

# Kapitola 2

## DML - Select 1

### 2.1 Zadanie cvičenia

1. Vypíšte všetky údaje o všetkých študentoch.
2. Vypíšte menný zoznam všetkých študentoch 2.ročníka.
3. Vypíšte menný zoznam študentov narodených v rokoch 1985–1989
4. Vypíšte menný zoznam študentov, ktorí študujú na detašovanom pracovisku Prievidza. ( druhý znak štúdinej skupiny je 'P')
5. Predchádzajúci výpis utried'te podľa priezviska.
6. Vypíšte menný zoznam študentov, ktorí študujú predmet BI06 a usporiadajte ich.
7. Vypíšte všetky kombinácie `prednasajuci` / `cis_predm`, ktoré sa nachádzajú v relácii `zap_predmety` tak, aby sa eliminovali duplikáty.
8. K predchádzajúcemu výpisu doplňte meno učiteľa a názov predmetu.
9. Vypíšte mená učiteľov, ktorí učia (sú prednášajúcimi) študentov druhého ročníka bakalárskeho štúdia (t.j. číslo štúijného odboru je z intervalu  $<100, 199>$  ) .
10. Vypíšte názvy predmetov študenta s priezviskom 'Balaz'.
11. Vypíšte počet riadkov v tabuľke `zap_predmety`.
12. Vypíšte počet študentov, ktorí majú zapísaný predmet 'Zaklady databazovych systemov'.
13. Vypíšte menný zoznam študentov spolu s dátumom narodenia.
14. Vypíšte počet kreditov študenta s osobným číslom 500439 za absolvované predmety.
15. Vypíšte menný zoznam študentov druhého ročníka spolu s ich vekom.

## 2.2 Pomôcky

### 2.2.1 Časti príkazu SELECT

```

SELECT      - čo chceme vypísať
FROM        - zoznam spojení tabuliek
WHERE       - štandardné podmienky
GROUP BY    - skupiny pre výpočet agregračných funkcií
HAVING      - dodatočné podmienky s agregračnými funkciami
ORDER BY    - triedenie

```

### 2.2.2 Potlačenie duplícít

```
Select DISTINCT zoznam_stlpcov
```

```

1 select st_skupina
2   from student
3* order by st_skupina
SQL> /

```

```

ST_SKU
-----
5PI011
5PI021
5PI021
5ZI000
5ZI000
5ZI000
5ZI012
5ZI021
5ZI030
5ZI031
5ZIA21
5ZM021
5ZM021
5ZM023
5ZM031
...

```

```

1 select DISTINCT st_skupina
2   from student
3* order by st_skupina
SQL> /

```

```

ST_SKU
-----
5PI011
5PI021
5ZI000
5ZI012
5ZI021
5ZI030
5ZI031
5ZIA21
5ZM021
5ZM023
5ZM031
5ZN021
5ZP012
5ZP021
5ZR013
...

```

### 2.2.3 Aliasy a plné mená stĺpcov

#### 1. Alias tabuľky a pomenovanie stĺpca

Princíp	Príklad
<b>Ak nie je zavedený alias tabuľky</b>	
nazov_stlpca	meno
nazov_tabulky.nazov_stlpca	os_udaje.meno
schema.nazov_tabulky.nazov_stlpca	vajsova.os_udaje.meno
<b>Ak je zavedený alias tabuľky</b>	
nazov_stlpca	meno
alias_tabulky.nazov_stlpca	ou.meno

(a) Bez aliasu tabuľky

```
select meno
  from os_udaje;
```

```
select os_udaje.meno
  from os_udaje;
```

```
select vajsova.os_udaje.meno
  from vajsova.os_udaje;
```

(b) S aliasom tabuľky

```
select meno
  from os_udaje ou;
```

```
select ou.meno
  from os_udaje ou;
```

## 2. Alias stĺpca

V časti Select za výrazom môžeme dopísať alias stĺpca, ktorý sa stane názvom stĺpca vo výsledku. Tento alias nie je možné použiť v inej časti selectu.

```
SELECT vyraz AS alias_stlpca
```

```
SQL> select os_cislo, rocnik + 1 AS novy_rocnik
      2  from student;
```

