programs are often available |
| Staff | A staff member is a worker in the store, fulfilling a specific role | N/A | There are many staff |
| Shifts | The amount of time measured in hours a staff member is allocated | N/A | A staff member works many shifts |
| PaymentHistory | Information recorded regarding the chosen method of payment made previously by a customer | N/A | Every shift is paid individually |
| Customers | A person who chooses to place an order at the store | N/A | Many customers make orders |
| ItemsOrdered | List of the items ordered by a specific customer from the menu | N/A | An Order consists of one to many items |
| MenuItems | List of the items available for the customer to choose from when placing their order | N/A | The menu offers many items |
| IngredientList | List of ingredients used in each menu item, given to inform customers | N/A | A menu item consists of several ingredients |
| Ingredients | An individual item used in recipes to make each menu item | N/A | Many Ingredients are used on different menu items |
| Suppliers | External businesses which are used to purchase from the necessary goods used to make the stores products | N/A | The business uses many suppliers to source products |
| IngredientOrders | Lists of ingredients sent to suppliers to be purchased by the store | N/A | The business makes many ingredient orders to maintain stock levels |

## Relationship Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Customers | 1..1 | Has > | 1..\* | Orders |
| Orders | 0..\* | Has > | 0..1 | DiscountPrograms |
| Orders | 1..1 | Has > | 1..\* | ItemsOrdered |
| ItemsOrdered | 0..\* | Has > | 1..\* | MenuItems |
| MenuItems | 0..\* | Has > | 1..\* | IngredientList |
| IngredientList | 0..\* | Has > | 1..\* | Ingredients |
| Ingredients | 0..\* | Has > | 1..\* | Suppliers |
| Suppliers | 1..1 | < Has | 0..\* | IngredientOrders |
| IngredientOrders | 0..\* | Has > | 1..\* | Ingredients |
| Orders | 0..\* | CreatedBy > | 1..2 | Staff |
| Staff | 1..1 | Has > | 0..\* | Shifts |
| Staff | 1..1 | Has > | 0..\* | PaymentHistory |
| PaymentHistory | 1..1 | Has > | 1..1 | Shifts |

