

2005-10-18

SQL:**Výhody používání jazyka SQL:**

- **Snížení ceny na zaškolení pracovníků** – tvůrci a uživatelé aplikací mohou snadněji přecházet od jednoho databázového prostředí k jinému.
- **Přenositelnost vytvořené aplikace** – aplikace vytvořená v konkrétním databázovém prostředí může být provozovaná v jiném prostředí bez změny, což umožňuje odladit danou aplikaci na PC a výsledek pak přenést na vyšší kategorii počítačů.
- **Délka života aplikace se zvyšuje** - v případě nutnosti je možno přejít na jiný databázový prostředek a prodloužit tak životnost dané aplikace.
- **Společný přístup k datům v heterogenním prostředí** – distribuované databáze provozované pod různým SŘBD.

Příkazy jazyka pro manipulaci dat DML – SELECT:

Základem dotazovacího jazyka SQL je příkaz SELECT.

Syntaxe příkazů SELECT:

```
SELECT [DISTINCT] { * | <seznam-sloupců>
FROM <jméno-tabulky> [, <jméno-tabulky>]...
[WHERE <výběrová-podmínka>]
[GROUP BY <seznam-sloupců>]
[HAVING <výběrová-podmínka>]]
[ORDER BY <jméno-sloupce>
[ASC | DESC] [, <jméno-sloupce> [ASC | DESC]]...]
```

Význam jednotlivých klauzulí příkazu SELECT:

- **GROUP BY** obsahuje definice výrazů, podle kterých budou řádky seskupeny
- **HAVING** omezuje souhrnné řádky definované pomocí GROUP BY, které splňují zadanou podmínku
- **ORDER BY** umožňuje seřazení řádků ve výsledku dle zadaných kritérií
- Klíčové slovo **DISTRICT** odstraňuje duplicitní řádky výběru

Projekce – výběr sloupců:

```
SQL> SELECT jmeno, bydliště FROM student;
```

JMENO	BYDLISTE
-----	-----
Karel Novák	Praha 1
Jiří Dvořák	Praha 9
Karel Franěk	Kolín 13
Josef Havlát	Praha 1

```
SQL>
```

Modifikovaný text v záhlaví vybraných sloupců:

Řešení pomocí klauzule AS příkazu SELECT.

```
SQL> SELECT jmeno AS "Jméno studenta" FROM student;
```

Jméno studenta

Karel Novák
Jiří Dvořák
Karel Franěk
Josef Havlát

```
SQL>
```

Problém činí délka vybíraného sloupce, která současně vymezuje délku nového (výstižnějšího) **ALIAS** pojmenování.

Výpis modifikovaných (agregovaných) údajů:

```
SQL> SELECT jmeno, bydliště, stipendium*1.2
```

AS "Stipendium v SK" FROM student;		
JMENO	BYDLISTE	Stipendium v SK
-----	-----	-----
Karel Novák	Praha 1	8400
Jiří Dvořák	Praha 9	0
Karel Franěk	Kolín 13	6000
Josef Havlát	Praha 1	2400
SQL>		

SELEKCE – výběr řádků (specifikováním podmínky):

SQL> SELECT * FROM student WHERE stipendium>2000;					
CIND	JMENO	BYDLISTE	DATNAR	ROCNIK	STIP
-----	-----	-----	-----	-----	-----
111/99	Karel Novák	Praha 1	10.01.80	3	7000
333/01	Karel Franěk	Kolín 13	12.03.85	3	5000
SQL>					

SELEKCE – odstranění duplicitních řádků:

SQL> SELECT bydliště FROM student;					
BYDLISTE					

Praha 1					
Praha 9					
Kolín 13					
Praha 1					
SQL> SELECT DISTINCT bydliště FROM student;					
BYDLISTE					

Kolín 13					
Praha 1					
Praha 9					
SQL>					

SELEKCE – predikáty ve výběrové podmínce:

SQL> SELECT * FROM student WHERE stipendium BETWEEN 5000 AND 7000;					
CIND	JMENO	BYDLISTE	DATNAR	ROCNIK	STIP
-----	-----	-----	-----	-----	-----
111/99	Karel Novák	Praha 1	10.01.80	3	7000
333/01	Karel Franěk	Kolín 13	12.03.85	3	5000
SQL>					

Výběry na podkladě testu náležitosti do množiny hodnot:

SQL> SELECT jmeno, bydliště FROM student WHERE bydliště IN (‘Praha 1’, ‘Praha 3’, ‘Praha 9’);	
JMENO	BYDLISTE
-----	-----
Karel Novák	Praha 1
Jiří Dvořák	Praha 9
Josef Havlát	Praha 1
SQL>	

