

Jazyk SQL – SELECT

Michal Valenta

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze
©Michal Valenta, 2021

BI-DBS, LS 2020/2021

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



1 Úvod do jazyka SQL

2 Dotazování v SQL

- Jednoduché dotazy
- Podmínky a operátory
- Spojení tabulek
- Aritmetické operace

3 Relační algebra a SQL

Jazyk SQL

- Říkáme, co chceme získat, ne jak se to má technicky dělat.

Jazyk SQL

- Říkáme, co chceme získat, ne jak se to má technicky dělat.
- Intuitivně srozumitelný zápis.
Připomíná jednoduché anglické věty.
Nepřipouští se žádné zkratky.

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.

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)

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 ...

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.

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, ...)

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.

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

Rozdělení SQL

- Jazyk pro definici dat (Data Definition Language, **DDL**)

Rozdělení SQL

- Jazyk pro definici dat (Data Definition Language, **DDL**)
 - ▶ Jazyk pro definice pohledů

Rozdělení SQL

- Jazyk pro definici dat (Data Definition Language, **DDL**)
 - ▶ Jazyk pro definice pohledů
 - ▶ Jazyk pro definice integritních omezení

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**)

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**)

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**)

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**

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, triggerů

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)

Obsah

1 Úvod do jazyka SQL

2 Dotazování v SQL

- Jednoduché dotazy
- Podmínky a operátory
- Spojení tabulek
- Aritmetické operace

3 Relační algebra a SQL

Jednoduchý dotaz na tabulku

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

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

Jednoduché dotazy v SQL

D1. Vypiš tabulku jmen režisérů a let, 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

Jednoduché dotazy v SQL

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

Jednoduché dotazy v SQL

D1. Vypiš tabulku jmen režisérů a let, 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
Ohnivě léto	Čáp F.	1939
Dceruška k ...	Bínovec V.	1940
Eva tropí ...	Frič M.	1939
Cesta do hlubin ...	Frič M.	1939

Jednoduché dotazy v SQL

D1. Vypiš tabulku jmen režisérů a let, 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
Ohnivě 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 let, 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
Ohnivě 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.

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
Ohnivě léto	Čáp F.	1939
Dceruška k ...	Bínovec V.	1940
Eva tropí ...	Frič M.	1939
Cesta do hlubin ...	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
Ohnivě 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

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
Ohnivě 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.

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
Ohnivě 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 (ale nemusí nutně být).

Jednoduché dotazy v SQL

D2. Najdi filmy natočené před rokem 1940.

Jednoduché dotazy v SQL

D2. Najdi filmy natočené před rokem 1940.

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

Jednoduché dotazy v SQL

D2. Najdi filmy natočené před rokem 1940.

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

Výsledek:

NAZEV_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

Podmínky pomocí WHERE

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

Podmínky pomocí WHERE

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

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

Podmínky pomocí 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:

NAZEV_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

Podmínky pomocí 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:

NAZEV_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;
```

Podmínky pomocí 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:

NAZEV_F	REZISER	ROK
Nevinná	Bínovec V.	1939
Kristián	Frič M.	1939
Madla zpívá ...	Bínovec V.	1940
Ohnivě 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;
```

doplňek 1

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

doplňek 2

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

Operátory porovnání

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

Operátory porovnání

=	je rovno
<> nebo !=	není rovno
<	je menší
>	je větší
<=	je menší nebo rovno
>=	je větší 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**.

Operátory porovnání

=	je rovno
<> nebo !=	není rovno
<	je menší
>	je větší
<=	je menší nebo rovno
>=	je větší 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.

Operátory porovnání

=	je rovno
<> nebo !=	není rovno
<	je menší
>	je větší
<=	je menší nebo rovno
>=	je větší 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**.

Operace s hodnotou NULL

Operace s hodnotou NULL

Všechny datové typy (domény) mají “bottom” prvek **NULL**.

Operace s hodnotou NULL

Všechny datové typy (domény) mají “bottom” prvek **NULL**.

Má význam “UNKNOWN”, “NEUVEDENO”, “N/A”. Není to nula ani prázdný řetězec!

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:

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.

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.

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.

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:

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

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`

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

Spojení tabulek

SELECT – základní syntaxe

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

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

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

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

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

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

Alternativa 2 (Pozor na nechtěné atributy stejného jména!)

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

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

Alternativa 2 (Pozor na nechtěné atributy stejného jména!)

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

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

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

Alternativa 2 (Pozor na nechtěné atributy stejného jména!)

