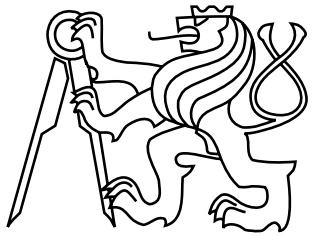
Jazyk SQL 3 - DML a DDL

Michal Valenta

Katedra softwarového inženýrství FIT
České vysoké učení technické v Praze
©Michal Valenta, 2010

BI-DBS, ZS 2010/11

https://edux.fit.cvut.cz/courses/BI-DBS/



SQL DML

- příkazy INSERT, UPDATE, DELETE, MERGE
- transakční zpracování
  - záleží na nastavení klienta (autocommit on|off)
  - předpokládejme autocommit off, potom:
    - transakce je tvořena sekvencí DML a SELECT příkazů
    - potvrzena příkazem COMMIT
    - odvolána příkazem ROLLBACK
    - ostatní session vidí v průběhu transakce staré hodnoty, dokud tato není potvrzena
    - příkazy DDL nebo DCL provedou implicitní COMMIT
  - systematický výklad transakčního zpracování bude podán v samostatné přednášce
- při zpracování DML příkazů si DBMS hlídá platnost integritních omezení

Aktualizace v SQL – příklady

Smazání řádku / řádek

DELETE FROM Filmy
WHERE jmeno_f = 'Puška'; - - pozor na správně formulovanou podmínku

zmněna hodnoty řádku / řádek

UPDATE Zakaznici SET jmeno = 'Gotzová'
WHERE rod_c = '4655292130'; - - pozor na správně formulovanou podmínku

UPDATE Zákazníci SET jméno = 'Müller'
WHERE jméno = 'Muller';

UPDATE více řádek pomocí vnořeného dotazu

Doplň k tabulce Zakaznici redundantní atribut pocet_pujcek a jednorázově dopočti jeho hodnoty.

```
ALTER TABLE zakaznici
Add Pocet_pujcek Number; - - přidáme sloupec to tabulky zákazníci

UPDATE Zákazníci Z - - jednorázově do něj doplníme hodnoty
SET Pocet_pujcek = (SELECT count(*) from Vypucky V
                    WHERE V.rod_c = Z.rod_c);
```

Pokud se databáze mění, je třeba udržovat tuto informaci konzistentní!!!. Lze toho dosáhnout například pomocí triggerů.

INSERT

Vložení jednoho řádku do tabulky Zakaznici.

```
INSERT INTO Zakazníci (rod_c, jmeno)
                VALUES ('4804230160','Novák');
```

Vložení více řádek vnořeným SELECTem

```
CREATE TABLE Kolik_kopii - - neprve vytvoříme tabulku
(rod_c CHAR(10), pocet SMALLINT);

INSERT INTO Kolik_kopii
SELECT rod_c, COUNT(c_kopie) FROM Vypujcky
GROUP BY rod_c; - - ... a pak ji naplníme daty
nebo celé v jednom příkazu:
CREATE TABLE Kolik_kopii
(rod_c CHAR(10), pocet SMALLINT)
AS SELECT rod_c, COUNT(c_kopie) FROM Vypujcky
GROUP BY rod_c;
```

Pokračování přednášky

... je prozatím k dispozici v méně učesané formě v samostaném PDF souboru.
Do konce semestru se pokusím překlopit slides do stejného formátu.

Pohledy - charakteristika

- pohled je virtuální relace
- v datové slovníku je uložen ve formě SELECT příkazu, kterým je definován
- z hlediska dotazování je pohled zaměnitelný s tabulkou
- DML operace nad pohledy v omezené míře
  - musí se jednat o tzv “simple view” (neobsahuje operace join, agregace, výrazy, ...)
  - DML nesmí být zakázany v jeho definici – klauzule READ ONLY
  - je doporučeníhodné používat klauzuli WITH CHECK OPTION
- DML nad tzv. “complex views” lze realizovat pomocí instead-of triggerů
- k čemu pohledy slouží:
  - odstínění informací, které uživatel/role nemá vidět
  - zpřehlednění složitých dotazů
  - snazší vývoj aplikací
- pohledy nepřinášejí výkonové zrychlení – k tomu lze použít tzv. materializované pohledy (MATERIALIZED VIEWS)

Pohledy - charakteristika

Definice pohledu – syntaxe

```
CREATE VIEW jméno-pohledu [(v-jméno-atr[,v-jméno-atr]...)]
AS dotaz
WITH CHECK OPTION;
```

Pohled obsahující pouze Pražáky

```
CREATE VIEW Prazaci AS
SELECT c_ct, jmeno, adresa
FROM Zakaznici WHERE adresa LIKE '%PRAHA%';

DROP VIEW Prazaci;

CREATE VIEW Dluznici (rod_c, pocet_vypujcek) AS
SELECT rod_c, COUNT(c_kopie) FROM Vypucky
GROUP BY rod_c;

SELECT * from Dluznici
where pocet_vypujcek > 5;
```