

# Aplikace databázového zpracování

KAPITOLA

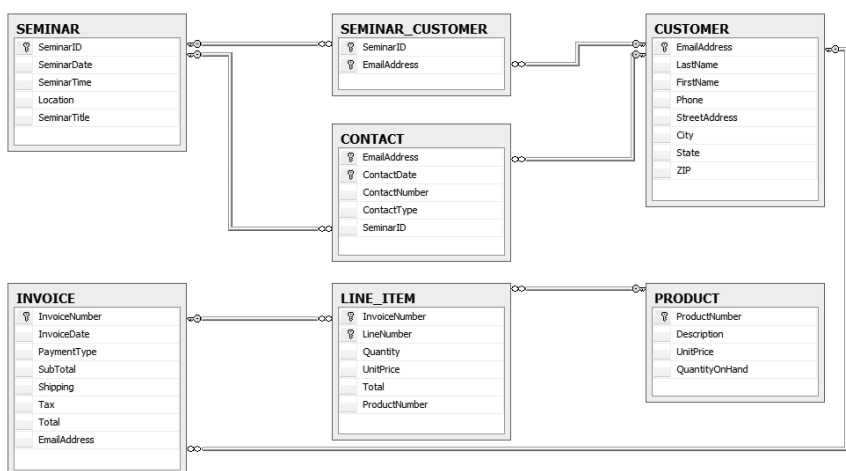
# 7

## Témata kapitoly

- Principy a nastavení webového databázového zpracování
- Základní koncepce jazyka XML (Extensible Markup Language)

V této kapitole představíme témata, která vycházejí ze základů, s nimiž jsme se seznámili v prvních šesti kapitolách této knihy. Když jsme již navrhli a vytvořili databázi, můžeme ji nasadit v praxi. V této kapitole se podíváme na různé aplikace, které využívají databázové zpracování, a zaměříme se přitom zejména na webové databázové zpracování. Setkáme se také s jazykem XML (Extensible Markup Language), který značně rozšiřuje možnosti webových aplikací.

V této kapitole budeme pokračovat v práci s databází Heather Sweeney Designs (HSD), kterou jsme modelovali v kapitole 4, navrhli v kapitole 5 a implementovali v kapitole 6. Databáze se nazývá HSD a databázový diagram systému SQL Server pro tuto databázi je znázorněn na obrázku 7.1.



Obrázek 7.1: Diagram databáze HSD

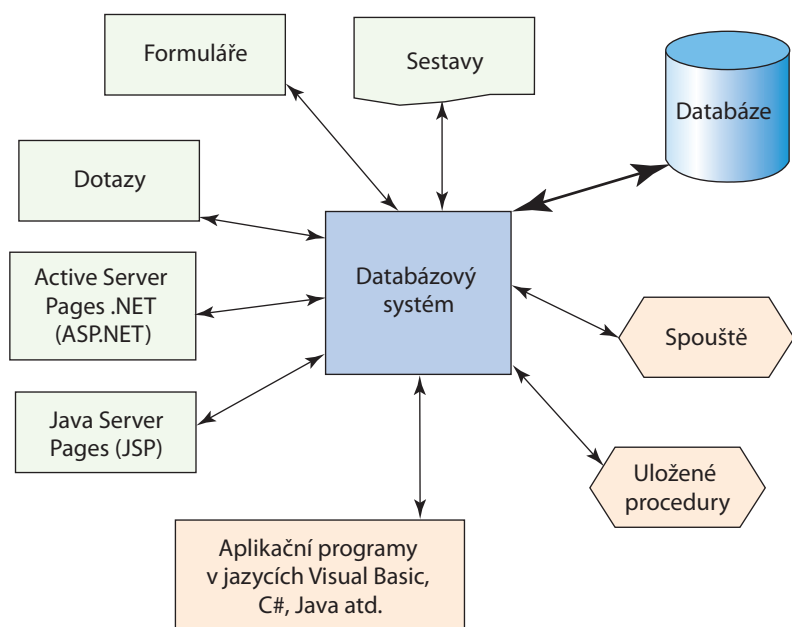
## Prostředí zpracování databáze

Databáze se značně liší velikostí i rozsahem od jednouživatelských databází po velké databáze sdílené více organizacemi, jako jsou systémy rezervace letenek. Na obrázku 7.2, který kopíruje obrázek 6.2, je patrné, že databáze se liší rovněž způsobem svého zpracování.

Některé databáze mají pouze několik formulářů a sestav. Zpracování jiných databází zajišťují aplikace pomocí internetových technologií jako **Active Server Pages .NET (ASP.NET)** a **Java Server Pages (JSP)**. S dalšími databázemi pracují aplikační programy napsané v jazycích Visual Basic .NET, Java, C# nebo jiných. Kromě toho mohou ke zpracování databází sloužit uložené procedury a spouště. V této kapitole se budeme zabývat všemi uvedenými typy databázového zpracování.

### Dotazy, formuláře a sestavy

V této knize se zaměřujeme na tvorbu a zpracování databází pomocí databázových systémů. Popsali jsme například nutnost implementovat pravidla, jako jsou kaskádové aktualizace či odstranění.



Obrázek 7.2: Prostředí zpracování databáze

Databáze spravované databázovým systémem lze používat pomocí databázových aplikací. Základem aplikací jsou dotazy, formuláře a sestavy. Generátory dotazů, formulářů a sestav mohou být integrovány do databázového produktu, jako je Microsoft Access, nebo mohou fungovat jako samostatné programy. Můžeme používat funkce pro práci s dotazy, formuláři a sestavami systému Microsoft Access a zároveň se připojovat k databázi v jiném databázovém systému, jako je SQL Server. V tomto případě „databáze“ systému Microsoft Access ve skutečnosti funguje jako aplikace, která používá připojenou, ale samostatnou databázi. Na obrázku 7.3 je znázorněna databáze Microsoft Access 2010 navržena tak, aby fungovala jako aplikace spolupracující s databází HSD v systému Microsoft SQL Server 2012, kterou vidíme na obrázku 6.1 a jejíž diagram je uveden na obrázku 7.1.<sup>1</sup> V aplikaci představené na obrázku 7.3 se systém Microsoft Access 2010 připojuje k systému SQL Server pomocí ovladače ODBC (Open Database Connectivity). Připojením ODBC se budeme zabývat dále v této kapitole.

The screenshot displays the Microsoft Access 2010 interface. The title bar reads 'HSD : Database (Access 2007 - 2010) - Microsoft Access'. The ribbon at the top includes 'File', 'Home', 'Create', 'External Data', and 'Database Tools'. The 'Database Tools' ribbon is active, showing options like Filter, Sort & Filter, Records, Find, and Text Formatting. On the left, the 'All Access Objects' pane shows a list of tables (dbo\_CONTACT, dbo\_CUSTOMER, dbo\_INVOICE, dbo\_LINE\_ITEM, dbo\_PRODUCT, dbo\_SEMINAR, dbo\_SEMINAR\_CUSTOMER) and one form (HSD Customer Data Input Form). The main window shows the 'HSD Customer Data Input Form' in 'Form View'. The form has the following fields and values: EmailAddress (Bobbi.Pearson@elsewhere.com), LastName (Pearson), FirstName (Bobbi), Phone (512-974-3344), StreetAddress (43 West 23rd Street), City (Austin), State (TX), and ZIP (78710). The status bar at the bottom indicates 'Record: 1 of 12' and 'No Filter'.

**Obrázek 7.3:** Aplikace HSD v systému Microsoft Access 2010

<sup>1</sup> Než připojíte projekt Microsoft Access 2010 k databázi SQL Server 2012, musíte zkontrolovat, zda je spuštěna služba SQL Server Browser. V systému Windows XP spusíte nástroj SQL Server Configuration Manager pomocí položky **Start → Programs → Microsoft SQL Server 2012 → Configuration Tools → SQL Server Configuration Manager**. V systému Microsoft Windows Server 2008 R2 nebo Windows 7 vyberte položku **Start → All Programs → Microsoft SQL Server 2012 → Configuration Tools → SQL Server Configuration Manager**. V oblasti SQL Server Services vyhledejte položku SQL Server Browser a zkontrolujte, zda je tato služba spuštěna (State = Running) a zda se automaticky spouští při spuštění počítače (Start Mode = Automatic). Pokud nikoli, klepněte na ikonu **SQL Server Browser** pravým tlačítkem myši. V zobrazené místní nabídce klepněte na příkaz **Properties** a zobrazí se dialogové okno SQL Server Browser Properties. Na stránce vlastností Log On klepněte na tlačítko **Start** a poté klepněte na tlačítko **Apply**. Klepnutím na kartu **Service** zobrazíte stránku vlastností Service. Klepnutím vyberte řádek **Start Mode** a z rozevřacího seznamu Start Mode vyberte položku **Automatic**. Klepněte na tlačítko **Apply** a poté zavřete dialogové okno SQL Server Browser Properties klepnutím na tlačítko **OK**. Kromě toho musí mít uživatelé příslušná oprávnění k použití databáze, jak jsme vysvětlili v kapitole 6.

Na obrázku 7.3 si můžeme všimnout, že tabulky databáze HSD uložené v systému SQL Server jsou zobrazeny v oblasti Tables navigačního podokna a v oblasti Forms navigačního podokna je vytvořen a uložen formulář na zadávání dat o zákaznících s názvem „HSD Customer Data Input Form“. Tento formulář uživatelům poskytuje přístup k datům v tabulce CUSTOMER. Tato aplikace může samozřejmě obsahovat i dotazy a sestavy.

Nyní se zamysleme nad některými úkoly, které musí databázový systém (např. Microsoft Access) provádět na pozadí při implementaci příkazů zpracování databáze. Předpokládejme například, že vytvoříme dotaz k odstranění dat z tabulky, která má vztah 1:N k jiné tabulce s nastavením On Delete Cascade. Dále předpokládejme, že druhá tabulka má vztah 1:N ke třetí tabulce s nastavením Enforce Referential Integrity, ale nikoli On Delete Cascade. V souladu s těmito vlastnostmi vztahů musí systém Microsoft Access při spuštění tohoto dotazu k odstranění dat odstranit řádky z první a druhé tabulky.