OS_CISLO	NOVY_ROCNIK
501512	4
501469	3
501414	3
550545	2
550020	4
501567	1
501319	3

```
      1  select os_cislo, rocnik + 1 AS novy_rocnik
      2  from student
      3* where novy_rocnik = 1
SQL> /
where novy_rocnik = 1
      *
ERROR at line 3:
ORA-00904: "NOVY_ROCNIK": invalid identifier
```

```
      1  create table pom_tab
      2  AS
      3  select os_cislo, rocnik + 1 AS novy_rocnik
      4* from student
SQL> /
```

Table created.

```
SQL> desc pom_tab
```

Name	Null?	Type
OS_CISLO	NOT NULL	NUMBER(38)
NOVY_ROCNIK		NUMBER

### 2.2.4 Spojenia

```
tabulka_A {LEFT | RIGHT | FULL } JOIN tabulka_B
{ ON ( podmienky_spojenia)
  |
  USING( zoznam_stlpcov)
}
```

- **USING** - je možné použiť len v prípade, že atribúty primárneho a cudzieho kľúča majú rovnaké meno. Pri stĺpcoch použitých v klauzule USING nie je možné používať alias tabuľky.

```
tabulka_A JOIN tabulka_B USING ( zoznam_stlpcov )
```

```
SELECT rod_cislo, st.os_cislo, ou.meno, ou.priezvisko
FROM os_udaje ou JOIN student st USING ( rod_cislo );
```

```
SELECT rod_cislo, st.os_cislo, ou.meno, ou.priezvisko,
       sto.popis_odboru, sto.popis_zamerania
FROM os_udaje ou JOIN student st USING ( rod_cislo )
     JOIN st_odbory sto USING ( st_odbor, st_zameranie );
```

- **ON** - je možné využiť vždy. V tomto prípade je nutné používať alias tabuľky.

```
tabulka_A JOIN tabulka_B ON ( podmienky_spojenia )
```

```
SELECT ou.rod_cislo, st.os_cislo, ou.meno, ou.priezvisko
FROM os_udaje ou JOIN student st
     ON ( ou.rod_cislo = st.rod_cislo );
```

```
SELECT ou.rod_cislo, st.os_cislo, ou.meno, ou.priezvisko,
       sto.popis_odboru, sto.popis_zamerania
FROM os_udaje ou JOIN student st ON ( ou.rod_cislo = st.rod_cislo )
     JOIN st_odbory sto ON ( st.st_odbor = sto.st_odbor
                          AND st.st_zameranie = sto.st_zameranie );
```

- Pomocou zátvoriek je možné zmeniť poradie spojení.

```
SELECT st.os_cislo, ou.meno, ou.priezvisko
FROM os_udaje ou JOIN student st
     ON ( ou.rod_cislo = st.rod_cislo)
     JOIN zap_predmety zp ON (zp.os_cislo = st.os_cislo);
```

```
SELECT st.os_cislo, ou.meno, ou.priezvisko
FROM os_udaje ou JOIN
     ( student st JOIN zap_predmety zp ON (zp.os_cislo = st.os_cislo) )
     ON ( ou.rod_cislo = st.rod_cislo);
```

### 2.2.5 Triedenie

V časti **ORDER BY** doplníme zoznam stĺpcov v poradí v akom chceme aplikovať triedenie spolu so smerom triedenia (**ASC**, **DESC**). Pričom usporiadanie od najmensej hodnoty po najväčiu je defaultné triedenie ( **ASC** ).

```
ORDER BY stlpec { ASC | DESC } [ , ... ]
```

```
select meno, priezvisko, rocnik
from os_udaje JOIN student USING (rod_cislo)
ORDER BY rocnik DESC, priezvisko;
```

MENO	PRIEZVISKO	R
-----	-----	-
Peter	Kapustny	3
Jan	Krnac	3
Peter	Novak	3
Branislav	Balaz	2
Dusan	Bucany	2
Jaroslav	Cipak	2
Marek	Durica	2
Alojz	Gazo	2
Miroslav	Gmuca	2
Lubomir	Kominek	2
Rastislav	Kontros	2
...		

