

DML - Select 2

7.1 Zadanie cvičenia

1. Vypíšte vek najstaršieho študenta v čase ukončenia štúdia. (použite stĺpec **ukoncenie** z tabuľky **student**)
2. Vypíšte menný zoznam študentov, ktorí majú budúci mesiac narodeniny. (použite systémový dátum)
3. Ku každému predmetu vypíšte nasledovnú štatistiku - minimálnu, maximálnu známku, celkový počet študentov v školskom roku 2008.
4. Vypíšte menný zoznam študentov, ktorí majú priemer menší ako 3.(Známku prevedte najprv na koeficient, nevyplnenú známku počítajte za 4.)
5. Vypíšte názvy predmetov, ktoré majú zapísané aspoň štyria študenti v školskom roku 2006.
6. Vypíšte menný zoznam študentov, ktorí opakovali nejaký predmet.
7. Vypíšte menný zoznam študentov spolu s počtom predmetov, ktoré študujú v školskom roku 2008 .
8. Vypíšte zoznam predmetov, ktoré nemá nikto zapísané v školskom roku 2006, ale sú v tabuľke **st_program** daného roku.
9. Vypíšte koľko dní ubehlo medzi zápočtom a skúškou u tých predmetov študentov, ktorí majú zadaný aj zápočet aj skúšku.
10. Vypíšte zoznam študentov a ich predmety, u ktorých prešiel aspoň jeden mesiac medzi zápočtom a skúškou.
11. Vypíšte menný zoznam študent, ktorí nikdy neopakovali žiaden predmet.
12. (a) Vypíšte celkový počet študentov.
(b) Ku každému ročníku vypíšte počet študentov.
(c) Ku každému názvu odboru vypíšte počet študentov.
13. Vypíšte zoznam voliteľných predmetov študenta s osobným číslom 500439. (Voliteľný predmet je každý predmet, ktorý nie je povinný alebo povinne voliteľný podľa odboru a zamerania študenta. Pri povinných predmetoch je vždy **st_program.st_zameranie** = 0, pričom platia pre všetky zamerania.)



- Povinnosť predmetov v danom školskom roku pre jednotlivé odbory a zamerania môžeme nájsť v stĺpci `st_program.typ_povin`:
 - P - povinný
 - S - povinne voliteľný
 - V - voliteľný
- Predmet, ktorý je platný pre všetky zamerania daného odboru bude v tabuľke `st_program` pre daný odbor, typ povinnosti, v zadanom školskom roku len JEDEN RIADOK a v stĺpci `st_program.st_zameranie` bude hodnota 0

14. Vypíšte všetky názvy všetkých povinných predmetov z roku 2008 a počet študentov, ktoré majú tento predmet. Ak ho nemá zapísaný ani jeden študent vypíšte aspoň názov predmetu.

7.2 Doplnujúce úlohy z modelu LETISKO

1. Select:

- (a) Vypíšte menný zoznam cestujúcich, ktorí zrušili nejakú letenku minulý mesiac.
- (b) Ku každému letu vypíšte presný dátum a čas odletu a id_krajiny odletu a príletu.
- (c) Vypíšte čísla letov a presný čas odletu všetkých letov, ktoré letia budúci mesiac z Bratislavy do mesta Rome.
- (d) Pre ľubovoľne zvolený let vypíšte voľnú kapacitu pre jednotlivé triedy.
- (e) Vypíšte osobné údaje cestujúcich, ktorí mali v jednom čase zakúpené viaceré letenky v aktuálnom roku. (z pohľadu samotného letu - t.j. v skutočnosti mohli využiť len jednu letenku.)
- (f) Vypíšte osobné údaje cestujúcich, ktorí mali v jednom čase zakúpené viaceré letenky toho istého letu v aktuálnom roku. (z pohľadu samotného letu - t.j. v skutočnosti mohli využiť len jednu letenku.)
- (g) Vypíšte zoznam aktuálnych zamestnancov spoločnosti Ryanair.
- (h) Vypíšte všetky lety, ktoré budú budúci mesiac do Francúzska.
- (i) K jednotlivým leteckým spoločnostiam vypíšte zarobenú sumu za nezrušené letenky minulého roka.
- (j) K jednotlivým leteckým spoločnostiam a mesiacom minulého roka vypíšte počet zakúpených (nezrušených) leteniek.
- (k) Vypíšte zoznam lietadiel spoločnosti Ryanair, ktoré sú voľné na budúci štvrtok.
- (l) Vypíšte zoznam letov, ktoré neboli využité ani na 20% kapacity letu.
- (m) Vypíšte zoznam cestujúcich, ktorý leteli viac ako 3 krát počas posledného pol roka.
- (n) Vypíšte zoznam cestujúcich, u ktorých medzi poslednými dvoma letmi prešli viac ako 2 mesiace.
- (o) Vypíšte zoznam letov, ktoré boli využité na viac ako 30% kapacity letu.
- (p) Vypíšte koľko zaplatili jednotliví cestujúci za letenky minulého roka.
- (q) K jednotlivým rokom vypíšte najčastejšie vybratú destináciu podľa počtu realizovaných letov.
- (r) K jednotlivým rokom vypíšte najčastejšie vybratú destináciu podľa počtu realizovaných leteniek.
- (s) K jednotlivým mesiacom minulého roka vypíšte najčastejšie lietanú trasu podľa počtu realizovaných letov.
- (t) K jednotlivým mesiacom minulého roka vypíšte najčastejšie lietanú trasu podľa počtu predaných leteniek.
- (u) K jednotlivým krajinám vypíšte počet cestujúcich z danej krajiny, ak nie je nikto, tak vypíšte aspoň názov krajiny.

