

**akční dotaz**

Dotaz, který kopíruje nebo mění data. Akční dotazy zahrnují dotazy připojovací, odstraňovací, vytvářející tabulku a aktualizací. Odstraňovací a aktualizací dotazy mění existující data; dotazy připojovací a vytvářející tabulku kopírují existující data. Naproti tomu výběrové dotazy předávají datové záznamy. Předávací dotaz SQL může být také akčním dotazem.

### **agregační funkce**

Funkce jako **Sum**, **Count**, **Avg** a **Var**, které můžete použít k výpočtu součtů. Pokud chcete vypočítat různé statistiky můžete při zadávání výrazů a při programování použít agregační funkce SQL (včetně čtyř výše uvedených) a agregační funkce domény.

**alias**

Visual Basic: Náhradní slovo, které jste dali externí proceduře, aby nedošlo ke konfliktu s klíčovým slovem, obecnou proměnnou, konstantou aplikace nebo názvem, který není povolen konvencemi aplikace Visual Basic.

Strukturovaný jazyk dotazu (SQL): Náhradní název, které jste dali poli nebo výrazu v příkazu SELECT (použitím klauzule AS).

**znaková sada ANSI**

8-bitová znaková sada použitá systémem Microsoft Windows, která umožňuje použít až 256 znaků (0 až 255) pomocí klávesnice. Znaková sada ASCII je podsada sady ANSI.

**připojovací dotaz**

Akční dotaz, který přidává nové záznamy na konec existující tabulky nebo dotazu. Připojovací dotazy nevracejí záznamy (řádky).

**argument**

Konstanta, proměnná nebo výraz, který nahrazuje dodatečnou informaci pro akci, proceduru nebo metodu. Argument s operandem ve výrazu použijete, chcete-li dodat informaci požadovanou některými akcemi.

## **znaková sada ASCII**

Americký standardní kód pro výměnu informací (ASCII), 7-bitová znaková sada široce používaná odpovídá písmenům a symbolům nacházejícím se na amerických klávesnicích. Znaková sada ASCII je stejná jako prvních 128 znaků (0 – 127) ve znakové sadě ANSI.

### **hvězdička**

Znak (\*) používaný jako zástupný znak ve výrazech SQL LIKE, které místo něj přijmou jakýkoliv řetězec znaků. Například výraz LIKE `"*město"` vrátí záznamy obsahující řetězec "město" následovaný žádným nebo více znaky.

Když vytvoříte dotaz nebo zadáte příkaz SQL, použijete hvězdičku, chcete-li zahrnout všechna pole, která aktuálně existují v podkladové tabulce nebo dotazu. Například `SELECT * FROM MojeTabulka` vrátí všechna pole (sloupce) z tabulky.

### **pole typu automatické číslo**

Datový typ pole, který pro každý záznam automaticky ukládá jednoznačné číslo, jakmile je záznam přidán do tabulky. Pole typu automatické číslo vždy používá datový typ **Long** a čísla vytvořená tímto polem nelze upravovat. (Říká se mu také pole typu počítadlo.)

## **základní tabulka**

Tabulka v databázi Microsoft Jet. Tabulka definuje strukturu relační databáze a je objektem, který ukládá data do záznamů a polí. Se strukturou základní tabulky můžete pracovat pomocí objektů DAO nebo příkazů SQL pro definici dat, a data v základní tabulce můžete upravit pomocí objektů **Recordset** nebo akčních dotazů.

### **záložka**

Vlastnost objektu **Recordset**, která obsahuje binární řetězec určující aktuální záznam. Přiřadíte-li hodnotu **Bookmark** proměnné a pak se přesunete na jiný záznam, předchozí záznam opět zaktualizujete nastavením vlastnosti **Bookmark** na řetězcovou proměnnou.

### **datový typ Boolean**

Hodnota **True/False** nebo ano/ne. Hodnoty **Boolean** jsou v databázi Microsoft Jet obvykle uloženy v polích Bit; některé databáze ale tento datový typ přímo nepodporují.

**výraz Boolean**

Výraz, který odpovídá buď hodnotě **True** nebo **False**.

**datový typ Byte**

Základní datový typ používaný pro malá kladná celá čísla v rozmezí 0 až 255.

### **kartézský součin**

Výsledek spojení dvou relačních tabulek dávající všechny možné nařazené kombinace řádků z první tabulky se všemi řádky z druhé tabulky.

Obecně získáte Kartézský součin provedením SQL příkazu SELECT odkazujícího se na dvě nebo více tabulek v parametru FROM a neobsahujícího parametry WHERE nebo JOIN, které určují, jak budou tabulky spojeny.

### **postupné odstranění**

Pro vztahy, které si vynucují referenční integritu mezi tabulkami, jde o možnost, kdy odstranění záznamu z primární tabulky způsobí automatické odstranění všech souvisejících záznamů v související tabulce nebo tabulkách.

Například vytvoříte vztah mezi tabulkou (primární) Zákazníci a tabulkou (cizí) Objednávky se zapnutou možností postupného odstranění. Když odstraníte záznam v tabulce Zákazníci, všechny zakázky spojené se zákazníkem budou také odstraněny v cizí tabulce Objednávky.

### **postupná aktualizace**

Pro vztahy, které si vynucují referenční integritu mezi tabulkami, jde o možnost, kdy změna primárního klíče v záznamu v primární tabulce způsobí automatickou aktualizaci cizího klíče ve všech souvisejících záznamech v související cizí tabulce nebo tabulkách.

Například vytvoříte vztah mezi tabulkou (primární) Zákazníci a tabulkou (cizí) Objednávky se zapnutou možností postupné aktualizace. Když změníte primární klíč v záznamu v tabulce Zákazníci, všechny zakázky spojené s tímto zákazníkem budou také změněny v cizí tabulce Objednávky.

**kaskáda**

Proces, kdy jedna akce spustí další. Například je-li definován vztah postupné aktualizace pro dvě (nebo více) tabulek, aktualizace primárního klíče v primární tabulce automaticky spustí (postupně provede) změny u cizí tabulky.

### **rozlišující malá a velká písmena**

Schopný rozlišit malá a velká písmena. Hledání s tímto rozlišením najde pouze takový text, který přesně odpovídá zadanému vzoru i co se týče velikosti písmen. Takové hledání by například považovalo za odlišné řetězce "ZeroLengthStr" a "zerolengthstr". Operace databáze Microsoft Jet nerozlišují malá a velká písmena. Toto rozlišení mají však některé další databázové řídicí systémy.

**třída**

Formální definice objektu. Třída se chová jako šablona, podle které je při běhu programu vytvořen příslušný objekt. Třída definuje vlastnosti objektu a metody používané pro ovládání chování objektu.

**kolekce**

Objekt, který obsahuje sadu souvisejících objektů. Umístění objektu v kolekci se může změnit , kdykoliv v kolekci dojde ke změnám; proto se umístění jakéhokoliv určitého objektu v kolekci může měnit.

**sloupec**

Vizuální zobrazení pole v mřížce. Sloupec definuje datový typ, velikost a další atributy jednoho pole nebo řádku (záznamu) dat. Všechny sloupce dohromady definují řádek (záznam) v databázi. Jednotlivý sloupec obsahuje data v tabulce související typem a účelem; to znamená, že definice sloupce se nemění řádek od řádku.

**potvrdit**

Přijmout nevyřizenou transakci. Používáte-li zpracování transakce a začne-li transakce používat DAO, žádná ze změn provedených v transakci se nezapiše do databáze, dokud nepotvrdíte nebo nepřijmete transakci.

## **shodný**

Stav objektu **Recordset** typu vícenásobné tabulky, který umožňuje provést pouze ty aktualizace, jejichž výsledkem je shodný vzhled dat. Například v objektu **Recordset**, který je spojením dvou nebo více tabulek (vztah 1:mnoho), by shodný dotaz neumožnil nastavit mnohostranný klíč na hodnotu, která není v jednostranné tabulce.

### **kopírovací vyrovnávací paměť**

Místo vytvořené databázovým jádrem Microsoft Jet pro obsah záznamu, který je otevřen pro úpravy. Metoda **Edit** zkopíruje aktuální záznam do kopírovací vyrovnávací paměti; metoda **AddNew** smaže vyrovnávací paměť pro nový záznam a nastaví výchozí hodnoty; a metoda **Update** uloží data z kopírovací vyrovnávací paměti do databáze, přičemž nahradí aktuální záznam nebo vloží nový záznam. Jakýkoliv příkaz, který nastaví nebo přesune ukazatel na aktuální záznam, způsobí vymazání kopírovací vyrovnávací paměti. Například použití metody **MoveNext** nebo změna vlastnosti **Index** tabulky způsobí ztrátu obsahu kopírovací vyrovnávací paměti.

**spojovací řetězec**

Řetězec používaný k definici zdroje dat pro externí databázi. Spojovací řetězec je obvykle přiřazen vlastnosti **Connect** objektu **QueryDef**, **TableDef**, **Connection** nebo **Database** nebo jako argument pro metodu **OpenDatabase**.

### **datový typ AutoNumber**

Atribut pole, který pro každý záznam automaticky ukládá jednoznačné číslo, jakmile je záznam přidán do tabulky. Pole typu automatické číslo používá datový typ **Long** a jeho vlastnost **Attributes** je nastavena na **dbAutoIncrement**. Číslo vytvořené polem typu automatické číslo nelze upravovat. (Říká se mu také počítadlo.)

**Poznámka:** Pro replikované tabulky v replikovaných databázích se pole typu automatické číslo neztvětšuje po 1. Místo toho je jí přiřazena náhodná hodnota mezi -2,000,000,000 a 2,000,000,000.

**kritéria**

Sada omezujících podmínek, jako je = "Dánsko" (znamenající rovnající se Dánsku) nebo > 30000, používaných při tvorbě dotazů nebo filtru, chcete-li zobrazit určitou skupinu záznamů.