## Attributes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name | Attributes | Description | Data Type & Length | Nulls | Multi-valued | Derived | Default |
| Orders | OrderNumber | Unique ID number for the order | INT | NO | No | No |  |
|  | OrderDateTime | Date and Time Order was placed | DATETIME | NO | No | No | GETDATE() |
|  | CustomerID | Customer’s unique ID | INT | NO | No | No |  |
|  | StaffID | Staff member that processed order’s unique ID | INT | NO | No | No |  |
|  | OrderType | Type of Order, PD – Phone Delivery, PP – Phone Pickup, WI – Walk-in Order | CHAR(2) | NO | No | No | PD |
|  | OrderDescription | Any notes on the order | VARCHAR(250) |  | No | No |  |
|  | OrderStatus | Current Status of the order | VARCHAR(10) | NO | No | No | ORDERED |
|  | PaymentMethod | Method used to pay for order | VARCHAR(4) | NO | No | No | CASH |
|  | Subtotal | Total Before Discounts and Tax | FLOAT | NO | No | No | 0.0 |
|  | DiscountID | Unique ID of any discount applied | INT |  | No | No |  |
|  | DiscountAmount | Total Amount Saved | FLOAT |  | No | No |  |
|  | TaxAmount | Tax Amount Charged | FLOAT | NO | No | No |  |
|  | TotalAmount | Total cost of order | FLOAT | NO | No | No |  |
|  | PaymentApprovalNo | Approval Number of Card Payments | VARCHAR(50) |  | No | No |  |
|  | DeliveryDriverID | Unique ID of Staff member that delivered order | DATETIME |  | No | No |  |
|  | VerificationAnswered | Time the Verification Call was Answered | DATETIME |  | No | No |  |
|  | VerificationEnded | Time the verification call was Ended | INT |  | No | No |  |
| ItemsOrdered | OrderNo | ID of the order | INT | NO | No | No |  |
|  | ItemNo | ID of the item in the order | INT | NO | No | No |  |
| MenuItems | ItemNo | Unique ID for Menu Item | INT | NO | No | No |  |
|  | ItemName | Name of menu item | VARCHAR(50) | NO | No | No |  |
|  | Size | Size of Menu Item | CHAR | NO | No | No | S |
|  | Price | Price of Menu Item | FLOAT | NO | No | No |  |
| DiscountPrograms | DiscountID | Unique Discount ID | INT | NO | No | No |  |
|  | DiscountDescription | Description of Discount | VARCHAR(50) |  | No | No |  |
|  | StartDate | Date Discount Starts | DATE | NO | No | No | CONVERT (date, GETDATE()) |
|  | EndDate | Date Discount Ends | DATE |  | No | No |  |
|  | DiscountPercentage | Percentage Discounted | INT | NO | No | No | 20 |
|  | Requirements | Requirements before discount applies | VARCHAR(50) | NO | No | No |  |
| Customers | CustomerID | Unique ID for Customer | INT | NO | No | No |  |
|  | FirstName | Customer Firstname | VARCHAR(50) | NO | No | No |  |
|  | LastName | Customer Surname | VARCHAR(50) | NO | No | No |  |
|  | CustomerAddress | Address of Customer | VARCHAR(200) | NO | No | No |  |
|  | Phone | Customer’s Phone Number | VARCHAR(20) | NO | No | No |  |
| Suppliers | SupplierNo | Supplier’s Unique ID | INT | NO | No | No |  |
|  | SupplierName | Name of Supplier | VARCHAR(100) | NO | No | No |  |
|  | SupplierAddress | Address of Supplier | VARCHAR(200) | NO | No | No |  |
|  | Phone | Phone Number of Supplier | VARCHAR(20) | NO | No | No |  |
|  | ContactPerson | Contact Person at Supplier | VARCHAR(50) |  | No | No |  |
| IngredientOrders | OrderNo | ID of Ingredient Order | INT | NO | No | No |  |
|  | DateOrdered | Date Ingredient Ordered | DATETIME | NO | No | No | GETDATE() |
|  | DateReceived | Date Received | DATETIME |  | No | No |  |
|  | Status | Status of Order | VARCHAR(10) | NO | No | No | ORDERED |
|  | Description | Any notes on the order | VARCHAR(200) |  | No | No |  |
|  | SupplierNo | ID of the supplier of the order | INT | NO | No | No |  |
|  | IngredientCode | ID of the ingredient being ordered | INT | NO | No | No |  |
|  | Amount | Amount of the ingredient being ordered (in grams) | FLOAT | NO | No | No | 5000 |
|  | Price | Price for the amount ordered | FLOAT | NO | No | No |  |
| IngredientList | ItemNo | The Pizza we’re finding the ingredients for | INT | NO | No | No |  |
|  | IngredientCode | The Ingredient’s ID | INT | NO | No | No |  |
|  | Quantity | The quantity of ingredient used on the menu item | FLOAT | NO | No | No | 0.0 |
| Staff | StaffID | Staff’s Unique ID | INT | NO | No | No |  |
|  | FirstName | Staff’s Firstname | VARCHAR(50) | NO | No | No |  |
|  | LastName | Staff’s Last Name | VARCHAR(50) | NO | No | No |  |
|  | Address | Address of staff member | VARCHAR(200) | NO | No | No |  |
|  | Phone | Staff Contact Number | VARCHAR(15) | NO | No | No |  |
|  | TaxFileNumber | Staff TFN | VARCHAR(11) | NO | No | No |  |
|  | BankCode | Staff BSB | VARCHAR(7) | NO | No | No |  |
|  | BankName | Staff Bank | VARCHAR(50) | NO | No | No |  |
|  | BankAccNo | Staff Bank Account Number | VARCHAR(20) | NO | No | No |  |
|  | PaymentRate | Payment Rate of Staff Member – Hourly if In-Store, Per Delivery if Delivery | FLOAT | NO | No | No | 15 |
|  | Description | Any Notes on Employee | VARCHAR(100) |  | No | No |  |
|  | Type | Type of Employee – I for Instore, D for Delivery | CHAR | NO | No | No | I |
|  | DriversLicenceNo | If Delivery, Drivers Licence Number is required | INT |  | No | No |  |
| Shifts | ShiftID | Unique ID for a Shift | INT | NO | No | No |  |
|  | StaffID | Staff member ID performing Shift | INT | NO | No | No |  |
|  | StartingDateTime | DateTime Shift Started | DATETIME | NO | No | No |  |
|  | EndingDateTime | DateTime Shift Ended | DATETIME |  | No | No |  |
|  | HoursWorked | Hours worked | FLOAT |  | No | No |  |
|  | OrdersDelivered | Number of Orders Delivered during Shift | INT |  | No | No |  |
| PaymentHistory | PaymentID | Payment ID | INT | NO | No | No |  |
|  | StaffID | Staff Member being Paid | INT | NO | No | No |  |
|  | ShiftID | Shift being Paid | INT | NO | No | No |  |
|  | Amount | Total Amount being paid | FLOAT | NO | No | No |  |
|  | PaymentDateTime | DATETIME of payment | DATETIME | NO | No | No | GETDATE() |
|  | PaymentConfirmation | Confirmation ID of Payment | INT | NO | No | No |  |
| Ingredients | IngredientCode | Unique ID of Ingredient | INT | NO | No | No |  |
|  | IngredientName | Name of Ingredient | VARCHAR(50) | NO | No | No |  |
|  | IngredientType | Type of Ingredient | VARCHAR(25) | NO | No | No |  |
|  | IngredientDescription | Description of the Ingredient | VARCHAR(100) |  | No | No |  |
|  | StockLevelAtCurrentPeriod | Stock Level | FLOAT | NO | No | No |  |
|  | DateLastStocktakeWasTaken | Date Stock was last updated | DATETIME | NO | No | No |  |
|  | SuggestedStockLevel | Estimated Ideal Stock Level | FLOAT | NO | No | No |  |
|  | ReorderLevel | Orders must be placed if Stock Level below Reorder Level | FLOAT | NO | No | No |  |
|  | SupplierNo | Supplier that Ingredient can be acquired from | INT | NO | No | No |  |