2. Insert:

- (a) Osoba s priezviskom Kubica si chce objednať letenku prvej triedy na let 37235. Cenu nastavte na 130% originálnej ceny letenky.
- (b) Vytvorte 3 nové lety z Bratislavy do Milána, všetky potrebné údaje si vymylite.

3. Update:

- (a) Znížte cenu všetkých letov o 10% letov, kde je už len menej ako 14 do odletu, ale lietadlo je voľné.
- (b) Prebookujte let z letiska ZFQ do letiska VCE cestujúceho s menom Pavol a priezviskom Bablena z dňa 06.06.2015 na najbližší voľný termín.
- (c) Zmeňte priezvisko zamestnankyne s id_zamestnanca 135 na Vesela.

4. Delete:

- (a) Vymažte všetky údaje o zrušených letenkách starších ako 1 rok.
- (b) Vymažte všetky údaje o letoch starších ako 5 rokov.
- (c) Vymažte údaje o všetkých lietadlách typu 'Fokker 100', pričom nezrušíte žiaden let.
- (d) Vymažte duplicity - ponechajte iba prvú letenku, ak si cestujúci zakúpil viaceré letenky toho istého letu.

7.3 Pomôcky

7.3.1 Agregáčné funkcie

1. Min - minimum

```
Min(vyraz)
```

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

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

2. Max - maximum

```
Max(vyraz)
```

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

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

3. Sum - súčet

```
Sum(vyraz)
```

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

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

4. Avg - priemer

```
Avg(vyraz)
```

5. Count - počet riadkov

```
Count( * | [DISTINCT] vyraz )
```

- 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
```

7.3.2 GROUP BY

7.3.3 Zásady pre group by

1. Ak nevypisujeme nič okrem hodnôt agregáčnych funkcií, nie je potrebné GROUP BY.
2. Ak vypisujeme nejaký stlpec, či už priamo, alebo vo výraze, okrem agregáčnej funkcie, daný stlpec musí byť v GROUP BY.
3. V prípade potreby môžeme do GROUP BY doplniť ďalšie stĺpce (najčastejšie primárny kľúč, aby boli výsledky správne).

7.3.4 Count

1. Bez group by - počet riadkov tabuľky.

```
select COUNT(*) from zap_predmety;
```

Výsledok

11

os_cislo	cis_predm	skrok	vysledok	UC
501319	BL14	2005	C	EX001
501319	BE13	2005	D	KMT01
501319	BL11	2005		EX001
501319	BL11	2006	C	EX001
501201	IM16	2002	C	KMM01
501201	II08	2003		KDS01
501345	BA12	2002	C	KI003
501345	IS04	2003	D	KI001
550123	II07	2001		KI001
550123	IA07	2001	C	KMM02
550123	II17	2002	D	KI002

Počet všetkých riadkov v tabuľke

2. Zmena GROUP BY - iné výsledky.