### **křížový dotaz**

Dotaz, který vypočítá součet, průměr, počet nebo jiný typ součtu u záznamů, a výsledek pak seřadí podle dvou typů informace – jeden dole na levé straně mřížky (hlavičky řádků) a druhý nahoře (hlavičky sloupců). Například dotaz Čtvrtletní objednávky podle výrobků ve vzorové databázi Northwind je křížový dotaz.

### **datový typ Currency**

Datový typ, který je užitečný pro výpočty zahrnující peníze nebo výpočty s pevnou čárkou, ve kterých je mimořádně důležitá přesnost. Tento datový typ se používá pro uložení čísel s až 15 číslicemi nalevo od desetinné čárky a 4 číslice napravo. Protože datový typ **Currency** používá nespojité hodnoty pro všechny částky, při výpočtu součtů se neprojeví binární zaokrouhlování.

### **aktuální databáze**

Objekt **Database** vrácený funkcí **CurrentDB**( ). Odkaz `DBEngine.Workspaces(0).Databases(0)` vrací první otevřenou databázi. Toto pojetí se používá pouze u aplikace Microsoft Access.

### **aktuální index**

Pro indexovaný objekt **Recordset** typu tabulka jde o index nejnověji nastavený vlastností **Index**. Tento index je základem pro seřazení záznamů v objektu **Recordset** typu tabulka a používá ho metoda **Seek** k nalezení záznamů. Objekt **Recordset** může mít více než jeden index, ale v danou chvíli může použít pouze jeden index (ačkoliv na něm může objekt **TableDef** mít definováno několik indexů). Databázové jádro Microsoft Jet může při vyhodnocení dotazu používat více než jeden index.

### **aktuální záznam**

Záznam v objektu **Recordset**, který lze použít při úpravě nebo prozkoumání dat. Chcete-li změnit umístění aktuálního záznamu v sadě záznamů, použijte metody **Move**. Chcete-li změnit umístění aktuálního záznamu podle zvláštních kritérií, použijte metod **Find** (u objektu **Recordset** typu snímek nebo dynamická sada) nebo metodu **Seek** (u objektu **Recordset** typu tabulka).

Pouze jeden záznam v sadě záznamů **Recordset** může být aktuální záznam; ale sada **Recordset** nemusí mít žádný aktuální záznam. Například byl odstraněn záznam sady **Recordset** typu dynamická sada nebo když sada **Recordset** nemá žádné záznamy, aktuální záznam je nedefinovaný. V tomto případě operace, které se vztahují k aktuálnímu záznamu, budou mít za následek zachytitelnou chybu.

### **aktuální transakce**

Všechny změny provedené na objektu **Recordset** poté, co jste použili poslední metodu **BeginTrans** a předtím, než jste použili metodu **Rollback** nebo **CommitTrans**.

## **DAO objekt**

Objekt, který je definován knihovnou Data Access Objects (DAO). Můžete použít DAO objekty, jako jsou objekty **Database**, **TableDef**, **Recordset** a **QueryDef**, k označení objektů, které se používají v kódu ke třídění dat a práci s nimi, jako jsou tabulky a dotazy.

**definiční dotaz**

SQL dotaz, se kterým lze vytvořit, změnit nebo odstranit tabulku, nebo vytvořit nebo odstranit index v databázi.

**zdroj dat**

Pojmenovaný zdroj Open Database Connectivity (ODBC), který upřesňuje místo, typ ovladače a další parametry potřebné pro ODBC ovladač, aby měl přístup k ODBC databázi. Zdrojem dat může být jakýkoliv zdroj databázové informace.

## **datový typ**

Charakteristiky proměnné, které určují, jaký typ dat proměnná může obsahovat. Datové typy zahrnují **Big Integer, Binary, Byte, Boolean, Char, Currency, Date, Decimal, Double, Float, GUID, Integer, Long, Long Binary (OLE objekt), Memo, Numeric, Single, String, Text, Time, TimeStamp, VarBinary, Variant** (výchozí), uživatelem definované typy (vytvořené příkazem **Type**) a objektové datové typy, které zahrnují hostitelem definované objektové datové typy a DAO objektové typy.

**databázové jádro**

Taková část databázového systému, která získává data z uživatelských a systémových databází a také do nich data ukládá.

**datum vyjádřené znaky**

Jakákoliv sekvence znaků s platným formátem, která je obklopena číselnými znaky (#). Platné formáty zahrnují datový formát určený místním nastavením pro váš kód nebo univerzální datový formát.

Například #12/31/92# je datum vyjádřené znaky, které odpovídá datu 31.prosince 1992, pokud místním nastavením pro aplikace je angličtina-americká angličtina. Data vyjádřená znaky použijte, abyste maximalizovali přenositelnost mezi národními prostředími.

**data a časy**

Data a časy se ukládají vnitřně jako různé části reálného čísla.

Hodnota nalevo od desetinné čárky odpovídá datu mezi 30.prosincem 1899 a 30.prosincem 9999 včetně.  
Záporné hodnoty odpovídají datu předcházejícímu 30. prosinci 1899.

Hodnota napravo od desetinné čárky čas mezi 0:00:00 a 23:59:59 včetně. Poledne je reprezentováno číslem .5.

**DDL (Jazyk pro definici dat)**

Jazyk používaný k popisu atributů databáze, zvláště tabulek, polí, indexů a systému uložení.

### **výchozí pracovní prostor**

Objekt **Workspace**, který DAO automaticky vytvoří, když se aplikace prvně odvolá na DAO objekt. Na tento pracovní prostor **Workspace** je odkázáno pomocí `DBEngine.Workspaces(0)` nebo prostě `Workspaces(0)`.

**odstraňovací dotaz**

Akční dotaz, který odstraní sadu řádků, která odpovídají zadaným kritériím. Odstraňovací dotaz nevrací řádky.

**navrhovací čas**

Čas, během kterého vyvíjíte ve vývojovém prostředí aplikaci přidáváním ovladacích prvků, nastavením vlastností ovládacích prvků nebo formulářů atd. Naproti tomu, za běhu programu, zacházíte s aplikací jako kdybyste byli uživatel.

### **datový typ Double**

Základní datový typ, který obsahuje čísla s plovoucí čárkou s dvojitou přesností ve formátu IEEE. Proměnná **Double** je uložena jako 64-bitové (8-bajtové) číslo v rozmezí od  $1.79769313486231E308$  do  $-4.94065645841247E-324$  pro záporné hodnoty, od  $4.94065645841247E-324$  do  $1.79769313486231E308$  pro kladné hodnoty, a 0.

**DLL (dynamicky připojovaná knihovna)**

Sada rutin, které lze zavolat z procedur a jsou nahrány a spojeny s aplikací za běhu programu.

## **dynamická sada**

Typ objektu **Recordset**, který vrací dynamickou sadu ukazatelů přímo na databázová data. Jako objekt **Recordset** typu tabulka nebo snímek, typ dynamická sada vrací data v záznamech (řádky) a v polích (sloupce). Na rozdíl od typu tabulka, objekt **Recordset** typu dynamická sada může být výsledkem dotazu, který spojí dvě nebo více tabulek. Záznamy v objektu **Recordset** typu dynamická sada lze aktualizovat, je-li vlastnost **Updatable** objektu **Recordset** nastavena na **True**, lze-li aktualizovat měněný objekt **Field** a není-li uzamčena datová stránka obsahující aktuální záznam. Datová stránka je uzamčena, když je použita metoda **Update** (když vlastnost **LockEdits** je **False**) nebo metoda **Edit** (když vlastnost **LockEdits** je **True**).

**shodné spojení**

Spojení, ve kterém jsou kombinovány záznamy ze dvou tabulek a jsou přidány k objektu **Recordset** pouze tehdy, když jsou ve spojených polích shodné hodnoty.

**výraz**

Jakákoliv kombinace operátorů, konstant, číselných hodnot, funkcí a názvů polí, ovládacích prvků a vlastností, které jsou vyhodnoceny jako jednotlivé hodnoty. Výrazy můžete použít jako nastavení pro mnoho vlastností a akčních argumentů, pro nastavení kritérií nebo definici počítaných polí v dotazech.

**externí tabulka**

Tabulka v databázi mimo otevřenou databázi Microsoft Jet. Externí tabulka může mít formu jiného souboru nebo může být tabulkou v jiné relační databázi.

**výhradní**

Typ přístupu, které chrání data v databázi sdílené po síti. Když otevřete databázi ve výhradním režimu, ostatním znemožníte otevření databáze.

**externí databáze**

Buď ODBC databáze jako je Microsoft SQL Server, která je umístěna na vzdáleném serveru, nebo jedna z externích databází jako je Paradox, dBASE, FoxPro, Microsoft Excel, Microsoft Access, Lotus 1-2-3, HTML a Text.

**pole**

Kategorie informace uložené v tabulce v databázi – sloupec dat. Prvek databázové tabulky, která obsahuje specifickou část informace, jako je příjmení.

## **datové typy pole**

Následující tabulka uvádí seznam datových typů **Field**.

<b>Datový typ DAO Field:</b>	<b>Konstanta:</b>
Big Integer	<b>dbBigInt</b>
Binary	<b>dbBinary</b>
Boolean	<b>dbBoolean</b>
Byte	<b>dbByte</b>
Char	<b>dbChar</b>
Currency	<b>dbCurrency</b>
Date/Time	<b>dbDate</b>
Decimal	<b>dbDecimal</b>
Double	<b>dbDouble</b>
Float	<b>dbFloat</b>
GUID	<b>dbGUID</b>
Integer	<b>dbInteger</b>
Long	<b>dbLong</b>
Long Binary (OLE objekt)	<b>dbLongBinary</b>
Memo	<b>dbMemo</b>
Numeric	<b>dbNumeric</b>
Single	<b>dbSingle</b>
Text	<b>dbText</b>
Time	<b>dbTime</b>
TimeStamp	<b>dbTimeStamp</b>
VarBinary	<b>dbVarBinary</b>

### **vlastnosti pole**

Atributy pole, které určují, jaký druh dat pole obsahuje. Například **Size** a **Type**.

