

Ćwiczenie 1

Model konceptualny i fizyczny bazy danych.

Zadanie 1

Zaprojektować bazę danych realizującą, w sposób wyczerpujący, określony system rzeczywisty. Projekt wykonać w oparciu o model konceptualny i fizyczny. W modelu winny być wykorzystane wszelkie znane studentowi typy relacji w relacyjnych systemach baz danych.

Przykładowe systemy/projekty:

| OCI | system | OCI | system |
|-----|---|-----|--|
| 0 | sklep internetowy | 5 | system rezerwacji (wizyty u lekarza/hotel) |
| 1 | wypożyczalnia / biblioteka | 6 | system bankowy |
| 2 | system CMS (zarządzanie treścią/artykulami) | 7 | forum internetowe |
| 3 | system ERP (zarządzanie zadaniami/ludźmi) | 8 | dziennik szkolny |
| 4 | portal społecznościowy | 9 | serwis typu: Q&A |

OCI – ostatnia cyfra indeksu

Pomoc

Tworzenie modelu konceptualnego w programie Power Designer.

- stworzyć nowy model konceptualny File- New następnie należy wybrać Conceptual Data Model
- wybierając z menu Model->Properties nadać odpowiednią nazwę dla modelu (zakładka General)
- wykorzystując pasek narzędziowy Palette wstawić co najmniej 3 nowe encje (Entity)
- poprzez dwukrotne kliknięcie na polu wybranej encji przejść do edycji jej właściwości: nadać odpowiednią nazwę (zakładka General) oraz stworzyć kilka kolumn (zakładka Attributes)
- przy tworzeniu kolumn podać nazwę (Name), typ danych (Data Type) oraz zaznaczyć odpowiednie opcje (M-Mandatory - wymagana obecność - NOT NULL, P-Primary - dla klucza pierwotnego, D - Display - pokaż w polu encji w modelu)
- zaprojektować pozostałe encje
- wykorzystując opcje Relationship z paska Palett stworzyć relacje pomiędzy odpowiednimi encjami zgodnie z wcześniejszymi założeniami poprzez przeciągnięcie myszką
- poprzez dwukrotne kliknięcie na relację przejść do edycji właściwości relacji, w zakładce General podać nazwę relacji, w zakładce Detail określić pozostałe parametry relacji

Tworzenie modelu fizycznego w programie Power Designer.

- na podstawie stworzonego modelu konceptualnego przejść do generacji modelu fizycznego Tools->Generate Physical Data Model (Ctrl+G)
- wybrać odpowiedni system zarządzania bazą danych pod który wykonujemy projekt np. DBMS: Sybase SQL Anywhere 5,5 oraz podać nazwę tworzonego modelu
- w razie potrzeby w zakładce Detail ustawić dodatkowe opcje
- poprzez wciśnięcie OK. wygenerować model fizyczny
- w razie potrzeby poprawić błędy i ponownie wygenerować model fizyczny

Pytania kontrolne

- 1) Co to jest model konceptualny bazy danych.
- 2) Co to jest model fizyczny bazy danych.
- 3) Jakież są relacje między tabelami.

Ćwiczenie 2

Tworzenie bazy danych, wprowadzanie i modyfikowanie rekordów.

Zadanie 1

Na podstawie modelu z poprzedniego ćwiczenia wykonać następujące zadania:

- 1) Wygenerować kod SQL do tworzenia zaprojektowanej bazy danych.
- 2) Utworzyć bazę danych Sybase SQL Anywhere oraz w innym dowolnie wybranym systemie bazodanowym.
- 3) Wypełnić bazę danych odpowiednią ilością rekordów za pomocą polecenia INSERT
- 4) Stworzyć zapytania SQL niezbędne do przyszłej realizacji określonych przypadków użycia wybranego systemu/projektu.
- 5) Stworzyć kopię zapasową bazy danych i zachować na przyszłe ćwiczenia.

Pomoc

Generowanie pliku z kodem SQL do tworzenia bazy danych w programie Power Designer.

