Grupa 1

Krukowski

Włoch

Świątkowski

Widoki dla klienta

* wolne stoliki
* menu
* menu z owocami morza

Widoki dla sprzedawcy:

* Menu z owocami morza
* Zamówienia, w których znajdują się owoce morza
* Złożone zamówienia
* Dostępność składników do produktów, które są aktualnie w menu
* Dostępność stolików (patrz wolne stoliki dla klienta)

Widoki dla CEO:

* wynagrodzenie
* wydatki
* zysk

1. Menu dla klienta

CREATE VIEW Menu

AS SELECT ProductName, ProductPrice

FROM Products

JOIN Categories

ON Products.CategoryID = Categories.CategoryID

WHERE CategoryName != ‘Seafood’;

2. Menu z owocami morza dla sprzedającego

CREATE VIEW Menu

AS SELECT ProductName, ProductPrice

FROM Products

JOIN Categories

ON Products.CategoryID = Categories.CategoryID

WHERE CategoryName = ‘Seafood’;

3. Zamówienia odnośnie owoców morza dla sprzedawcy

CREATE VIEW Seafruits

AS SELECT Categories.CategoryName,

Products.ProductName,

Products.ProductPrice,

Orders.OrderDate

FROM Categories

JOIN Products ON Products.CategoryID=Categories.CategoryID

JOIN OrderDetails ON OrderDetails.ProductID=Products.ProductID

JOIN Orders ON Orders.OrderID=OrderDetails.OrderID

WHERE Categories.CategoryName=’Seafood’

AND GETDATE() > SuppliesToOrder.SupplyDeliveryDate

4. Dostępność stolików dla klienta i sprzedającego

CREATE VIEW FreeTables

AS SELECT Tables.TableID, Tables.Seats, Locals.LocalName

FROM Tables

JOIN Locals ON Locals.LocalID=Tables.LocalID

JOIN ReservedTables ON ReservedTables.TableID=Tables.TableID

JOIN TableReservation ON TableReservation.ReservationID=ReservedTables.ReservationID

WHERE Tables.TableID NOT IN ReservedTables.TableID

5. Wynagrodzenie (widok dla CEO)

CREATE VIEW PaymentManagement

AS SELECT Employees.EmployeeID, LastName, FirstName, Locals.LocalName, Locals.LocalID, SalaryBrutto, SalaryNetto

FROM Employees

JOIN Staff ON Employees.EmployeeID=Staff.EmployeeID

JOIN Locals ON Staff.LocalID=Locals.LocalID

6. Wydatki (widok dla CEO)

CREATE VIEW Expenses

AS

SELECT ‘Podatki’ AS NazwaWydatku, SUM(Products.ProductPrice \* Taxes.Percentage) AS expense\_price

FROM Products

JOIN Taxes

ON Products.TaxID = Taxes.TaxID

UNION ALL

SELECT ‘Koszty utrzymania’, SUM(Cost)

FROM CostsOfMaintenance

UNION ALL

SELECT ‘Wynagrodzenie’, SUM(SalaryBrutto)

FROM Employees

UNION ALL

SELECT ‘Wydatki na produkty’, SUM(PriceOfUnit \* Quantity)

FROM SuppliesToOrder

UNION ALL

SELECT ‘Prowizja’, SUM(Commision \* OrderDetails.Quantity \* ProductPrice \* (1 - OrderDetails.Discount))

FROM MethodOfPayment

JOIN Orders ON MethodOfPayment.PaymentID=Orders.PaymentID

JOIN OrderDetails ON Orders.OrderID=OrderDetails.OrderID

7. Zysk (dla CEO)

CREATE VIEW Profits

AS SELECT SUM(Quantity \* ProductPrice \* (1 - Discount)) - (SELECT SUM(expense\_price) FROM Expenses) AS Profit

FROM OrderDetails

8. Dostępność składników do produktów (dla sprzedającego)

CREATE VIEW AvailableIngredients

AS SELECT Products.ProductName,

CASE

WHEN (Supplies.QuantityPerPackage \* Supplies.PackagesInStock) >= ProductDetails.Amount THEN “1”

ELSE “0”

END AS IsAvailable

FROM Products

JOIN ProductDetails ON Products.ProductID = ProductDetails.ProductID

JOIN Supplies ON Supplies.SupplyID = ProductDetails.SupplyID

9. Złożone zamówienia dla sprzedawcy

CREATE VIEW SubmittedOrders

AS SELECT ProductName, Orders.OrderDate, Orders.OrderID, Locals.LocalID, OrderDetails.ProductPrice

FROM Products

JOIN OrderDetails ON Products.ProductID=OrderDetails.ProductID

JOIN Orders ON Orders.OrderID=OrderDetails.OrderID

JOIN Locals ON Orders.LocalID=Locals.LocalID

ORDER BY Orders.OrderDate DESC