# Mapping of the EER model to the relational model

DBDL formatting

## Orders

Orders (OrderNumber, OrderDateTime, CustomerID, StaffID, OrderType, OrderDescription, OrderStatus, PaymentMethod, Subtotal, DiscountID, DiscountAmount, TaxAmount, TotalAmount, PaymentApprovalNo, VerificationAnswered, VerificationEnded, DeliveryDriverID)

Primary Key OrderNumber

Foreign Key CustomerID references Customer (CustomerID)

Foreign Key StaffID references Staff (StaffID)

Foreign Key ItemsOrderedID references ItemsOrdered (ItemsOrderedID)

Foreign Key DiscountID references DiscountPrograms (DiscountID)

Foreign Key DeliveryDriverID references DeliveryShifts(StaffID)

## DiscountPrograms

DiscountPrograms (DiscountID, DiscDescription, StartDate, EndDate, DiscountPercentage, Requirements)

Primary Key DiscountID

## Staff

Driver (StaffID, FirstName, LastName, Address, Phone, TaxFileNumber, BankCode, BankName, BankAccNo, PaymentRate, Description, Type, DriversLicenceNo)

Primary Key StaffID

## Shifts

Shifts (ShiftID, StaffID, StartingDateTime, EndingDateTime, HoursWorked, OrdersDelivered)

Primary Key ShiftID

Foreign Key StaffID references Staff(StaffID)

## PaymentHistory

PaymentHistory (PaymentID, StaffID, ShiftID, Amount, PaymentDateTime, PaymentConfirmation)

Primary Key PaymentID

Foreign Key StaffID references Staff(StaffID)

Foreign Key ShiftID references Shifts(ShiftID)

## Customers

Customer (CustomerID, FirstName, LastName, CustomerAddress, Phone)

Primary Key CustomerID

## ItemsOrdered

ItemsOrdered (OrderNo, ItemNo)

Foreign Key OrderNo references Orders(OrderNo)

Foreign Key ItemNo references MenuItems(ItemNo)

## MenuItems

MenuItems (ItemNo, ItemName, Size, Price)

Primary Key ItemNo

## IngredientList

IngredientList (ItemNo, IngredientCode, Quantity)

Foreign Key ItemNo References MenuItems(ItemNo)

Foreign Key IngredientCode references Ingredients(IngredientCode)