- po wczytaniu modelu fizycznego wybieramy Databases->Generate databases (Ctrl+G)
- podajemy nazwę pliku i ścieżkę gdzie ma zostać utworzony
- w razie potrzeby ustawiamy dodatkowe opcje
- tworzymy plik poprzez wciśnięcie przycisku OK.
- po wciśnięciu przycisku Edit możemy przejrzeć i ewentualnie zmodyfikować uzyskany kod

Pytania kontrolne

- 1) Dokładnie zapoznać się z wygenerowanym kodem (i zrozumieć ;))
- 2) Dokładnie opisać stworzone zapytania SQL.
- 3) Na czym polega kaskadowe usuwanie rekordów. Czym różni się od zwykłego usuwania.
- 4) Na czym polega i jakie znasz sposoby tworzenia kopii zapasowych bazy danych.

Ćwiczenie 3

Podstawowe konstrukcje i obsługa formularzy w ASP

Zadanie 1

Zapoznać się z podstawowymi konstrukcjami w języku ASP i wykonać proste czynności do z obrazowania ich działania. Wykorzystać konstrukcje: **IF**, **FOR...NEXT**, **FOR...EACH**, **SELECT CASE** i inne.

Zadanie 2

Zapoznać się z wbudowanymi obiektami ASP i ich metodami. W szczególności: **Request** (Form QueryRequest ServerVariables), **Response** (Write, Redirect, CacheControl, Expires), **Server** (ScriptTimeout, Execute, Transfer, HTML Encode, URLEncode), **Session** (SessionID, TimeOut, Abandon)

Zadanie 3

Stworzyć formularz składający się minimum z 8pól, wykorzystaj typy takie **TEXT**, **TEXTAREA**, **HIDDEN**, **CHECKBOX**, **RADIO**, **SELECT**, **PASSWORD**, **SUBMIT**. Zrealizować zadanie polegające na wypełnieniu formularza a następnie zaprezentowaniu danych wpisanych przez użytkownika w przyjaznej tabelarycznej formie. Określić dodatkowo adres IP użytkownika, wersje przeglądarki internetowej oraz systemu operacyjnego. Do zaprogramowania formularza wykorzystać maksymalnie dwa pliki ASP.

Pomoc:

Opis wbudowanych obiektów ASP:

- **Application** - przechowuje informacje o stanie aplikacji.
- **Session** - przechowuje informacje dotyczące jednego użytkownika, jest "magazynem" informacji każdego użytkownika odwiedzającego stronę.
- **Request** - zawiera wszystkie informacje, które są wysyłane z przeglądarki do serwera. Zawiera dane, które zostały wysłane przy użyciu formularza (form).
- **Response** - wysyła HTML i różne informacje, łącznie z cookies i nagłówki, z powrotem do przeglądarki (klienta).
- **Server** - zwiększa funkcjonalność Active Server Pages.
- **ObjectContext** - używany jest do zatwierdzania albo do przerywania transakcji, zarządzanej przez usługi składników, która została zainicjowana przez skrypt umieszczony na stronie ASP

Podstawowe konstrukcje:

```
<% @Language=VBScript %>
<!--#include file="../get_me.asp"-->
```

Warunek If.

```
If warunek Then instrukcje [Else instrukcje ]
```

Przykład:

```
If Hour(Now) < 12 Then
    Response.Write "Good morning!"
ElseIf Hour(Now) >= 18 Then
    Response.Write "Good evening!"
Else
    Response.Write "Good afternoon!"
End If
```

Warunek Select Case:

```
Select Case wyrażenie
[Case wartość
```

```
[instrukcje] ] ...  
[Case Else wartosc  
[instrukcje] ]  
End Select
```

Przykład:

```
Select Case Hour(Now)  
    Case 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11  
        Response.Write "Good morning!"  
    Case 12, 13, 14, 15, 16, 17  
        Response.Write "Good afternoon!"  
    Case Else  
        Response.Write "Good evening!"  
End Select
```

Pętla For ... Next

```
For licznik = start To koniec [ Step krok ]  
[instrukcje]  
[Exit For]  
[instrukcje]  
Next
```

