**Projekt bazy danych przechowującej informacje potrzebne do prowadzenia sklepu internetowego**

Wykonali:

Danie Dobrowolski

Tomasz Janik

Łukasz Kosmaty

**Cel**

Stworzenie bazy danych mogącej przechowywać informację dotyczące sklepu internetowego, wraz z funkcjonalnościami umożliwiającymi prostą administrację.

**Główne założenia**

* Możliwość sprzedaży usług, jak i przedmiotów.
* System magazynowania przedmiotów.
* Możliwość odtworzenia cen produktów.
* Społecznościowy aspekt sklepu (system newsów, możliwość logowania do witryny, oceny użytkowników).
* Obsługa różnych metod płatności.
* Usługi sprzedawane w sklepie są aktywne przez określony czas.
* System umożliwiający zarządzanie pracownikami.

**Możliwości**

* Prowadzenie pełnej dokumentacji sprzedaży.
* Dodawanie nowych użytkowników.
* Nadzorowanie magazynami.
* Łatwy dostęp do zamówień, które musimy wysłać.
* Informowanie użytkowników o zmienię statutu zamówienia.
* Automatyzacja niektórych zadań administracyjnych.

**Ograniczenia**

* Brak możliwości odtworzenia informacji o przecenach.
* Zakupiona usługa musi być aktywna tylko przez określony czas.
* Wspólne id dla przedmiotów i usług.

**Strategia pielęgnacji bazy danych**

Ze względu na ważne informację przechowywane w bazie kopie zapasowe powinny być wykonywane często. W naszym projekcie zdecydowaliśmy się następującą strategię:

* Co trzy dni wykonywana jest pełna kopia zapasowa
* Codziennie wykonywana jest różnicowa kopia zapasowa
* A co 30 minut kopia dzienników transakcji

Taka strategia zagwarantuje, to że będziemy w stanie odzyskać najważniejsze dane utracone w czasie awarii.

**Dokumentacja bazy danych**

|  |  |  |
| --- | --- | --- |
| Tabela | Pola | Przeznaczenie |
| ActiveServices | * ServicesId, PK INT * ClientsLogin, PK NVARCHAR * StartDate, DATETIME * EndDate, PK DATETIME | Informacje o aktywnych usługach użytkowników |
| BonusSalary | * EmployeesId, PK INT * ReceivedDate, PK DATE * BonusSalary, MONEY * ReceiveFor, NVARCHAR | Informacje o premiach i bonusach pracowników |
| Categories | * Id, PK INT * Category, NVARCHAR | Kategiore |
| Clients | * Login, PK NVARCHAR * PeopleId, INT * Password, NVARCHAR | Klienci sklepu |
| Comments | * NewsId, PK INT * ClientsLogin, PK NVARCHAR * AddDate, DATETIME * CommentBody, TEXT | Komentarze do newsów na stronie |
| Employees | * PeopleId, PK INT * HireDate, DATE * PositionsId, INT * BossId, INT | Dane pracowników zatrudnionych w sklepie |
| ExpiredServices | * ServicesId, PK INT * ClientsLogin, PK NVARCHAR * StartDate, DATETIME * EndDate, PK DATETIME | Zostają tutaj przeniesione usługi, które już wygasły |
| Items | * Id, PK INT * Name, NVARCHAR * CategoriesId, INT | Przedmioty sprzedawane w sklepie |
| ItemsInWarehouse | * ItemsId, PK INT * WarehouseId, PK INT * Quantity, INT | Informacje o tym, w jakim magazynie znajdują się dane przedmioty |
| Logs | * Id, PK INT * Date, DATETIME * Info, TEXT * Level, CHAR | Wszystkie ważne informacje dotyczące pracy sklepu.  Pole Level oznacza jaki typ informacji jest zawarta w Info.  Używane oznaczenia:  I – informacja  W – ostrzeżenie  B – błąd |
| News | * Id, PK INT * Topic, NVARCHAR * NewsBody, TEXT * Date, DATETIME | System postów na stronie sklepu |
| Orders | * Id, PK INT * PurchaseDate, DATETIME * City, NVARCHAR * Country, NVARCHAR * Address, NVARCHAR * ClientsLogin, NVARCHAR | Zamówienia złożone w sklepie |
| OrdersDetails | * OrdersId, PK INT * ProductsId, PK INT * UnitPrice, MONEY * Quantity, SMALLINT * Discount, INT | Szczegółowe informacje o zamówieniach |
| Status | * Id, PK INT * StatusInfor, TEXT | Typy statusów zamówień |
| Tabela | **Pola** | **Przeznaczenie** |
| OrdersStatus | * OrdersId, PK INT * StatusId, PK INT | Status zamówień |
| Payments | * Id, PK INT * PaymentsTypesId, INT * OrdersId, INT * PaymentDate, DATETIME * Value, MONEY | Informacje o przeprowadzonych płatnościach |
| PaymentsTyps | * Id, PK INT * Type, NVARCHAR * Provider, NVARCHAR * TermsUrl, NVARCHAR | Wszystkie dostępne płatności |
| People | * Id, PK INT * Name, NVARCHAR * Surname, NVARCHAR * Email, NVARCHAR * Phone, NVARCHAR * Country, NVARCHAR * City, NVARCHAR * Adress, NVARCHAR | Podstawowe informacje, używane później dla klientów i pracowników |
| Positions | * Id, PK INT * Position, NVARCHAR * Salary, MONEY | Stanowiska pracowników oraz ich wynagrodzenie |
| Products | * Id, PK INT * Description, TEXT | Tabela łącząca usługi i przedmioty w jedno. Wykorzystywana przy wyświetlaniu na stronie |
| ProductsPrices | * ProductsId, PK INT * StartDate, PK DATETIME * EndDate, DATETIME * UnitPrice, MONEY | Historia cen produktów wraz z aktualną ceną |
| PromotionCodes | * Id, PK INT * ProductsId, INT * Discount, INT * Code, NVARCHAR * ExipirDate, DATETIME * ClientsLogin, NVARCHAR * OrdersId, INT | Kody promocyjne na dane produkty. Kod jest użyty, jeśli w ClientsLogin nie ma nulla |
| Ratings | * ClientsLogin, PK NVARCHAR * ProductsId, PK INT * Rating, SMALLINT * AddDate DATETIME | Oceny użytkowników danych produktów |
| Sales | * ProductsId, PK INT * UnitPrice, MONEY * Until, DATETIME | Tymczasowe zmiany cen produktów. Dane zawarte w tej tabeli przysłaniają dane z tabeli ProductsPrices |
| Services | * Id, PK INT * Name, NVARCHAR * Time, INT * CategoriesId, INT | Usługi, które można zakupić |
| Warehouse | * Id, PK INT * Name, NVARCHAR * Country, NVARCHAR * City, NVARCHAR * Address, NVARCHAR * Phone, NVARCHAR | Dostępne magazyny |