Situace je ještě složitější, pokud během činnosti odstraňovacího dotazu jiný uživatel vytváří sestavu založenou na těchto třech tabulkách. Jak by měl systém Microsoft Access postupovat? Měl by během činnosti dotazu zobrazit sestavu s daty, která zatím zbývají? Nebo by měl systém Microsoft Access chránit sestavu před odstraněním na základě dotazu a nepovolit odstranění žádných dat, dokud není sestava dokončena? Měl by dotaz úplně zakázat, nebo provést něco zcela jiného?

Jako jednodušší příklad předpokládejme, že vytvoříme formulář, který v hlavní části obsahuje data z jedné tabulky a v podformuláři data z druhé tabulky. Nyní řekněme, že uživatel provede změny na pěti řádcích podformuláře, změní některá data v prvním formuláři a poté stiskne klávesu Escape. Které změny se skutečně dostanou do databáze? Žádné? Pouze změny dat v podformuláři? Nebo platí některá jiná možnost?

Dokonce i v případě jednoduchých dotazů, formulářů a sestav je správa těchto funkcí na pozadí složitá. Můžeme změnit vlastnosti databáze, které určují chování systému Microsoft Access v některých z těchto případů, ale musíme přitom vědět, jaké mají tyto změny důsledky. Databázové produkty podnikové třídy, jako jsou SQL Server, Oracle Database a MySQL, poskytují mnoho vlastností a funkcí, které vývojářům dovolují změnit chování databázového systému v uvedených situacích. (Mnoha z nich jsme se zabývali v kapitole 6.)

## Klient-server a tradiční aplikační zpracování

Databázové zpracování se v organizacích začalo využívat začátkem 70. let. Od té doby aplikační programy napsané v programovacích jazycích jako Visual Basic, C, C++, C# a Java zpracovaly již tisíce, možná až miliony databází. Všechny tyto jazyky umožňují do svého standardního kódu integrovat příkazy jazyka SQL.

Pokud má například aplikace firmy Heather Sweeney Designs zpracovávat objednávky online, musí plnit následující funkce:

1. Komunikuje s uživatelem, aby zjistila jeho zákaznický identifikátor.
2. Načte data z tabulky CUSTOMER.
3. Prezentuje uživateli formulář k zadání objednávky.
4. Získá data z tabulky PRODUCT a od zákazníka údaj o objednávaném množství.
5. Ověří stav zásob v tabulce PRODUCT.
6. Odebere položku tabulky PRODUCT ze skladových zásob.
7. V případě potřeby naplňuje zpětné objednávky.
8. Naplňuje vyzvednutí a doručení skladových zásob.
9. Aktualizuje data v tabulkách CUSTOMER, INVOICE a LINE\_ITEM (a pokud se prodej považuje za typ kontaktu se zákazníkem, aktualizuje také tabulku CONTACT).

Aplikace je napsána tak, aby reagovala na výjimky, jako jsou nedostupná data, chybná data, selhání komunikace a desítky dalších potenciálních problémů.

Kromě toho je aplikační program na zpracování objednávek potřeba navrhnout tak, aby s ním mohlo pracovat mnoho uživatelů současně – je možné, že se tuto aplikaci pokusí zároveň používat 50 nebo 100 uživatelů.

## Uložené procedury a spouště

Databázové produkty podnikové třídy, jako jsou SQL Server, Oracle Database, MySQL a DB2, zahrnují funkce, které vývojářům umožňují vytvářet moduly logických a databázových akcí zvané *spouště* a *uložené procedury*. Spouště a uložené procedury se píšou v jazycích, které poskytuje databázový systém.<sup>2</sup> Systém SQL Server například zahrnuje jazyk s názvem Transact-SQL (T-SQL) a společnost Oracle vyvinula jazyk PL/SQL. Programátoři mohou do kódu těchto programovacích jazyků vkládat příkazy jazyka SQL.

**Spoušť** (trigger) je program uložený v rámci databáze, který je databázovým systémem spuštěn, když nastane určitá událost. Těmito událostmi jsou obvykle příkazy SQL, které používají příkazy INSERT, UPDATE či DELETE. Příslušné události pak zpracuje logika spouště BEFORE, AFTER nebo INSTEAD OF. Můžeme se tedy setkat s kombinacemi spouští jako BEFORE DELETE, INSTEAD OF UPDATE a AFTER INSERT. (Uvědomme si, že se jedná pouze o několik příkladů – existuje devět možných kombinací logiky spouště a příkazů SQL.)

Různé databázové produkty podporují různé sady spouští. Například systém Oracle Database 11g Release 2 je kompatibilní se spouštěmi BEFORE, AFTER a INSTEAD OF. MySQL 5.5 nabízí pouze spouště BEFORE a AFTER. Z devíti možných kombinací spouští systém SQL Server 2012 podporuje spouště AFTER a INSTEAD OF, ale kombinuje tuto podporu s dalšími vlastními typy spouští, kterými se v této knize nebudeme zabývat.

**Uložená procedura** (stored procedure) je podobná podprogramu nebo funkci počítačového programu, ale je umístěna v databázi, kde zajišťuje jistou databázovou aktivitu. Příkladem pro databázi HSD by byla uložená procedura na aktualizaci sloupců v tabulce INVOICE pro určitou hodnotu InvoiceNumber při přidání záznamů LINE\_ITEM do tabulky INVOICE. Vyvolat uloženou proceduru, předat jí parametry a přijmout výsledky mohou aplikační programy, webové aplikace a interaktivní uživatelské dotazy.

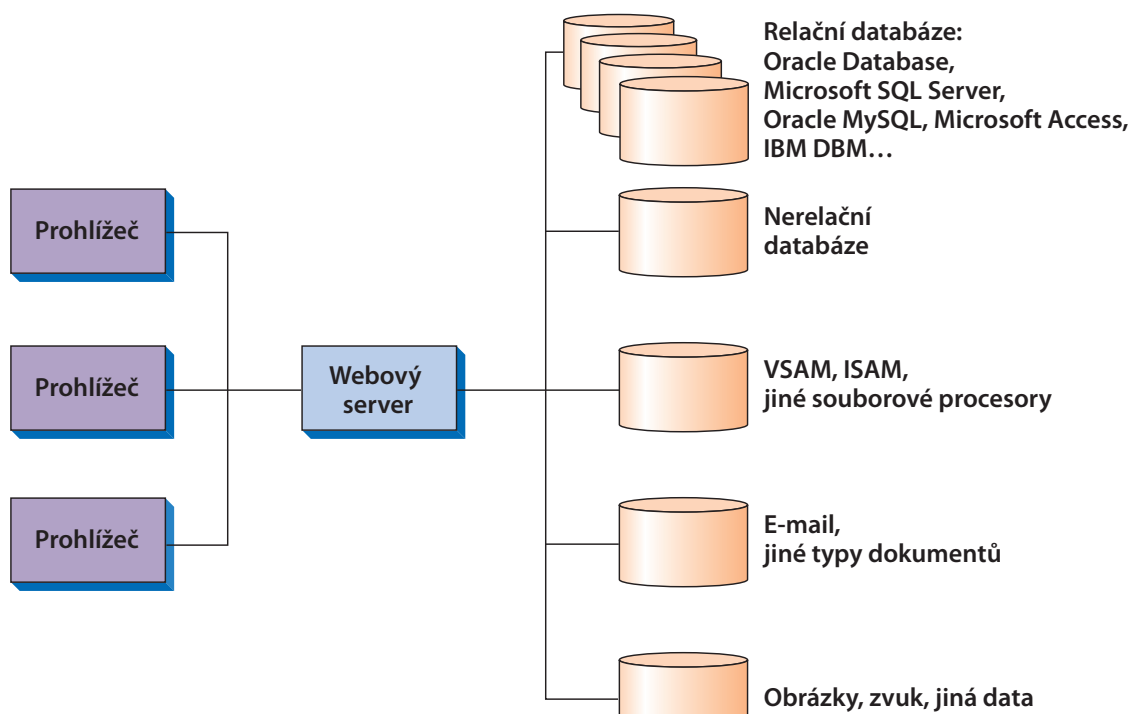
## Webové databázové zpracování

Současné databázové aplikace založené na internetové technologii se nacházejí v bohatém a složitém prostředí. Jak je patrné na obrázku 7.4, typický webový server musí publikovat aplikace, které pracují s mnoha různými typy dat. V tomto textu uvažujeme pouze o relačních databázích, ale existuje také mnoho jiných datových typů.

Pro přístup k databázovým serverům vzniklo několik standardních rozhraní. Každý databázový produkt má své **rozhraní API (Application Programming Interface)**. Rozhraní API je kolekce objektů, metod a vlastností, které umožňují spouštět funkce databázového systému z programového kódu. Každý databázový systém má bohužel vlastní rozhraní API a tato rozhraní se mezi jednotlivými databázovými systémy liší. Aby se programátoři nemuseli učit pracovat s mnoha odlišnými rozhraními, byly vytvořeny oborové standardy pro přístup k databázím.

<sup>2</sup> Další informace o spouštích, uložených procedurách a jejich využití naleznete v knize: David M. Kroenke a David J. Auer, *Database Processing: Fundamentals, Design, and Implementation* (Zpracování databází: principy, návrh a implementace), 12. vydání (Upper Saddle River, NJ: Prentice Hall, 2012): kapitoly 10, 10A a 10B.

Standard **ODBC (Open Database Connectivity)**, který jsme již zmínili v souvislosti s přístupem aplikace systému Microsoft Access 2010 k databázi SQL Server, vznikl v 90. letech, aby bylo možné zpracovávat data relačních databází nezávisle na konkrétním databázovém systému. V polovině 90. let společnost Microsoft přišla s objektově orientovaným rozhraním **OLE DB**, které zahrnuje funkci datového serveru. Rozhraní OLE DB neposkytuje přístup pouze k relačním databázím, ale umožňuje přistupovat i k mnoha jiným typům dat. Rozhraní OLE DB je snadno přístupné programátorům, kteří pracují s jazyky typu C, C# a Java. Toto rozhraní však není k dispozici pro uživatele jazyka Visual Basic (VB) a skriptovacích jazyků. Společnost Microsoft proto vyvinula sadu objektů **Active Data Objects (ADO)**, která umožňuje využívat rozhraní OLE DB z libovolného jazyka včetně jazyků Visual Basic (VB), VBScript a JScript. Po rozhraní ADO následovalo rozhraní **ADO.NET**, což je zdokonalená verze rozhraní ADO vyvinutá v rámci platformy .NET společnosti Microsoft.



