



Jazyk SQL

HISTORIE (1)

- ❑ V polovině 70. let vznikají databázové systémy založené na relacích a relační algebře, proto bylo nutné vytvořit sadu příkazů, kterými by se relační databáze ovládaly – vznikl jazyk **SEQUEL** (Structured English Query Language)
- ❑ V roce 1977 byl v upravené formě **SEQUEL/2** použit v databázovém systému **SYSTÉM R**
- ❑ V roce 1979 vznikl relační databázový systém – **ORACLE**
- ❑ IBM uvedla v roce 1981 nový systém **SQL/DS** a v roce 1983 **DB2**

HISTORIE (2)

- ❑ Vznikají databázové systémy dalších firem – **PROGRESS**
INFORMIX SYBASE
- ❑ V roce 1986 se jazyk SQL stává standardem – **SQL86**
- ❑ SQL 86 obsahuje nedostatky, nejsou v něm obsaženy důležité prvky - v roce 1992 vzniká nový standard **SQL-92** nebo také **SQL92**.
- ❑ V současné době **SQL3** – řeší objektový přístup a objektové databáze obecně

PŘÍKAZY JAZYKA SQL

- ❑ SQL je neprocedurální jazyk – v příkazech popisuje „co“ chceme získat, a ne „jak“
- ❑ SQL příkazy se dělí do dvou základních skupin
 - příkazy pro manipulaci s daty – **DML**
 - příkazy pro definici dat – **DDL**

PŘÍKLAD DATABÁZE

kniha

ISBN	nazev	ID_autor	nakladatel	pocet_stran
0001	Vekslák	14331	1	278
0002	Hříšnice	14332	2	336
0003	Svatoušek	14333	3	286
0004	Matka	14332	2	288
0005	Pidilidi	14334	2	136

autor

ID_autor	jmeno	prijmeni
14331	Jan	Frybort
14332	Petra	Hammersfahrová
14333	Madeline	Hunter
14334	Mary	Nortonová

nakladatel

nakladatel	nazev
1	Šulc
2	Albatros
3	Alpress

Příkazy pro manipulaci s daty – DML

□ SELECT, INSERT, UPDATE, DELETE

□ Příkaz **SELECT**

- umožňuje výběr a prohlížení záznamů uložených v tabulkách

- Syntaxe:

SELECT	<i><seznam výstupních položek></i>
FROM	<i><seznam tabulek></i>
WHERE	<i><podmínka></i>
GROUP BY	<i><seznam položek></i>
HAVING	<i><skupinová podmínka></i>
ORDER BY	<i><kritéria třídění></i>

DML

- Výpis všech informací v tabulce

```
SELECT * FROM kniha
```

ISBN	nazev	ID_autor	nakladatel	pocet_stran
0001	Vekslák	14331	1	278
0002	Hříšnice	14332	2	336
0003	Svatoušek	14333	3	286
0004	Matka	14332	2	288
0005	Pidilidi	14334	2	136

DML – projekce

- Výběr sloupců z tabulky

- pomocí příkazu **SELECT** omezíme sloupce

```
SELECT nazev, pocet_stran FROM kniha
```

nazev	pocet_stran
Vekslák	278
Hříšnice	336
Svatoušek	286
Matka	288
Pidilidi	136

DML - restrikce

- Omezení počtu řádků

```
SELECT * FROM kniha WHERE nakladatel=2
```

ISBN	nazev	ID_autor	nakladatel	pocet_stran
0002	Hříšnice	14332	2	336
0004	Matka	14332	2	288
0005	Pidilidi	14334	2	136

DML

- Kombinace projekce a restrikce:

```
SELECT nazev FROM kniha WHERE pocet_stran > 280
```

nazev
Hříšnice
Svatouše k
Matka

DML – ORDER BY

- Třídění řádků na výstupu

```
SELECT * FROM kniha ORDER BY nazev
```

ISBN	nazev	ID_autor	nakladatel	pocet_stran
0002	Hříšnice	14332	2	336
0004	Matka	14332	2	288
0005	Pidilidi	14334	2	136
0003	Svatoušek	14333	3	286
0001	Vekslák	14331	1	278

- vybere z tabulky všechny řádky a seřadí je abecedně podle názvu

DML

- ❑ Souhrnné informace za celou tabulku
- ❑ Agregáční funkce - SUM (součet), AVG (průměr), MIN (minimum), MAX (maximum), COUNT (počet)

```
SELECT COUNT (*), AVG (pocet_stran), MIN (počet_stran),  
MAX (počet_stran) FROM kniha
```