Przykład:

```
For i=0 to 5  
    Response.Write "Witaj świecie!"  
Next
```

Pętla While ... Wend

```
While warunek  
[instrukcje]  
Wend
```

Przykład:

```
i=0  
While i <= 5  
    Response.Write "Witaj świecie!"  
    i=i+1  
Wend
```

Pętla Do ... Loop

```
Do [{While | Until} warunek ]  
[instrukcje]  
[Exit Do]  
[instrukcje]  
Loop [{While | Until} warunek ]
```

Przykład:

```
Do  
    Response.Write "Witaj świecie!"  
    i=i+1  
Loop Until i > 5
```

Pętla For ... Each

```
For Each element In grupa|zbiór  
[instrukcje]  
[Exit For]  
[instrukcje]  
Next
```

Przykład:

```
For Each item In Request.Form
    Response.Write Request.Form(item)
Next
```

Formularze HTML:

```
<input type="text" name="pole_tekstowe" value="domyślna wartość" />
```

```
<input type="password" name="hasło" value="" />
```

```
<input type="checkbox" name="opcja" value="1" checked="checked"/> Opis opcji
```

```
<input type="radio" name="plec" group="plec" value="k" /> kobieta
<input type="radio" name="plec" group="plec" value="m" checked="checked" />
mężczyzna
```

```
<select name="auto">
    <option value="volvo" select="selected">Volvo</option>
    <option value="saab">Saab</option>
</select>
```

```
<textarea name="opis" cols="10" rows="4">domyślna wartość</textarea>
```

```
<input type="submit" name="zapisz" value="Wyślij formularz" />
```

Tabela w HTML:

```
<table>
    <tr>
        <td>W1 K1</td>
        <td>W1 K2</td>
    </tr>
    <tr>
        <td>W2 K1</td>
        <td>W2 K2</td>
    </tr>
</table>
```

Pytania kontrolne

- 1) Jakie znasz metody przesyłania informacji między skryptami. Podaj wady i zalety poszczególnych.
- 2) Wymienić i omówić podstawowe kolekcje i metody obiektów Response, Request, Session i Server.

Ćwiczenie 4

Obsługa sesji i plików cookies. Prosty system logowania. Obsługa plików tekstowych.

Zadanie 1

Stworzyć prosty system logowania oparty na maksymalnie 2 plikach ASP. Jeden plik z ukrytą treścią dostępną tylko dla zalogowanych użytkowników. Drugi z formularzem do logowania. W przypadku próby nieautoryzowanego wejścia na stronę zabezpieczoną logowaniem przekierować do strony z formularzem. System oprzeć o sesję.

Zadanie 2

Stworzyć analogiczny system jak w zadaniu 1 ale oparty o pliki cookies.

Zadanie 3

Wybrać jeden z wcześniej stworzonych systemów logowania i wzbogacić je o system logów serwera. Dla błędnego logowania użytkownika w pliku ERR.LOG zapisać datę logowania, adres ip, oraz użyte w formularzu dane. Natomiast dla poprawnego zalogowania w pliku OK.LOG zapisać tylko datę logowania i pseudonim użytkownika. W plikach każdy kolejny wiersz ma być nowym rekordem.

Pomoc

SESJE

Właściwości obiektu Session:

- **SessionID** - zwraca wartość identyfikatora sesji, który jest unikalny do momentu restartu serwera WWW
- **TimeOut** - określa maksymalny czas (w min) nieaktywności użytkownika po jakim sesja zostaje przerwana
- **LCID** - identyfikator lokalny, który może być czytany i ustawiany podczas sesji
- **CodePage** - numer strony kodowej (iso-8859-2 ma wartość 28592)

Metoda **Abandon** - powoduje usunięcie wszystkich zasobów obiektu Session

Zapis danych do sesji:

```
Session("Imie") = "Jan"  
Session("Wiek") = 28
```

CIASTEczKA

Zapis danych do ciasteczka:

```
Response.Cookies("Imie") = "Mariusz"  
Response.Cookies("Produkt") = "Telewizor"  
Response.Cookies("Produkt")("DataZamowienia") = CStr(Now())  
Response.Cookies("Produkt").Expires = DateAdd("m", 1, Now())
```