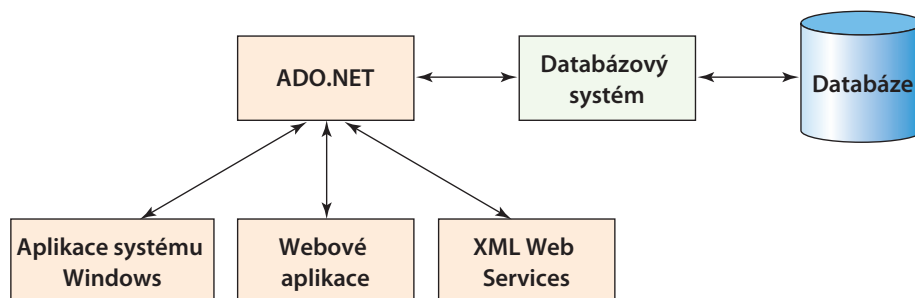
**Obrázek 7.4:** Prostředí webového databázového zpracování

Na webových stránkách se používá technologie ASP.NET, která umožňuje vytvářet webové databázové aplikace. Technologie ASP.NET, která je znázorněna na obrázku 7.2 v rámci prostředí databázového zpracování, používá jazyk HTML (Hypertext Markup Language) a jazyky platformy Microsoft .NET při tvorbě webových stránek, které dokážou načítat a zapisovat databázová data a přenášet je přes veřejné a soukromé sítě pomocí internetových protokolů. Technologie ASP.NET je k dispozici ve webovém serveru společnosti Microsoft, který se nazývá IIS (Internet Information Services). ASP.NET patří do architektury Microsoft .NET a vychází z rozhraní ADO.NET. Nasazení rozhraní ADO.NET ilustruje obrázek 7.5.<sup>3</sup>

Na obrázku 7.2 je uvedena i technologie JSP. JSP představuje kombinaci jazyků HTML a Java, která plní stejnou funkci jako technologie ASP a kompiluje stránky do servletů jazyka Java. Stránky JSP se často používají v open-source webovém serveru **Apache**. Další oblíbenou kombinací webových vývojářů je webový server Apache s databází MySQL a jazykem Perl nebo PHP. Tato kombinace se

<sup>3</sup> Další informace o architektuře Microsoft .NET a rozhraní ADO.NET naleznete v knize: David M. Kroenke a David J. Auer, *Database Processing: Fundamentals, Design, and Implementation* (Zpracování databází: principy, návrh a implementace), 12. vydání (Upper Saddle River, NJ: Prentice Hall, 2012): kapitola 11.

nazývá **AMP** (Apache–MySQL–PHP/Perl). Když funguje v operačním systému Linux, označuje se zkratkou **LAMP**, a v případě operačního systému Windows se můžeme setkat se zkratkou **WAMP**.<sup>4</sup>



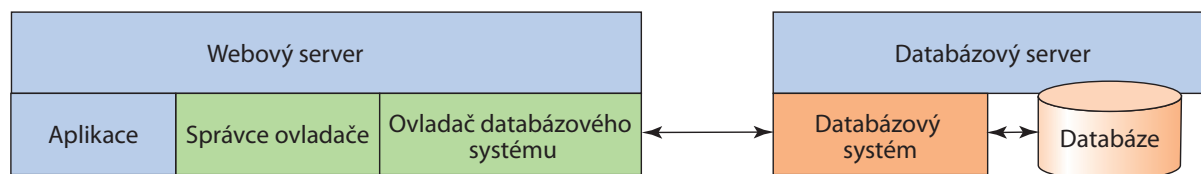
**Obrázek 7.5:** Role rozhraní ADO.NET

Pokud je ve webovém prostředí databázového zpracování webový server spuštěn ve stejném počítači jako databázový systém, jedná se o **architekturu s dvěma vrstvami**. (Jednu vrstvu představují prohlížeče a druhou počítač s webovým serverem a databázovým systémem.) Případně může webový server fungovat v jiném počítači než databázový systém, a tedy jde o **architekturu se třemi vrstvami**. Vysoce výkonné aplikace mohou používat mnoho počítačů s webovým serverem a v některých systémech může více počítačů zajišťovat i provoz databázového systému. Pokud v druhém případě počítače s databázovým systémem zpracovávají stejnou databázi, říkáme takovému systému *distribuovaná databáze*. (K tématu distribuovaných databází se dostaneme v kapitole 8.)

## ODBC

Standard ODBC programátorům umožňuje programovat instrukce pro různé databázové produkty a používat přitom standardní příkazy ODBC. Tyto instrukce jsou předány ovladači ODBC, který je překládá pro rozhraní API konkrétního databázového systému. Ovladač poté přijímá výsledky z databázového systému a převádí tyto výsledky do formy, která je součástí standardu ODBC.

**Architektura ODBC** – základní **architektura ODBC** v prostředí webového serveru se třemi vrstvami – *ale před rozhraními OLE DB a ADO* – je znázorněna na obrázku 7.6. Aplikační program označovaný jako **správce ovladače ODBC** i **ovladač ODBC databázového systému** (v tomto případě ovladač pro více vrstev) se nacházejí na webovém serveru. Ovladač databázového systému odesílá požadavky na zdroje dat, které se nacházejí na databázovém serveru. Podle standardu ODBC je **zdrojem dat** databáze a příslušný databázový systém, operační systém a síťová platforma.



Příkazy SQL

**Obrázek 7.6:** Architektura ODBC webového serveru se třemi vrstvami

<sup>4</sup> Informace o JSP, JDBC a souvisejících technologiích a nástrojích naleznete v knize: David M. Kroenke a David J. Auer, *Database Processing: Fundamentals, Design, and Implementation* (Zpracování databází: principy, návrh a implementace), 12. vydání (Upper Saddle River, NJ: Prentice Hall, 2012): kapitola 11.

Aplikace zadává požadavky, které umožňují vytvořit připojení ke zdroji dat, zadat příkazy SQL a přijmout výsledky, zpracovat chyby a zahájit, potvrdit a vzít zpět transakce. Architektura ODBC poskytuje standardní formát každého z těchto požadavků a definuje také standardní sadu chybových kódů a zpráv.

Správce ovladače slouží jako prostředník mezi aplikací a ovladačem databázového systému. Když aplikace požádá o připojení, ovladač určí typ databázového systému, který obsluhuje daný zdroj dat ODBC, a načte tento ovladač do paměti (pokud zatím není načten).

**Ovladač** zpracovává požadavky ODBC a odesílá danému typu zdroje dat konkrétní příkazy jazyka SQL. Ke každému typu zdroje dat patří odlišný ovladač. Ovladač odpovídá za to, že budou správně spuštěny standardní příkazy ODBC. Ovladač také převádí chybové kódy a zprávy zdroje dat na kódy a zprávy standardu ODBC.

ODBC identifikuje dva typy ovladačů: pro jednu vrstvu a pro více vrstev. **Ovladač pro jednu vrstvu** zpracovává volání ODBC i příkazy jazyka SQL. **Ovladač pro více vrstev** zpracovává volání ODBC, ale požadavky SQL předává přímo databázovému serveru. Může sice požadavek SQL přeformátovat tak, aby odpovídal verzi jazyka příslušného zdroje dat, ale kód jazyka SQL přitom nezpracovává.

**Vytvoření názvu zdroje dat ODBC** – zdroj dat je datová struktura standardu ODBC, která identifikuje databázi i databázový systém, který ji zpracovává. Datové zdroje se dělí do tří typů: souborové, systémové a uživatelské. **Souborový zdroj dat** je soubor, který mohou uživatelé databáze sdílet. Jediný požadavek spočívá v tom, že uživatelé mají stejný ovladač databázového systému a oprávnění pro přístup k databázi. **Systémový zdroj dat** je umístěn v jediném počítači. Se systémovým zdrojem dat může pracovat operační systém a libovolný uživatel tohoto systému (s příslušnými oprávněními). **Uživatelský zdroj dat** je dostupný pouze uživateli, který jej vytvořil. Každý vytvořený zdroj dat dostává **název DSN (data source name)**, pomocí něž lze na zdroj dat odkazovat.

Nejlepší variantou pro internetové aplikace je vytvoření systémového zdroje dat na webovém serveru. Uživatelé pomocí svého prohlížeče přistupují k webovému serveru, který následně pomocí systémového zdroje dat naváže připojení k databázovému systému a databázi.

Potřebujeme nyní vytvořit systémový zdroj dat pro databázi HSD firmy Heather Sweeney Designs, abychom ji mohli používat v aplikaci webového databázového zpracování. Databázi HSD jsme vytvořili v systému SQL Server 2012 a systémový zdroj dat musí poskytnout připojení k databázovému systému SQL Server 2012. V operačním systému Windows lze systémový zdroj dat vytvořit pomocí nástroje **ODBC Data Source Administrator**.