**filtr**

Sada kritérií použitá na záznamy, aby se vytvořila podsada záznamů.

**cizí klíč**

Jedno nebo více polí, které odkazují na pole primárního klíče v jiné tabulce. Cizí klíč ukazuje, jak jsou tabulky spojeny – data v polích cizího a primárního klíče musí souhlasit. Například seznam platných čísel částí by obsahoval cizí klíč pro inventární tabulku obsahující odkazy na platná čísla částí. Používá se při vytvoření referenční integrity pro databázi.

**cizí tabulka**

Tabulka, která poskytuje cizí klíč pro jinou tabulku v databázi. Obecně se cizí tabulka používá k vytvoření nebo vynucení si referenční integrity. Cizí tabulka je obvykle na straně "mnoho" ve vztahu 1:mnoho. Příkladem cizí tabulky je tabulka objednávek zákazníků.

## **formulář**

Okno nebo dialog. Formuláře jsou kontejnery pro ovládací prvky. Formulář s rozhráním pro vícenásobné dokumenty (MDI) se také chová jako kontejner pro formuláře následníků a některé ovládací prvky.

**posun dopředu**

Přesune se směrem ke konci (**EOF**) objektu **Recordset**.

### **Recordset typu pouze vpřed**

Objekt **Recordset**, ve kterém lze záznamy prohledat od začátku do konce; umístění aktuálního záznamu nelze přesunout zpátky na první záznam. Sady záznamů typu pouze vpřed omezují dobu zpracování na vzdálených databázích. **Recordset** typu pouze vpřed můžete použít například na spojené vzdálené tabulce, abyste rychle zpracovali data v jednom průchodu, jako když sestavujete vlastní upravenou zprávu.

### **procedura Function**

Procedura, která provede zvláštní úkol v programu Visual Basic a vrátí hodnotu. Procedura **Function** začíná příkazem **Function** a končí příkazem **End Function**.

**skupina**

Kolekce uživatelských účtů v objektu **Workgroup**, určená názvem skupiny a osobním číslem (PID). Práva přiřazená skupině se používají pro všechny uživatele ve skupině.

## **datový typ GUID**

Globálně jedinečný identifikátor/Univerzálně jedinečný identifikátor. Jedinečný určující řetězec používaný s voláními vzdálených procedur. Každé rozhraní a třída objektu používá **GUID** pro identifikaci. **GUID** je 128-bitová hodnota. Například 12345678-1234-1234-1234-123456789ABC je syntakticky správný **GUID**. Řetězce **GUID** u klienta a serveru musí souhlasit, aby se klient a server propojil. Prodejci objektů mohou vyžadovat, aby Microsoft alokoval jednu nebo více sad z 256 řetězců **GUID** pro své výhradní použití. Jinak, máte-li síťovou kartu, můžete spustit nástroj nazvaný Uuidgen.exe, který nabízí sadu 256 řetězců **GUID** založených na denní době, datu a jedinečném čísle obsaženém v síťové kartě.

### **neshodný**

Stav objektu **Recordset** typu vícenásobné tabulky, který umožňuje aktualizovat všechna pole (sloupce). Například v objektu **Recordset**, vytvořeném spojením dvou tabulek do spojení 1:mnoho (jako v tabulkách **Zákazníci** a **Objednávky**), můžete aktualizovat **Objednávky.IDZákazníka** tak, aby nesouhlasil s **Zákazníci.IDZákazníka**, pokud referenční integrita umožňuje aktualizaci.

## **index**

Dynamický křížový odkaz na jednu nebo více tabulkových datových polí (sloupců), které umožňují rychlejší získání určitých záznamů z tabulky. Jak jsou záznamy přidávány, měněny nebo odstraňovány, řídicí systém databáze automaticky aktualizuje index tak, aby odrážel změny.

Při použití objektu **Recordset** typu tabulka určuje aktuální index pořadí, ve kterém jsou datové záznamy předány objektu **Recordset**. Tabulka může mít pro svá data definováno několik indexů.

DAO objekt **Index** odpovídá indexu pro objekt **TableDef**.

### **vnitřní spojení**

Spojení, ve kterém záznamy ze dvou tabulek zkombinovány a přidány do sady záznamů **Recordset** pouze tehdy, když odpovídá určené podmínce. Například shodné spojení je vnitřní spojení, ve kterém se hodnoty spojených polí musí rovnat.

### **instalovatelný ISAM**

Ovladač, který lze zadat a který umožňuje přístup k formátům externích databází jako jsou dBASE, Microsoft Excel a Paradox. ISAM je zkratka pro Indexed Sequential Access Method. Databázové jádro Microsoft Jet instaluje (načítá) tyto ISAM ovladače, pokud se na ně odkáže aplikace. Umístění těchto ovladačů lze získat v registru Microsoft Windows.

## **instance**

Jakákoliv jedna sada objektů sdílí stejnou třídu. Například vícenásobné instance třídy **Form** sdílí stejný kód a jsou načteny se stejnými ovládacími prvky, se kterými byl vytvořena třída **Form**. Za běhu programu lze jednotlivé vlastnosti ovládacích prvků u každé instance nastavit na různé hodnoty.

**datový typ Integer**

Základní datový typ, který obsahuje celá čísla. Proměnná **Integer** je uložena jako 16-bitové (2-bajtové) číslo v rozmezí od -32,768 do 32,767.

**vnitřní konstanta**

Konstanta dodaná aplikací. Například konstanty Data Access Objects (DAO) jsou uvedeny v seznamu knihovny objektů DAO a lze si je prohlédnout Prohlížečem objektů.

## **spojení**

Databázová operace, která kombinuje některé nebo všechny záznamy ze dvou nebo více tabulek jako je shodné spojení, vnější spojení nebo samo spojení. Obecně se spojení vztahuje ke spojení mezi polem v jedné tabulce a polem se stejným datovým typem v jiné tabulce. Spojení vytvoříte SQL příkazem.

Když definujete vztah mezi dvěma tabulkami, vytvoříte spojení určením primárních nebo cizích tabulkových polí. Když přidáte tabulku k dotazu, musíte vytvořit spojení mezi příslušnými poli v SQL příkazu, který definuje dotaz.

### levé vnější spojení

Vnější spojení, ve kterém jsou všechny záznamy z levé strany operace LEFT JOIN v SQL příkazu v dotazu jsou přidány k objektu **Recordset**, i když neexistují shodné hodnoty ve spojeném poli z tabulky napravo. Záznamy z tabulky napravo jsou kombinovány s těmi z tabulky nalevo, pouze když ve spojených polích existují souhlasné hodnoty. Když záznamu na levé straně chybí odpovídající hodnota, řádek hodnot **Null** se připojí napravo.

**levé spojení**

Levé vnější spojení obsahuje všechny záznamy z první (levé) ze dvou tabulek, i když v druhé (pravé) tabulce neexistují odpovídající hodnoty pro záznamy.

**spojená tabulka**

Tabulka v jiné databázi spojené k databázi Microsoft Jet. Data pro spojené tabulky zůstává v externí databázi, kde s ním mohou pracovat ostatní aplikace. (Dříve známé jako *připojené tabulky*.)

### **uzamčený**

Stav datové stránky, objektu **Recordset** nebo **Database**, který jej dělá pouze ke čtení pro všechny uživatele kromě toho, který do něj právě zadává data.

**datový typ Long Binary**

Typ pole, který může obsahovat 1.2 GB dat. Pole Long Binary mohou obsahovat jakýkoliv typ binárních dat. (V programu Microsoft Access známý také jako datový typ *OLE objekt*).

### **datový typ Long**

Základní datový typ, který obsahuje dlouhá celá čísla. Proměnná **Long** je uložena jako 32-bitové (4-bajtové) číslo v rozmezí od -2,147,483,648 do 2,147,483,647.

### **dotaz vytvářející tabulku**

Akční dotaz, který vytváří novou tabulku z objektu **Recordset** existujícího dotazu.

**tabulka se stranou "mnoho"**

Tabulka, která se neobjeví jako strana "jedna" jakéhokoliv spojení v dotazu.

**datový typ Memo**

Datový typ pole. Taková pole mohou obsahovat až 1.2 GB textových dat.

**zpráva**

Soubor informací předaný z jedné aplikace do druhé.

**metoda**

Postup podobný funkci, který funguje u zvláštních objektů.

### **databázové jádro Microsoft Jet**

Řídící systém databáze, který získává data z uživatelských a systémových databází a také do nich data ukládá. Databázové jádro Microsoft Jet lze považovat za komponentu správce dat, pomocí kterého jsou sestaveny další systémy pro přístup dat jako jsou Microsoft Access a Visual Basic.

### **víceuživatelská databáze**

Databáze, která umožňuje více než jednomu uživateli mít přístup a upravit stejnou sadu dat ve stejnou dobu. V některých případech může být další "uživatel" jinou instancí aplikace nebo jinou aplikací spuštěnou na systému, která přistupuje ke stejným datům jako některé jiné aplikace.

**normalizovat**

Minimalizovat výskyt duplicitních informací v relační databázi pomocí efektivního návrhu tabulek.

### **prázdná hodnota (null)**

Hodnota, která vyjadřuje chybějící nebo neznámá data. Prázdné hodnoty mohou být uloženy v polích, ve kterých nejsou známy informace a ve výrazech a dotazech. Ve Visual Basicu, označuje klíčové slovo **Null** Prázdnou hodnotu. Některá pole, jako například pole obsahující primární klíč, nesmí obsahovat **prázdné hodnoty**.

