Jazyk SQL – SELECT

Michal Valenta

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze

BI-DBS, LS 2021/2022

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

Jazyk SQL

- Říkáme, co chceme získat, ne jak se to má technicky dělat.
- Intuitivně srozumitelný zápis.
  Připomíná jednoduché anglické věty.
  Nepřipouští se žádné zkratky.
- Podpora odsazování, lámání na více řádek = snadná čitelnost.
- Klíčová slova a názvy objektů nejsou case sensitive, 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, ...)
- Poslední čistě relační standard byl 1992.
- Implementace mají svoje odchylky – většinou podstatná část standardu 92.

Rozdělení SQL

- Jazyk pro definici dat (Data Definition Language, DDL)
  - Jazyk pro definice pohledů
  - Jazyk pro definice integritních omezení
- Jazyk pro manipulaci dat a dotazování (Data Manipulation Language, DML)
- Jazyk pro přiřazení přístupových práv (Data Control Language, DCL)
- Jazyk pro řízení transakcí (Transaction Control Language, TCL)
- Data Dictionary
- Jazyk modulů – packages, procedury, triggery

Schéma relací použitých v příkladech

KINA(NAZEV_K, ADRESA, JMENO_V)

FILMY(NAZEV_F, REZISER, ROK)

ZAKAZNICI(ROD_C, JMENO, ADRESA)

ZAMESTNANCI(OSOBNI_C, ADRESA, JMENO, PLAT, ROD_C)

HERCI(ROD_C, JMENO, ADRESA, SPECIALIZACE)

KOPIE(C_KOPIE, NAZEV_F)

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

REZERVACE(NAZEV_F, ROD_C)

PREDSTAVENI(NAZEV_K, NAZEV_F, DATUM)

OBSAZENI(NAZEV_F, ROD_C_HERCE, ROLE)

Jednoduchý dotaz na tabulku

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

D7. Najdi jména režisérů a let, kdy natočili nějaký film.

```
SELECT reziser, rok
FROM filmy1;
```

Zdroj:

NAZEV_F	REZISER	ROK
Pátý element	Luc Besson	1997
Vrchní, prchni	Ladislav Smoljak	1980
Pretty Woman	Garry Marshall	1990
Zločiny a poklesky	Woody Allen	1989
Povídky z New Yorku	Woody Allen	1989
September	Woody Allen	1987
Zlaté časy rádia	Woody Allen	1987
Hollywood v koncích	Woody Allen	2002
Samotáři	David Ondtříšek	2000

Výsledek:

REZISER	ROK
Luc Besson	1997
Ladislav Smoljak	1980
Garry Marshall	1990
Woody Allen	1989
Woody Allen	1989
Woody Allen	1987
Woody Allen	1987
Woody Allen	2002
David Ondříček	2000

Jednoduché dotazy v SQL

D7. Vypiš tabulku jmen režisérů a roků, kdy natočili nějaký film.

```
SELECT DISTINCT reziser, rok
FROM filmy1:
```

Zdroj:

NAZEV_F	REZISER	ROK
Pátý element	Luc Besson	1997
Vrchní, prchni	Ladislav Smoljak	1980
Pretty Woman	Luc Besson	1990
Zločiny a poklesky	Woody Allen	1989
Povídky z New Yorku	Woody Allen	1989
September	Woody Allen	1987
Zlaté časy rádia	Woody Allen	1987
Hollywood v koncích	Woody Allen	2002
Samotáři	David Ondříček	2000

Výsledek:

REZISER	ROK
Luc Besson	1997
Ladislav Smoljak	1980
Garry Marshall	1990
Woody Allen	1989
Woody Allen	1987
Woody Allen	2002
David Ondříšek	2000

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

Jednoduché dotazy v SQL

D7. Vypiš tabulku jmen režisérů a let, kdy natočili nějaký film.

```
SELECT reziser, rok
FROM filmy1
ORDER BY reziser ASC, rok DESC;
```

