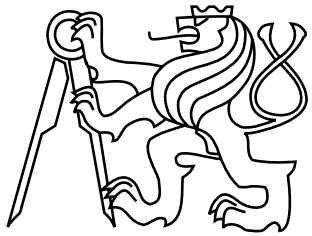
Jazyk SQL 1

Michal Valenta

Katedra softwarového inženýrství FIT
České vysoké učení technické v Praze

BI-DBS, ZS 2014/15

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



Principy SQL

- Říkáme, co chceme získat, ne jak se to má dělat.
- Intuitivně srozumitelný zápis.
  Připomíná jednoduché anglické věty.
  Nepřipouští se žádné zkratky.
- Snadná čitelnost => odsazování, více řádek.
- Kličová slova a názvy objektů nejsou case senzitvní, porovnání řetězců ano (pokud není v implementaci na úrovni session řečeno jinak – MySQL)
- Standardizace: 1986, 1992, 1999, 2003, 2005, 2008, 2011 ...
- Nový standard obvykle zahrnuje předchozí.
- Standardy postupně “nabalují“ další rysy (OO, XML, ...)
- Implementace mají svoje odchylky – většinou podstatná část standardu 92.

Přehled SQL

- jazyk pro definici dat (DDL)
  - jazyk pro definice pohledů
  - jazyk pro definice IO
- jazyk pro manipulaci dat a dotazování (DML)
- jazyk pro přiřazení přístupových práv (DCL)
- jazyk pro řízení transakcí (TCL)
- systémový katalog
- jazyk modulů

Schéma příkladu

KINA(NAZEV_K, ADRESA, JMENO_V)

FILMY(JMENO_F, REZISER, ROK)

ZAKAZNICI(ROD_C, JMÉNO, ADRESA)

ZAMESTNANCI(OSOBNI_C, ADRESA, JMENO, PLAT)

HERCI(ROD_C, JMENO, ADRESA, SPECIALIZACE)

KOPIE(C_KOPIE, JMENO_F)

VYPUJCKY(C_KOPIE,OSOBNI_C, ROD_C, CENA,DATUM_V)

REZERVACE(JMENO_F, ROD_C)

PREDSTAVENI(NAZEV_K, JMENO_F, DATUM)

OBSAZENI(JMENO_F, ROD_C_HERCE, ROLE)

SELECT nad 1 tabulkou

SELECT – základní syntaxe

SELECT specifikace_sloupců
FROM specifikace_zdroje
[WHERE podmínka_selekce]
[ORDER BY specifikace_řazení]

specifikace_sloupců =
{DISTINCT | ALL} * | {jm_sloupce [,jm_sloupce]...}

Jednoduché dotazy v SQL

D1. Vypiš tabulku jmen režisérů a roků, kdy natočili nějaký film.

```
SELECT REZISER, ROK FROM FILMY;
```

Zdroj:

NAZEV_F	REZISER	ROK
Nevinná	Bínovec V.	1939
Kristián	Frič M.	1939
Madla zpívá . . .	Bínovec V.	1940
Noční motýl	Čáp F.	1941
Ohnivé léto	Čáp F.	1939
Dceruška k . . .	Bínovec V.	1940
Eva tropí . . .	Frič M.	1939
Cesta do hlubin ...	Frič M.	1939

Výsledek:

REZISER	ROK
Bínovec V.	1939
Frič M.	1939
Bínovec V.	1940
Čáp F.	1941
Čáp F.	1939
Bínovec V.	1940
Frič M.	1939
Frič M.	1939

Jednoduché dotazy v SQL

D1. Vypiš tabulku jmen režisérů a roků, kdy natočili nějaký film.

```
SELECT REZISER, ROK FROM FILMY ORDER BY REZISER, ROK;
```

Zdroj:

NAZEV_F	REZISER	ROK
Nevinná	Bínovec V.	1939
Kristián	Frič M.	1939
Madla zpívá . . .	Bínovec V.	1940
Noční motýl	Čáp F.	1941
Ohnivé léto	Čáp F.	1939
Dceruška k . . .	Bínovec V.	1940
Eva tropí . . .	Frič M.	1939
Cesta do hlubin ...	Frič M.	1939

Výsledek:

REZISER	ROK
Bínovec V.	1939
Bínovec V.	1940
Bínovec V.	1940
Čáp F.	1939
Čáp F.	1941
Frič M.	1939
Frič M.	1939
Frič M.	1939

Jednoduché dotazy v SQL

D1. Vypiš tabulku jmen režisérů a roků, kdy natočili nějaký film.

SELECT DISTINCT REZISER, ROK FROM FILMY:

Zdroj:

NAZEV_F	REZISER	ROK
Nevinná	Bínovec V.	1939
Kristián	Frič M.	1939
Madla zpívá . . .	Bínovec V.	1940
Noční motýl	Čáp F.	1941
Ohnivé léto	Čáp F.	1939
Dceruška k . . .	Bínovec V.	1940
Eva tropí . . .	Frič M.	1939
Cesta do hlubin ...	Frič M.	1939

Výsledek:

REZISER	ROK
Bínovec V.	1939
Bínovec V.	1940
Čáp F.	1939
Čáp F.	1941
Frič M.	1939

DISTINCT se vztahuje vždy na celou klauzuli SELECT.
Řazení při použití DISTINCT bývá implicitní (nemusí nutně být).