## Ingredients

Ingredients (IngredientCode, IngredientName, IngredientType, IngredientDescription, StockLevelAtCurrentPeriod, DateLastStocktakeWasTaken, SuggestedStockLevel, ReorderLevel, SupplierNo)

Primary Key IngredientCode

Foreign Key SupplierNo references Suppliers(SupplierNo)

## Suppliers

Suppliers (SupplierNo, SupplierName, SupplierAddress, Phone, ContactPerson)

Primary Key SupplierNo

## IngredientOrders

IngredientOrders (OrderNo, DateOrdered, DateReceived, Status, Description, SupplierNo, IngredientCode, Amount, Price)

Primary Key OrderNo

Foreign Key SupplierNo references Suppliers(SupplierNo)

Foreign Key IngredientCode references Ingredients(IngredientCode)

# SQL Implementation

## SQL Database Script

-- Assignment 3

-- Jonathan Godley

-- c3188072

-- Drop Tables

DROP TABLE ItemsOrdered

DROP TABLE IngredientList

DROP TABLE IngredientOrders

DROP TABLE Ingredients

DROP TABLE Suppliers

DROP TABLE MenuItems

DROP TABLE PaymentHistory

DROP TABLE Shifts

DROP TABLE Orders

DROP TABLE Customers

DROP TABLE DiscountPrograms

DROP TABLE Staff

GO

-- Create Database

CREATE TABLE Staff(

StaffID INT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Address VARCHAR(200) NOT NULL,

Phone VARCHAR(15) NOT NULL,

TaxFileNumber VARCHAR(11) NOT NULL,

BankCode VARCHAR(7) NOT NULL,

BankName VARCHAR(50) NOT NULL,

BankAccountNumber VARCHAR(20) NOT NULL,

PaymentRate FLOAT NOT NULL DEFAULT 15, -- instore paid hourly, delivery paid per delivery

Description VARCHAR(100),

Type CHAR DEFAULT 'I' NOT NULL,

DriversLicenceNo int

)

CREATE TABLE DiscountPrograms(

DiscountID INT PRIMARY KEY,

DiscDescription VARCHAR(50),

StartDate DATE DEFAULT (CONVERT (date, GETDATE())) NOT NULL,

EndDate DATE ,

DiscountPercentage INT DEFAULT 20 NOT NULL CHECK(DiscountPercentage !> 80),

Requirements VARCHAR(50) NOT NULL

)

CREATE TABLE Customers(

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

CustomerAddress VARCHAR(200) NOT NULL,

Phone VARCHAR(20) NOT NULL

)

CREATE TABLE Orders(

OrderNumber INT PRIMARY KEY,

OrderDateTime DATETIME NOT NULL DEFAULT GETDATE(),

CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID),

StaffID INT FOREIGN KEY REFERENCES Staff(StaffID),

OrderType CHAR(2) NOT NULL DEFAULT 'PD', -- PP = Phone Pickup, WI = Walkin PD = PhoneDelivery

OrderDescription VARCHAR(250),

OrderStatus VARCHAR(10) NOT NULL DEFAULT 'ORDERED',

PaymentMethod VARCHAR(4) NOT NULL DEFAULT 'CASH', -- or 'CARD'

Subtotal FLOAT NOT NULL DEFAULT 0.0 CHECK(Subtotal >= 0),

DiscountID INT FOREIGN KEY REFERENCES DiscountPrograms(DiscountID),

DiscountAmount FLOAT CHECK(DiscountAmount >= 0),

TaxAmount FLOAT NOT NULL CHECK(TaxAmount >= 0),

TotalAmount FLOAT NOT NULL CHECK(TotalAmount >=0),

PaymentApprovalNumber VARCHAR(50),

VerificationAnswered DATETIME,

VerificationEnded DATETIME,

DeliveryDriverID INT FOREIGN KEY REFERENCES Staff(StaffID),

)

CREATE TABLE Shifts(

ShiftID INT PRIMARY KEY,

StaffID INT FOREIGN KEY REFERENCES Staff(StaffID),

StartingDateTime DATETIME NOT NULL,

EndingDateTime DATETIME,

HoursWorked FLOAT,

OrdersDelivered INT CHECK(OrdersDelivered >= 0)