Zdroj:

NAZEV_F	REZISER	ROK
Pátý element	Luc Besson	1997
Vrchní, prchni	Ladislav Smoljak	1980
Pretty Woman	Luc Besson	1990
Zločiny a poklesky	Woody Allen	1989
Povídky z New Yorku	Woody Allen	1989
September	Woody Allen	1987
Zlaté časy rádia	Woody Allen	1987
Hollywood v koncích	Woody Allen	2002
Samotáři	David Ondtříšek	2000

Výsledek:

REZISER	ROK
David Ondříšek	2000
Garry Marshall	1990
Ladislav Smoljak	1980
Luc Besson	1997
Woody Allen	2002
Woody Allen	1989
Woody Allen	1987

Jednoduché dotazy v SQL

D1. Vypiš tabulku filmů natočených před rokem 1990.

```
SELECT *
FROM filmy1
WHERE rok < 1990;
```

Výsledek:

NAZEV_F	REZISER	ROK
Vrchní, prchni	Ladislav Smoljak	1980
Zločiny a poklesky	Woody Allen	1989
Povídky z New Yorku	Woody Allen	1989
September	Woody Allen	1987
Zlaté časy rádia	Woody Allen	1987

Operátory porovnání

=	je rovno
<> nebo !=	není rovno
<	je menší
>	je větší
<=	je menší nebo rovno
>=	je vetší nebo rovno
BETWEEN	je v rozsahu
IN	je v seznamu hodnot
LIKE	shoda na část řetězce
EXISTS	je splněn poddotaz
	... a mnoho dalších

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

Podmínky pomocí WHERE

D8. Vypiš z tabulky filmy1 řádky týkající se filmů natočených v letech 1989 – 2000.

```
SELECT *
FROM filmy1
WHERE rok >= 1989 AND rok <= 2000;
```

Výsledek:

NAZEV_F	REZISER	ROK
Pátý element	Luc Besson	1997
Pretty Woman	Garry Marshall	1990
Zločiny a p...	Woody Allen	1989
Povídky z N...	Woody Allen	1989
Samotáři	David Ondtříšek	2000

alternativa

```
SELECT *
FROM filmy1
WHERE rok BETWEEN 1989 AND 2000;
```

doplněk 1

```
SELECT *
FROM filmy1
WHERE rok NOT BETWEEN 1989 AND 2000;
```

doplněk 2

```
SELECT *
FROM filmy1
WHERE NOT (rok BETWEEN 1989 AND 2000);
```

Operace s hodnotou NULL

Všechny datové typy (domény) mají “bottom” prvek NULL.
Má význam “UNKNOWN”, “NEUVEDENO”, “N/A”. Není to nula ani prázdný řetězec!
“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

Operace s hodnotou 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.

Spojení tabulek

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

D2. Najdi, co hrají ve kterých kinech, spolu s režiséry filmů a rokem jejich natočení.

```
SELECT filmy1.*, predstaveni1.*
FROM filmy1 JOIN predstaveni1
    ON (filmy1.jmeno_f = predstaveni1.jmeno_f);
```

Alternativa 1 (Bezpečnější než "alternativa 2", ale není v RA.)

```
SELECT filmy1.*, predstaveni1.*
FROM filmy1 JOIN predstaveni1 USING (jmeno_f);
```

Alternativa 2 (Pozor na nechtěné atributy stejného jména!)

```
SELECT filmy1.*, predstaveni1.*
FROM filmy1 NATURAL JOIN predstaveni;
```

Alternativa 3

```
SELECT filmy1.*, predstaveni1.*
FROM filmy1, predstaveni1
WHERE filmy1.jmeno_f = predstaveni1.jmeno_f;
```

Alternativa 3 je zápis dle standardu SQL86. Je stále použitelný. Dotaz je (syntakticky):
{Filmy1 x Predstaveni1} (Filmy1.nazev_f=Predstaveni1.nazev_f)[reziser].