Výběry na podkladě částečné specifikace textu:

- % **procento** specifikuje žádný, jeden nebo několik libovolných znaků (obdoba * v MS-DOS)
- _ **podtržítko** vyjadřuje právě jeden libovolný znak (obdoba ? v MS-DOS)

```
SQL> SELECT jmeno FROM student
WHERE jmeno LIKE 'K%';
JMENO
-----
Karel Novák
Karel Franěk
SQL>
```

Výběry na podkladě částečné specifikace textu:

```
SQL> SELECT jmeno FROM student
WHERE jmeno LIKE '%N%' OR jmeno LIKE '%D%';
JMENO
-----
Karel Novák
Jiří Dvořák
SQL>
```

Výběr s relačním operátorem pro nerovnost (!=):

```
SQL> SELECT * FROM student WHERE rocnik!='3';
CIND      JMENO      BYDLISTE    DATNAR      ROCNIK      STIP
-----
222/01     Jiří Dvořák  Praha 9     24.07.84     2           0
444/02     Josef Havlát  Praha 1     15.04.85     1          2000
SQL>
```

Výběry na podkladě hodnoty datumu:

```
SQL> SELECT * FROM student;
CIND      JMENO      BYDLISTE    DATNAR      ROCNIK      STIP
-----
111/99     Karel Novák  Praha 1     10.01.80     3          7000
222/01     Jiří Dvořák  Praha 9     24.07.84     2           0
333/01     Karel Franěk  Kolín 13    12.03.85     3          5000
444/02     Josef Havlát  Praha 1     15.04.85     1          2000
SQL> SELECT * FROM student WHERE (sysdate-datnar)>7300;
CIND      JMENO      BYDLISTE    DATNAR      ROCNIK      STIP
-----
111/99     Karel Novák  Praha 1     10.01.80     3          7000
222/01     Jiří Dvořák  Praha 9     24.07.84     2           0
SQL>
```

Třídění výstupů:

```
SQL> SELECT * FROM student ORDER BY bydliště ASC, rocnik DESC;
CIND      JMENO      BYDLISTE    DATNAR      ROCNIK      STIP
-----
333/01     Karel Franěk  Kolín 13    12.03.85     3          5000
111/99     Karel Novák  Praha 1     10.01.80     3          7000
444/02     Josef Havlát  Praha 1     15.04.85     1          2000
222/01     Jiří Dvořák  Praha 9     24.07.84     2           0
SQL>
```

Práce s více tabulkami:

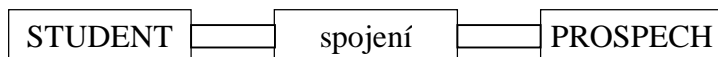
V tomto případě jde o operaci relační algebry – spojení, tj. spojení těch záznamů, které spolu „korespondují“. Pokud vybíráme stejnojmenné sloupce z těchto tabulek, je nutno použít úplnou identifikaci položek v tzv. „tečkové notaci“, tj. <jméno-tabulky>.<jméno-sloupce>.

```
SQL> SELECT * FROM prospech;
CIND      DATUM ZK    CPRED      ZNAMKA
```

```

-----
111/99    12.10.04    E0111    1
222/01    12.10.04    E0122    3
444/02    13.10.04    E0122    2
SQL>

```



jmeno, cpred, datumzk, znamka

```

SQL> SELECT jmeno AS "Jméno studenta",
          datumzk AS "Datum zkoušky",
          cpred AS "Číslo předmětu",
          znamka

```

```

FROM student,prospech

```

```

WHERE student.cind=prospech.cind;

```

```

Jméno studenta      Datum zkoušky      Číslo      ZNAMKA
-----
Karel Novák         12.10.04           E0111      1
Jiří Dvořák          12.10.04           E0122      3
Josef Havlát         13.10.04           E0122      2
SQL>

```

STUDENT:

CIND	JMENO	BYDLISTE	DATNAR
-----	-----	-----	-----
111/99	Karel Novák	Praha 1	10.01.80
222/01	Jiří Dvořák	Praha 9	24.07.84
333/01	Karel Franěk	Kolín 13	12.03.85
444/02	Josef Havlát	Praha 1	15.04.85

PROSPECH:

CIND	DATUMZK	CPRED	ZNAM
-----	-----	-----	-----
111/99	12.10.04	E0111	1
222/01	12.10.04	E0122	3
444/02	13.10.04	E0122	2

Jméno studenta	Datum zkoušky	Číslo	ZNAMKA
-----	-----	-----	-----
Karel Novák	12.10.04	E0111	1
Jiří Dvořák	12.10.04	E0122	3
Josef Havlát	13.10.04	E0122	2