### 2.2.6 Podmienky

- vyraz1
 

>
>=
=
<>
!=
<
<=

 vyraz2
- vyraz1 **BETWEEN** min\_hodnota **AND** max\_hodnota
- vyraz **IS** [NOT] **NULL**
- vyraz [NOT] **IN** ( mnozina )
- [ NOT] **EXISTS** ( mnozina )
- podmienka1 **AND** podmienka2 | podmienka1 **OR** podmienka2

## 2.3 Funkcie

### 1. Štandardné funkcie

- Podľa dátových typov: **TO\_CHAR**, **TO\_DATE**, **ROUND**, **SUBSTR**, **LENGTH**, ...
- Operátor **LIKE**
- podmienky sú vo **WHERE**

## 2. Agregáčné funkcie

- COUNT, SUM, MIN, MAX, AVG
- podmienky sú v HAVING
- spojené väčšinou s GROUP BY

### 2.3.1 Tabuľka dual

V DBS Oracle existuje tabuľka dual, na ktorej nie je možné nič meniť. Používa sa keď nepotrebujeme žiadne dáta z tabuliek, len potrebujeme zísť hodnotu nejakej funkcie pomocou selectu.

```
SQL> desc dual
```

Name	Null?	Type
DUMMY		VARCHAR2(1)

```
SQL> select * from dual;
```

```
D
-
X
```

### 2.3.2 Špeciálne funkcie

- **DECODE** - prekódovanie

```
DECODE ( vyraz, IF-hodnota, THEN-hodnota, [ ,..] , ELSE-hodnota )
```

```
SQL> select os_cislo
```

```
1      , decode ( st.stav, 'S', 'student', 'K','ukonceny', 'P', 'preruseny', 'vyluceny') a
2  from student st;
```

```
OS_CISLO STAV
-----
501512 student
501469 student
501414 student
550545 preruseny
550020 student
501567 vyluceny
501319 student
501555 student
501402 student
501096 vyluceny
501103 ukonceny
501333 student
...
```

### 2.3.3 Funkcie pre číselné dátové typy - NUMBER

- **ROUND** - zaokrúhlenie

```
ROUND( cislo, [ pocet_des_miest] )
```

```
SQL> select round(2.5), round(21.3), round(13.6) from dual;
```

```
ROUND(2.5) ROUND(21.3) ROUND(13.6)
-----
          3          21          14
```

```
SQL> select round(13.2458878, 2 ) from dual;
```

```
ROUND(13.2458878,2)
-----
          13.25
```

```
SQL> select round(657981.124, -2) from dual;
```

```
ROUND(657981.124,-2)
-----
        658000
```

- **TRUNC** - useknutie podľa parametra

```
TRUNC( cislo, [presnost])
```

```
SQL> select trunc( 153.3 ), trunc (169.9)
       2 from dual;
```

```
TRUNC(153.3) TRUNC(169.9)
-----
          153          169
```

```
SQL> select trunc( 15.26589, 1), trunc (15.26589, - 1 )
       2 from dual;
```

```
TRUNC(15.26589,1) TRUNC(15.26589,-1)
-----
          15.2          10
```

- **MOD** - zvyšok po delení

```
MOD( delenec, delitel )
```

```
SQL> select mod(5,2) ,mod(16,4) from dual;
```

```
MOD(5,2) MOD(16,4)
-----
          1          0
```

### 2.3.4 Funkcie pre reťazcové dátové typy - CHAR, VARCHAR

- **SUBSTR** - podreťazec

```
SUBSTR(ret'azca, pozícia, dĺžka)
```

*kde*

- **pozícia** začína od 1.
- V prípade zápornej pozície, pozícia je odrátaná od konca reťazca.
- Ak nie je udaná dĺžka reťazca, vypíše sa až do konca reťazca.

```
SQL> select substr( '5ZI021', 2, 1 ) as pracovisko
      2* from dual;
```

```
P
-
Z
```

```
SQL> select st_skupina, substr(st_skupina, 5,2) from student;
```

```
ST_SKU SUBSTR(S
-----
5ZI031 31
5ZP021 21
5ZI021 21
5ZI012 12
5ZM031 31
5ZI000 00
```

```
SQL> select st_skupina, substr(st_skupina, -2) from student
```

```
ST_SKU SUBSTR(S
-----
5ZI031 31
5ZP021 21
5ZI021 21
5ZI012 12
5ZM031 31
5ZI000 00
```

- **LENGTH** - dĺžka reťazca

```
LENGTH(ret'azec)
```

```
SQL> select priezvisko , length(priezvisko) from os_udaje;
```