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

Alternativa 3

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

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

Alternativa 2 (Pozor na nechtěné atributy stejného jména!)

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

Alternativa 3

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

Dotazy nad více tabulkami

D4. Najdi režiséry, jejichž některé filmy jsou rezervovány.

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

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

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

Alternativa 2 (Pozor na nechtěné atributy stejného jména!)

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

Alternativa 3

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

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

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

- Chci-li zadat projekci:

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

- Chci-li zadat projekci:

Kvalifikace v klauzuli SELECT

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

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

- Chci-li zadat projekci:

Kvalifikace v klauzuli SELECT

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

Zavedení aliasů a jejich použití

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

- Chci-li zadat projekci:

Kvalifikace v klauzuli SELECT

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

Zavedení aliasů a jejich použití

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


Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

- Chci-li zadat projekci:

Kvalifikace v klauzuli SELECT

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

Zavedení aliasů a jejich použití

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

Kvalifikace, aliasy

D5. Najdi co hrají v kterých kinech.

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

- Chci-li zadat projekci:

Kvalifikace v klauzuli SELECT

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

Zavedení aliasů a jejich použití

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

Spojení tabulky se sebou samou

Spojení tabulky se sebou samou

D6. Najdi dvojice zákazníků, kteří mají stejnou adresu.

Spojení tabulky se sebou samou

D6. Najdi dvojice zákazníků, kteří mají stejnou adresu.

```
SELECT Soused1.rod_c AS první, soused2.rod_c AS druhý  
FROM Zakaznici Soused1 JOIN Zakaznici Soused2 USING (adresa)  
WHERE Soused1.rod_c < Soused2.rod_c;
```

Spojení tabulky se sebou samou

D6. Najdi dvojice zákazníků, kteří mají stejnou adresu.

```
SELECT Soused1.rod_c AS první, soused2.rod_c AS druhý  
FROM Zakaznici Soused1 JOIN Zakaznici Soused2 USING (adresa)  
WHERE Soused1.rod_c < Soused2.rod_c;
```

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

Dotazy se spojením a selekcí

Dotazy se spojením a selekcí

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

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

nebo

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

nebo

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

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

nebo

```
SELECT P.nazev_k, F.nazev_f
FROM Filmy F JOIN Predstaveni P ON (F.nazev_f=P.nazev_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

Výrazy v klauzuli SELECT

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

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)
```

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)
```

- Ize použít běžné oprátory jako: '/', '**', '-', '+'

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 operátory jako: '/', '**', '-', '+'
- obvyklá priorita operátorů, jinak lze použít závorky

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 operátory jako: '/', '**', '-', '+'
- obvyklá priorita operátorů, jinak lze použít závorky
- existuje řada vestavěných funkcí (specifické pro různé implementace)

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 operátory jako: '/', '*', '-', '+'
- obvyklá priorita operátorů, jinak lze použít závorky
- existuje řada vestavěných funkcí (specifické pro různé implementace)
- výrazy mohou být i nad řetězci a datумы

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

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

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 operátory jako: '/', '*', '-', '+'
- obvyklá priorita operátorů, jinak lze použít závorky
- existuje řada vestavěných funkcí (specifické pro různé implementace)
- výrazy mohou být i nad řetězci a datумы
- 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:

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 operátory jako: '/', '*', '-', '+'
- obvyklá priorita operátorů, jinak lze použít závorky
- existuje řada vestavěných funkcí (specifické pro různé implementace)
- výrazy mohou být i nad řetězci a datумы
- 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, (COALESCE(plat,0))/24.65 AS "Plat v Euro"  
FROM Zamestnanci  
WHERE rod_c IS NULL;
```

Obsah

1 Úvod do jazyka SQL

2 Dotazování v SQL

- Jednoduché dotazy
- Podmínky a operátory
- Spojení tabulek
- Aritmetické operace

3 Relační algebra a SQL

Relační algebra \rightarrow SELECT 1/2

$R(\varphi) [A_1, A_2, \dots, A_j]$

Relační algebra \rightarrow SELECT 1/2

$R(\varphi)$ $[A_1, A_2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R WHERE  $\varphi$ 
```

Relační algebra \rightarrow SELECT 1/2

$R(\varphi) [A_1, A_2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R WHERE  $\varphi$ 
```

$(R_1 \times R_2 \times \dots \times R_k)(\varphi) [A_1, A_2, \dots, A_j]$

Relační algebra \rightarrow SELECT 1/2

$R(\varphi) [A_1, A_2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R WHERE  $\varphi$ 
```

$(R_1 \times R_2 \times \dots \times R_k)(\varphi) [A_1, A_2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R_1, R_2, ..., R_k  
WHERE  $\varphi$ 
```