**Błędy**

50001 – Kategoria nie istnieje.

50002 – Przedmiot jest niedostępny.

50003 – Nie można utworzyć takiego człowieka.

50004 – Rejestracja nowego użytkownika nie powiodła się.

50005 – Zatrudnienie pracownika nie powiodło się.

50006 – To stanowisko nie istnieje.

50007 – Cena produktu nie została znaleziona.

**Funkcje**

ufnTopSellers (@startDate DATETIME, @endDate DATETIME) RETURNS TABLE(ProductID INT, UnitsSold INT)

Zwraca najlepiej sprzedające się produkty w podanym przedziale czasowym.

ufnSalaries (@startDate DATETIME, @endDate DATETIME) RETURNS

TABLE(Name NVARCHAR(30), Surname NVARCHAR(30), Earnings MONEY)

Zwraca zarobki pracowników w podanym przedziale czasowym.

ufnGetProductName (@id INT) RETURNS VARCHAR(100)

Zwraca nazwę produktu po jego id.

ufnCheckStatus(@id INT) RETURNS TABLE(OrderID INT, Status TEXT, Name TEXT, PaymentDate DATETIME)

Zwraca status oraz informacje o płątności zamówienia o podanym id.

ufnIsCodeActive(@id INT) RETURNS BIT

Zwraca czy dany kod był już aktywowany, czy nie.

ufnGetEmployees(@positionID INT) RETURNS TABLE(

PositionID INT,

PositionName NVARCHAR(30),

EmployeeID INT,

Name NVARCHAR(30),

Surname NVARCHAR(30))

Zwraca wszystkich pracowników na stanowisku o podanym id.

ufnGetActiveClients(@start DATETIME, @end DATETIME) RETURNS TABLE(

ClientID INT,

ClientName NVARCHAR(30),

ClientSurname NVARCHAR(30), NumberOfOrders INT )

Zwraca użytkowników, którzy dokonali zakupu w podanym przedziale czasowym.

ufnCountItemsInWarehouse (@id INT, @country NVARCHAR(30) = NULL) RETURNS INT Zwraca dla podanego przedmiotu jego ilość w magazynie (z danego kraju).

ufnGetLogs(@date DATETIME) RETURNS TABLE (Info TEXT, Level CHAR(1))

Zwraca logi z podanego dnia.

ufnUsedPromotionCodes(@Login NVARCHAR(30)) RETURNS TABLE (

CodeID INT,

Code NVARCHAR(10),

ProductID INT,

Discount INT,

OrderID INT)

Zwraca kody użyte przez podanego użytkownika.