Kvalifikace, aliasy

D2. Najdi, co hrají ve kterých kinech, spolu s režiséry filmů a rokem jejich natočení.

```
SELECT *
FROM filmy1 JOIN predstaveni1
WHERE filmy1.jmeno_f=predstaveni1.jmeno_f;
```

- Chci-li zadat projekci:

Kvalifikace v klauzuli SELECT

```
SELECT predstaveni1.nazev_k, filmy1.*
FROM filmy1 JOIN predstaveni1
WHERE filmy1.jmeno_f=predstaveni1.jmeno_f
ORDER BY predstaveni1.nazev_k;
```

Zavedení aliasů a jejich použití

```
SELECT P.nazev_k, F.*
FROM filmy1 F JOIN predstaveni1 P
WHERE F.jmeno_f= P.jmeno_f
ORDER BY P.nazev_k;
```

Spojení tabulky se sebou samou

D9. Najdi dvojice zaměstnaců (reprezentovaných číslem zaměstnance id_zam), kteří mají stejnou pracovní pozici (atribut id_pozice).

```
SELECT z1.id_zam AS prvni, z2.id_zam AS druhy
FROM zamestnanci1 z1, zamestnanci z2
WHERE z1.id_pozice = z2.id_pozice and z1.id_zam < z2.id_zam;
```

- Zástupná jména (aliasy) zde slouží k "mentálnímu" zdvojení téže tabulky.

Dotazy se spojením a selekcí

D10. Najdi kina, kde hrají filmy natočené před rokem 1997 nebo v roce 2000, jejichž režisér není Smoljak a Marshall.

```
SELECT P.nazev_k, F.jmeno_f
FROM filmy1 F, predstaveni1 P
WHERE F.jmeno_f = P.jmeno_f
    AND (F.rok < 1997 OR F.rok = 2000)
    AND NOT (F.reziser = 'Ladislav Smoljak'
                OR F.reziser = 'Garry Marshall')
ORDER BY P.nazev_k;
```

nebo

```
SELECT p.nazev_k, f.jmeno_f
FROM filmy1 f JOIN predstaveni1 p
    ON (f.jmeno_f = p.jmeno_f AND (f.rok < 1997 OR f.rok = 2000))
WHERE NOT (f.reziser = 'Ladislav Smoljak'
              OR f.reziser = 'Garry Marshall')
ORDER BY P.nazev_k;
```

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

D11. Vypiš pro zahraniční zaměstnance jejich platy přepočtené na EUR.

```
SELECT jmeno, prijmeni plat/25.47 AS euro_plat
FROM zamestnanci
WHERE rodne_cislo IS NULL; - - cizinci (komentář v textu dotazu)
```

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

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

```
SELECT jmeno, prijmeni (COALESCE(plat,0))/25.47 AS euro_plat
FROM zamestnanci
WHERE rodne_cislo IS NULL;
```

Relační algebra → SELECT 1/2

R(φ) [A1,A2,..,Aj]

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

(R1xR2x...xRk)(φ)[A1,A2,...,Aj]

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

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

(R1 * R2 * ... * Rk)(φ)[A1,A2, ..,Aj]

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

Relační algebra $\rightarrow$ SELECT 2/2

{R1[t1 Θ t2]R2} (φ)[A1,A2,...,Aj]

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

Důležité pojmy

- Dotaz, struktura dotazu SELECT
- Spojování tabulek pomocí JOIN
- NULL hodnota
- Konverze rel. algebry do SQL SELECT
- Krom **množinových operací** a **VIEW** již nyní umíte o malinko více než relační algebra (navíc ORDER BY, vynechání klauzule DISTINCT, výrazy v klauzuli SELECT).

Nějaký praktický trénink?

- proseminář Jirky Hunky
- jeden pokročilý proseminář bude věnován nástroji DataGrip a také efektivní práci v psql
- volně stažitelné přílohy k učebnici DBS
- vaše semestrálka (3. část)