# CHANGELOG

Zmiana nazw niektórych checków

Usunięcie Orders.TotalCost

Zmiana Orders.ClientType oraz Orders.ClientID na Orders.CustomerID oraz Orders.DepID

Zmiana połączenia OrderDetails.MealID <-> Menu.MealID na

OrderDetails.MealID <-> Warehouse.MealID

Dodanie Orders.MealPrepared

# OPIS WIDOKÓW I KOD JE GENERUJĄCY

Przypomnienie ról systemowych:

* Chef - odczyt OrderDetails
* Accountant - generowanie faktur i raportów, przeglądanie Orders, kursy walut
* Waiter - dostep do Orders i OrdersDetails
* Menager - nadzoruje całość, kontrola Employees
* Administrator IT - kontrola nad całym systemem
* WarehouseMenager - nadzoruje zamówienia i dba o uzupełnianie warehouse

Każdy widok zawiera:

1. krótki opis
2. użytkownika, który może go wywołać / zobaczyć.
3. kod generujący widok.

### Lista Pracowników

* Wyświetla pełną listę pracowników z ich podstawowymi danymi.
* Może go wyświetlić Manager.

CREATE VIEW OurEmployees  
AS  
SELECT FirstName,  
 LastName,  
 BirthDate,  
 C.CityName,  
 C2.CountryName,  
 Address,  
 PostalCode,  
 Email,  
 HireDate,  
 J.JobName,  
 SuperiorID  
FROM Employees E  
 INNER JOIN Cities C ON E.CityID = C.CityID  
 INNER JOIN Countries C2 ON C.CountryID = C2.CountryID  
 INNER JOIN Jobs J ON E.JobID = J.JobID;  
GO;

### Lista zamówień z dzisiaj

* Wyświetla wszystkie zamówienia złożone aktualnego dnia.
* Może go wyświetlić Manager i Chef.

#### 

CREATE VIEW OrdersMadeToday AS  
SELECT \*  
FROM Orders  
WHERE YEAR(WhenOrdered) = YEAR(GETDATE())  
 AND MONTH(WhenOrdered) = MONTH(GETDATE())  
 AND DAY(WhenOrdered) = DAY(GETDATE());

### Lista zamówień na dzisiaj

* Wyświetla wszystkie zamówienia złożone na dzisiejszy dzień.
* Może go wyświetlić Manager i Chef.

CREATE VIEW OrdersForToday AS  
SELECT \*  
FROM Orders  
WHERE YEAR(WhenToBeDone) = YEAR(GETDATE())  
 AND MONTH(WhenToBeDone) = MONTH(GETDATE())  
 AND DAY(WhenToBeDone) = DAY(GETDATE());

### Aktualne / dzisiejsze menu + stan magazynu dla tych dań

* Wyświetla dzisiejsze menu oraz stan magazynu dla każdego wypisanego dania.
* Mogą go wyświetlić Manager oraz WarehouseManager.

CREATE VIEW TodaysMenu  
AS  
SELECT Name, UnitPrice, UnitsInStock  
FROM Warehouse AS W  
 INNER JOIN Menu M on W.MealID = M.MealID

### Dania, których zapas się skończył

* Wyświetla wszystkie dania, których nie ma w magazynie.
* Może go wygenerować WarehouseManager.

CREATE VIEW OutOfStock  
AS  
SELECT MealID, C.CategoryName, Name  
FROM Warehouse W  
 INNER JOIN Categories C on C.CategoryID = W.CategoryID  
WHERE UnitsInStock = 0;

### Zamówienia do przygotowania na najbliższą przyszłość (nie dalej niż do końca dnia oraz godzinę wstecz na wypadek jakichś opóźnień)

* Wyświetla zamówienia do końca dnia, jeszcze nie przygotowane
* Może go wyświetlić Chef

CREATE VIEW OrdersToCome  
AS  
SELECT OrderDetails.OrderID, MealID, Quantity  
FROM OrderDetails  
 INNER JOIN Orders O on O.OrderID = OrderDetails.OrderID  
WHERE DATEDIFF(day, WhenToBeDone, GETDATE()) = 0  
 AND DATEDIFF(MINUTE, WhenToBeDone, GETDATE()) < 60;

### Obecny stan Magazynu

* Wyświetla obecny stan magazynu.
* Może go wyświetlić: Manager, Chef oraz WarehouseManager.

CREATE VIEW WarehouseStatus  
AS  
SELECT MealID, Name, C.CategoryName, UnitsInStock, UnitPrice  
FROM Warehouse W  
 INNER JOIN Categories C on C.CategoryID = W.CategoryID;

### Ilość obecnie zajętych stolików oraz łączna ich ilość

* Wyświetla ilości obecnie zajmowanych stolików z podziałem na typy jakościowe
* Może go wyświetlić Manager oraz Waiter

CREATE VIEW TablesUsedCurrently  
AS  
SELECT COUNT(TU.TableID) as OccupiedTables, TClassName ClassName  
FROM TablesUsed TU  
 INNER JOIN Orders O ON TU.OrderID = O.OrderID  
 INNER JOIN Tables T ON T.TableID = TU.TableID  
 INNER JOIN TableClasses TC ON TC.TClassID = T.TClassID  
WHERE DATEDIFF(mi, WhenToBeDone + CAST(ForHowLong AS DATETIME), GETDATE()) < 0  
 AND DATEDIFF(mi, WhenToBeDone, GETDATE()) > 0  
GROUP BY TClassName