)

CREATE TABLE PaymentHistory(

PaymentID INT PRIMARY KEY,

StaffID INT FOREIGN KEY REFERENCES Staff(StaffID),

ShiftID INT FOREIGN KEY REFERENCES Shifts(ShiftID),

Amount FLOAT NOT NULL CHECK(Amount >= 0),

PaymentDateTime DATETIME NOT NULL DEFAULT GETDATE(),

PaymentConfirmation INT NOT NULL

)

CREATE TABLE MenuItems(

ItemNo INT PRIMARY KEY,

ItemName VARCHAR(50) NOT NULL,

Size CHAR NOT NULL DEFAULT 'S',

Price FLOAT NOT NULL

)

CREATE TABLE Suppliers(

SupplierNo INT PRIMARY KEY,

SupplierName VARCHAR(100) NOT NULL,

SupplierAddress VARCHAR(200) NOT NULL,

Phone VARCHAR(20) NOT NULL,

ContactPerson VARCHAR(50))

CREATE TABLE Ingredients(

IngredientCode INT PRIMARY KEY,

IngredientName VARCHAR(50) NOT NULL,

IngredientType VARCHAR(25) NOT NULL,

IngredientDescription VARCHAR(100),

StockLevelAtCurrentPeriod FLOAT NOT NULL CHECK(StockLevelAtCurrentPeriod >= 0), --quantity in grams

DateLastStocktakeWasTaken DATETIME NOT NULL,

SuggestedStockLevel FLOAT NOT NULL CHECK(SuggestedStockLevel >= 0),

ReorderLevel FLOAT NOT NULL CHECK(ReorderLevel >= 0),

SupplierNo INT FOREIGN KEY REFERENCES Suppliers(SupplierNo)

)

CREATE TABLE IngredientOrders(

OrderNo INT PRIMARY KEY,

DateOrdered DATETIME NOT NULL DEFAULT GETDATE(),

DateReceived DATETIME,

Status VARCHAR(10) NOT NULL DEFAULT 'ORDERED',

Description VARCHAR(200),

SupplierNo INT FOREIGN KEY REFERENCES Suppliers(SupplierNo),

IngredientCode INT FOREIGN KEY REFERENCES Ingredients(IngredientCode),

Amount FLOAT NOT NULL DEFAULT 5000 CHECK(Amount > 0),

Price FLOAT NOT NULL CHECK(Price > 0))

CREATE TABLE IngredientList(

ItemNo INT FOREIGN KEY REFERENCES MenuItems(ItemNo),

IngredientCode INT FOREIGN KEY REFERENCES Ingredients(IngredientCode),

Quantity FLOAT NOT NULL DEFAULT 0.0 CHECK(Quantity >= 0)

)

CREATE TABLE ItemsOrdered(

OrderNo INT FOREIGN KEY REFERENCES Orders(OrderNumber),

ItemNo INT FOREIGN KEY REFERENCES MenuItems(ItemNo),

)

GO

## SQL Statements

-- Insert Proper Data into Tables, 3 per table

INSERT INTO Staff VALUES (1,'Tom','Yates','23 Fake Street','0222223333','123-456-789','123-456','CommBank','54215654',16.50,NULL,'I',NULL)

INSERT INTO Staff VALUES (2,'Samantha','Jackson','12 Fake Street','0222223344','123-456-779','123-456','CommBank','54215657',16.50,NULL,'I',NULL)

INSERT INTO Staff VALUES (3,'James','McDeliveryperson','13 Fake Street','0222223355','123-456-888','323-456','Greater Building Society','54215659',2.50,'Our only delivery driver','D',00001234)

GO

INSERT INTO DiscountPrograms VALUES (1,'25% off your order','2017-05-03','2018-01-01',25,'Order must contain 3+ Pizzas')

INSERT INTO DiscountPrograms VALUES (2,'15% off your order','2017-08-03','2017-09-03',15,'Order must contain 2+ Pizzas')

INSERT INTO DiscountPrograms VALUES (3,'5% off your order','2016-05-03',NULL,5,'No Requirements')

GO

INSERT INTO Customers VALUES (1,'Steven','Jacobs','31 Fake Avenue','0401666666')

