

Databázový systém MySQL

1. MYSQL.....	2
1.1 SQL.....	2
1.1.1 SQL základní příkazy.....	2
<i>Příkazy pro manipulaci s daty.....</i>	<i>2</i>
<i>Příkazy pro definici dat.....</i>	<i>2</i>
<i>Příkazy pro řízení dat.....</i>	<i>2</i>
1.2 SŘBD.....	2
2. RELAČNÍ DATABÁZE.....	2
2.1 RELAČNÍ TABULKA.....	2
2.2 ZÁKLADNÍ OPERACE RELAČNÍ ALGEBRY.....	3
1.1.2 Projekce.....	3
1.1.3 Selekce.....	3
1.1.4 Spojení.....	3
<i>spojení na rovnost (vnitřní).....</i>	<i>3</i>
<i>Theta spojení (vnitřní).....</i>	<i>3</i>
<i>Vnější spojení (outer join).....</i>	<i>3</i>
2.3 DATOVÉ TYPY RELAČNÍ DATABÁZE.....	4
1.1.5 Celá čísla.....	4
1.1.6 Čísla s pohyblivou desetinou čárkou.....	4
1.1.7 Řetězce.....	4
1.1.8 Datum a čas.....	4
3. PŘÍKAZY MYSQL.....	5
3.1 DATABÁZE.....	5
1.1.9 Založení databáze.....	5
1.1.10 Výpis databází MySQL.....	5
1.1.11 Nastavení aktivní databáze.....	5
1.1.12 Název aktuální databáze.....	5
1.1.13 Smazání databáze.....	5
3.2 TABULKY.....	5
1.1.14 Vytvoření tabulky.....	5
1.1.15 Vytvoření dočasné tabulky.....	5
1.1.16 Výpis popisu tabulky.....	5
1.1.17 Změny v tabulce.....	6
<i>Nový sloupec.....</i>	<i>6</i>
<i>Smazání sloupce.....</i>	<i>6</i>
1.1.18 Přejmenování tabulky.....	6
3.3 PRÁCE S DATY.....	7
1.1.19 Vkládání záznamů.....	7
1.1.20 Výpis záznamů.....	7
1.1.21 Mazání záznamů.....	7
4. VÝPIS SQL TABULKY POMOCÍ PHP.....	7

1. MySQL

- je multiplatformní databázový systém
- komunikuje s databází pomocí jazyka SQL

1.1 SQL

- Structured Query Language (strukturovaný dotazovací jazyk)
- umožňuje ovládat databázi pomocí příkazů – dotazů

1.1.1 SQL základní příkazy

Příkazy pro manipulaci s daty

- SELECT – vybírá data z databáze, umožňuje výběr podmnožiny a řazení dat
- INSERT – vkládá do databáze nová data
- UPDATE – mění data v databázi (editace)
- DELETE – odstraňuje data (záznamy) z databáze

Příkazy pro definici dat

- CREATE – vytváření nových objektů
- ALTER – změny existujících objektů
- DROP – odstraňování objektů

Příkazy pro řízení dat

- GRANT – příkaz pro přidělení oprávnění uživateli k určitým objektům.
- REVOKE – příkaz pro odnětí práv uživateli.
- BEGIN – zahájení transakce.
- COMMIT – potvrzení transakce.
- ROLLBACK – zrušení transakce, návrat do původního stavu.

1.2 SŘBD

- je softwarové vybavení, které zajišťuje práci s databází (tvoří rozhraní mezi aplikačními programy a uloženými daty)
- řeší zejména ukládání dat, hledání v datech, souběžný přístup a vzájemnou provázanost dat.
- musí být schopno efektivně pracovat s velkým množstvím dat
- nejčastější dnes používané SŘBD jsou relační databáze - například MySQL

2. Relační databáze

- je databázový systém založený na relačním modelu dat a relační algebře
- relační databázi tvoří kolekce více tabulek, jejich funkčních vztahů, indexů a dalších součástí
- data jsou uspořádána do tabulek (relací), nad kterými jsou definovány přípustné operace
- software pro řízení databáze je právě SŘBD

2.1 Relační tabulka

- je struktura záznamů s pevně stanovenými položkami (sloupce - řádky)
- tabulky (relace) tvoří základ relační databáze
- každý sloupec má definován jednoznačný název, typ a rozsah, neboli doménu

Záznam se stává n-ticí (řádkem) tabulky, pokud jsou v různých tabulkách sloupce stejného typu, pak tyto sloupce mohou vytvářet vazby mezi jednotlivými tabulkami. Tabulky se poté naplňují vlastním obsahem - konkrétními daty.