**prázdné pole**

Pole, které neobsahuje žádné znaky nebo hodnoty. Prázdné pole není totožné s řetězcem nulové délky ("") nebo s polem s hodnotou 0. Pole je prázdné, pokud je obsah pole neznámý. Například pole datum ukončení v tabulce úkolů je prázdné, dokud není dokončen daný úkol.

**numerický výraz**

Jakýkoliv výraz, jehož výsledkem je číslo. Může to být libovolná kombinace proměnných, konstant, funkcí a operátorů, jejichž výsledkem je číslo.

**prohlížeč objektů**

Dialogové okno, které umožňuje prohlížení obsahu knihovny objektů, kde lze získat informace o poskytovaných objektech.

### **objektový výraz**

Výraz, který definuje určitý objekt. Tento výraz může obsahovat kterýkoliv kontejner objektu. Jestliže vaše aplikace obsahuje například objekt **Application**, který obsahuje objekt **Document**, který obsahuje objekt **Text**, jsou následující výrazy platnými objektovými výrazy:

```
Application.Document.Text
```

```
Application.Text
```

```
Document.Text
```

```
Text
```

### **knihovna objektů**

Knihovna DLL s jedním nebo více typy knihovních zdrojů, které mají obvykle rozšíření .olb. Pro prohlížení obsahu knihovny objektů, kde můžete získat informace o poskytovaných objektech, použijte prohlížeč objektů.

**objektová proměnná**

Proměnná, obsahující odkaz na určitý objekt.

### **ODBC (Open Database Connectivity)**

Standardní protokol, který umožňuje aplikacím připojení k různým externím databázovým serverům nebo souborům. Ovladače ODBC použité v databázovém jádru Microsoft Jet umožňují přístup na Microsoft SQL Server a několik dalších externích databází.

Aplikační programové rozhraní (API) protokolu ODBC může být také použito pro přístup k ovladačům ODBC a k databázím, k nimž se lze připojit bez použití databázového jádra Microsoft Jet.

### **datový zdroj ODBC**

Termín, vyjadřující název databáze nebo databázového serveru, použitého jako datový zdroj. Na datové zdroje ODBC se odkazuje jejich názvy datových zdrojů (DSN). Datové zdroje mohou být vytvořeny pomocí ovládacího panelu Windows nebo pomocí metody **RegisterDatabase**.

**ovladač ODBC**

Knihovna DLL, která se používá pro připojení určitého datového zdroje ODBC k jiné (klientské) aplikaci.

**OLE**

Objektově založená technologie firmy Microsoft, která umožňuje vytváření aplikací obsahujících komponenty z různých jiných aplikací.

**objekt ActiveX**

Objekt, který je předáván jiným aplikacím nebo programovacím nástrojům pomocí automatického rozhraní (původně OLE automatizace). Například aplikace Microsoft Access může vytvářet a pracovat s aktivnímiX objekty v aplikacích Microsoft Word nebo Microsoft Excel.

**datový typ objektu OLE**

Datový typ pole, použitý pro objekty vytvořené v jiných aplikacích, které mohou být propojeny nebo vloženy do databáze Microsoft Jet. Objekt pole typu OLE můžete například použít pro uložení sady obrázků.

**vztah 1:mnoho**

Spojení mezi dvěma tabulkami ve kterém:

- Hodnota primárního klíče každého záznamu v primární tabulce souhlasí s hodnotou odpovídajícího pole nebo polí několika záznamů ve svázané tabulce.
- Hodnota primárního klíče každého záznamu ve svázané tabulce souhlasí s hodnotou odpovídajícího pole nebo polí právě jednoho záznamu v primární tabulce.

**vztah 1:1**

Spojení mezi dvěma tabulkami ve kterém:

- Hodnota primárního klíče každého záznamu v primární tabulce souhlasí s hodnotou odpovídajícího pole nebo polí právě jednoho záznamu ve svázané tabulce.
- Hodnota primárního klíče každého záznamu ve svázané tabulce souhlasí s hodnotou odpovídajícího pole nebo polí právě jednoho záznamu v primární tabulce.

### **optimistické(zamykání)**

Typ zamykání, ve kterém datová stránka obsahující jeden nebo více záznamů, včetně editovaných záznamů je nedostupná ostatním uživatelům pouze pokud je záznam aktualizován pomocí metody **Update**, ale je dostupná mezi metodami **Edit** a **Update**. Optimistické zamykání se používá při přístupu na ODBC databáze nebo pokud je vlastnost **LockEdits** objektu **Recordset** nastavena na **False**.

### **vnější spojení**

Spojení, ve kterém všechny záznamy jedné tabulky jsou přidány do objektu **Recordset**, dokonce i tehdy, pokud ve spojovaném poli druhé tabulky nejsou žádné odpovídající hodnoty. Záznamy z druhé tabulky jsou zkombinovány se záznamy první tabulky pouze pokud jsou ve spojovaných polích odpovídající hodnoty. Pokud neexistuje odpovídající záznam, jsou připojeny prázdné záznamy.

**stránka**

Část databáze, ve které jsou uloženy datové záznamy. V závislosti na velikosti záznamů, může stránka obsahovat více než jeden záznam. V databázích Microsoft Jet (.mdb) je stránka dlouhá 2048 (2K) bajtů.

## **parametr**

Prvek, obsahující hodnotu, kterou můžete měnit tak, aby byl ovlivněn výsledek dotazu. Například dotaz, který vrací data o zaměstnancích může obsahovat parametr jméno zaměstnance. Potom můžete používat jeden objekt **QueryDef** pro vyhledávání dat o zaměstnancích pomocí nastavení parametru na určité jméno před tím, než spustíte dotaz.

**parametrický dotaz**

Dotaz, který vyžaduje zadání jedné nebo více hodnot kritérií, jako například Redmond pro název města, před tím, než je spuštěn. Parametrický dotaz není v přesném slova smyslu samostatný druh dotazu, spíše zvyšuje flexibilitu ostatních dotazů.

**oddělovač**

Slouží k identifikaci částí příkazů nebo výrazů a následné kontrole těchto částí pomocí příslušných jazykových zvyklostí.

### **předávací dotaz**

Speciální SQL dotaz, který se používá pro zaslání příkazů přímo na SQL databázový server (například Microsoft SQL Server). Pomocí předávacích dotazů pracujete přímo s tabulkami na serveru místo toho, že byste k nim byli připojeni. Předávací dotazy se používají pro provádění SQL dotazů a speciálních systémových příkazů napsaných pomocí verze SQL jazyka určené pouze pro databázový server.

Předávací dotaz může a nemusí vracet záznamy. Pokud vrací záznamy, jsou vždy vráceny ve formě snímku.

## **oprávnění**

Jeden nebo více parametrů, určujících jaký druh přístupu k datům nebo objektům databáze má uživatel. Například, jestliže má uživatel právo číst data pro určitou tabulku nebo dotaz, může zde prohlížet nebo vyhledávat data, ale nemůže je upravovat.

### **stálý objekt**

Objekt uložený v databázi; například databázová tabulka nebo objekt **QueryDef**. Objekty **Recordset** typu dynamická sada nebo snímek nejsou stálé objekty, neboť jsou vytvářeny v paměti podle potřeby.

**osobní číslo (PID)**

Alfanumerický, 4-20 znaků dlouhý řetězec, ve kterém se rozlišují velká a malá písmena, který používá databázové jádro Microsoft Jet v kombinaci s názvem registrace pro identifikaci uživatele nebo skupiny v objektu **Workgroup**. Osobní číslo (PID) a název registrace zadáváte při vytváření nového uživatele nebo skupiny.

### **pesimistické(zamykání)**

Typ zamykání, ve kterém datová stránka obsahující jeden nebo více záznamů, včetně editovaných záznamů je nedostupná ostatním uživatelům při použití metody **Edit** a zůstává nedostupná, dokud nepoužijete metodu **Update**. Pesimistické zamykání je umožněno, pokud je vlastnost **LockEdits** objektu **Recordset** nastavena na **True**.

**primární klíč**

Jedno nebo více polí, jejichž hodnota nebo hodnoty jednoznačně identifikuje každý záznam v tabulce. V relaci se primární klíč používá pro odkaz na určité záznamy v jedné tabulce z jiné tabulky. Pokud je na záznam odkaz z jiné tabulky, nazývá se primární klíč cizí klíč. Primární klíč může být pouze jeden. V tabulce Zaměstnanci může být použito například čísla sociálního pojištění jako primárního klíče.

### **primární tabulka**

Strana "jedna" dvou relačních tabulek v relaci 1:mnoho s cizí tabulkou. Obecně se tabulky primárního klíče používá pro vytvoření nebo vynucení referenční integrity.

**vlastnost**

Pojmenovaný atribut určitého objektu. Vlastnosti určují charakteristiky objektu jako je velikost, barva a umístění na obrazovce nebo stav objektu jako je zpřístupněn nebo zneprístupněn.

### **pseudo index**

Dynamický křížový odkaz jednoho nebo více datových polí tabulky (sloupců), který umožňuje ODBC tabulka (tabulka serveru) bez toho, že by byl upraven jednoznačný index. Pseudo index není aktuální index tabulky na serveru, ale křížový odkaz, který používá databázové jádro Microsoft Jet při aktualizacích tabulky. Pseudo index je nutno použít pouze v případě, že ODBC tabulka nemá jednoznačný index, který umožňuje, aby byla tabulka aktualizována. Pseudo index nepotřebujete, pokud již tabulka má jednoznačný index nebo pokud nepotřebujete tabulku aktualizovat. (Tabulky nemusí mít jednoznačný index, pokud jsou vytvořeny pouze ke čtení.) V předchozí verzi se pseudo indexy nazývaly *parametry indexu*.