PRIEZVISKO	LENGTH(PRIEZVISKO)
-----	-----
Novak	5
Steinmuller	11
Toth	4
Ratroch	7
Biely	5
Balaz	5
Kapustny	8

- **CONCAT**, operátor || - spojenie reťazcov

```
CONCAT(ret'azec1, ret'azec2 )
```



```
SQL> select concat( meno, priezvisko )
      2 from os_udaje;
```

```
CONCAT(MENO,PRIEZVISKO)
```

```
-----
```

```
PeterNovak
StanislavSteinmuller
JanosToth
MarekRatroch
BohuslavBiely
BranislavBalaz
PeterKapustny
MarekDurica
MartinKluciar
LukasSatrapa
JanKrnac
JurajPapun
```

```
SQL> select concat(concat(meno, ' ') , priezvisko ) as cele_meno
      2 from os_udaje;
```

```
CELE_MENO
```

```
-----
```

```
Peter Novak
Stanislav Steinmuller
Janos Toth
Marek Ratroch
Bohuslav Biely
Branislav Balaz
Peter Kapustny
Marek Durica
Martin Kluciar
Lukas Satrapa
Jan Krnac
Juraj Papun
```

Reťazce je možné spojiť nielen pomocou funkcie CONCAT, ale aj pomocou operátora ||.

!

```
ret'azec1 || ret'azec2 [ || retazec3 ...]
```

```
SQL> select  meno || ' ' || priezvisko as cele_meno
      2 from os_udaje;
```

```
CELE_MENO
```

```
-----
```

```
Peter Novak
Stanislav Steinmuller
Janos Toth
Marek Ratroch
Bohuslav Biely
Branislav Balaz
Peter Kapustny
Marek Durica
```

Martin Kluciar  
 Lukas Satrapa  
 Jan Krnac  
 Juraj Papun

- **UPPER** - prevod na veľké písmo

```
UPPER(ret'azec)
```

```
SQL> select priezvisko, upper(priezvisko) from os_udaje;
```

PRIEZVISKO	UPPER(PRIEZVISKO)
Novak	NOVAK
Steinmuller	STEINMULLER
Toth	TOTH
Ratroch	RATROCH
Biely	BIELY
Balaz	BALAZ
Kapustny	KAPUSTNY
Durica	DURICA
Kluciar	KLUCIAR
Satrapa	SATRAPA

- **LOWER** - prevod na malé písmo

```
LOWER(ret'azec)
```

```
SQL> select priezvisko, lower(priezvisko) from os_udaje;
```

PRIEZVISKO	LOWER(PRIEZVISKO)
Novak	novak
Steinmuller	steinmuller
Toth	toth
Ratroch	ratroch
Biely	biely
Balaz	balaz
Kapustny	kapustny
Durica	durica
Kluciar	kluciar

- Operátor **LIKE**

```
stlpec LIKE vzor
```

Vzor	Význam
%	ľubovoľný počet znakov
-	jeden znak

```
select meno, priezvisko
from os_udaje
where priezvisko LIKE 'K%';
```

MENO	PRIEZVSKO
Peter	Kapustny
Martin	Kluciar
Jan	Krnac
Rastislav	Kontros
Rudolf	Kovac
Lubomir	Kominek
Peter	Korenciak

```
-- znak 3 na ľubovoľnej pozícii
SQL> select distinct st_skupina
2   from student
3   where st_skupina like '%3%';
```

```
ST_SKU
-----
5ZI031
5ZI030
5ZM023
5ZSN23
5ZM031
5ZR032
5ZR013
```

```
-- znak 3 po štyroch ľubovoľných znakoch
SQL> select distinct st_skupina
2   from student
3   where st_skupina like '____3%';
```

```
ST_SKU
-----
5ZI031
5ZI030
5ZM031
5ZM031
5ZR032
```

### 2.3.5 Funkcie pre dátumové dátové typy - DATE

- **SYSDATE** - aktuálny dátum a čas

```
SYSDATE
```

```
SQL> select sysdate from dual;
```

```
SYSDATE
-----
04-FEB-15
```

