08-10-2020

Jesper Pilgård Klausen

h5 - infrastruktur

Webshop Separat dokumentation

SQL-database og vedligeholdelses-script

Dette er dokumentationen for en fiktiv webshop. Den består af en SQL database, hvor der hertil er det oprettet et Python script til opdatering og vedligeholdelse af databasen. Man skal kunne se/opdatere/fjerne/indsætte i tabeller i databasen via dette script.

Der er brugt OOP (Objekt orienteret programmering) til denne opgave.

# SQL-databasen er designet med følgende tabeller:

CREATE TABLE Product (

ProductCode int PRIMARY KEY,

ProductName nvarchar(50),

ProductLine nvarchar(50),

ProductDescription nvarchar(50),

ProductVendor nvarchar(50),

BuyPrice float,

QuantityInStock int FOREIGN KEY REFERENCES Stock(QuantityInStock));

Create table Stock (

QuantityInStock int PRIMARY KEY,

ProductCode int FOREIGN KEY REFERENCES Product(ProductCode),

LocationID int FOREIGN KEY REFERENCES Location(LocationID),

Color nvarchar(50),

Size nvarchar(50));

CREATE TABLE Location (

LocationID int Primary Key,

WarehouseID int);

CREATE TABLE Profiles (

ProfileID int PRIMARY KEY,

ProfileUsername nvarchar (50),

ProfilePassword int,

FirstName nvarchar (50),

LastName nvarchar (50),

City nvarchar (50),

Country nvarchar (50),

PhoneNumber int,

Company nvarchar (50));

CREATE TABLE Orders (

OrderNumber int PRIMARY KEY,

ProductCode int Foreign KEY REFERENCES Product(ProductCode),

OrderDate TIMESTAMP,

RequiredDate DATE,

ShippedDate DATE,

QuantityOrdered int,

PriceEach float,

Comments Text);

CREATE TABLE OrderDetails (

OrderNumber int PRIMARY KEY,

ProductCode int FOREIGN KEY REFERENCES Product(ProductCode),

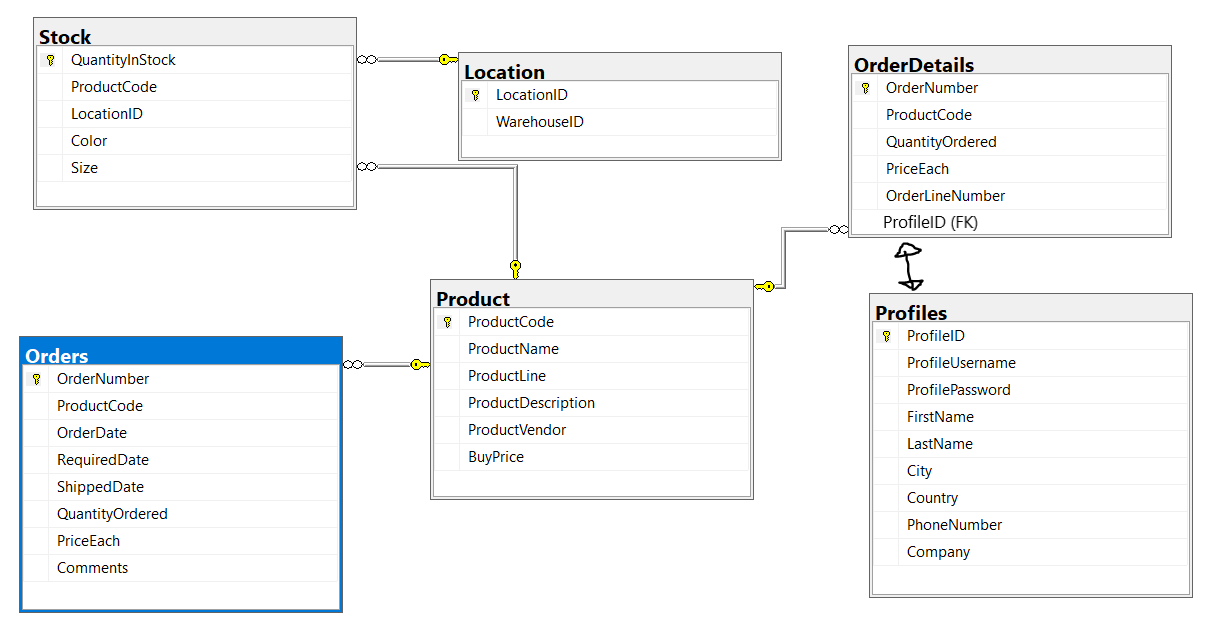
ProfileID int FOREIGN KEY REFERENCES Profile(ProfileID),

QuantityOrdered int,

PriceEach float,

OrderLineNumber int);

SQL databasen er oprettet med indhold der er relevant for en webshop. Den er bygget på 6 forskellige tabeller, der refererer til hinanden på kryds og tværs. Her ses hvordan primary/foreign keys er fordelt i diagrammet nedenfor (Eller i SQL Querien på forrige side).



# Der er oprettet følgende classes i python scriptet:

**ProductClass** er lavet ud fra SQL tabellen: ”Product”. Alle rows er defineret i \_\_init\_\_ og herefter er der oprettet variabler til at eksekverer SQL queries. Insert into, Update, Delete, Select er defineret hver for sig i ProductClass.

**OrderDetailsClass** er oprettet på samme hvis. Bare med alle rows OrderDetails tabellen.

**DatabaseObject** er en class der er oprettet med forbindelsen til SQL databasen. Denne indeholder også muligheden for at kunne commit ændringerne og eksekvere de queries. I dette tilfælde er det de andre classes der benytter denne meget. Denne bliver genbrugt mange andre gange, og er derfor god at have pakket ind i en class.

Scriptet består af fire filer for nemmere overblik:  
  
**Webshop\_Menu**Det er her scriptet køres fra. Her er main menuen defineret med alle If/elif statements, til når man trykker sig frem og tilbage i menuen. De andre funktioner og classes, samt selve teksten til menuen er importeret fra andre filer.

**WebshopMain**  
Det er her selve forbindelsen til SQL’en er defineret i en class. Dertil er muligheden for eksekvering af SQL Queries muligt via denne fil. Denne bliver kaldt fra

**Classes**Det er her ProductClass og OrderDetailsClass er defineret og bliver kaldt fra. De er beskrevet på forrige side.

**TextMenuFunctions**Denne indeholder teksten til selve menuen, så det ikke fylder i samme script. Dette er fin måde at holde det mere overskueligt på. Det er pakket ind i funktioner, så hver del af menuen har hver sin funktion (med kun tekst) som bliver kaldt ovre i Webshop\_Menu.

**Unittest**Der er oprettet en **unittest**. Formålet med denne er at teste om forbindelsen til SQL-databasen kan oprettes, samt lukker den pænt ned igen. Jeg har forsøgt at lave en unit test der skulle opdatere en tabel i databasen, men den kom aldrig i mål og er derfor ikke med.