Relační algebra \rightarrow SELECT 1/2

$R(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R WHERE  $\varphi$ 
```

$(R1 \times R2 \times \dots \times R_k)(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1, R2, ..., R_k  
WHERE  $\varphi$ 
```

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1 CROSS JOIN ... R_k  
WHERE  $\varphi$ 
```

Relační algebra \rightarrow SELECT 1/2

$R(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R WHERE  $\varphi$ 
```

$(R1 \times R2 \times \dots \times R_k)(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1, R2, ..., R_k  
WHERE  $\varphi$ 
```

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1 CROSS JOIN ... R_k  
WHERE  $\varphi$ 
```

$(R1 * R2 * \dots * R_k)(\varphi) [A1, A2, \dots, A_j]$

Relační algebra \rightarrow SELECT 1/2

$R(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R WHERE  $\varphi$ 
```

$(R1 \times R2 \times \dots \times R_k)(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1, R2, ..., R_k  
WHERE  $\varphi$ 
```

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1 CROSS JOIN ... R_k  
WHERE  $\varphi$ 
```

$(R1 * R2 * \dots * R_k)(\varphi) [A1, A2, \dots, A_j]$

```
SELECT DISTINCT A1, A2, ..., A_j  
FROM R1 NATURAL JOIN R2 ... NATURAL JOIN R_k  
WHERE  $\varphi$ 
```

Relační algebra \rightarrow SELECT 2/2

$\{R1[t1 \ominus t2]R2\} (\varphi)[A1,A2,...,Aj]$

Relační algebra \rightarrow SELECT 2/2

$\{R1[t1 \Theta t2]R2\} (\varphi)[A1,A2,...,Aj]$

```
SELECT DISTINCT A1, A2,...,Aj  
FROM R1 JOIN R2 ON (R1.t1  $\Theta$  R2.t2)  
WHERE  $\varphi$ 
```


Relační algebra \rightarrow SELECT 2/2

$\{R1[t1 \Theta t2]R2\} (\varphi)[A1,A2,...,Aj]$

```
SELECT DISTINCT A1, A2,...,Aj  
FROM R1 JOIN R2 ON (R1.t1  $\Theta$  R2.t2)  
WHERE  $\varphi$ 
```

$\{R1 \times R2\}(R1.t1 \Theta R2.t2)(\varphi)[A1,A2,...,Aj]$

Relační algebra \rightarrow SELECT 2/2

$\{R1[t1 \Theta t2]R2\} (\varphi)[A1,A2,...,Aj]$

```
SELECT DISTINCT A1, A2,...,Aj  
FROM R1 JOIN R2 ON (R1.t1  $\Theta$  R2.t2)  
WHERE  $\varphi$ 
```

$\{R1 \times R2\}(R1.t1 \Theta R2.t2)(\varphi)[A1,A2,...,Aj]$
což je totéž jako

Relační algebra \rightarrow SELECT 2/2

$\{R1[t1 \Theta t2]R2\} (\varphi)[A1,A2,...,Aj]$

```
SELECT DISTINCT A1, A2,...,Aj  
FROM R1 JOIN R2 ON (R1.t1  $\Theta$  R2.t2)  
WHERE  $\varphi$ 
```

$\{R1 \times R2\}(R1.t1 \Theta R2.t2)(\varphi)[A1,A2,...,Aj]$

což je totéž jako

$R1 \times R2(R1.t1 \Theta R2.t2 \text{ and } \varphi)[A1,A2,...,Aj]$

Relační algebra \rightarrow SELECT 2/2

$\{R1[t1 \ominus t2]R2\} (\varphi)[A1,A2,...,Aj]$

```
SELECT DISTINCT A1, A2,...,Aj  
FROM R1 JOIN R2 ON (R1.t1  $\ominus$  R2.t2)  
WHERE  $\varphi$ 
```

$\{R1 \times R2\}(R1.t1 \ominus R2.t2)(\varphi)[A1,A2,...,Aj]$

což je totéž jako

$R1 \times R2(R1.t1 \ominus R2.t2 \text{ and } \varphi)[A1,A2,...,Aj]$

```
SELECT DISTINCT A1, A2,...,Aj  
FROM R1 CROSS JOIN R2  
WHERE (R1.t1  $\ominus$  R2.t2 and  $\varphi$ )
```

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 klauzuly SELECT).

Nějaký praktický trénink?

- prosemináře 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)