Postup zpracování jednoduchého dotazu SELECT:

Relační databázový stroj ORACLE vyhodnocuje dotaz SELECT v následujícím pořadí:

- Zpracuje klauzuli **FROM**, tj. výběr požadované zdrojové tabulky či tabulek.
- Zpracuje klauzuli **WHERE**, tj. jsou procházeny jednotlivé řádky (záznamy) jeden za druhým a vybírány pouze ty, které splňují výběrovou podmínku.
- Realizuje klauzuli **SELECT**, tj. výběr požadovaných sloupců z tabulky, je-li uvedena klauzule **DISTINCT**, budou odstraněny duplicitní řádky.
- Byl-li specifikován požadavek seřazení výběru, bude vykonána klauzule **ORDER BY**.

Práce s více tabulkami:

```

SQL> SELECT * FROM prospech;

```

```

Jméno studenta      Datum zkoušky      Číslo      ZNAMKA
-----
111/99              12.10.04           E0111      1
222/01              12.10.04           E0122      3
444/02              13.10.04           E0122      2

```

```

SQL> SELECT * FROM predmety;
CPRED  NAZEV  KATEDRA

```

```

-----
E111      Účetnictví   KOF
E115      DTP          KII
E122      Databáze     KII
SQL>

```

```

SQL> SELECT jmeno,datumzk,prospech.cpred,nazev,znamka
      FROM student,prospech,predmety
      WHERE student.cind=prospech.cind AND prospech.cpred=predmety.cpred;
JMENO      DATUMZK      CPRED      NAZEV      ZNAMKA
-----
Karel Franěk 12.10.04    E0111     Účetnictví 1
Karel Novák  12.10.04    E0122     Databáze   3
Josef Havlát 13.10.04    E0122     Databáze   2
SQL>

```

Při spojování tabulek kladen na klíčové prvky požadavek – musí být stejného datového typu a stejné velikosti, ale nemusí mít stejný název.

```

SQL> SELECT jmeno,datumzk,prospech.cpred,nazev,katedra,znamka
      FROM student,prospech,predmety
      WHERE student.cind=prospech.cind
      AND prospech.cpred=predmety.cpred
      AND katedra='KII';
JMENO      DATUMZK      CPRED      NAZEV      KATEDRA      ZNAMKA
-----
Jiří Dvořák 12.10.04    E0122     Databáze   KII           3
Josef Havlát 13.10.04    E0122     Databáze   KII           2
SQL>

```

Práce se skupinami příkazů:

Přehled nepoužívanějších funkcí:

COUNT(*)	počet řádků ve skupině
COUNT(<jméno-sloupce>)	počet hodnot ve sloupci, počítají se pouze platné hodnoty (nikoliv NULL)
SUM(<jméno-sloupce>)	součet hodnot ve specifikovaném numerickém poli
MIN(<jméno-sloupce>)	nejmenší hodnota pole ve sloupci
MAX(<jméno-sloupce>)	největší hodnota pole ve sloupci
AVG(<jméno-sloupce>)	průměrná hodnota polí v číselném sloupci

Zjištění četnosti v rámci stanoveného kritéria:

Počet studentů bydlících v Praze 1:

```

SQL> SELECT COUNT(*) AS "Počet studentů z Prahy 1"
      FROM student
      WHERE bydliště='Praha 1';
Počet studentů z Prahy 1
-----
2
SQL>

```

Počet vyzkoušených studentů:

```

SQL> SELECT * FROM prospech;
CIND      DATUM ZK      CPRED      ZNAMKA
-----
111/99    12.10.04      E0111      1
222/01    12.10.04      E0122      3
444/02    13.10.04      E0122      2
111/99    13.10.04      E0122      3
SQL> SELECT COUNT(DISTINCT cind) FROM prospech;
COUNT(DISTINCTCIND)
-----

```

3
SQL>

SQL> **SELECT * FROM student;**

CIND	JMENO	BYDLISTE	DATNAR	ROCNIK	STIP
-----	-----	-----	-----	-----	-----
111/99	Karel Novák	Praha 1	10.01.80	3	7000
333/01	Karel Franěk	Kolín 13	12.03.85	3	5000
444/02	Josef Havlát	Praha 1	15.04.85	1	2000
222/01	Jiří Dvořák	Praha 9	24.07.84	2	0

SQL> **SELECT rocnik,COUNT(*) FROM student**

GROUP BY rocnik;

R COUNT(DISTINCTCIND)

-- -----

1 1

2 1

3 2

SQL>