- **TO\_CHAR** - prevod dátumu na reťazec

```
TO_CHAR( datum, format)
```

```
SQL> select to_char( sysdate, 'DD.MM.YYYY') from dual;
```

```
TO_CHAR(SY
-----
10.02.2015
```

```
SQL> select to_char(sysdate, 'DD.MM.YYYY HH24:MI:SS') as datum
2   from dual;
```

```
DATUM
-----
04.02.2015 14:12:02
```

```
SQL> select sysdate, to_char( sysdate, 'D') , to_char( sysdate, 'DD') from dual
      2 from dual;
```

```
SYSDATE    T TO
-----  - --
10-FEB-15  3 10
```

Formát	Význam
D	Poradové číslo dňa v týždni
DD	Číslo dňa v mesiaci
MM	Číslo mesiaca
YYYY	Rok - štyri číslice
RRRR	Rok - štyri číslice
HH	Hodiny - 12 hodinový formát
HH24	Hodiny - 24 hodinový formát
MI	Minúty
SS	Sekundy

- **TO\_DATE** - prevod z reťazca na dátum

```
TO_DATE(ret'azec, formát)
```

```
SQL> select to_date ( '31.12.2015', 'DD.MM.YYYY') from dual;
```

```
TO_DATE('
-----
31-DEC-15
```

- **ADD\_MONTHS** - pridaj k dátumu zadany počet mesiacov

```
ADD_MONTHS(dátum, pocet_mesiacov)
```

```
SQL> select sysdate, add_months( sysdate, 2), add_months(sysdate, -2) from dual;
```

```
SYSDATE    ADD_MONTH ADD_MONTH
-----  -----  -----
10-FEB-15  10-APR-15  10-DEC-14
```

- **LAST\_DAY** - posledný deň v mesiaci

```
LAST_DAY(dátum)
```

```
SQL> select sysdate, last_day( sysdate ) from dual;
```

```
SYSDATE    LAST_DAY(
-----  -----
10-FEB-15  28-FEB-15
```

```
SQL> select last_day( to_date( '15.5.2015', 'DD.MM.YYYY')) from dual;
```

```
LAST_DAY(
-----
31-MAY-15
```

- **TRUNC** - useknutie dátumu

```
TRUNC ( datum, [ format ] )
```

```
1 select to_char( sysdate , 'DD.MM.YYYY HH24:MI:SS') as d1,
2       to_char( trunc(sysdate), 'DD.MM.YYYY HH24:MI:SS') as d2,
3       to_char( trunc(sysdate, 'DD'), 'DD.MM.YYYY HH24:MI:SS') as d3
4* from dual
SQL> /
```

D1	D2	D3
10.02.2015 13:16:34	10.02.2015 00:00:00	10.02.2015 00:00:00

```
1 select to_char( sysdate , 'DD.MM.YYYY HH24:MI:SS') as d1,
2       to_char( trunc(sysdate, 'MI'), 'DD.MM.YYYY HH24:MI:SS') as d4,
3       to_char( trunc(sysdate, 'YYYY'), 'DD.MM.YYYY HH24:MI:SS') as d5
4* from dual
SQL> /
```

D1	D4	D5
10.02.2015 13:17:17	10.02.2015 13:17:00	01.01.2015 00:00:00

- **EXTRACT** - získanie časti dátumu

```
EXTRACT ( co FROM datum)
```

```
1* select sysdate, extract ( month from sysdate ) from dual
SQL> /
```

SYSDATE	EXTRACT(MONTHFROMSYSDATE)
10-FEB-15	2

## 2.3.6 NULL hodnoty

- Podmienky s NULL hodnotami

```
- vyraz IS NULL
```

```
SQL> select os_cislo, meno, priezvisko
2 from os_udaje ou JOIN student st USING ( rod_cislo )
3 where ukoncenie IS NULL;
```

OS_CISLO	MENO	PRIEZVISKO
550020	Peter	Kapustny
501333	Martin	Kluciar
501402	Marek	Durica
501555	Marek	Durica
501319	Branislav	Balaz
501469	Stanislav	Steinmuller

– **vyraz IS NOT NULL**

```
1 select os_cislo, meno, priezvisko
2 from os_udaje ou JOIN student st USING ( rod_cislo )
3* where ukoncenie IS NOT NULL
SQL> /
```

OS_CISLO	MENO	PRIEZVISKO
501103	Martin	Kluciar
501096	Martin	Kluciar
500438	Miroslav	Gmuca
500433	Juraj	Tuma
500422	Martin	Skuta
500428	Peter	Stelbasky
500432	Zdenko	Olzbut
501567	Branislav	Balaz
500429	Peter	Minarik