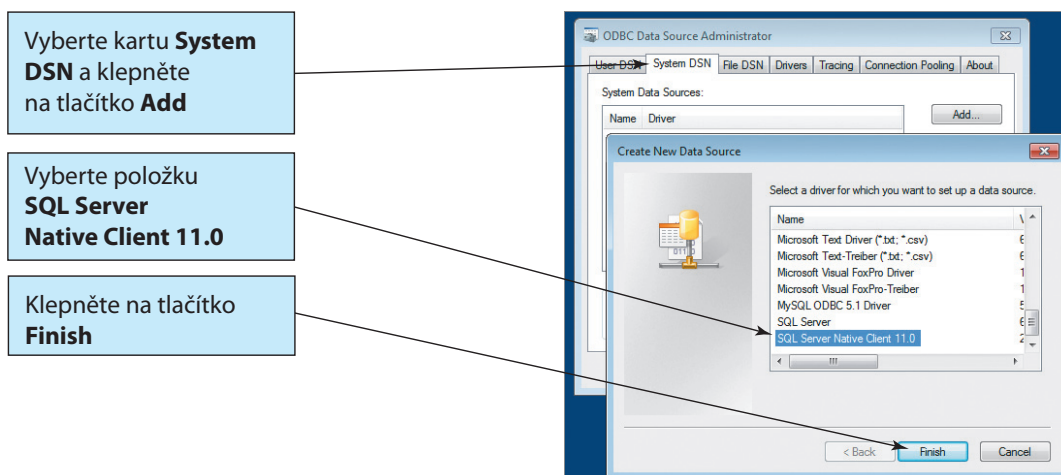
V systému Windows 7 lze nástroj ODBC Data Source Administrator<sup>5</sup> spustit takto:

1. Otevřete ovládací panel systému Windows klepnutím na tlačítko **Start** a příkaz **Control Panel**.
2. V okně Control Panel klepněte na položku **System and Security**.
3. V okně System and Security klepněte na možnost **Administrative Tools**.
4. V okně Administrative Tools poklepejte na ikonu zástupce **Data Sources (ODBC)**.

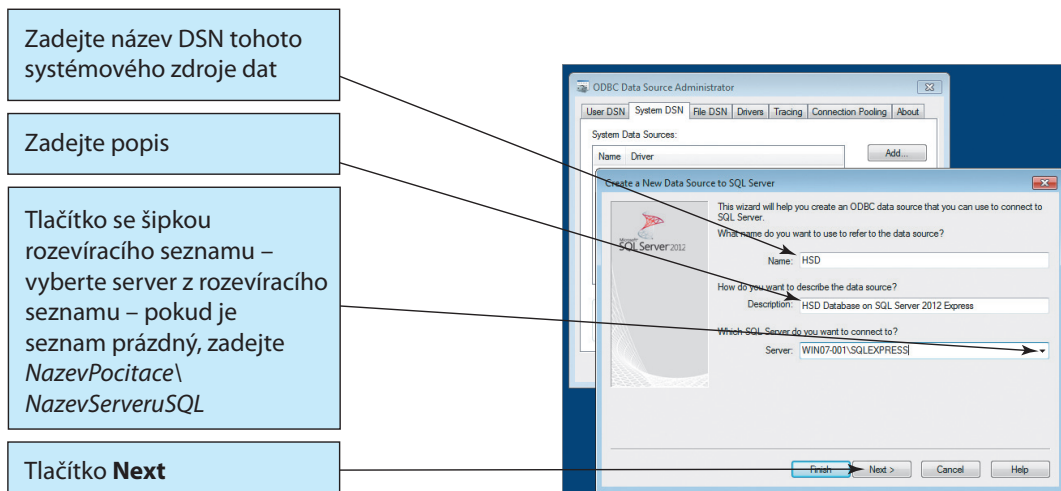
Zde můžete vytvořit systémový zdroj dat s názvem HSD, který bude pracovat s databází HSD firmy Heather Sweeney Designs v databázovém systému Microsoft SQL Server 2012:

<sup>5</sup> Upozornění: v 64bitových verzích systémů Microsoft Vista a Microsoft Windows 7 ve skutečnosti existuje *více verzí* nástroje ODBC Data Source Administrator. Musíte použít správnou verzi podle toho, zda pracujete s 32bitovou nebo 64bitovou verzí databázového systému. Věci se ještě komplikují tím, že všechny verze mají stejný název souboru *odbcad32.exe*! Pokyny na této stránce popisují spuštění 64bitové verze, která je kompatibilní s 64bitovými programy. Pokud se pomocí standardu ODBC připojujete k 32bitové verzi systému Microsoft Access 2010, Microsoft SQL Server 2012, Oracle Database 11g Release 2 nebo MySQL 5.5, musíte použít 32bitovou verzi nástroje ODBC Data Source Administrator v umístění **C:\Windows\SysWOW64\odbcad32.exe**. Jestliže je vše nastaveno správně, ale webové stránky se přesto správně nezobrazují, pravděpodobně je to způsobeno nesprávnou verzí tohoto nástroje.

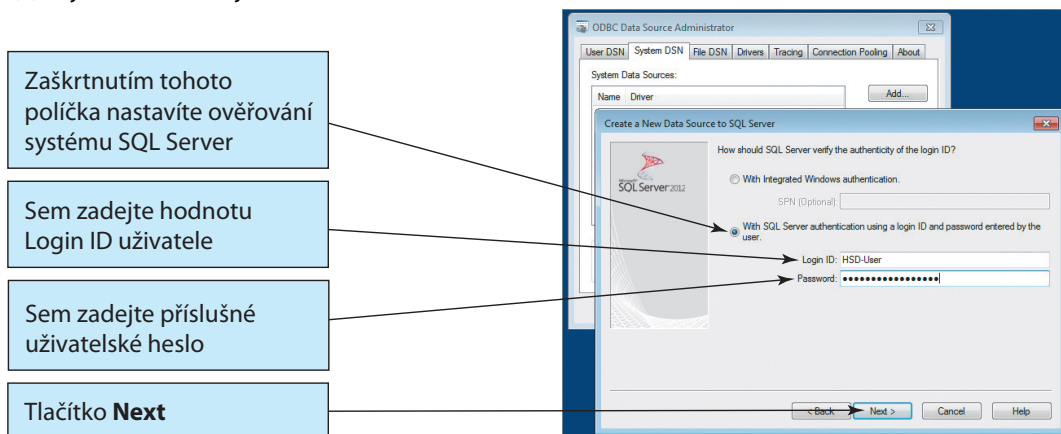
1. V nástroji ODBC Data Source Administrator klepněte na kartu **System DSN** a poté klepněte na tlačítko **Add**.
2. V dialogovém okně Create New Data Source je nutné navázat připojení k systému SQL Server 2012. Vyberte tedy možnost **SQL Server Native Client 11**, jak je zřejmé na obrázku 7.7 ze systému Windows 7.
3. Klepněte na tlačítko **Finish**. Zobrazí se dialogové okno Create New Data Source to SQL Server znázorněné na obrázku 7.8(a).
4. V dialogovém okně Create New Data Source to SQL Server zadejte informace o databázi HSD z obrázku 7.8(a) (všimněte si databázového serveru vybraného z rozevíracího seznamu Server) a poté klepněte na tlačítko **Next**.
  - **Poznámka:** Pokud se v rozevíracím seznamu Server nezobrazí položka serveru SQL, zadejte ji ručně ve tvaru **NazevPocitace\NazevServeruSQL**.
5. V dalším kroku, který je znázorněn na obrázku 7.8(b), vyberte klepnutím hodnotu přepínače **SQL Server authentication** a poté do pole Login ID zadejte hodnotu **HSD-User** a do pole Password hodnotu **HSD-User+password** – tyto přihlašovací údaje jsme vytvořili v kapitole 6. Po zadání těchto dat klepněte na tlačítko **Next**.
  - **Poznámka:** Pokud hodnoty Login ID a Password nejsou správné, zobrazí se v této fázi chybová zpráva. Zkontrolujte, zda jste správně vytvořili přihlášení k systému SQL Server podle popisu v kapitole 6 a zda jste na tomto místě uvedli správná data.
6. Podle obrázku 7.8(c) změňte zaškrtnutím políčka výchozí databázi, nastavte výchozí databázi na **HSD** a poté klepněte na tlačítko **Next**.
7. Na obrázku 7.8(d) je patrné, že na další stránce není nutné nastavovat žádné další možnosti, takže stačí klepnout na tlačítko **Finish**. Zobrazí se dialogové okno ODBC Microsoft SQL Server Setup, které vidíte na obrázku 7.8(e).
8. V dialogovém okně Microsoft SQL Server Setup vyzkoušejte připojení klepnutím na tlačítko **Test Data Source**.
9. Pokud jsou všechna nastavení správná, objeví se dialogové okno **SQL Server ODBC Data Source Test** z obrázku 7.8(f), které informuje o tom, že testy byly úspěšně dokončeny. Klepněte na tlačítko **OK**.
10. Klepněte na tlačítko **OK** dialogového okna ODBC Microsoft SQL Server Setup.
11. Dokončený systémový zdroj dat HSD je znázorněn na obrázku 7.9. Klepnutím na tlačítko **OK** ukončete nástroj ODBC Data Source Administrator.



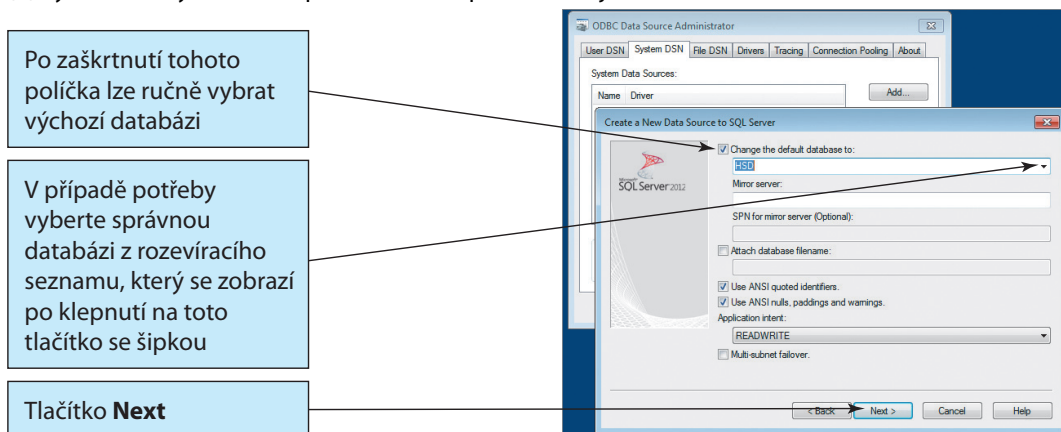
Obrázek 7.7: Dialogové okno Create New Data Source