COUNT(*)	AVG(pocet_stran)	MIN(počet_stran)	MAX(počet_stran)
5	264,8	136	336

DML – GROUP BY

- ❑ Souhrnné informace za skupiny
- ❑ `SELECT nakladatel, COUNT (*), AVG (pocet_stran), MIN (počet_stran), MAX (počet_stran) FROM kniha GROUP BY nakladatel`

nakladatel	COUNT (*)	AVG(počet_stran)	MIN(počet_stran)	MAX(počet_stran)
1	1	278	278	278
2	3	253,3	136	336
3	1	286	286	286

- ❑ seskupí řádky podle `nakladatel`
- ❑ souhrnná funkce je vypočtena pro každou vzniklou skupinu

DML - HAVING

- omezuje, které souhrnné řádky budou ve výsledku
- `SELECT nakladatel, SUM(počet_stran) FROM kniha GROUP BY nakladatel HAVING SUM(počet_stran) > 280`

nakladatel	SUM(pocet_stran)
2	760
3	286

SPOJOVÁNÍ TABULEK

- ❑ spojování se provádí spojovací podmínkou, ta určuje, které řádky se mají navzájem spojit
- ❑ podmínka se zadává jako rovnost sloupců, z nichž každý je v jedné tabulce

SPOJOVÁNÍ TABULEK

- `SELECT` `nazev`, `jmeno`, `prijmeni` `FROM` `kniha`, `autor`
`WHERE` `kniha.ID_autor` = `autor.ID_autor`

nazev	jmeno	prijmeni
Vekslák	Jan	Frýbort
Hříšnice	Petra	Hammersfahrová
Svatoušek	Madeline	Hunter
Matka	Petra	Hammersfahrová
Pidilidi	Mary	Nortonová

INSERT

- ❑ přidává nové řádky do tabulky
- ❑ `INSERT INTO autor VALUES (14335 , 'Jaroslav' , 'Hašek')`
do tabulky autor přidá nový řádek
- ❑ Neznámou hodnotu nahrazujeme speciální hodnotou NULL
- ❑ `INSERT INTO autor VALUES (14335 , NULL , 'Hašek')`

UPDATE

- ❑ opravuje dříve zadanou hodnotu v databázi
- ❑ nedochází ke změně počtu řádků
- ❑ `UPDATE autor SET prijmeni = 'Novotná'`
`WHERE ID_autor = 14332`
- ❑ u autora 14332 změní jméno na Novotná

DELETE

- ❑ Smaže z tabulky všechny záznamy, které vyhovují podmínce
- ❑ `DELETE FROM autor WHERE ID_autor = 14335`
- ❑ z tabulky autor smaže autora 14335

Příkazy definici dat – DDL

- ❑ CREATE TABLE
- ❑ DROP TABLE
- ❑ ALTER TABLE

VYTVOŘENÍ TABULKY

❑ CREATE TABLE

❑ CREATE TABLE autor

```
ID_autor    int NOT NULL
```

```
jmeno      varchar(40)
```

```
prijmeni   varchar(50)
```

❑ Vlastnost NOT NULL definuje, že položka ID_autor nemůže obsahovat prázdnou položku

ODSTARNĚNÍ TABULKY

- ❑ DROP TABLE
- ❑ DROP TABLE autor
- ❑ Vymaže tabulku autor

ZMĚNA TABULKY

- ❑ ALTER TABLE
- ❑ Slouží pro manipulaci s existující tabulkou

- ❑ přidání sloupce

ALTER TABLE *autor*

ADD (*adresa* varchar(5))

přidá do tabulky *autor* sloupec *adresa*

ZMĚNA TABULKY

- ❑ modifikace sloupce

ALTER TABLE autor

MODIFY (adresa varchar(10))

sloupec *adresa* nebude řetězec o délce 5 znaků ale 10 znaků

- ❑ Smazání sloupce

ALTER TABLE autor

DROP adresa

smaže sloupec *adresa*

NASTAVENÍ PRÁV

- ❑ příkaz GRANT
- ❑ GRANT <práva> - SELECT, INSERT, UPDATE, DELETE
 ON <jméno tabulky>
 TO <jméno uživatele>

Př:

GRANT UPDATE (jmeno, prijmeni) ON autor TO geust

- ❑ uživatel guest má právo aktualizovat sloupec jméno a příjmení

ODEBRÁNÍ PRÁV

- příkaz REVOKE

REVOKE DELETE ON autor FROM guest

- uživateli guest bylo odejmuto právo mazání