– S NULL hodnotami nikdy nepoužívajte relačné operátory, pretože výsledkom bude vždy FALSE.

```
1 select os_cislo, meno, priezvisko
2 from os_udaje ou JOIN student st USING ( rod_cislo )
3* where ukoncenie = NULL
SQL> /
```

no rows selected

```
1 select os_cislo, meno, priezvisko
2 from os_udaje ou JOIN student st USING ( rod_cislo )
3* where ukoncenie <> NULL
SQL> /
```

no rows selected

- Funkcia NVL - nahradenie null hodnoty

```
SQL> select vysledok, nvl(vysledok, 'F') from zap_predmety;
```

```
V N
- -
C C
F
F
A A
C C
D D
F
F
A A
```

- COALESCE- nahradenie null hodnoty

```
coalesce (vyraz1, vyraz2 [,...] )
```

```
SQL> select vysledok, coalesce(vysledok, 'F') from zap_predmety;
```

```
V C
- -
C C
  F
  F
A A
C C
D D
  F
  F
A A
```

### 2.3.7 Množiny

vyráz `NOT IN ( mnozina )`

alebo

`NOT EXISTS ( vymenovana_mnozina )`

- Vymenovaná množina

```
select meno, priezvisko
from os_udaje JOIN student USING (rod_cislo)
where rocnik IN ( 1, 2);
```

```
select meno, priezvisko
from os_udaje JOIN student USING (rod_cislo)
where rocnik NOT IN ( 1, 2);
```

- Množina udaná pomocou vnoreného selectu.

– IN

```
select ou.meno, ou.priezvisko
from os_udaje ou JOIN student st USING (rod_cislo)
where st.os_cislo NOT IN
      ( select zp.os_cislo
        from zap_predmety zp
        where zp.cis_predm = 'BI06'
      );
```

– EXISTS

```
select ou.meno, ou.priezvisko
from os_udaje ou JOIN student st USING (rod_cislo)
where NOT EXISTS
      ( select 'x'
        from zap_predmety zp
        where zp.cis_predm = 'BI06'
          and zp.os_cislo = st.os_cislo
      );
```

### 2.3.8 Agregáčné funkcie

- **Min** - minimum

```
Min( vyraz )
```

```
SQL> select min(os_cislo) from student;
```

```
MIN(OS_CISLO)
-----
          500422
```

- **Max** - maximum

```
Max( vyraz )
```

```
SQL> select max(os_cislo) from student;
```

```
MAX(OS_CISLO)
-----
          550945
```

- **Sum** - súčet

```
Sum( vyraz )
```

```
SQL> select sum(ects)
2   from zap_predmety
3   where os_cislo = 501103
4   ;
```

```
SUM(ECTS)
-----
          19
```

- **Avg** - priemer

```
Avg( vyraz )
```

```
SQL> select avg(ects)
2   from zap_predmety;
```

```
AVG(ECTS)
-----
4.99793388
```

- **Count** - počet riadkov

– Počet všetkých riadkov skupiny

```
Count(*)
```



```
SQL> select count(*) from zap_predmety;
```

```
      COUNT(*)  
-----  
         484
```

- Počet riadkov skupiny, kde `stlpec` nemá NULL hodnotu.

```
Count(stlpec)
```

```
SQL> select count(vysledok) from zap_predmety;
```

```
      COUNT(VYSLEDOK)  
-----  
         295
```

- Počet unikátnych hodnôt `stlpca` v skupine.

```
Count(DISTINCT stlpec)
```

```
SQL> select count(DISTINCT vysledok) from zap_predmety;
```

```
      COUNT(DISTINCTVYSLEDOK)  
-----  
         6
```

### 2.3.9 Agregáčné funkcie a group by

Pravidlá pre GROUP BY:

1. Pokiaľ chceme vypísať len jednu hodnotu agregáčnej funkcie, nie je potrebné group by.
2. Do GROUP BY je potrebné dať všetko, čo je selectnuté, okrem agregáčnej funkcie.
3. Okrem toho je možné doplniť aj niečo iné.
4. Aby sme zabezpečili správnosť výsledkov, nezabudnime do GROUP BY doplniť primárny kľúč toho, za ktorú skupinu vypočítavame agregáčnú funkciu. (Aby sme napr. nespočítavali menovcov dokopy)