(a) Pojmenování zdroje dat ODBC



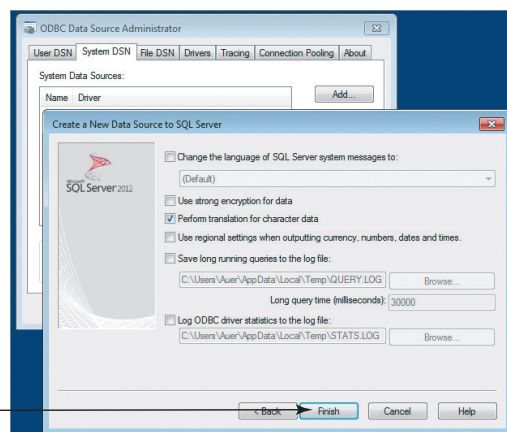
(b) Výběr metody ověřování pro uživatelské přihlašovací jméno



(c) Výběr výchozí databáze

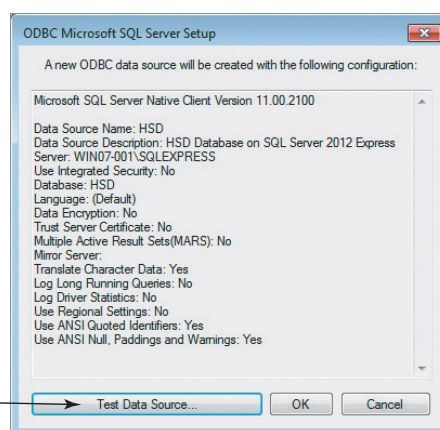
Tlačítko **Finish**

(d) Další možnosti nastavení



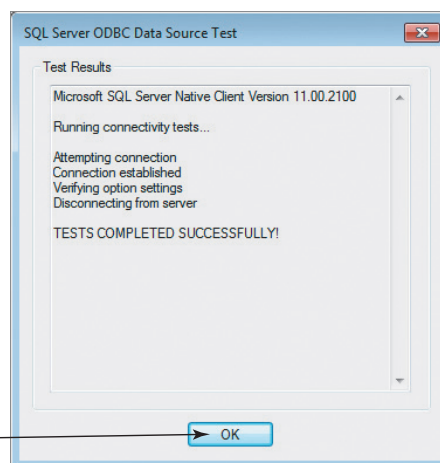
Tlačítko  
**Test Data Source**

(e) Testování zdroje dat

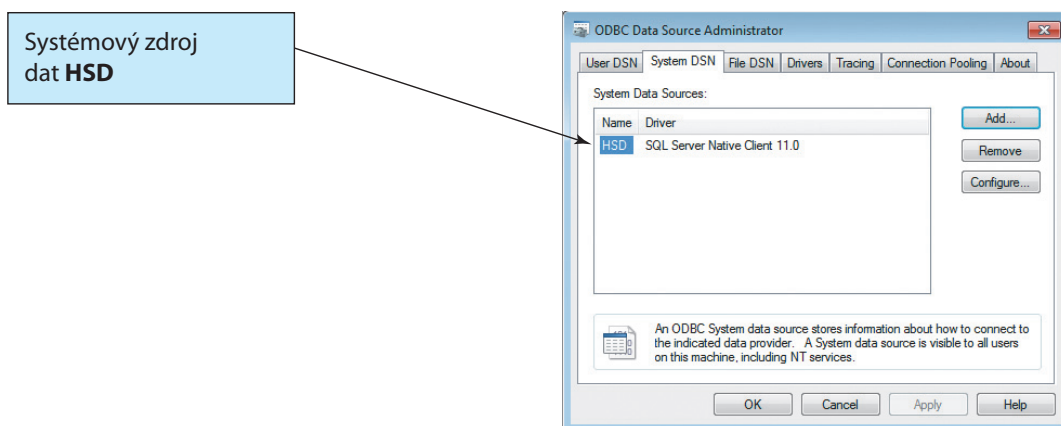


Tlačítko **OK**

(f) Úspěšně vyzkoušený zdroj dat



**Obrázek 7.8:** Dialogové okno Create New Data Source to SQL Server



Obrázek 7.9: Dokončený systémový zdroj dat HSD

## Webové zpracování pomocí webového serveru Microsoft IIS

Když jsme vytvořili zdroj dat ODBC, podívejme se nyní na webové databázové zpracování. K tomu potřebujeme webový server, který umožní ukládat a poskytovat webové stránky. Mohli bychom použít server Apache protokolu HTTP (je k dispozici na stránkách nadace Apache Software Foundation na adrese <http://www.apache.org>). Jedná se o nejpoužívanější webový server a některá z jeho verzí funguje prakticky v každém existujícím operačním systému. Avšak vzhledem k tomu, že v „Kurzů systému Access“ předpokládáme operační systém Windows a databázový systém Microsoft Access 2010, založíme svůj web na webovém serveru Microsoft IIS. Jedna výhoda tohoto webového serveru pro uživatele systémů Windows XP Professional, Windows Vista a Windows 7 spočívá v tom, že server IIS je již součástí operačního systému: v systému Windows XP je k dispozici IIS verze 5.1, systém Windows Vista obsahuje IIS verze 7.0 a v systému Windows 7 nalezneme IIS verze 7.5. Server IIS standardně není nainstalován, ale lze jej kdykoli doinstalovat. Každý uživatel si tedy může vyzkoušet tvorbu a zpracování webových stránek ve své pracovní stanici systému Windows.

### Poznámka

Tato část kapitoly týkající se webového databázového zpracování je sestavena s ohledem na co nejširší použití. Po menších úpravách lze následujícím postupem zprovoznit i webový server Apache, pokud je v systému k dispozici. Kdykoli je to možné, volíme produkty a technologie, které jsou dostupné v mnoha operačních systémech.

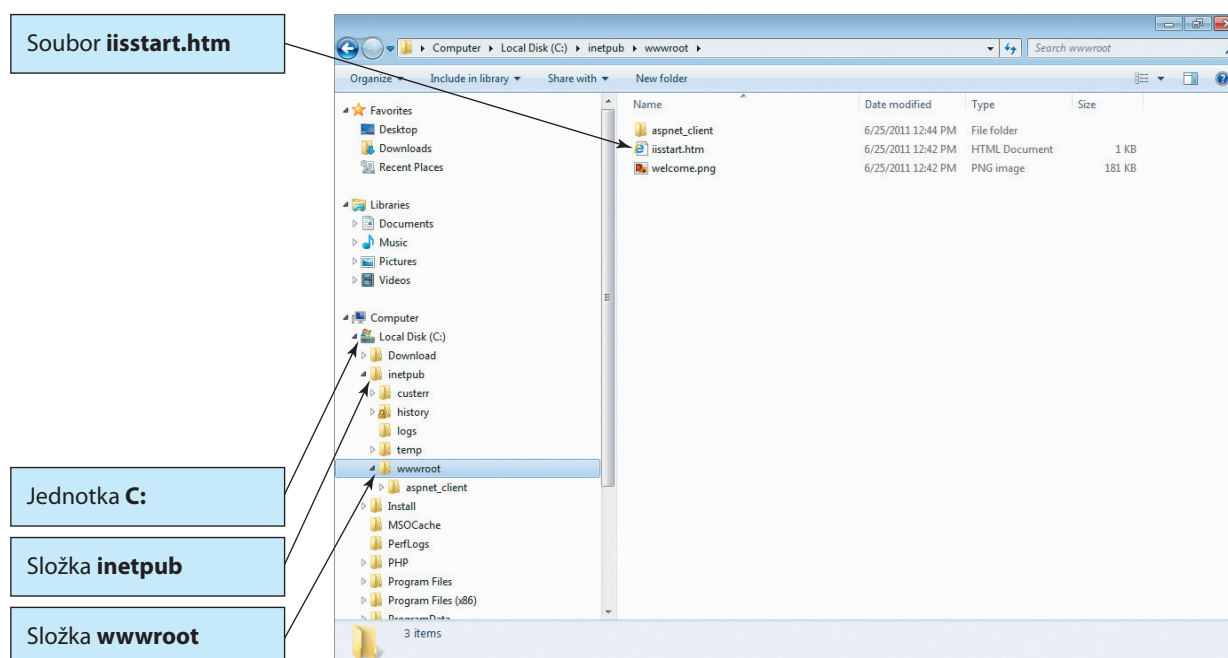
Po své instalaci server IIS vytvoří na disku C: složku **inetpub** v umístění C:\inetpub. V rámci složky inetpub se nachází složka **wwwroot**, kde server IIS ukládá základní webové stránky používané webovým serverem. Obrázek 7.10 znázorňuje tuto adresářovou strukturu po instalaci serveru IIS, přičemž v podokně souborů jsou zobrazeny soubory ze složky wwwroot. Všimněte si, že je nutné správně nastavit vlastnosti zabezpečení složky wwwroot, aby do ní měli uživatelé přístup. Skupině Windows Users je nutné ke složce wwwroot poskytnout oprávnění *Modify* a *Write*.

Ke správě serveru IIS slouží program, který se v systému Windows XP nazývá **Internet Information Services** a v systémech Windows Vista a Windows 7 má název **Internet Information Services Manager**. Chcete-li spustit některý z těchto programů, otevřete složku **Control Panel** a vyberte položku **Administrative Tools**. Pod položkou Administrative Tools je k dispozici ikona zástupce nástroje Internet Information Services nebo Internet Information Services Manager. Obrázek 7.11 představuje okno nástroje Internet Information Services Manager.