**dotaz**

Formalizovaný příkaz pro databázi pro získání sady záznamů nebo provedení určité akce se sadou záznamů tak, jak je to uvedeno v dotazu. Například následující SQL dotaz vrací záznamy:

```
SELECT CompanyName FROM Publishers WHERE Region = 'NY'
```

Můžete vytvářet a spouštět výběrové, akční, křížové, parametrické a speciální SQL dotazy.

**pouze ke čtení**

Typ přístupu k datům, při kterém mohou být informace vyhledávány, ale nesmí být upravovány.

**záznam**

Sada souvisejících dat o osobě, místě, události nebo jiné položce. Data v tabulce jsou ukládána v záznamech (řádcích) v databázi. Každý záznam sestává ze sady souvisejících polí (sloupců) – každé pole definuje jednu část informace pro daný záznam. Dohromady definuje záznam jednu určitou jednotku získatelné informace z databáze.

## referenční integrita

Pravidla, která jsou nastavena pro vytvoření a zachování vazeb mezi tabulkami při přidávání, změnách nebo odstraňování záznamů. Použití referenční integrity znemožňuje uživatelům přidávat záznamy do propojené tabulky, která nemá primární klíč, měnit hodnoty v primární tabulce tak, že by v připojené tabulce zůstaly nsvázané záznamy a odstraňovat záznamy z primární tabulky, pokud existují odpovídající svázané záznamy.

Pokud pro relaci vyberete volbu **dbRelationDeleteCascade** nebo **dbRelationUpdateCascade**, databázové jádro Microsoft Jet umožní provádění změn a odstraňování záznamů, ale mění a odstraňuje svázané záznamy, aby byla nadále zachována pravidla.

**relace**

Vazba vytvořená mezi společnými poli (sloupci) ve dvou tabulkách. Relace může být 1:1, mnoho:mnoho nebo 1:mnoho.

**opakovaný dotaz**

Nové spuštění dotazu tak, aby odrážel změny provedené v záznamech, vyhledal nově přidané záznamy a vyřadil odstraněné záznamy.

**vyhrazené slovo**

Slovo, které je součástí určitého jazyka, jakým je například SQL jazyk databázového jádra Microsoft Jet. Vyhrazenými slovy mohou být názvy příkazů, předdefinované funkce a datové typy, metody, operátory a objekty. Například SELECT, UPDATE, BETWEEN, SET a INSERT. Nepoužívejte vyhrazená slova pro názvy proměnných nebo objektů.

### **pravé spojení**

Pravé vnější spojení zahrnuje všechny záznamy z druhé (pravé) ze dvou tabulek, dokonce i pokud neexistují žádné odpovídající hodnoty pro záznamy v první (levé) tabulce.

LEVÉ SPOJENÍ můžete například použít u tabulek Oddělení (levá) a Zaměstnanci (pravá) pro výběr všech oddělení, včetně těch, která nemají přiřazeny žádné zaměstnance. Chcete-li vybrat všechny zaměstnance, včetně těch, kteří nejsou zařazeni do žádného oddělení, použijte PRAVÉ SPOJENÍ.

### **pravé vnější spojení**

Vnější spojení, ve kterém všechny záznamy z pravé strany operace PRAVÉHO SPOJENÍ v SQL dotazu jsou přidány do objektu **Recordset**, i v případě, že neexistují žádné odpovídající hodnoty v připojovaných polích levé tabulky. Záznamy z levé tabulky se zkombinují se záznamy z pravé tabulky pouze pokud existují odpovídající hodnoty v připojovaných polích.

**řádek**

Sada souvisejících sloupců nebo polí, kde jsou uložena data. Řádek má v prostředí databázového jádra Microsoft Jet stejný význam jako záznam. Tabulka se skládá z nula nebo více řádků dat.

**čas běhu programu**

Čas, kdy běží program. Během tohoto času pracujete s programem podle uživatelské potřeby.

**chyba při běhu programu**

Chyba, která nastane při běhu programu. Chyba při běhu programu nastane, pokud se některý příkaz pokouší provést neplatnou operaci.

## rozsah

Vlastnost proměnné nebo procedury, která určuje, jaké oblasti z kterých modulů jí rozpoznají. Existují tři úrovně rozsahu: public, module, a procedure. Proměnné, které deklaruji jako **Public** jsou dostupné kterémukoliv modulu, zatímco proměnné, které deklaruji v určitém modulu mohou být použity pouze v rámci tohoto modulu. Také proměnné, které deklaruji v proceduře **Sub** nebo **Function**, mohou být použity pouze v této dílčí proceduře.

### **pracovní skupina se zabezpečením**

Objekt **Workgroup** databáze Microsoft Jet, do kterého se uživatelé přihlašují uživatelským jménem a heslem a ve kterém je přístup do databázových objektů omezen v souladu s oprávněními stanovenými pro uživatelskou registraci a skupiny.

## **zabezpečení**

Sada funkcí, použitá pro nastavení nebo zamezení přístupu určitých uživatelů nebo skupin uživatelů k datům a objektům databáze.

### **výběrový dotaz**

Dotaz, který se pokouší získat odpověď z dat uložených v tabulkách a vrací objekt **Recordset** bez toho, že by byla data měněna. Poté, co jsou data získána do sady **Recordset**, můžete data prohlížet a měnit v základních tabulkách. Na rozdíl od toho, akční dotazy mohou provádět změny do dat, ale nevrací datové záznamy.

### **vlastní spojení**

Spojení, při kterém jsou záznamy z určité tabulky zkombinovány s jinými záznamy ze stejné tabulky, pokud existují odpovídající hodnoty v propojovaných polích. Vlastní spojení může být buď vnitřní spojení nebo vnější spojení. Vlastní spojení je užitečné, pokud pokládáte hierarchické dotazy, jako je například dotaz, ukazující všechny vedoucí pracovníky a pracovníky, kteří pod ně spadají.

**server**

Databázový řídicí systém, navržený pro sdílení dat klientských aplikací; servery a klienti jsou obvykle propojeni pomocí sítě. Databázový server obvykle obsahuje a spravuje centrální zásobu dat, kterou mohou vzdálené klientské aplikace prohledávat a manipulovat s ní.

## **relace**

Relace představuje postup operací prováděných databázovým jádrem Microsoft Jet. Relace začíná přihlášením uživatele a končí jeho odhlášením. Všechny operace provedené během relace tvoří jednu transakci a jsou závislé na přístupových právech určených přihlašovací jménem uživatele a heslem. Relace jsou implementovány jako objekt **Workspace** pomocí DAO.

### **datový typ Short**

Základní datový typ databázového jádra Microsoft Jet, který odpovídá datovému typu **Integer** definovanému ve Visual Basicu pro aplikace. Proměnná typu **Short** je uložena jako 16-bitové (2-bajtové) číslo v rozsahu hodnot od -32,768 do 32,767.

### **datový typ Single**

Základní datový typ, který ukládá čísla s pohyblivou desetinnou čárkou s jednoduchou přesností ve formátu IEEE. Proměnná typu **Single** je uložena jako 32-bitové (4-bajtové) číslo v rozsahu hodnot od  $-3.402823E38$  do  $-1.401298E-45$  pro záporné hodnoty, od  $1.401298E-45$  do  $3.402823E38$  pro kladné hodnoty a 0.

## **snímek**

Pevná kopie sady záznamů získaných z databáze a zkopírovaných do paměti. Snímek získaný ze zdroje dat připojeného k Microsoft Jet nelze aktualizovat. Objekty **Recordset** typu snímek mohou být vytvořeny ze základní tabulky, z dotazu nebo z jiné sady **Recordset**. Všechny SQL předávací dotazy vrací snímky.

**pořadí třídění**

Princip sekvenčního zpracování používaný k alfabtickému nebo numerickému třídění dat. Pořadí třídění může být buď vzestupné nebo sestupné.

**speciální SQL dotaz**

Dotaz, který může být vytvořen pouze pomocí SQL příkazu. Poddotazy a předávací dotazy, sjednocovací a definiční dotazy jsou speciální SQL dotazy.

### **SQL příkaz/řetězec**

Výraz, který definuje příkaz jazyka SQL, jakým je například SELECT, UPDATE nebo DELETE a může obsahovat klauzule typu WHERE a ORDER BY. SQL řetězce a příkazy jsou většinou použity v dotazech, v objektech **Recordset** a agregovaných funkcích, ale mohou být použity také pro vytvoření nebo modifikaci struktury databáze.

**standardní odchylka**

Parametr, který určuje způsob, jakým je pravděpodobnostní funkce vystředěna okolo svého středu a který je roven druhé odmocnině momentu ve kterém je odchylka umocněna ze svého středu. Používá se k popisu nastavení hodnot pro určení jak se hodnoty liší od aritmetického středu nebo průměru.

### **standardní pravidla pro pojmenování**

Soubor určitých pravidel pro pojmenování DAO objektů. Jména mohou být až 64 znaků dlouhá a mohou obsahovat jakékoliv kombinace písmen, číslic, mezer a speciálních znaků s výjimkou tečky (.), vykřičníku (!), obrácených uvozovek (') a hranatých závorek ([ ]). Nemůžete také použít vedoucí mezery a řídicí znaky (ASCII hodnoty od 0 do 31).

## řetězcové porovnávání

Použití operátoru pro určení zda je jeden řetězec větší nebo roven jinému řetězci. Pokud použijete v sekci Deklarace modulu **Option Compare Text**, řetězcové porovnávání nebere v úvahu malá a velká písmena. Pokud použijete **Option Compare Binary**, porovnávání bude zohledňovat malá a velká písmena. Pokud použijete **Option Compare Database**, bude porovnávací metoda nastavena podle aktuální databáze.