INSERT INTO Customers VALUES (2,'Tyler','Thomas','21 Fake Avenue','0401666678')

INSERT INTO Customers VALUES (3,'Mary','Shelley','1 Fake Avenue','0401666656')

GO

INSERT INTO Orders VALUES (1,'01/01/17 20:50:59.990',1,1,'PD',NULL,'READY',

'CASH',17.50,NULL,NULL,1.75,19.25,NULL,'01/01/17 20:56:59.990','01/01/17 20:58:23.990',3)

INSERT INTO Orders VALUES (2,'10/19/17 19:59:59.990',1,2,'WI','Had to recook order','COOKING',

'CASH',35.00,NULL,NULL,3.50,38.50,NULL,NULL,NULL,NULL)

INSERT INTO Orders VALUES (3,'05/05/17 21:59:59.990',2,2,'WI',NULL,'COMPLETE',

'CARD',17.50,3,0.88,1.66,18.28,'1535WAF3543532',NULL,NULL,NULL)

INSERT INTO Orders VALUES (4,'10/19/17 19:59:59.990',1,2,'WI',NULL,'COOKING',

'CASH',35.00,NULL,NULL,3.50,38.50,NULL,NULL,NULL,NULL)

GO

INSERT INTO Shifts VALUES (1,1,'01/01/17 12:00:01.290','01/01/17 20:00:03.990',8,NULL)

INSERT INTO Shifts VALUES (2,2,'01/01/17 15:00:02.990','01/01/17 21:30:02.990',6.5,NULL)

INSERT INTO Shifts VALUES (3,3,'10/10/17 18:30:30.290','10/10/17 21:30:29.990',3,18)

INSERT INTO Shifts VALUES (4,3,'10/15/17 18:30:30.290','10/15/17 21:30:29.990',3,18)

INSERT INTO Shifts VALUES (5,3,'11/11/17 18:30:30.290','11/11/17 21:30:29.990',3,20)

INSERT INTO Shifts VALUES (6,3,'11/15/17 18:30:30.290','11/15/17 21:30:29.990',3,20)

GO

INSERT INTO PaymentHistory VALUES (1,1,1,132,'01/01/17 12:10:01.290',1452342)

INSERT INTO PaymentHistory VALUES (2,2,2,107.25,'01/01/17 12:10:03.590',234324145)

INSERT INTO PaymentHistory VALUES (3,3,3,45,'10/10/17 12:10:08.292',152334242)

INSERT INTO PaymentHistory VALUES (4,3,4,45,'10/15/17 12:10:08.292',152334242)

INSERT INTO PaymentHistory VALUES (5,3,5,50,'11/11/17 12:10:08.292',152334242)

INSERT INTO PaymentHistory VALUES (6,3,6,50,'11/15/17 12:10:08.292',152334242)

GO

INSERT INTO MenuItems VALUES (1,'Pepperoni','S',17.50)

INSERT INTO MenuItems VALUES (2,'Supreme','S',17.50)

INSERT INTO MenuItems VALUES (3,'Meatlovers','S',17.50)

GO

INSERT INTO Suppliers VALUES (1,'General Supplies','3 Factory Way','0249341234','Jacob')

INSERT INTO Suppliers VALUES (2,'Stuff Co', '2 Factory Way','0249343334','Sam')

INSERT INTO Suppliers VALUES (3,'Cheese Corp', '32 Factory Way','0249341233','Adam')

GO

INSERT INTO Ingredients VALUES (1,'Mozzarella','CHEESE','Fresh Mozzarella Cheese',3202.55,'02/01/17 12:30:01.290',15000,5000,3)

INSERT INTO Ingredients VALUES (2,'Bacon','MEAT','Diced Bacon Pieces',1202.55,'02/01/17 12:35:01.290',5000,2500,1)

INSERT INTO Ingredients VALUES (3,'Capsicum','VEGETABLE','Diced Red Capsicum Pieces',2442.05,'02/01/17 12:40:01.290',5000,2500,2)

GO

INSERT INTO IngredientOrders VALUES (1,'02/01/17 12:30:01.290',NULL,'ORDERED','AUTOMATED ORDER',3,1,12000,60)