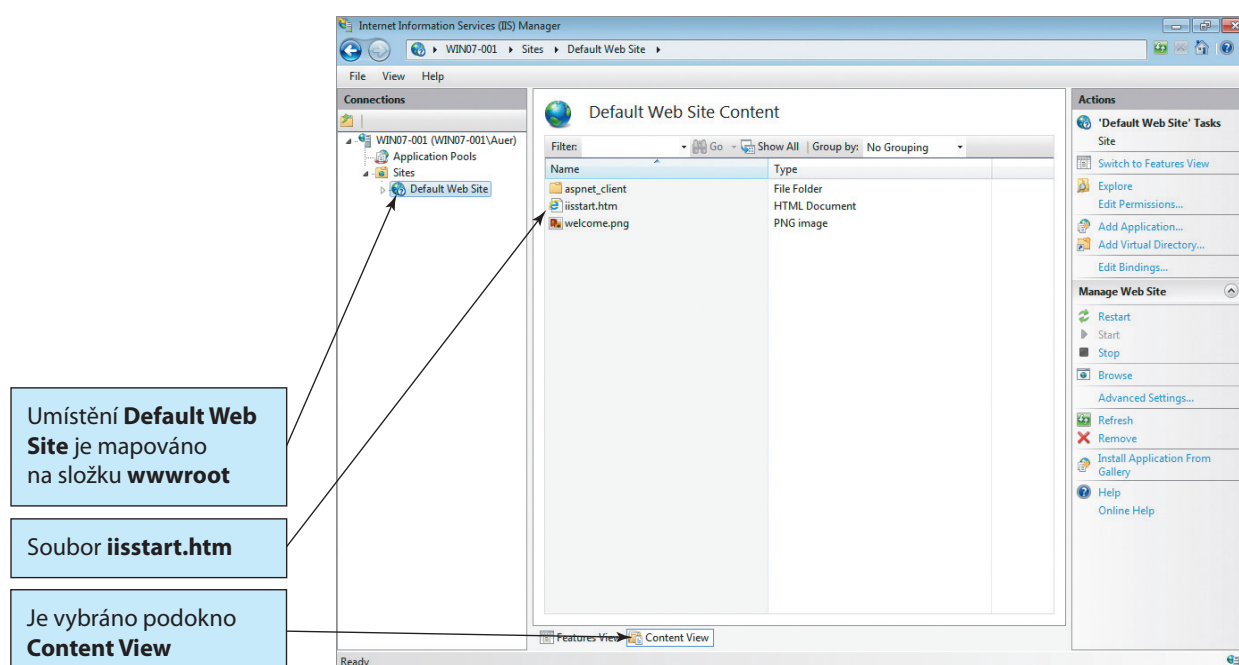
Všimněte si, že ve složce **Default Web Site** na obrázku 7.11 jsou umístěny stejné soubory, které se na obrázku 7.10 nacházejí ve složce wwwroot – jedná se o výchozí soubory, které server IIS vytváří

při své instalaci. V systému Windows XP soubor **iisstart.asp** generuje webovou stránku, kterou lze zobrazit v prohlížeči Internet Explorer (nebo jiném webovém prohlížeči, který s tímto webovým serverem naváže spojení přes Internet). Když webový server kontaktujete z pracovní stanice, kde je server IIS spuštěn, zavolá soubor iisstart.asp soubor **localstart.asp**, který poskytuje speciální webovou stránku pro administrátora webového serveru. Systémy Windows Vista a Windows 7 fungují podobně, ale místo uvedeného souboru se používá soubor iisstart.htm.

Chcete-li instalaci webového serveru vyzkoušet, otevřete okno webového prohlížeče, zadejte adresu URL *http://localhost* a stiskněte klávesu **Enter**. V systému Windows 7 se zobrazí webová stránka znázorněná na obrázku 7.12. Pokud webový prohlížeč nezobrazí odpovídající webovou stránku, není webový server nainstalován správně.



**Obrázek 7.10:** Složka wwwroot serveru IIS



**Obrázek 7.11:** Správa serveru IIS



**Obrázek 7.12:** Výchozí webová stránka serveru IIS

Nyní vytvoříme malý web, který budeme používat při databázovém zpracování databáze HSD. Nejdříve ve složce `wwwroot` vytvoříme novou podsložku s názvem `DBC` (database concepts). Tato nová složka bude sdružovat všechny webové stránky, které vyvineme na základě postupů a cvičení v této knize. V dalším kroku vytvoříme podsložku složky `DBC` s názvem `HSD`. Tato složka bude obsahovat webový server HSD. Tyto složky lze vytvořit pomocí Průzkumníku Windows, jak je patrné na obrázku 7.10.

## Začínáme s webovými stránkami v jazyce HTML

Nejjednodušší webové stránky lze vytvářet pomocí **jazyka HTML (Hypertext Markup Language)**. Termín *hypertext* se vztahuje k tomu, že do webových stránek lze vkládat odkazy na jiné objekty, jako jsou webové stránky, mapy, obrázky, a dokonce i zvukové soubory a videosekvence. Po klepnutí na příslušný odkaz je možné bezprostředně přejít na další objekt, který se zobrazí ve webovém prohlížeči. Samotný jazyk HTML sestává ze standardní sady **syntaktických pravidel HTML** a **dokumentových značek HTML**, které webové prohlížeče dokážou interpretovat a zobrazit pomocí nich na obrazovce konkrétní obsah.

Značky jsou obvykle párové a zahrnují příslušnou počáteční a odpovídající koncovou značku, která obsahuje symbol lomítka ( / ). Textový odstavec je tedy formátován pomocí značek `<p>{zde je text odstavce}</p>` a hlavní nadpis je zapsán ve tvaru `<h1>{zde je text nadpisu}</h1>`. Některé značky nevyžadují párovou koncovou značku, protože mohou stát samostatně. Chceme-li například do webové stránky vložit vodorovnou čáru, použijeme příslušnou značku `<hr />` (horizontal rule). Tyto samostatné stojící značky musí jako v uvedeném příkladu obsahovat symbol lomítka.

Pravidla jazyka HTML jsou definována ve standardech **konsorcia W3C (World Wide Web Consortium)** a podrobnosti současných a navržených standardů jsou k dispozici na adrese <http://www.w3.org> (tento web také obsahuje vynikající návody jazyka HTML<sup>6</sup>). Na webu W3C nalezneme

<sup>6</sup> Další informace o jazyce HTML nalezneme na webu W3C na adrese <http://www.w3.org>. Dobré kurzy jazyka HTML připravil David Raggett: „Getting Started with HTML“ (Začínáme s jazykem HTML – <http://www.w3.org/MarkUp/Guide/>), „More Advanced Features“ (Pokročilejší funkce – <http://www.w3.org/MarkUp/Guide/Advanced.html>) a „Adding a Touch of Style“ (Formátování pomocí stylů – <http://www.w3.org/MarkUp/Guide/Style.html>).

aktuální standardy jazyka HTML, **jazyka XML (Extensible Markup Language)**, kterým se budeme zabývat v další části této kapitoly, a kombinace obou jazyků s názvem **XHTML**. Úplný rozbor těchto standardů přesahuje rozsah našeho textu. V této kapitole budeme vycházet z aktuálního standardu HTML 4.01 (obvykle v takzvané „přísné“ formě).

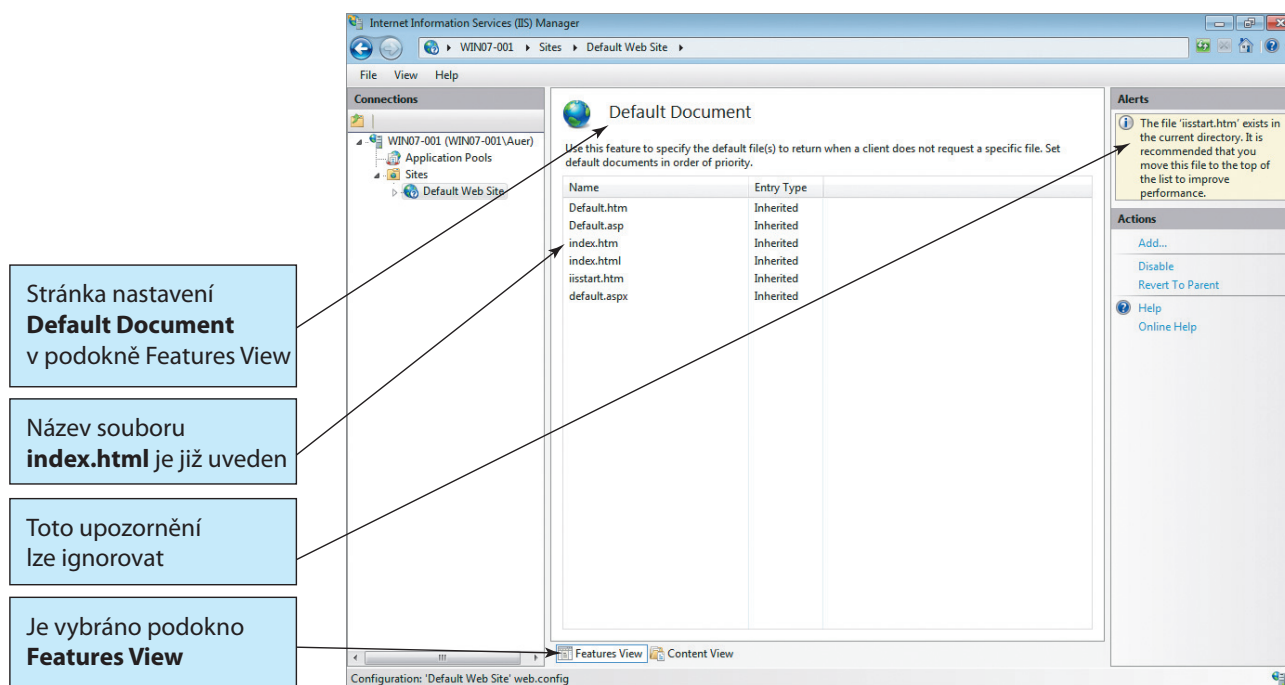
V této kapitole vytvoříme jednoduchou domovskou stránku webu Heather Sweeney Designs ve formátu HTML a umístíme ji do složky HSD. Zakrátko popíšeme některé z mnoha dostupných editorů webových stránek, ale v praxi k vytvoření webových stránek stačí jednoduchý textový editor. Svou první webovou stránku napíšeme v textovém editoru společnosti Microsoft, který se nazývá Notepad (Poznámkový blok) a pracuje s textem v kódování ASCII. Jeho výhoda spočívá v tom, že je k dispozici v každé verzi operačního systému Windows.