### **datový typ String**

Základní datový typ, který ukládá znakové informace. Proměnná typu **String** je buď pevné nebo proměnné délky a obsahuje jeden znak v jednom bajtu. Řetězce pevné délky jsou deklarovány určitou délkou a mohou obsahovat 1 až přibližně 64K ( $2^{16}$ ) znaků. Řetězce proměnné délky mohou být dlouhé až 2 miliardy ( $2^{31}$ ) znaků (pro Microsoft Windows verze 3.1 a starší přibližně 64K [ $2^{16}$ ] znaků), nebo méně v závislosti na množství paměti.

### **řetězcový výraz**

Jakýkoliv výraz, jehož výsledkem je nepřetržitá sekvence znaků. Prvky výrazu mohou obsahovat funkce, které vrací řetězec, řetězcové literály, řetězcové konstanty, řetězcové proměnné, řetězce typu **Variant** nebo funkce, které vrací řetězce typu **Variant (Var Type 8)**.

### **strukturovaný dotazovací jazyk (SQL)**

Jazyk, používaný při dotazování, aktualizaci a řízení relační databáze. Jazyk SQL může být použit pro získávání, třídění a filtrování určitých dat, získaných z databáze.

Příkazy SQL SELECT můžete použít kdekoliv, kde je akceptován název tabulky, dotazu nebo pole. Příkaz SQL můžete například použít na místě názvu tabulky při použití metody **OpenRecordset**.

**syntaktická chyba**

Chyba, která nastane pokud vložíte řádek zdrojového kódu, který Visual Basic nerozpozná.

**Poznámka** Syntaktická pravidla pro jednotlivá klíčová slova jsou definována v Syntaktické sekci příslušného hesla nápovědy. Chcete-li získat nápovědu ke klíčovému slovu ve vývojovém prostředí, vyberte klíčové slovo a stiskněte klávesu F1.

**systemový objekt**

Databázový objekt definovaný databázovým jádrem Microsoft Jet, jako je například tabulka MSysIndexes.

## **tabulka**

Základní jednotka pro ukládání dat v relační databázi. V tabulce jsou data ukládány do záznamů (řádků) a polí (sloupců) a týká se obvykle určité kategorie objektů, jakými jsou například zaměstnanci nebo součástky. Nazývá se také základní tabulka.

Sada záznamů **Recordset** typu tabulka je logickým zobrazením základní tabulky v databázi.

**dočasný disk**

Disk nebo adresář, určený systémovou proměnnou operačního systému TEMP, do kterého databázové jádro Microsoft Jet ukládá dočasné soubory a výsledná nastavení. Nazývá se také dočasná jednotka. Přestože systémová proměnná TEMP může odkazovat na RAM disk, nedoporučuje se to.

## **TEMP**

Systémová proměnná TEMP je nastavena souborem autoexec.bat při spuštění počítače. Obecně proměnná TEMP odkazuje na oblast pevného disku, kterou používají Microsoft Windows a další programy, jako je například databázové jádro Microsoft Jet, pro ukládání informací, které nemusí být uschovány poté, co vypnete počítač. Například následující řádka v souboru autoexec.bat určuje systémovou proměnnou TEMP jako dočasný adresář složku D:\TempArea:

```
SET TEMP=D:\TempArea
```

**datový typ Text**

Datový typ pole. Textová pole mohou obsahovat až 255 znaků nebo počet znaků, určený vlastností **Size** objektu **Field**, pokud je menší. Pokud je vlastnost **Size** textového pole nastavena na 0, textové pole může obsahovat až 255 znaků dat.

## **transakce**

Řada změn provedených do databázových dat a schémat. Začátek transakce označíte příkazem **BeginTrans**, transakci ukončíte pomocí příkazu **CommitTrans** a chcete-li vrátit zpět všechny změny před použitím **BeginTrans**, použijte příkaz **Rollback**.

Transakce jsou nepovinné a mohou být vnořovány až do pěti úrovní. Transakce zvyšují rychlost operací, které mění data a umožňují jednoduše vrátit zpět změny.

Transakce jsou globalní vzhledem k databázovému objektu **Workspace**.

### **sjednocovací dotaz**

Speciální SQL výběrový dotaz, který vytváří objekt **Recordset** typu snímek, obsahující data ze všech určených záznamů ve dvou nebo více tabulkách s vyloučením duplicitních záznamů. Chcete-li do výběru zahrnout duplicitní záznamy, připojte klíčové slovo ALL.

Například výsledkem sjednocovacího dotazu tabulek Zákazníci a Dodavatelé je sada záznamů **Recordset** typu snímek, obsahující všechny dodavatele, kteří jsou zároveň zákazníky.

**aktualizace**

Proces, který ukládá změny provedené v datech do záznamu. Dokud není záznam uložen, jsou změny ukládány do dočasného záznamu nazvaného vyrovnávací paměť pro kopírování. Klauzule UPDATE v příkazu SQL mění hodnoty dat v jednom nebo více záznamech (řádcích) v databázové tabulce.

**aktualizační dotaz**

Akční dotaz, který mění sadu záznamů podle vámi zadaných kritérií. Aktualizační dotaz nevrací žádné záznamy.

### **uživatelská registrace**

Registrace určená uživatelským jménem a osobním číslem(PID), která je vytvořena pro řízení přístupu do databázových objektů v objektu **Workgroup** databáze Microsoft Jet.

**ověření**

Proces kontrolující, zda vložená data splňují určité podmínky nebo omezení.

### **ověřovací vlastnosti**

Vlastnosti, používané k nastavení podmínek pro pole a záznamy tabulky. Ověřovací vlastnosti zahrnují **ValidationRule**, **Required**, a **AllowZeroLength**.

### **ověřovací pravidlo**

Pravidlo, které nastavuje omezení nebo podmínky, jak mají být ukládána data do jednoho nebo více polí. Ověřovací pravidla je možno nastavit pro objekt **Field** nebo **TableDef**. Ověřovací pravidlo je použito ve chvíli, kdy aktualizujete záznam, který obsahuje pole, vyžadující ověření. Pokud je pravidlo porušeno, objeví se chybové hlášení.

## **proměnná**

Pojmenované místo v paměti, obsahující data, která mohou být modifikována při běhu programu. Každá proměnná má název, který ji jednoznačně identifikuje v rámci úrovně oboru platnosti. Datový typ může a nemusí být určen.

Názvy proměnných:

- Musí začínat písmenem.
- Nesmí obsahovat vloženou tečku nebo znak deklarující typ.
- Musí být jednoznačné v rámci stejného oboru platnosti.
- Nesmí být delší než 255 znaků.

### **datový typ Variant**

Speciální datový typ, který může obsahovat čísla, řetězce nebo datумы stejně jako speciální hodnoty **Empty** a **Null**. Funkce **VarType** určuje, jak se s daty v proměnné typu **Variant** zachází. Všechny proměnné se stávají proměnnou typu variant, pokud jim není explicitně přiřazen jiný typ.

**výraz typu variant**

Jakýkoliv výraz, jehož výsledkem je číslo, řetězec nebo datum nebo speciální hodnoty **Empty** a **Null**.

## **zástupné znaky**

Hvězdička (\*), otazník (?), označení čísla(#), vykřičník (!), pomlčka (-) a hranaté závorky ([ ]) jsou zástupné znaky. Tyto znaky můžete použít v dotazech a výrazech tak, aby obsahovaly všechny záznamy, názvy souborů nebo další položky, které začínají určitými znaky nebo splňují určitý vzorec. Zástupné znaky můžete také použít pro další upřesnění vyhledávání při použití příkazu SQL.

<b>Znak</b>	<b>Příklad</b>	<b>Použití</b>
*	<i>wh*</i> vyhledá what, white, a why <i>*at</i> vyhledá cat, bat, a what	Nahrazuje libovolný počet znaků a může být použit kdekoliv ve znakovém řetězci.
?	<i>b?ll</i> vyhledá ball, bell, a bill	Nahrazuje jakýkoliv jednotlivý znak.
#	<i>1#3</i> vyhledá 103, 113, 123	Nahrazuje jakoukoliv jednotlivou číslici.
[ ]	<i>b[ae]ll</i> vyhledá ball a bell ale ne bill	Nahrazuje jakýkoliv jednotlivý znak uzavřený v hranatých závorkách.
!	<i>b![ae]ll</i> vyhledá bill a bull ale ne bell nebo ball	Nahrazuje jakýkoliv jednotlivý znak, který není uveden v seznamu.
-	<i>b[a-c]d</i> vyhledá bad, bbd, a bcd	Nahrazuje kterýkoliv jednotlivý znak z uvedeného rozsahu.

**Poznámka** Zástupné znaky \* (hvězdička), ? (otazník), # (označení čísla) a [ (levá hranatá závorka) mohou nahrazovat samy sebe pouze pokud jsou uzavřeny v hranatých závorkách.

**pracovní skupina**

Skupina uživatelů ve víceuživatelském prostředí, kteří sdílejí data ve stejném informačním souboru pracovní skupiny (obvykle soubor System.mdw ).

### **správce pracovní skupiny**

Člen skupiny Správci v objektu **Workgroup** databáze Microsoft Jet. Správce pracovní skupiny má vždy úplná práva pro všechny databázové objekty vytvořené v pracovní skupině **Workgroup**.

### **řetězec nulové délky**

Řetězec, který neobsahuje žádné znaky (""). Funkce **Len** pro řetězec nulové délky vrátí 0.

**asynchronní dotaz**

Typ dotazu, ve kterém dotazy SQL odpoví okamžitě, i kdyby výsledky byli stále nedokončené. Toto umožní aplikaci pokračovat ve zpracování dalších procesů, přestože dotaz není dokončen.

**datový zdroj ODBC připojený k Microsoft Jet**

Datový zdroj ODBC, který je dostupný pomocí Data Access Objects (DAO) a databazového jádra Microsoft Jet.

**datový zdroj ODBCDirect**

Datový zdroj, který je zpřístupněn použitím funkcí DAO ODBCDirect, bez použití databázového jádra Microsoft Jet.