Jednoduché dotazy v SQL

D2. Najdi filmy natočené před rokem 1940.

```
SELECT * FROM FILMY
WHERE ROK < 1940;
```

Výsledek:

JMENO_F	REZISER	ROK
Nevinná	Bínovec V.	1939
Kristián	Frič M.	1939
Ohnivé léto	Čáp F.	1939
Eva tropí . . .	Frič M.	1939
Cesta do hlubin ...	Frič M.	1939

Klauzule WHERE

D3. Vypiš z tabulky Filmy řádky tykající se filmů natočených v letech 1938-1940.

```
SELECT *
FROM Filmy
WHERE rok>=1938 AND rok<=1940;
```

Výsledek:

JMENO_F	REZISER	ROK
Nevinná	Bínovec V.	1939
Kristián	Frič M.	1939
Madla zpívá . . .	Bínovec V.	1940
Ohnivé léto	Čáp F.	1939
Dceruška k . . .	Bínovec V.	1940
Eva tropí . . .	Frič M.	1939
Cesta do hlubin ...	Frič M.	1939

alternativa

```
SELECT * FROM Filmy
WHERE rok BETWEEN 1938 AND 1940;
```

doplněk 1

```
SELECT * FROM Filmy
WHERE rok NOT BETWEEN 1938 AND 1940;
```

doplněk 2

```
SELECT * FROM Filmy
WHERE NOT (rok BETWEEN 1938 AND 1940);
```

Logické operátory

=	je rovno
<> nebo !=	není rovno
<	je menší
>	je větší
<=	je menší nebo rovno
>=	je vetší nebo rovno
Between	
Like	
	... a mnoho dalších

- V klauzuli WHERE lze používat též logické spojky AND, OR a NOT.
- Pro správné vyjádření pořadí vyhodnocení se použíají závorky.
- Klauzule WHERE se vyhodnotí jako výraz typu boolean. Tedy TRUE, FALSE nebo NULL.

Logické operace, výrazy a hodnota NULL

Všechny datové typy (domény) mají “bottom” prvek NULL.
Má význam “UNKNOWN”, “NEUVEDENO”, “N/A” apod.. Nezaměňovat s 0 !
“Tříhodnotová” logika je na operacích AND, OR a NOT definována takto:

A	B	A and B	A or B	not A
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
TRUE	NULL	NULL	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	NULL	FALSE	NULL	TRUE
NULL	TRUE	NULL	TRUE	NULL
NULL	FALSE	FALSE	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Pokud alespoň jeden člen porovnání (=, <>, <, >, <=, >=) nabývá hodnoty NULL, pak je výsledek porovnání NULL.
Hodnota aritmetického výrazu, kde alespoň jeden člen je NULL je také NULL.
Totéž by mělo platit u řetězců. Například výraz: ’A’||NULL||’B’ by měl být NULL.

POZOR: v implementacích to u řetězců nemusí být pravda:

Oracle: ’A’||NULL||’B’ = AB PostgreSQL: ’A’||NULL||’B’ = NULL

V praxi je lepší se NULL hodnotě ve výrazech vyhnout s pomocí operátorů is null, is not null a coalesce.

SELECT – spojení

SELECT – základní syntaxe

SELECT specifikace_sloupců
FROM specifikace_zdroje
[WHERE podmínka_selekce]
[ORDER BY specifikace_řazení]

specifikace_zdroje =
tabulka | pohled | (vnořený_dotaz) |
specifikace_zdroje [{LEFT | RIGHT | FULL} [OUTER]] JOIN
    specifikace_zdroje ON (podmínka_spojení) |
specifikace_zdroje [{LEFT | RIGHT | FULL} [OUTER]] JOIN
    specifikace_zdroje USING (seznam_sloupců) |
specifikace_zdroje [{LEFT | RIGHT | FULL} [OUTER]]
    NATURAL JOIN specifikace_zdroje |
specifikace_zdroje CROSS JOIN specifikace_zdroje |
specifikace_zdroje , specifikace_zdroje

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

```
SELECT DISTINCT reziser
FROM Filmy JOIN Rezervace ON (Filmy.jmeno_f=Rezervace.jmeno_f);
```

Alternativa 1

```
SELECT DISTINCT reziser
FROM Filmy JOIN Rezervace USING (jmeno_f);
```

Alternativa 2

```
SELECT DISTINCT reziser
FROM Filmy NATURAL JOIN Rezervace;
```

Alternativa 3

```
SELECT DISTINCT reziser
FROM Filmy, Rezervace WHERE Filmy.jmeno_f=Rezervace.jmeno_f;
```

Alternativa 3 je zápis dle standardu SQL86. Je stále použitelný. Dotaz je formulován (syntakticky): Filmy x Rezervace (Filmy.nazev_f=Rezervace.nazev_f)[reziser].