## Webová stránka index.html

Vytvořený soubor bude mít název **index.html**. Název index.html je nutné použít proto, že se z hlediska webových serverů jedná o speciální název. Název souboru index.html patří mezi několik málo názvů, které *většina* webových serverů zobrazí automaticky, když dostanou požadavek s adresou URL bez uvedení konkrétního názvu souboru. Bude se tedy jednat o novou výchozí stránku, kterou naše webová databázová aplikace bude zobrazovat. Všimněte si však zvýraznění „*většina* webových serverů“ v předchozí větě. Servery Apache, IIS 7.0 a IIS 7.5 (viz obrázek 7.13) jsou nakonfigurovány tak, aby soubor index.html rozpoznávaly automaticky. Pro server IIS 5.1 to však neplatí.<sup>7</sup>

## Vytvoření webové stránky index.html

Nyní můžeme vytvořit webovou stránku index.html, která zahrnuje základní příkazy jazyka HTML. Její kód je uveden na obrázku 7.14. Obrázek 7.15 znázorňuje kód HTML v editoru Notepad.



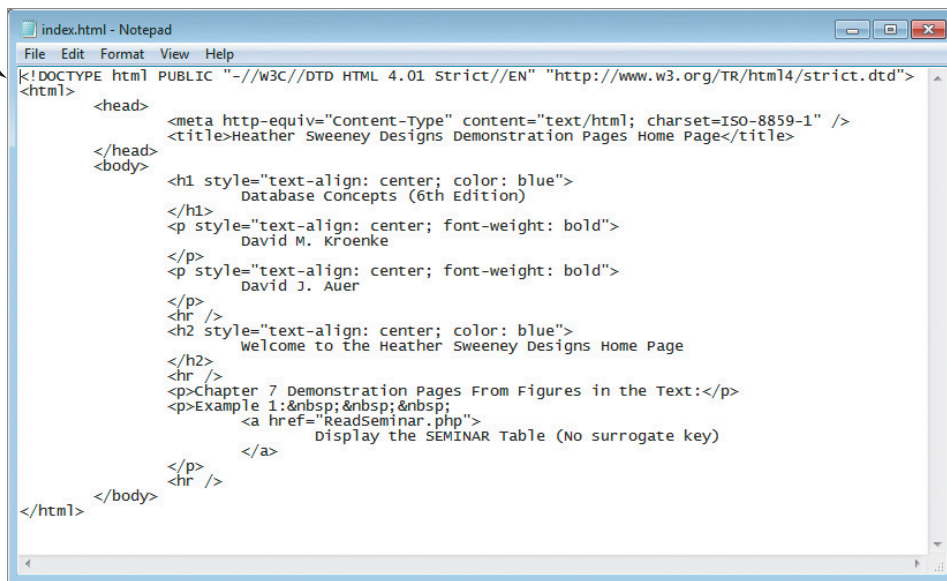
**Obrázek 7.13:** Soubor index.html v nástroji IIS Manager systému Windows 7

<sup>7</sup> Používáte-li kombinaci operačního systému Windows XP a serveru IIS 5.1, musíte stránku index.html přidat do seznamu rozpoznávaných souborů. To lze provést v serveru Internet Information Services. V okně Internet Information Services klepněte pravým tlačítkem myši na objekt **Web Sites**. V zobrazené místní nabídce klepněte na příkaz **Properties**, který zobrazí dialogové okno Web Sites Properties. Poté vyberte kartu **Documents** a klepnutím na tlačítko **Add** zobrazíte dialogové okno Add Default Document. Poté zadejte název dokumentu (název souboru) **index.html** a klepněte na tlačítko **OK**. Nakonec dialogové okno Web Sites Properties zavřete.

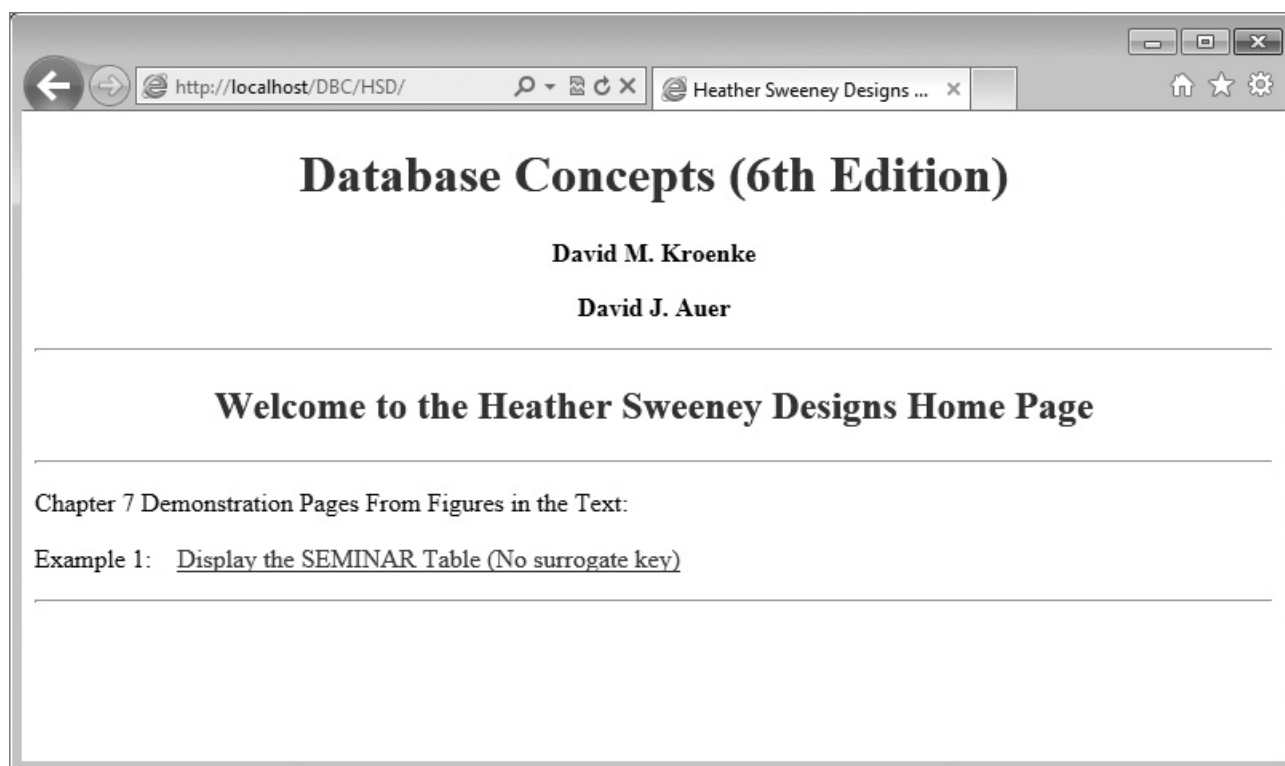
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    <title>Heather Sweeney Designs Demonstration Pages Home Page</title>
  </head>
  <body>
    <h1 style="text-align: center; color: blue">
      Database Concepts (6th Edition)
    </h1>
    <p style="text-align: center; font-weight: bold">
      David M. Kroenke
    </p>
    <p style="text-align: center; font-weight: bold">
      David J. Auer
    </p>
    <hr />
    <h2 style="text-align: center; color: blue">
      Welcome to the Heather Sweeney Designs Home Page
    </h2>
    <hr />
    <p>Chapter 7 Demonstration Pages From Figures in the Text:</p>
    <p><b>Example 1:<br>
      <a href="ReadSeminar.php">
        Display the SEMINAR Table (No surrogate key)
      </a>
    </p>
    <hr />
  </body>
</html>
```

**Obrázek 7.14:** Kód HTML souboru index.html ve složce HSD

Kód HTML souboru **index.html** –  
k uspořádání a lepší  
čitelnosti kódu slouží  
odsazování



**Obrázek 7.15:** Kód HTML souboru index.html v editoru Notepad



Obrázek 7.16: Webová stránka index.html ve složce HSD

Pokud nyní zadáme adresu URL `http://localhost/DBC/HSD`, objeví se webová stránka znázorněná na obrázku 7.16.

## Poznámka

V kódu HTML souboru index.html se nachází následující segment:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

jedná se o **deklaraci typu dokumentu (DTD – document type declaration)** jazyka HTML/XML, která umožňuje zkontrolovat a ověřit obsah vytvořeného kódu. Deklarací DTD se budeme věnovat dále v této kapitole. Prozatím deklaraci uvedeme v kódu v té podobě, jak je zapsána.

## Webové databázové zpracování pomocí jazyka PHP

Když jsme nyní nastavili svůj jednoduchý web, můžeme rozšířit jeho možnosti pomocí webového vývojového prostředí, které dovoluje propojit webové stránky s databází. Tuto možnost nabízí několik technologií. Vývojáři, kteří pracují s produkty společnosti Microsoft, obvykle používají architekturu .NET a technologii ASP.NET. Vývojáři, kteří pracují s webovým serverem Apache, mohou upřednostňovat tvorbu souborů JSP ve skriptovacím jazyce JavaScript, nebo mohou používat programovací jazyk Java v prostředí Java Enterprise Edition (Java EE).

**Skriptovací jazyk PHP** – v této kapitole budeme používat skriptovací jazyk **PHP**. Jeho zkratka PHP vychází ze slov **PHP:Hypertext Processor** (dříve se vysvětlovala jako *Personal Hypertext Processor*). Jedná se o skriptovací jazyk, který lze integrovat do webových stránek. Jazyk PHP je

mimořádně oblíbený. V létě 2007 tento jazyk fungoval na více než 2 milionech internetových domén<sup>8</sup> a podle indexu programátorské komunity TIOBE z roku 2012 se jazyk PHP mezi nejoblíbenějšími programovacími jazyky zařadil na šesté místo (následoval postupně za těmito jazyky: C, Java, C++, Objective-C a C#).<sup>9</sup> V roce 2008 zaujal jazyk PHP čtvrté místo (za jazykem Visual Basic), zatímco roku 2010 jazyk PHP na jistou dobu obsadil třetí místo. Jazyk PHP si podle všeho trvale udržuje svou oblibu mezi programátory a návrháři webových stránek. Lze se jej snadno naučit a používat jej ve většině prostředí webových serverů a s většinou databází. Jedná se zároveň o produkt typu open-source, který si můžete volně stáhnout z webu PHP (<http://www.php.net>).

**Integrované vývojové prostředí (IDE) Eclipse** – k psaní jednoduchých webových stránek sice stačí základní textový editor, jako je Notepad, ale při tvorbě složitějších stránek je vhodnější přejít na **integrováné vývojové prostředí (IDE – Integrated Development Environment)**. Prostředí IDE poskytuje robustnější a uživatelsky přívětivější postupy, jak vytvářet a spravovat webové stránky. Pracujete-li s produkty společnosti Microsoft, pravděpodobně používáte prostředí Visual Studio (nebo jeho volně dostupné vydání Visual Studio 2010 Express Edition, které je k dispozici na adrese <http://www.microsoft.com/express/>). Jestliže pracujete s jazykem JavaScript nebo Java, můžete dát přednost prostředí NetBeans IDE (které je dostupné na adrese <http://www.netbeans.org>).

V této kapitole opět využijeme výsledků vývojářské komunity open-source a použijeme prostředí **Eclipse IDE**. Prostředí Eclipse poskytuje architekturu, kterou mohou doplňkové moduly upravovat k mnoha účelům. V případě jazyka PHP můžeme pracovat v prostředí Eclipse po úpravách projektu **Eclipse PDT (PHP Development Tools) Project**, jehož hlavním cílem je zajistit vývojové prostředí jazyka PHP v prostředí Eclipse (obecné informace naleznete na stránce <http://www.eclipse.org/projects/project.php?id=tools.pdt> a soubory ke stažení jsou k dispozici na adrese <http://www.eclipse.org/pdt/downloads/> – stáhněte si aktuální stabilní sestavení pro svůj operační systém).<sup>10</sup>

Obrázek 7.17 představuje soubor index.html vytvořený v prostředí Eclipse IDE. Porovnejte tuto verzi s verzí z obrázku 7.16, která vznikla v editoru Notepad.

**Vytvoření stránky ReadSeminar.php** – když jsme nyní nastavili svůj jednoduchý web, můžeme do jeho webových stránek začít integrovat kód PHP. Nejdříve vytvoříme stránku, která bude načítat data z databázové tabulky a zobrazí výsledky v rámci webové stránky. Konkrétně ve složce HSD vytvoříme webovou stránku s názvem ReadSeminar.php, která bude spouštět následující dotaz SQL:

```
/* *** SQL-QUERY-CH07-01 *** */
SELECT * FROM SEMINAR;
```

Tato stránka ve webovém prohlížeči zobrazí výsledek dotazu bez náhradního klíče tabulky SeminarID. Kód jazyků HTML a PHP stránky ReadSeminar.php je znázorněn na obrázku 7.18 a stejný kód v prostředí Eclipse vidíme na obrázku 7.19.

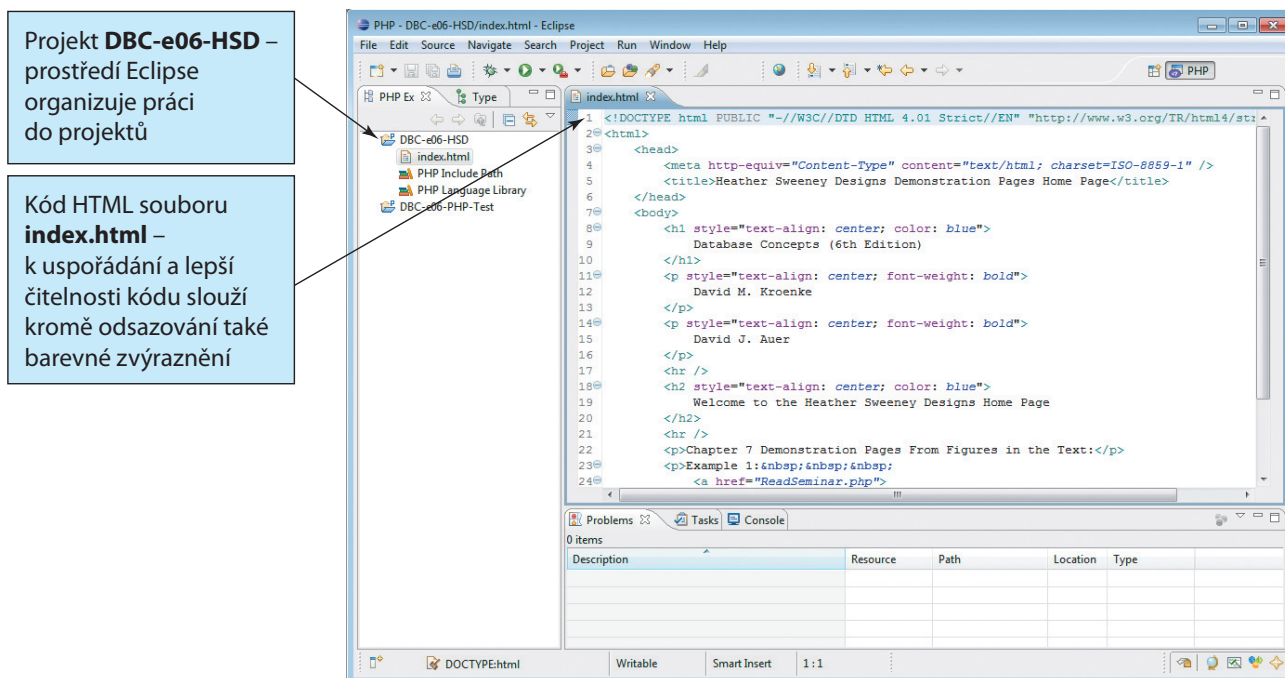
Pokud do webového prohlížeče zadáte adresu URL <http://localhost/DBC/HSD> a potom na zobrazené stránce klepnete na odkaz **Example 1: Display the SEMINAR Table (No surrogate key)**, zobrazí se webová stránka z obrázku 7.20.

Kód v souboru ReadSeminar.php kombinuje jazyk HTML (který se provádí v pracovní stanici uživatele) a příkazy jazyka PHP (které vykonává webový server). Příkazy, které jsou na obrázku 7.18 uzavřeny mezi značky `<?php` a `?>`, představují programový kód, který realizuje počítač webového serveru. Zbývající část kódu je v jazyce HTML, který je generován a odeslán do prohlížeče klienta. Příkazy na obrázku 7.18:

<sup>8</sup> Viz <http://www.php.net/usage.php>.

<sup>9</sup> Viz <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.

<sup>10</sup> Jak je uvedeno na stránce ke stažení nástrojů PDT, je také nutné nainstalovat prostředí JRE neboli Java Runtime Environment (v době psaní této knihy se jedná o verzi JRE 7 Update 4 od společnosti Oracle), které naleznete na adrese <http://java.com/en/download/manual.jsp>. Všimněte si také, že verze prostředí Eclipse pro operační systém Windows se nainstaluje tak jako většina programů pro tento systém: je nutné vytvořit složku s názvem EclipsePDT ve složce Program Files na jednotce C: (úplná cesta bude mít tvar C:\Program Files\EclipsePDT), rozbalit soubory Eclipse PDT do této složky a poté na ploše vytvořit zástupce spustitelného souboru eclipse.exe.



Obrázek 7.17: Kód jazyka HTML souboru index.html v prostředí Eclipse IDE

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/
      html; charset=UTF-8">
    <title>ReadSeminar PHP Page</title>
    <style type="text/css">
      h1 {text-align: center; color: blue}
      h2 {font-family: Ariel, sans-serif;
        text-align: left; color: blue}
      p.footer {text-align: center}
      table.output {font-family: Ariel, sans-serif}
    </style>
  </head>
  <body>

```

představují normální kód HTML. Při odeslání do prohlížeče příkazy: (1) nastaví název okna prohlížeče na *ReadSeminarPHP Page*, (2) definují styly použité v nadpisech, tabulce výsledků a zápatí, a (3) zajistí některé další funkce související s jazykem HTML.<sup>11</sup> Další skupina příkazů je obsažena mezi značkami `<?php` a `?>` a jedná se tedy o kód jazyka PHP, který bude spuštěn na webovém serveru. Všimněte si také, že všechny příkazy PHP musí podobně jako příkazy SQL končit středníkem (;).

<sup>11</sup> Styly umožňují řídit vizuální prezentaci webové stránky a definují se v sekci HTML mezi značkami `<style>` a `</style>`. Další informace o stylech naleznete v kurzu Davida Raggetta s názvem „Adding a Touch of Style“ (Formátování pomocí stylů) na adrese <http://www.w3.org/MarkUp/Guide/Style.html>.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ReadSeminar PHP Page</title>
    <style type="text/css">
      h1 {text-align: center; color: blue}
      h2 {font-family: Ariel, sans-serif; text-align: left; color: blue}
      p.footer {text-align: center}
      table.output {font-family: Ariel, sans-serif}
    </style>
  </head>
  <body>
    <?php
      // Get connection
      $Conn = odbc_connect('HSD', 'HSD-User', 'HSD-User+password');

      // Test connection
      if (!Conn)
      {
        exit ("ODBC Connection Failed: " . $Conn);
      }
      // Create SQL statement
      $SQL = "SELECT * FROM SEMINAR";

      // Execute SQL statement
      $RecordSet = odbc_exec($Conn,$SQL);

      // Test existence of recordset
      if (!$RecordSet)
      {
        exit ("SQL Statement Error: " . $SQL);
      }
    ?>

    <!-- Page Headers -->
    <h1>
      The Heather Sweeney Designs SEMINAR Table
    </h1>
    <hr />
    <h2>
      SEMINAR
    </h2>
    <?php
      // Table headers
      echo "<table class='output' border='1'>
        <tr>
          <th>SeminarDate</th>
          <th>SeminarTime</th>
          <th>Location</th>
          <th>SeminarTitle</th>
        </tr>";

      // Table data
      while($RecordSetRow = odbc_fetch_array($RecordSet))
      {
        echo "<tr>";
        echo "<td>" . $RecordSetRow['SeminarDate'] . "</td>";
        echo "<td>" . $RecordSetRow['SeminarTime'] . "</td>";
        echo "<td>" . $RecordSetRow['Location'] . "</td>";

```