**ODBCDirect**

Technologie, která vám umožní zpřístupnit ODBC data přímo použitím funkcí DAO, bez použití databázového jádra Microsoft Jet.

**správce ovladačů ODBC**

Funkce, která vytváří rozhraní mezi hostitelským jazykem a určitým ovladačem klientského datového zdroje.

**poddotaz**

Poddotaz je příkaz SELECT, umístěný uvnitř jiného příkazu SELECT, SELECT...INTO, INSERT...INTO, DELETE, nebo UPDATE nebo uvnitř jiného poddotazu.

**cílové pole**

Oblast dotazu, která bude obsahovat výsledky vašeho dotazu.

**omezení**

Vymezení možných hodnot, které mohou uživatelé vložit do pole.

**rozptyl**

Druhá mocnina směrodatné odchytky. Je to míra množství, kterou se všechny hodnoty ve skupině liší od průměrné hodnoty skupiny.

**řídící znak**

Každá kombinace kláves, která začíná klávesou CTRL nebo ESC.

## **Data Access Objects (DAO)**

Programové rozhraní pro zpřístupnění a manipulaci s objekty databáze.

**serverový kurzor**

Kurzory umístěné na serveru, na rozdíl od kurzorů umístěných na klientském počítači. Zatímco klientské kurzory kopírují klíčová nastavení na pracovní stanici, serverové kurzory používají zdroje databázového serveru pro klíčová nastavení kurzoru.

**pole**

Proměnná, která obsahuje konečný počet prvků, jež mají společné jméno a typ dat. Každý prvek pole je identifikován jedinečným číslem indexu. Provedené změny na jednom prvku v poli neovlivňují jiné prvky.

## **konstanta**

Pojmenovaná položka, která má stálou hodnotu při celém běhu programu na rozdíl od proměnné, jejíž hodnota se může během provádění programu měnit. Každá hostitelská aplikace může definovat svou vlastní sadu konstant. Další konstanty mohou být definovány příkazem **Const**. Konstanty mohou být použity kdekoliv ve zdrojovém kódu namísto skutečných hodnot. Konstanta může být řetězcem nebo číslem, jinou konstantou nebo jakoukoliv kombinací, která zahrnuje aritmetické nebo logické operátory kromě příkazu **Is** a umocňování. Například:

```
Const strdb = "c:\Program Files\Microsoft Office\Access\Samples\Northwind.mdb"
```

## **databáze Microsoft Jet**

Databáze vytvořená databázovým jádrem Microsoft Jet. Přípona názvu souboru databáze Microsoft Jet je .mdb.

**klastrovaný index**

Fyzické uspořádání řadků je stejné jako indexované uspořádání.

### **dávková aktualizace**

Model kurzoru pro klienty, kteří pracují s kurzory, ale nepotřebují provádět aktualizaci po každém řádku. Klient aktualizuje více řádků, které jsou ukládány lokálně a aktualizace probíhá v dávkách. Tento model kurzoru také umožňuje klientovi přerušit spojení se serverem a obnovit ho se stejným nebo dokonce jiným serverem.

K použití dávkové aktualizace v DAO 3.5 musíte využít pracovní prostor ODBCDirect, ve chvíli, kdy je spojení otevřeno nastavit vlastnost **DefaultCursorDriver** na dbUseClientBatchCursor a **Recordset** musí být otevřena metodou **OpenRecordset** s parametrem typ uzamykání nastaveným na dbOptimisticBatch.

**datové zřetězení**

Proces, který užívá dávkové aktualizace optimistického typu k uložení dat ze serveru na lokální kurzor, ukončí spojení se serverem, otevře nové spojení k jinému serveru se stejným informačním schématem a potom uloží data na novém serveru.

**dynamický kurzor**

Dynamická sada řádků, kterou můžete použít k připojení, změně nebo vymazání řádků ze základní tabulky nebo tabulek databáze. Dynamický kurzor obsahuje sloupce z jedné nebo více tabulek v databázi. Sada řádků se může měnit.

**cursor množiny klíčů**

Sada řádků, kterou můžete použít k připojení, změně nebo vymazání řádků ze základní tabulky nebo tabulek databáze. Pohyb uvnitř sady není omezen. Kurzor množiny klíčů obsahuje sloupce z jedné nebo více tabulek v databázi. Sada řádků se nemůže měnit.

### **statický kurzor**

Výsledná sada, kde přítomnost, pořadí, a hodnoty jsou vždy neměnné, dokud je kurzor otevřený. Jestliže jiní uživatelé aktualizují, odstraní nebo vloží nějaké řádky, kurzor nezobrazí tyto změny dokud neskončíte a neotevřete ho znovu.

## **HTTP**

Hypertextový převodní protokol (Hypertext Transfer Protocol). Skrytý protokol Internetu, který doručí informace prostřednictvím World Wide Web. Protokol umožňuje uživateli použít klientský program pro vložení URL (nebo klepnout na hyperlink) a získat text, grafiku, zvuk a jiné digitální informace z Web serveru. Adresy URL souborů na Web serverech začínají **http://**.

## **FTP**

Protokol pro převod souborů (File Transfer Protocol). Protokol, který umožňuje uživateli přenos souborů z jednoho místa do jiného přes Internet. Adresy URL souborů na FTP serverech začínají **ftp://**.

**intranet**

Síť uvnitř organizace, která používá technologii Internetu (například HTTP nebo FTP protokoly). Intranet lze použít k orientaci mezi předměty, dokumenty, Web stránkami a jinými cíli pomocí technologie odkazů.

**Internet**

Celosvětová síť tisíců menších počítačových sítí a milionů počítačů využívaných pro komerční, vzdělávací, vládní a soukromé účely. Internet je jakýmsi elektronickým městem se zdánlivými knihovnami, sklady, kanceláři, uměleckými galeriemi atd.

**pracovní prostor ODBCDirect**

Pracovní prostor, který používá ODBCDirect k přímému přístupu k zdroji dat ODBC, bez použití databázového jádra Microsoft Jet.

**pracovní prostor Microsoft Jet**

Pracovní prostor, který používá databázové jádro Microsoft Jet pro přístup ke zdroji dat. Zdroj dat může být databázový soubor Microsoft Jet (.mdb), ODBC databáze - například databáze systému Paradox, nebo ISAM.

**knihovna kurzorů client batch**

Knihovna, která umožní kurzorům na straně klienta přístup k aplikacím databáze ODBCDirect. Tato knihovna pracuje se všemi čtyřmi typy kurzorů (množina klíčů, statický, dynamický, a forward-only) a poskytuje řadu jiných funkcí včetně schopnosti ukončit spojení a provést dávkovou aktualizaci optimistického typu.

**datový typ odkaz**

Datový typ pole, který se používá pro pole ukládající odkazy. Tato pole mohou obsahovat text nebo kombinace textu a čísel uložených jako text, a ten se používá jako adresa odkazu. Adresa může mít až tři části (každá část může obsahovat až 2048 znaků) a užívá syntaxe:

*displaytext#address#subaddress*

### **složený dotaz**

Dotaz, který je složený alespoň z jednoho akčního dotazu (dotaz, který kopíruje nebo mění data) a alespoň jednoho výběrového dotazu (dotaz, který vrátí **Recordset** bez změny dat). V DAO se složený dotaz vytvoří vložením dvou nebo více příkazů SQL (oddělených středníkem) do vlastnosti **SQL** objektu **QueryDef**.

**datový typ Big Integer**

Datový typ pro ukládání čísel se znaménkem s přesností 19 (se znaménkem) nebo 20 (bez znaménka), stupeň 0 (se znaménkem:  $-263 \leq n \leq 263-1$ ; bez znaménka:  $0 \leq n \leq 264-1$ ).

**datový typ Binary**

Datový typ pro ukládání binárních údajů pevné délky. Maximální délka je 255 bajtů.

### **datový typ Char**

Datový typ pro ukládání znakového řetězce pevné délky. Délka je určena vlastností **Size**.

**datový typ Decimal**

Datový typ pro ukládání číselné hodnoty se znaménkem s přesností  $p$  a stupněm  $s$  ( $1 \leq p \leq 15$ ;  $0 \leq s \leq p$ ).

**datový typ Float**

Datový typ pro ukládání přibližné číselné hodnoty se znaménkem s přesností mantisy 15 (nula nebo absolutní hodnota -10308 do 10308).

**datový typ Numeric**

Datový typ pro ukládání přesné číselné hodnoty se znaménkem, s přesností  $p$  a stupněm  $s$  ( $1 \leq p \leq 15$ ;  $0 \leq s \leq p$ ).

**datový typ Time**

Datový typ pro ukládání hodnoty času. Hodnota je závislá na nastavení času zdroje dat.

**datový typ VarBinary**

Datový typ, který ukládá binární data proměnné délky. Maximální délka je 255 bajtů.

**datový typ TimeStamp**

Datový typ, který ukládá systémový čas. Hodnota je závislá na nastavení času zdroje dat.

**procedurální dotaz**

Príkaz SQL, který provede uloženou proceduru.

**kolize**

Konflikt, který nastane při aktualizaci dávky.

Pokud klient čte data ze serveru a pak se pokusí tato data změnit v dávkové aktualizaci. Pokud před skutečným provedením dávky jiný klient změní originální data, dojde ke kolizi, neboť první klient se snaží změnit data, která na serveru již neexistují.

**kurzor pouze vpřed**

Výsledná sada, kde je možné přesouvat kurzor pouze vpřed o určitý počet záznamů, nebo na poslední záznam. Kurzor nelze posouvat směrem vzad na začátek sady ani na předchozí záznamy. Přítomnost, pořadí a hodnoty jsou vždy neměnné, dokud je kurzor otevřený. Jestliže jiní uživatelé aktualizují, odstraní nebo vloží nějaké řádky, kurzor nezobrazí tyto změny dokud neskončíte a neotevřete ho znovu.



