INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# DUM 11 téma: Databázové jazyky a servery

**ze sady:** 3 **tematický okruh sady:** III. Databáze
**ze šablony:** 7 – Kancelářský software **určeno pro:** 4. ročník
**vzdělávací obor:** 18-20-M/01 Informační technologie
**vzdělávací oblast:** odborné vzdělávání
**metodický list/anotace:** viz VY_32_INOVACE_07311ml.pdf

# Komunikace s databází

Spolu s přenesením samotných dat do prostředků informačních technologií se i změnil způsob přístupu k nim. U papírových kartoték a „databází“ přistupoval k datům přímo uživatel a pracoval s jejich hmotnou a jedinou podobou. V pozdějších fázích pak mezi koncového uživatele a kartotéční lístky vstupuje specializovaný zaměstnanec – například knihovník, který tvoří prostředníka a jednak alespoň částečně datový sklad ochrání před záměrným nebo nedbalostním poškozením a také přebírá od uživatele nutnost znát vnitřní struktury a postupy práce datového skladu.

## *SŘBD*

U počítačových databází tento úkol přebíhá specializovaný software, obecně označovaný jako systém řízení báze dat – SŘBD. Ten zajišťuje několik vzájemně provázaný, však pro uživatele oddělených činností. Především řídí ukládání dat na datová média databázového stroje (můžeme říci serveru). K tomu může ale nemusí využívat prostředky hostitelského operačního systému a souborových systémů. Při jejich využití je programování takového systému sice jednodušší, část výkonu se však ztrácí na nutných voláních služeb hostitelského operačního systému, ten pak ovlivňuje i celkovou spolehlivost a rychlost celého SŘBD. Nikoliv nepodstatná je i otázka bezpečnosti, když jakkoliv kvalitně nastavené zabezpečení pomocí samotných databázových nástrojů je zcela neefektivní při získání přístupu k hostitelskému OS a přímo datovým souborů.

Druhým přístupem je realizace například vlastního – databázově optimalizovaného – souborového systému, což ale klade vyšší nároky na autory SŘBD a samozřejmě zvyšuje jeho cenu. Zcela samostatně pak stojí otázka zda lze daný SŘBD provozovat jako jednu z mnoha aplikací na hostitelském stroji, nebo vyžaduje pro svůj běh některou formu výhradního přístupu. I zde je první varianta jednodušší, levnější, ovšem na úkor výkonnosti a druhá naopak.

## *Komunikace uživatele a SŘBD*

Pokud přijmeme myšlenku oddělení uživatele od samotných dat, musí existovat způsob, jak koncový uživatel – byť zprostředkovaně – k datům přístup získá, jinak by ztratila smysl existence samotného SŘBD. Tato forma je v informatice nazývána serverovou komunikací s různou mírou abstrakce (či zapouzdření). Rozlišujeme pak různé typy přístupu, v závislosti na poloze samotných dat, jejich správy, aplikace která je využívá a koncového uživatele:

- terminálový přístup – uživatel na jedné straně, prostřednictvím komunikační linky připojen na server, kde je spuštěna uživatelská aplikace, která dále komunikační linkou spolupracuje s databázovým serverem. Ke stanici uživatele přichází textový či grafický výstup uživatelské aplikace.
- klient-server – uživatel má na svém počítači spuštěnu klientskou část aplikace, která zajišťuje textovou či grafickou prezentaci dat, vstup a výstup a základní kontroly či formátování. Komunikační linkou na server putují předzpracovaná data a serverová část aplikace komunikuje s databází.
- stand-alone aplikace – uživatel má na svém počítači kompletní aplikaci, která komunikuje s databázovým serverem.

Všechny tyto varianty mohou být realizovány s databázovým serverem na stejném stroji jako je samotná uživatelská aplikace, nebo odděleným. Historicky je terminálový přístup nejstarší, přesto se k němu v dnešní době vývojově opět vracíme, prostřednictvím tzv. webových aplikací. Nejčastěji používanou variantou jsou aplikace klient-server.

Pro komunikaci uživatelské aplikace a databázového serveru bylo vyvinuto mnoho protokolů a postupů – mnohdy přímo autory konkrétních databázových systémů. Výběr použité značky/autora databázového systému pak jednoznačně určil i způsob komunikace s ním a často i rozhodl o dodavateli konkrétní uživatelské aplikace – například proto, že použité protokoly nebyly zveřejňovány a pro daný typ databáze mohl aplikace vyvíjet pouze její autor.

Tento stav rozhodně nepřispíval konkurenčnímu prostředí v oboru vývoje aplikací a v informačních technologiích není obvyklý. Již v 70. letech byl tedy vytvořen obecný univerzální jazyk pro komunikaci s databázemi, který měl být především platformou nezávislý a odstranil pevnou vazbu mezi autorem databázového systému a uživatelských aplikací k nim. Je samozřejmé, že takový přístup nevyhovoval některým zaběhnutým autorům (a některým nevyhovuje dodnes), obrovská poptávka po aplikacích, které by šlo snadno vyvíjet nezávisle na zvoleném SŘBD však nakonec zafungovala.