Odczytywanie zawartości ciasteczka

```
Request.Cookies("Imie")  
Request.Cookies("Produkt")("DataZamowienia")  
  
Request.Cookies("Produkt").HasKeys
```

CIASTEczKA (.NET)

Zapis danych do ciasteczka:

```
Dim Cookie As HttpCookie
Cookie = New HttpCookie("dane_uzytkownika")

Cookie.Values.Add("Imie", "Jan")
Cookie.Values.Add("Nazwisko", "Kowalski")

Response.AppendCookie(Cookie)
```

Odczytywanie zawartości ciasteczka:

```
Dim Cookie As HttpCookie
Cookie = Request.Cookies("dane_uzytkownika")
```

PLIKI

Metody TextStream:

- **Read** - odczytuje określoną ilość znaków z obiektu TextStream i zwraca ją w postaci łańcucha
- **ReadLine** - odczytuje całą linię z obiektu TextStream i zwraca ją w postaci łańcucha bez znaku końca linii
- **ReadAll** - odczytuje cały plik TextStream i w rezultacie go zwraca
- **Write** - zapisuje łańcuch do pliku TextStream
- **WriteLine** - zapisuje łańcuch do pliku TextStream i dodaje znak końca linii
- **Close** - zamyka obiekt TextStream

Właściwości TextStream:

- **AtEndOfLine** - zwraca wartość TRUE jeśli wskaźnik pliku TextStream znajduje się przed znakiem końca linii, a FALSE w przeciwnym wypadku
- **AtEndOfStream** - zwraca wartość TRUE jeśli wskaźnik pliku znajduje się na końcu pliku
- **Column** - zwraca wartość określającą bieżącą kolumnę, w której znajduje się wskaźnik pliku
- **Line** - zwraca wartość określającą bieżącą linię, w której znajduje się wskaźnik pliku

Zapis do pliku:

```
Const odczyt = 1, zapis = 2, dodaj = 8
Const TristateTrue = 0
Dim objFS, objTextStream, strLine

Set objFS = Server.CreateObject("Scripting.FileSystemObject")

If objFS.FileExists("C:plik.txt") = True Then
    Set objTextStream = objFS.OpenTextFile("C:plik.txt", dodaj, False, TristateTrue)
Else
    Set objTextStream = objFS.CreateTextFile("C:plik.txt", False, False)
End If
objTextStream.WriteLine "Data otwarcia i zapisu pliku: " & Now & "."
objTextStream.Close

Set objTextStream = Nothing
Set objFS = Nothing
```

Odczyt z pliku:

```
Const odczyt = 1, zapis = 2, dodaj = 8
Const TristateTrue = 0
Dim objFS, objTextStream, strLine

Set objTextStream = objFS.OpenTextFile("C:plik.txt", odczyt, TristateTrue)
Response.Write "<P>Zawartość pliku jest następująca:</P>" & VbCrLf
Do While objTextStream.AtEndOfStream <> True
    strLine = objTextStream.ReadLine
    strLine = Server.HtmlEncode(strLine)
```

```
        Response.Write strLine & "<BR>" & VbCrLf
Loop
objTextStream.Close

Set objTextStream = Nothing
Set objFS = Nothing
```


Ćwiczenie 5 i 6

Zadanie 1

Połączyć się z własną bazą danych i wykonać polecenia SQL przygotowane na wcześniejszych zajęciach.

Zadanie 2

Wybrać jedną z tabel w bazie danych i przedstawić ją w postaci tabeli HTML o odpowiadającej jej liczbie kolumn. Nagłówki kolumn tabeli HTML mają być linkami służącymi do sortowania danych w tej tabeli.

Zadanie 3

Przygotować formularz do wprowadzania nowego rekordu do wybranej w zadaniu 2 tabeli. Zabezpieczyć pola przed wysłaniem pustego formularza tam gdzie jest to wymagane przez pole bazy danych (NOT NULL). Dodatkowo użyć funkcji dzięki, której do bazy danych nie będą zapisywane zbędne spacje. Wykorzystać własność AUTO_INCREMENT dla pola klucza pierwotnego edytowanej tabeli.