**tabulka konfliktů**

Tabulka, která se vytvoří, když dojde k synchronizačnímu konfliktu mezi dvěma replikami. Tabulka konfliktů se objeví pouze v replice s odmítnutými změnami.

## **replikace databáze**

Proces množení databáze, který umožní, že dvě nebo více kopií (replik) jedné databáze bude synchronizováno. Změny dat v replikované tabulce v jedné replice jsou posílány a zanášeny do ostatních replik v rámci sady replik. Změny vzhledu databáze provedené v řídicí replice jsou posílány a zanášeny do všech replik v rámci sady.

**výchozí synchronizační partner**

Synchronizátor, který se používá k automatické synchronizaci repliky s ostatními replikami v rámci sady. Když se vytvoří replika, synchronizátor spravující zdrojovou repliku se stane výchozím synchronizačním partnerem pro novou repliku.

### **výchozí synchronizační rozvrh**

Šablona, používaná synchronizátorem k definici rozvrhu mezi naposledy spravovanou replikou a jejím výchozím synchronizačním partnerem, když nemá naposledy spravovaná replika žádný synchronizační rozvrh. Jakmile je rozvrh definován, lze ho měnit.

### **řídící replika**

Databáze, ke které byly přidány systémové tabulky, systémová pole a replikační vlastnosti. Řídící replika je první replikou v sadě replik. Změny ve struktuře databáze lze provádět pouze prostřednictvím řídícího repliky. Repliky ve stejné sadě se mohou pořadě stávat řídící replikou, ale současně může existovat pouze jedna řídící replika v každé sadě replik.

**všeobecně jednoznačný identifikátor (GUID)**

Pole o velikosti 16 bajtů, používané pro zřízení jednoznačného identifikátoru pro repliky. Tyto identifikátory se rozsáhle používají k identifikaci replik, sad replik, tabulek, záznamů a jiných objektů.

**Upozornění** V aplikaci Microsoft Access se vodítka GUID označují jako replikační ID.

**lokální objekt**

Tabulka, dotaz, formulář, sestava, makro nebo modul, které zůstávají v replice, ve které byly vytvořeny. Ani objekt, ani změny v objektu, nejsou rozšiřovány do ostatních členů sady replik.

**spravovaná replika**

Replika s přiřazeným synchronizátorem. Synchronizátor dodává a přijímá změny do a z ostatních replik v rámci sady.

**Upozornění** Není-li přiřazen k replice žádný synchronizátor, je možno přesto provádět synchronizaci v aplikačním kódu nebo použitím jiného synchronizátoru. Z takovéto repliky však nelze nastavit synchronizační rozvrh, dokud k ní nebude přiřazen synchronizátor.

### **replikační správce Microsoft**

Aplikace, která poskytuje grafické uživatelské rozhraní pro správu umístění, synchronizačního rozvrhu a vztahu jedné nebo více replik s ostatními replikami v rámci sady. Replikační správce řídí nastavení synchronizátoru, která ovládají replikaci, zatímco synchronizátor provádí aktuální synchronizační činnosti. Replikační správce je součástí softwarového kompletu Microsoft Office 97, Developer Edition.

**replikace objektu**

Proces množení objektu (tabulky, dotazu, sestavy, formuláře, makra nebo modulu) takový, že dvě nebo více kopií stejného objektu je synchronizováno. Změny provedené ve struktuře objektu v řídicí replice se odesílají a používají se ve všech replikách v rámci sady.

**synchronizace na požádání**

Výměna dat mezi dvěma replikami, ke které dojde na základě výslovné žádosti uživatele.

**replika**

Kopie databáze, včetně jejích tabulek, dotazů, formulářů, sestav, maker a modulů. Replika je členem sady replik a může být synchronizována s ostatními replikami v rámci sady. Změny dat v replikované tabulce jedné repliky jsou odesílány a používány v ostatních členech v rámci sady replik.

**Upozornění** Řídící replika se také považuje za repliku.

**člen sady replik**

Replika, která patří do sady replik. Členové sady replik sdílejí stejný návrh databáze a mohou být jeden s druhým synchronizováni.

**sada replik**

Repliky, které sdílejí stejný návrh databáze a jedinečný identifikátor sady replik. Mezi členy sady replik dochází k synchronizaci.

**topologie sady replik**

Pořadí, ve kterém jsou změny předávány z repliky do repliky. Topologie je důležitá, protože ovlivňuje, jak rychle se změny v jiné replice objeví v dané replice.

### **replikovaná databáze**

Databáze, ke které byly přidány další tabulky, pole a vlastnosti za účelem záznamu informací o změnách dat a návrhů replikovaných objektů v databázi.

**replikovat**

Vytvoření repliky nebo kopie něčeho – například databáze.

**replikovaný objekt**

Tabulka, dotaz, formulář, sestava, makro nebo modul, který se nachází ve všech replikách v rámci sady replik. Replikovaný objekt je možno změnit pouze v řídicí replice a tyto změny jsou během synchronizace rozšířeny do ostatních replik v rámci sady replik.

## **replikace**

Proces vytvoření a synchronizace replik v oddělených místech. Replikace přidá do databáze systémová pole, systémové tabulky a replikační vlastnosti, takže změny provedené v datech nebo v návrhu databáze se odesílají a používají ve všech členech sady replik.

**doba zdržení**

Počet dní, po které bude systém zdržovat pro repliku určité systémové replikační informace. Jestliže doba zdržení vyprší předtím, než je replika synchronizována, dojde k selhání synchronizace.

**rozvrh**

Časové okamžiky, kdy synchronizátor bude inicializovat synchronizaci s ostatními replikami. Minimální časový interval mezi plánovanými synchronizacemi je 15 minut. Je možno nastavit různé časové rozvrhy mezi dvěma libovolnými synchronizátory.

**plánovaná synchronizace**

Výměna dat mezi dvěma replikami, ke které dochází v předem určené době.

**synchronizace**

Proces aktualizace dvou replik, při kterém jsou vyměněny všechny aktualizované záznamy a objekty. Výměna dat mezi dvěma replikami může být jednosměrná nebo obousměrná a může být ošetřena synchronizátorem.

**synchronizační konflikt**

Situace, při které uživatelé dvou replik změnili stejný záznam. Jeden z aktualizovaných záznamů je vybrán jako "vítěz" a záznam v "prohrávající" replice je aktualizován tak, aby obsahoval vítězná data. Druhý aktualizovaný záznam je umístěn do tabulky konfliktů v replice, která prohrála.

**synchronizační chyba**

Problém, ke kterému dojde během synchronizace, a který brání tomu, aby změny záznamu byly úspěšně dokončeny. To je mnohem vážnější problém než konflikt a měl by být vyřešen co možno nejdříve.

**synchronizátor**

Program, který sleduje změny v replice a ošetřuje aktuální výměnu dat mezi replikami. Synchronizátor pracuje v pozadí a zobrazuje se na hlavním panelu. Synchronizátory sady replik je možno vidět v synchronizačním okně v replikačním správci.

**Upozornění** Předcházející verze replikačního správce používala pro synchronizátory pojem transportéry.

**základní počítač**

Počítač použitý v topologii typu hvězda. Replika se zkopíruje z centrálního serveru a umístí se na základním počítači. Tato replika může být použita pro vytvoření dalších replik pro ostatní počítače. Viz *také* hub server, hvězdicová topologie.

**složka odkládací schránky**

Sdílené místo v síti kam může synchronizátor uložit nebo odkud si může vyzvednout změny návrhu a dat.

**hub server**

Počítač použitý ve hvězdicové topologii. Replika sídlící na tomto počítači může být použita jako synchronizační partner s ostatními replikami. Viz *také* základní počítač, satelitní počítač nebo hvězdicová topologie.

**satelitní počítač**

Počítač ve hvězdicové topologii na kterém se nachází replika. Tato replika je dle výchozího nastavení zkopírována ze základního počítače a zesynchronizována s replikou na centrálním serveru. Viz *také* základní počítač, hub server, hvězdicová topologie.

**sdílená síťová složka**

Viz složka odkládací schránky.

### **hvězdicová topologie**

Typ organizační mapy pro počítače spojené do sítě. V případě replikace zahrnuje hub server, základní počítač a rozložení členů sady replik na ostatních satelitních počítačích. Topologie je důležitá pro stanovení synchronizačního rozvrhu. *Viz také* základní počítač, hub server, satelitní počítač.

### **synchronizační okno**

Oblast replikačního správce, ve které se zobrazují synchronizátory. Je možné přesouvat synchronizátory a zmenšit nebo zvětšit zobrazení, aby bylo možné vidět větší nebo menší počet synchronizátorů. Také je možno klepnout na řádek mezi dvěma synchronizátory za účelem úpravy jejich synchronizačního rozvrhu. Vysvětlení různých ikon v synchronizačním okně lze získat klepnutím na heslo **Legenda** v nabídce **Zobrazit**.

### **částečná replika**

Databáze, která obsahuje pouze dílčí sadu záznamů úplné repliky. V částečné replice je možno nastavovat filtry a určovat relace, které definují, která dílčí sada záznamů v úplné replice by měla být přítomna v databázi.

**přímá synchronizace**

Typ synchronizace ke které dojde, když během synchronizačního procesu mohou být dvě repliky otevřeny současně.

**nepřímá synchronizace**

Typ synchronizace, ke které dojde, když jeden synchronizátor odloží změny do složky odkládací schránky, aby si je jiný synchronizátor vyzvedl a použil do repliky, kterou spravuje.

### **nespravovaná replika**

Replika, která není spravovaná synchronizátorem. Nespravovaná replika se zobrazuje v synchronizačním okně se zvláštní ikonou. Z nespravované repliky není možno nastavit synchronizační rozvrh, ale přesto takovéto repliky mohou být synchronizovány jiným synchronizátorem nebo prostřednictvím kódu DAO.