Byl vytvořen SQL – Structured Query Language (strukturovaný dotazovací jazyk), někdy též (chybně) uváděn jako Simple Query Language (jazyk jednoduchých dotazů). Jeho nosnou myšlenkou je blízkost přirozenému jazyku. Základní jednotkou jazyka je „dotaz“ –

query. Takovým dotazem mohou být ale i příkazy pro databázový server, což ale nic nemění na uvedeném označení.

K příslušnému dotazu patří od databázového serveru „odpověď“ – response. Na každý dotaz musí server odpovědět, i kdyby se jednalo jen o odpověď „OK, provedl jsem“, případně „nerozumím“. Dotazy pro server jsou pokládány prostřednictvím formalizovaného jazyka, velmi blízkého angličtině. Většinu dotazů tak lze – s větší či menší tolerancí – přečíst jako smysluplnou anglickou větu. Například „SELECT car FROM garage WHERE color = „black““ – tedy „Vyber auto s garáže, které má černou barvu“.

Kromě samotné odpovědi zasílá server většinou i počet řádků, které byly daným dotazem „ovlivněny“ – buď nalezeny, nebo u změnových dotazů změněny a podobně. Klientská aplikace tak již při přebírání odpovědi ví, kolik prostoru pro ni musí vytvořit, případně zda čekat delší dobu na zbytek. Většina databázových serverů informuje i o délce provádění dotazu. Ve školní praxi se budeme setkávat s dobou desetin ms, u rozsáhlých databází ale mohou složité dotazy trvat i desetiny nebo jednotky sekund. Pokud se podobných dotazů opakuje i několik za sebou, je zpoždění patrné i pro člověka (ovšem stále menší, než kdyby se jednalo o starší typ databáze) a existuje dokonce samostatný vědní obor, který se zabývá optimalizací SQL dotazů pro jejich rychlejší vyřízení. Na stejnou věc se totiž lze mnohdy „zeptat“ několika způsoby, tato problematika je ale mimo rozsah učiva střední školy.

## *Složky SQL*

Dotazovací jazyk SQL je složen z několika částí, rozlišených podle svého účelu v komplexu činností komunikace s databází. Jedná se o:

- DML – Data Manipulation Language – jazyk pro manipulaci s daty, dotazování, vkládání a modifikace dat
- DDL – Data Definition Language – jazyk pro vytváření databázové struktury a její změny
- DCL – Data Control Language – jazyk pro řízení přístupu, přidělování a odebírání oprávnění

Každý příkaz SQL lze zařadit do některé z uvedených skupin. Zároveň je možné příkazy v rámci jednoho dotazu spojovat a vnořovat. Výsledek jednoho může být podmínkou druhého.

Logika zpracování se pak přenáší na databázový server, který klientovi vrátí již agregovanou formu výsledku.

## Databázové servery

Všechny dnes běžně používané databázové systémy jsou buď na SQL založeny, nebo minimálně dokážu krom svého vnitřního jazyka reagovat i na SQL. Jeho standardizace v některých drobnostech není zcela dokonalá a několik autorů SŘBD chápe jeho implementaci svým vlastním způsobem. Na takové odlišnosti bude čtenář v dalších kapitolách příslušně upozorněn. Jak již bylo řečeno – uživatel s aplikací kompatibilní s SQL si v zásadě může zvolit jakýkoliv SQL databázový server, který k ní použije.

V praxi se setkáme se systémy:

- Microsoft Access – jednoduchá klientská databáze typu stand-alone, kdy uživatel má na svém počítači kromě aplikace i celé vývojové prostředí, vlastní datové soubory a uživatelské rozhraní.
- Oracle – mohutný databázový systém – první který používal SQL, používá se na samostatných databázových serverech pro nejrozsáhlejší a nejvýkonnejší databázové úlohy jaké se v praxi vyskytují
- MS SQL – databázový systém od firmy Microsoft, používaná v komerční sféře pro méně náročné projekty, navenek komunikuje svou mírně upravenou variantou SQL
- MySQL – open source databázový systém pro většinu platforem, provozovaný nejčastěji v rámci GNU/Linux. Rozděluje samostatnou úroveň databázového komunikačního serveru a úroveň datového média pro fyzické uložení dat.
- PostgreSQL – také open-source databázový systém monolitického typu, blízký systému Oracle. Typická je jeho konzervativnost a snaha o maximální dodržování stanradů.

## Škálovatelnost

Při výběru databázového serveru je podstatná i možnost škálovatelnosti – tedy postupného rozšiřování výkonu databázového stroje přidáváním HW komponent a jednotek.

Představa databázového serveru jako jednoho fyzického počítače pak přestává korespondovat s realitou, kdy i skupina počítačů tvoří vnější rozhraní pro dotazy a další desítky a stovky kusů HW zajišťují provádění dotazů a správu fyzických úložišť. Z vnějšího pohledu se ale celý systém chová jako jediný databázový server a koncový uživatel toto rozdělení pozná pouze rychlostí odezvy (a samozřejmě náklady na jeho pořízení a provoz).

Drtivá většina zmíněných SŘBD škálovatelnost nativně podporuje.

# Úkoly pro samostatnou práci

- Zařaďte databázi MySQL do konceptu LAMP

- Najděte, jaký databázový server se nativně používá pro aplikace v Pascal/Delphi

Zdroje:

- Archiv autora