2.2 Základní operace relační algebry

1.1.2 Projekce

- z původní množiny záznamů vrátí pouze vybrané atributy
- za SELECT je uveden seznam polí (atributů), která se zahrnou do výsledné množiny záznamů
- klausule ORDER BY uspořádá data abecedně podle pole Zákazník_Příjmení a následně podle pole Zákazník_Jméno

```
SELECT Zakaznik_Prijmeni, Zakaznik_Jmeno
FROM Tab_Zakaznici
WHERE Zakaznik_Prijmeni = 'Vomáčka'
ORDER BY Zakaznik_Prijmeni, Zakaznik_Jmeno;
```

1.1.3 Selekce

- vytvoření relace s tímž schématem a ponechání prvků z původní relace splňující podmínku

1.1.4 Spojení

- nejběžnější případ relačních operací
- propojuje množiny záznamů na základě porovnání polí

spojení na rovnost (vnitřní)

- spojení na základě operátoru rovnosti
- vrátí pouze ty záznamy, ve kterých si vzájemně odpovídají hodnoty

```
SELECT Tab_Objednavky.IDObjednavka,
Tab_Vyroby_Na_Objednavkach.IDVyrobek,
Tab_Vyroby_Na_Objednavkach.Mnozstvi_Vyrobkou,
Tab_Vyroby_Na_Objednavkach.Jednotkova_Cena
FROM Tab_Objednavky
INNER JOIN Tab_Vyroby_Na_Objednavkach
ON Tab_Objednavky.IDObjednavka = Tab_Vyroby_Na_Objednavkach.IDObjednavka
WHERE (((Tab_Objednavky.IDObjednavka)=4));
```

Theta spojení (vnitřní)

- spojení na základě operátorů <>, >, >=, <, <=
- používají se většinou při porovnávání záznamů, jejichž nějaká hodnota je např. větší než průměr hodnot všech záznamů

Vnější spojení (outer join)

- vrátí všechny záznamy, které by vrátilo spojení vnitřní plus všechny záznamy z jedné nebo obou výchozích množin záznamů, přičemž místo chybějících hodnot (vzájemně neodpovídajících) se vypíše hodnoty Null
- existují vnější spojení levá, pravá a plná, směr spojení se rozlišuje podle pořadí v jakém jsou množiny uvedeny v příkazu SELECT
- plné vnější spojení vrátí všechny záznamy z obou zadaných množin, a spojí vzájemně záznamy, které splňují podmínky

- následující příklad tedy v obou případech vrátí všechny záznamy z množiny A a z množiny B jen ty, které splňují podmínky uvedené v <podmínka>

```
SELECT * FROM A LEFT OUTER JOIN B ON <podmínka>
SELECT * FROM B RIGHT OUTER JOIN A ON <podmínka>
```

- plné vnější spojení vrátí všechny záznamy z obou zadaných množin, a spojí vzájemně záznamy, které splňují podmínky v <podmínka>

```
SELECT * FROM A FULL OUTER JOIN B ON <podmínka>
```

Projekce vybírá sloupce, operace **selekce** vybírá řádky (záznamy) a operace **spojení** provádí spojení dvou tabulek přes zvolenou položku nebo skupinu položek.

2.3 Datové typy relační databáze

1.1.5 Celá čísla

Název	Rozsah hodnot	Bez znaménka	Místo v paměti
TINYINT	± 128	0 – 255	1 bajty
SMALLINT	± 32 tisíc	0 – 65 535	2 bajty
MEDIUMINT	± 8 mil	0 – 16 mil	3 bajty
INT (INTEGER)	± 2 miliardy	0 – 4 miliardy	4 bajty
BIGINT	± 9*10 ¹⁸	0 – 18*10 ¹⁸	8 bajty
BIT (BOOL)	Synonymum pro TINYINT		

1.1.6 Čísla s pohyblivou desetinnou čárkou

Název	Rozsah hodnot	Místo v paměti
FLOAT	± 3,402823466E+38	4 bajty
DOUBLE	±1,17976931348623157E+308	8 bajty
DOUBLE PRECISION (REAL)	- II -	
DECIMAL (M, D)	Rozsah nastavíme parametry M a D	M + 2 bajtů
DEC (NUMERIC)	- II -	

1.1.7 Řetězce

Název	Délka	Místo v paměti
CHAR	0 – M (0 - 255) znaků	1 bajt
VARCHAR	0 – M (0 - 255) znaků	1 + 1 bajt
TINYBLOB/ TINYTEXT	255 znaků	1 bajtů
BLOB/ TEXT	65 535 znaků	2 + 2 bajty
MEDIUMBLOB/ MEDIUMTEXT	16 mil znaků	3 + 3 bajty
LOB/ LONGTEXT	4 miliardy znaků	4 + 4 bajty
ENUM	65 535 znaků	2 bajty
SET	64 znaků	8 bajtů