Zadanie 4

Przygotować formularz do edycji wybranego rekordu tabeli. Przy dodawaniu jak i edycji rekordu wykorzystywać różnego rodzaju pola HTML do typów danych, np. do pól typu BOOLEAN wykorzystać CHECKBOX, do VARCHAR – INPUT, do TEXT – TEXTAREA etc.

Zadanie 5

Przygotować akcję do usuwania rekordu tabeli wraz z potwierdzeniem czy użytkownik jest pewny.

Pomoc

Obiekty ADO:

- **Connection** - stanowi połączenie z źródłem danych
- **Recordset** - zawiera rekordy zwrócone po wykonaniu zapytania na bazie danych
- **Field** - zawiera dane z pojedynczej kolumny i informacje na temat tych danych; obiekt Recordset zawiera grupę Fields, która z kolei zawiera wszystkie obiekty Field
- **Error** - zawiera rozszerzoną informację o zwracanych błędach; wszystkie błędy dostępu do bazy danych mogą być dostępne przez grupę Errors
- **Command** - umożliwia definiowanie specjalnych komend, które są wykorzystywane do wykonania operacji na bazie danych kilka razy wraz ze zmianą parametrów
- **Parameter** - jest to pojedynczy parametr lub argument skojarzony z obiektem Command bazującym na sparametryzowanym zapytaniu lub procedurze przechowywanej

Połączenie do bazy danych przez ODBC:

```
Option Explicit
Response.Expires = 0
Dim objConn, objRS, strQuery
Dim strConnection

Set objConn = Server.CreateObject("ADODB.Connection")
strConnection = "DSN=Northwind;Database=Northwind;UID=sa;PWD=;"
objConn.Open strConnection
```

Połączenie do bazy danych przez sterownik:

```
objConn = Server.CreateObject("ADODB.Connection")
strConnection = "Driver={SQL Server};Server=adres;Database=baza;Trusted_Connection=Yes;"
```

Dla połączenia się z bazą MS Access:

```
strConnection = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &  
Server.MapPath(vsciezka) & ";"
```

Pobieranie danych z bazy danych (metoda Execute obiektu Connection):

```
strQuery = "SELECT ProductName, UnitPrice FROM Products ORDER BY ProductName"  
objRS = objConn.Execute(strQuery)  
  
While Not objRS.EOF  
    Response.Write(objRS.Fields.Item("ProductName").Value & " ("  
    Response.Write(objRS.Fields.Item("UnitPrice").Value & ") <br />")  
    objRS.MoveNext()  
End While  
  
objRS.Close  
objConn.Close  
Set objRS = Nothing  
Set objConn = Nothing
```

Pobieranie danych z bazy danych (metoda Open obiektu Recordset):

```
Set objRS = Server.CreateObject("ADODB.Recordset")  
Set objRS.ActiveConnection = objConn  
  
objRS.Open("SELECT ProductName, UnitPrice FROM Products ORDER BY ProductName")  
  
While Not objRS.EOF  
    Response.Write(objRS.Fields.Item("ProductName").Value & " ("  
    Response.Write(objRS.Fields.Item("UnitPrice").Value & ") <br />")  
    objRS.MoveNext()  
End While  
  
objRS.Close  
objConn.Close  
Set objRS = Nothing  
Set objConn = Nothing
```

Dodawanie rekordu:

```
Const adOpenKeyset = 1, adUseServer = 3, adLockOptimistic = 3  
  
objRS.CursorLocation = adUseServer  
objRS.CursorType = adOpenKeyset  
objRS.LockType = adLockOptimistic  
  
If objRS.EOF Then  
    objRS.AddNew()  
    objRS("ProductName") = "Zeszyt"  
    objRS("UnitPrice") = "2.3"  
    objRS.Update()  
End If
```

Usuwanie rekordu:

```
While Not(objRS.EOF)  
    If objRS.Fields.Item("ProductName").Value = "Zeszyt" Then  
        objRS.Delete()  
        objRS.Update()  
    End If  
    objRS.MoveNext()  
Wend
```

Ćwiczenie 7

Kolokwium