UfnGetPrice(@id INT,@date DATETIME = NULL)

RETURNS MONEY

Zwraca aktulaną (lub z podanego dnia) cenę podanego produktu.

**Procedury**

uspDisplayErrors – zwraca poprawny błąd z bloków catch.

uspAddItem (@name NVARCHAR(30), @description TEXT, @category NVARCHAR(30) – dodawanie przedmiotu

uspAddService (@name NVARCHAR(30), @time INT, @description TEXT, @category NVARCHAR(30)) – dodawanie usługi

uspSellItem (@id INT, @warehouse INT = NULL, @quantity INT = 1) – usunięcie przedmiotu z magazynu

uspAddPeople (@id INT OUTPUT, @name NVARCHAR(30), @surname NVARCHAR(30), @email NVARCHAR(30),

@phone NVARCHAR(12), @country NVARCHAR(30), @city NVARCHAR(30), @adress NVARCHAR(30)) - tworzenie nowego człowieka (podstawy dla pracownika/klienta) – procedura pomocnicza. @id OUT jest to id stworzonego człowieka

uspAddClients (@login NVARCHAR(30), @password NVARCHAR(30) ,@name NVARCHAR(30), @surname NVARCHAR(30), @email NVARCHAR(30), @phone NVARCHAR(12) = NULL, @country NVARCHAR(30) = NULL, @city NVARCHAR(30) = NULL, @adress NVARCHAR(30) = NULL) – dodawanie nowego klienta

uspAddEmployees (@name NVARCHAR(30), @surname NVARCHAR(30), @email NVARCHAR(30), @position NVARCHAR(30), @phone NVARCHAR(12) = NULL, @country NVARCHAR(30) = NULL, @city NVARCHAR(30) = NULL, @adress NVARCHAR(30) = NULL, @hireDate DATE = NULL, @BossId INT = NULL) – dodawanie nowego pracownika

uspInitOrder(@person INT, @Cauntry NVarchar(30), @Address NVarchar(30),@City NVarchar(30) – tworzenie zamówienia

**Widoki**

|  |  |  |
| --- | --- | --- |
| Nazwa | Pola | Opis |
| Top5Commentators | * ClientsLogin * Count | 5 najlepszych komentujących użytkowników |
| Top5LastComments | * CommentBody | 5 ostatnich komentarzy |
| ItemsToSend | * Id * Country * City * Address * ProductsId * Quantity | Przedmioty, które należy wysłać |
| TopBuyers | * ClientsLogin * TotalSpend | 5 najlepszych kupujących |
| SortByRating | * ProductsId * AverageRating | Produkty posortowane po ocenach |
| ActiveClients | * ClientId * ServiceId * Name * Surname * StartDate * EndDate | Klienci i ich aktywne usługi |
| SumItems | * Id * Name * Sum | Sumy wszystkich przedmiotów w magazynach |
| LineOfAuthority | * BossId * EmployeeId | Zależności służbowe |

**Triggery**

LogPayments(INSTED OF INSERT) – dodaje informację o zakończonych płatnościach.

MoveToExpiredServices(AFTER DELETE) – przenosi zakończone usługi do tabeli z wygasłymi usługami

AddNewsAfterSales(AFTER INSERT) – dodaje newsa po dodaniu przeceny

LogEndSales(AFTER DELETE) – dodaje informację o zakończoniu przeceny

OrderDetailsGetPrice(INSTED OF INSERT) – uzupełnia ceny produktu podczas dodawania

NewPriceInsert(INSTED OF INSERT) – zapewnia dobre dodanie nowych cen produktów

**Dodatkowe informacje**

W bazie danych znajdują się też dwa sql job. Usuwają one wygasłe usługi oraz nieaktywne przeceny.

Dodawanie nowych produktów (usług/przedmiotów) oraz ludzi (pracowników/klientów) powinno odbyć się przez specjalnie do tego przygotowane procedury.

Id produktu, usługi czy przedmiotu są wspólne, tz. jeśli dany przedmiot jest sprzedawany to w tabeli produkty (Products) znajduje się to samo id co w tabeli przedmioty (Items).

Na diagramie ER nie są przedstawione wszystkie zależności, ponieważ nie byliśmy w stanie wygenerować zależności produkty <- przedmioty, usługi.