1.1.8 Datum a čas

Název	Formát	Místo v paměti
DATE	RRRR-MM-DD	3 bajty
TIME	HH:MM:SS	3 bajty
DATETIME	RRR-MM-DD HH:MM:SS	8 bajty
TIMESTAMP	RRR-MM-DD HH:MM:SS	4 bajty
YEAR	RRRR	1 bajty

3. Příkazy MySQL

3.1 Databáze

1.1.9 Založení databáze

```
CREATE DATABASE nazev_databaze;
```

- příkaz vytvoří databázi se jménem "*nazev_databaze*" (délka názvu může být max. 65 znaků)
- abychom mohli databázi používat, musíme v ní vytvořit jednu či více tabulek příkazem CREATE TABLE (viz níže)
- při práci v příkazovém řádku, musíme určit aktivní databázi příkazem USE (viz níže)

1.1.10 Výpis databází MySQL

```
SHOW DATABASES;
```

- příkaz vám zobrazí jména databází ve vašem spuštěném MySQL

1.1.11 Nastavení aktivní databáze

```
USE nazev_databaze;
```

- databázi "*nazev_databaze*" nastavíme takto jako aktivní a můžeme s ní pracovat

1.1.12 Název aktuální databáze

```
SELECT DATABASE();
```

- vrátí název aktuální databáze

1.1.13 Smazání databáze

```
DROP DATABASE nazev_databaze;
```

- vymaže celou databázi se jménem "*nazev_databaze*", tedy všechny tabulky a data v nich uložená

3.2 Tabulky

1.1.14 Vytvoření tabulky

```
CREATE TABLE nazev_tabulky (nazev_sloupce datovy_typ,... );
```

- v databázi, která je právě aktivní, vytvoříme novou tabulku
- délka názvu tabulky (a sloupců) může být max. 65 znaků

1.1.15 Vytvoření dočasné tabulky

```
CREATE TEMPORARY TABLE nazev_tabulky (nazev_sloupce datovy_typ,...);
```

- takto vytvoříme dočasnou, která po uzavření spojení s databází zanikne

1.1.16 Výpis popisu tabulky

```
DESCRIBE nazev_tabulky;
```

```
SHOW COLUMNS FROM nazev_tabulky;
```

- příkaz nám zobrazí definici požadované tabulky (názvy + datové typy + modifikátory)

1.1.17 Změny v tabulce

```
ALTER TABLE nazev_tabulky prikaz1, prikaz2, prikaz3, prik...;
```

- provede nějaký příkaz/příkazy s tabulkou "nazev_tabulky", viz dále...

Nový sloupec

```
.. ADD nazev_noveho_sloupce datovy_typ;  
.. ADD COLUMN nazev_noveho_sloupce datovy_typ;
```

- příkaz přidá do tabulky nový sloupec

```
ALTER TABLE knihovna ADD COLUMN vydavatel VARCHAR(10);
```

Smazání sloupce

```
.. DROP nazev_odstranovaneho_sloupce;  
.. DROP COLUMN nazev_odstranovaneho_sloupce;
```

- příkaz odebere požadovaný sloupec

```
ALTER TABLE knihovna DROP vydavatel;
```

1.1.18 Přejmenování tabulky

```
.. RENAME novy_nazev_tabulky;
```

- příkaz přejmenuje požadovanou tabulk

```
ALTER TABLE knihovna RENAME knihovnicka;
```

3.3 Práce s daty

1.1.19 Vkládání záznamů

```
INSERT INTO nazev_tabulky VALUES (seznam_hodnot);
```

1.1.20 Výpis záznamů

```
SELECT pozadavky FROM podminky_vyberu;
```

1.1.21 Mazání záznamů

```
DELETE FROM nazev_tabulky WHERE podminka;
```

4. Výpis SQL tabulky pomocí PHP

```
<?
$spojeni = mysql_connect (localhost,"jmeno","heslo");
mysql_select_db(noviny, $spojeni);
$vysledek = mysql_query(
"SELECT * FROM clanky ORDER BY datum",
$spojeni);
/* Konec přímé práce s databází. */

/* Cyklem procházím řádky výsledku a vytahuju z něj hodnoty do nového pole $zaznam */
while ($zaznam = mysql_fetch_array($vysledek) ):
echo "<p><a href='vypisclanku.php?id='";
echo $zaznam["id"];
echo ">";
echo $zaznam["titulek"];
echo "<br> ";
echo $zaznam["uvod"];
echo "<br> ";
echo "Autor: ";
echo $zaznam["autor"];
echo "</p>";
endwhile;
?>
```

- *Jméno a heslo*
- *Zápis SQL příkazu*
- *Název tabulky „clanky“*
- *Jména polí v tabulce*