INSERT INTO IngredientOrders VALUES (2,'02/01/17 12:35:01.290','02/03/17 12:35:01.290','ORDERED','AUTOMATED ORDER',1,2,4000,80)

INSERT INTO IngredientOrders VALUES (3,'02/01/17 12:40:01.290',NULL,'ORDERED','AUTOMATED ORDER',2,3,3000,30)

GO

INSERT INTO IngredientList VALUES (2,1,300)

INSERT INTO IngredientList VALUES (2,3,150)

INSERT INTO IngredientList VALUES (3,1,300)

INSERT INTO IngredientList VALUES (3,2,200)

GO

INSERT INTO ItemsOrdered VALUES (1,3)

INSERT INTO ItemsOrdered VALUES (2,3)

INSERT INTO ItemsOrdered VALUES (2,2)

INSERT INTO ItemsOrdered VALUES (3,3)

INSERT INTO ItemsOrdered VALUES (4,2)

INSERT INTO ItemsOrdered VALUES (4,3)

GO

-- Queries

--Q.1 For a staff with id number xxx, print his/her 1stname, lname, and hourly payment rate.

SELECT FirstName, LastName, PaymentRate FROM Staff WHERE staffID = 2;

GO

--Q.2 List the ingredient details of a menu item named xxx.

--note: understood the question as show all details for ingredients in pizza xxx, hence the use of ing.\* rather than picking and choosing

SELECT mi.ItemName AS 'Pizza', ing.\* FROM IngredientList il, MenuItems mi, Ingredients ing

WHERE mi.ItemName = 'Meatlovers' AND mi.ItemNo = il.ItemNo AND il.IngredientCode = ing.IngredientCode;

GO

--Q.3 List all the order details of the orders that are made by the customer with first name xxx via phone between date yyy and zzz.

SELECT O.\* FROM Orders O, Customers C

WHERE O.CustomerID = C.CustomerID AND C.Firstname = 'Steven'

AND O.OrderType IN ('PP','PD') AND O.OrderDateTime > '01/01/17' AND O.OrderDateTime < '01/01/18';

GO

--Q.4 Print the salary paid to a delivery staff named xxx in current month.

SELECT SUM(PH.Amount) AS 'Current Month Salary' FROM Staff ST, Shifts SH, PaymentHistory PH -- print the salary paid

WHERE ST.FirstName = 'James' AND ST.LastName = 'McDeliveryperson' -- delivery staff named xxx

AND SH.StaffID = ST.StaffID AND SH.StaffID = PH.StaffID AND PH.ShiftID = SH.ShiftID

AND YEAR(SH.EndingDateTime) = YEAR(getdate()) AND MONTH(SH.EndingDateTime) = MONTH(getdate());-- in current month

GO

--Q.5 List the menu item that is mostly ordered in current month.

SELECT TOP 1 MenuItems.ItemName AS 'Most Ordered Pizza', COUNT(ItemsOrdered.ItemNo) AS Occurrences -- List the menu item that is mostly ordered

FROM ItemsOrdered, MenuItems, Orders

WHERE MenuItems.ItemNo = ItemsOrdered.ItemNo AND ItemsOrdered.OrderNo = Orders.OrderNumber

AND YEAR(Orders.OrderDateTime) = YEAR(getdate()) AND MONTH(Orders.OrderDateTime) = MONTH(getdate())-- in current month

GROUP BY MenuItems.ItemName, ItemsOrdered.ItemNo

ORDER BY ItemsOrdered.ItemNo DESC;

GO

--Q.6 List the ingredient(s) that was/were supplied by the supplier with supplier ID xxx on date yyy

SELECT Ingredients.IngredientName, Ingredients.IngredientDescription, Ingredients.IngredientType -- List the ingredient(s) that was/were supplied

FROM IngredientOrders, Ingredients

WHERE IngredientOrders.IngredientCode = Ingredients.IngredientCode

AND IngredientOrders.SupplierNo = 1 -- by the supplier with supplier ID xxx

AND YEAR(IngredientOrders.DateReceived) = YEAR('02/03/17')

AND MONTH(IngredientOrders.DateReceived) = MONTH('02/03/17')

AND DAY(IngredientOrders.DateReceived) = DAY('02/03/17');-- on date yyy

GO