Dotazy – kvalifikace, alias

D5. Najdi co hrají v kterých kinech.

```
SELECT *
FROM Filmy JOIN Predstaveni
    ON (Filmy.jméno_f=Predstaveni.jméno_f);
```

- Chci-li zadat projekci :

Kvalifikace v klauzuli select

```
SELECT Prestaveni.nazev_k, Filmy.*
FROM Filmy JOIN Predstaveni ON (Filmy.jmeno_f=Predstaveni.jmeno_f)
ORDER BY Predstaveni.název_k;
```

Zavedení aliasů a jejich použití

```
SELECT P.název_k, F.*
FROM Filmy F JOIN Predstaveni P ON (F.jmeno_f= P.jmeno_f)
ORDER BY P.název_k;
```

Dotazy na spojení tabulky sama se sebou

D6. Najdi rdvojice zákazníků,kteří mají stejnou adresu.

```
SELECT Soused1.rod_c AS prvni, soused2.rod_c AS druhy
FROM Zakazníci Soused1 JOIN Zakaznici Soused2 Using (adresa)
WHERE Soused1.rod_c < Soused2.rod_c;
```

- Zástupná jména (aliasy) zde slouží k odlišení dvou instancí téže tabulky.

Dotazy se spojením a selekcí

D7. Najdi kina, kde hrají filmy natočené v roce 1936 nebo 1939, jejichž režisér není Frič ani Bínovec.

```
SELECT P.nazev_k, F.jmeno_f
FROM Filmy F JOIN Predstaveni P ON (F.jmeno_f=P.jmeno_f)
WHERE (F.rok = 1936 OR F.rok 1939)
      AND NOT (F.reziser = 'Frič' OR F.reziser = 'Bínovec');
```

... nebo

```
SELECT P.nazev_k, F.jmeno_f
FROM Filmy F JOIN Predstaveni P ON (F.jmeno_f=P.jmeno_f AND
      (F.rok = 1936 OR F.rok 1939))
WHERE NOT (F.reziser = 'Frič' OR F.reziser = 'Bínovec');
```

Lze napsat ještě mnoho dalších syntaktických variací tohoto dotazu.
Ta první je zřejmě srozumitelnější a logičtější než varianta druhá.
Dobře fungující optimalizátor by měl pro obě varianty sestavit stejný prováděcí plán.

Výrazy v klauzuli SELECT

D8. Vypiš pro zahraniční zaměstnance platy přepočtené na EUR. Zahraniční zaměstnanci nemají rodné číslo.

```
SELECT jmeno, plat/24.65 AS “Plat v Euro”
FROM Zamestnanci
WHERE rod_c IS NULL; - - cizinci (komentář v textu dotazu)
```

- lze použít běžné oprátory jako: ’/’, ’*’, ’-’, ’+’
- obvyklá priorita operátorů, jinak lze použít závorky
- existuje řada vestavěných funkcí (specifické pro různé implementace)
- výrazy mohou být i nad řetězci a datumy
- NULL se propaguje do výsledku, tj. je-li jeden z oparandů NULL, je výsledkem NULL tedy:

...někteří cizinci nepobírají plat, chceme vidět, že berou 0 Eur:

```
SELECT jmeno, (coalesce(plat,0))/24.65 AS “Plat v Euro”
FROM Zamestnanci
WHERE rod_c IS NULL;
```

Relační algebra → SELECT 1/2

R(φ) [A1,A2,..,Aj]

```
SELECT A1,
A2,...,Aj
FROM R
WHERE φ
```

(R1xR2x...xRk)(φ)[A1,A2,...,Aj]

```
SELECT A1, A2,...,Aj
FROM R1, R2,...,Rk
WHERE φ
```

```
SELECT A1, A2,...,Aj
FROM R1 CROSS JOIN R2 ... CROSS JOIN Rk
WHERE φ
```

(R1 * R2 * ... * Rk)(φ)[A1,A2, ..,Aj]

```
SELECT A1,A2,...,Aj
FROM R1 NATURAL JOIN R2 ... NATURAL JOIN Rk
WHERE φ
```

Relační algebra → SELECT 2/2

R1[t1 Θ t2]R2 (φ)[A1,A2,...,Aj]

```
SELECT A1, A2,...,Aj
FROM R1 JOIN R2 ON (R1.t1 Θ R2.t2)
WHERE φ
```

R1xR2(R1.t1 Θ R2.t2)(φ)[A1,A2,...,Aj]

což je totéž jako

R1xR2(R1.t1 Θ R2.t2 and φ)[A1,A2,..,Aj]

```
SELECT A1, A2,...,Aj
FROM R1 CROSS JOIN R2
WHERE (R1.t1 Θ R2.t2 and φ)
```