```
select os_cislo, count(*) from zap_predmety
GROUP BY os_cislo;
```

Výsledok

501319 4

501201 2

501345 2

550123 3

os_cislo	cis_predm	skrok	vysledok	UC
501319	BL14	2005	C	EX001
501319	BE13	2005	D	KMT01
501319	BL11	2005		EX001
501319	BL11	2006	C	EX001
501201	IM16	2002	C	KMM01
501201	II08	2003		KDS01
501345	BA12	2002	C	KI003
501345	IS04	2003	D	KI001
550123	II07	2001		KI001
550123	IA07	2001	C	KMM02
550123	II17	2002	D	KI002

Počet zapísaných predmetov študenta

3. Zmena GROUP BY - iné výsledky.

```
SELECT os_cislo, skrok, count(*) FROM zap_predmety
GROUP BY os_cislo, skrok;
```

Výsledok

550123 2002 1

501319 2005 3

501319 2006 1

501201 2002 1

501201 2003 1

501345 2002 1

501345 2003 1

550123 2001 2

os_cislo	cis_predm	skrok	vysledok	UC
501319	BL14	2005	C	EX001
501319	BE13	2005	D	KMT01
501319	BL11	2005		EX001
501319	BL11	2006	C	EX001
501201	IM16	2002	C	KMM01
501201	II08	2003		KDS01
501345	BA12	2002	C	KI003
501345	IS04	2003	D	KI001
550123	II07	2001		KI001
550123	IA07	2001	C	KMM02
550123	II17	2002	D	KI002

Počet predmetov študenta v jednom školskom roku.

4. Zmena GROUP BY - iné výsledky.

```
SELECT count(*) FROM zap_predmety
GROUP BY uc;
```

Výsledok

3
1
2
1
1
1
1
1
1

os_cislo	cis_predm	skrok	vysledok	UC
501319	BL14	2005	C	EX001
501319	BL11	2005		EX001
501319	BL11	2006	C	EX001
501201	II08	2003		KDS01
501345	IS04	2003	D	KI001
550123	II07	2001		KI001
550123	II17	2002	D	KI002
501345	BA12	2002	C	KI003
501319	BE13	2005	D	KMT01
501201	IM16	2002	C	KMM01
550123	IA07	2001	C	KMM02

Počet študentov učiteľov

7.3.5 Count(stlpec)

Potlačenie NULL hodnôt v stĺpci.

```
select os_cislo, count(vysledok) from zap_predmety
GROUP BY os_cislo;
```

Výsledok

501319 3
501201 1
501345 2
550123 2

os_cislo	cis_predm	skrok	vysledok	UC
501319	BL14	2005	C	EX001
501319	BE13	2005	D	KMT01
501319	BL11	2005		EX001
501319	BL11	2006	C	EX001
501201	IM16	2002	C	KMM01
501201	II08	2003		KDS01
501345	BA12	2002	C	KI003
501345	IS04	2003	D	KI001
550123	II07	2001		KI001
550123	IA07	2001	C	KMM02
550123	II17	2002	D	KI002

7.3.6 Nedostatočné group by.

Zlé group by môže viesť k nesprávnym dátam.

```
SELECT meno, priezvisko, count(*) FROM pom
GROUP BY meno, priezvisko;
```

Výsledok

Branislav	Balaz	2
Jaroslav	Cipak	1
Alojz	Gazo	1
Martin	Kluciar	3
Frantisek	Murgas	2
Juraj	Papun	1
Marek	Ratroch	1
Janos	Toth	1

MENO	PRIEZVISKO	ROD_CISLO	OS_CISLO
Branislav	Balaz	830514/5341	501567
Branislav	Balaz	830514/5341	501319
Jaroslav	Cipak	840821/8027	500425
Alojz	Gazo	840410/6777	500426
Martin	Kluciar	791229/5431	501096
Martin	Kluciar	791229/5431	501103
Martin	Kluciar	791229/5431	501333
Frantisek	Murgas	870913/3326	501381
Frantisek	Murgas	900913/3326	550945
Juraj	Papun	890310/2145	501345
Marek	Ratroch	850130/3695	550545
Janos	Toth	860907/1259	501414

Zlé group by môže viesť k nesprávnym dátam.

```
SELECT meno, priezvisko, count(*) FROM pom
GROUP BY meno, priezvisko , rod_cislo;
```

Výsledok

Branislav	Balaz	2
Jaroslav	Cipak	1
Alojz	Gazo	1
Martin	Kluciar	3
Frantisek	Murgas	1
Frantisek	Murgas	1
Juraj	Papun	1
Marek	Ratroch	1
Janos	Toth	1

MENO	PRIEZVISKO	ROD_CISLO	OS_CISLO
Branislav	Balaz	830514/5341	501567
Branislav	Balaz	830514/5341	501319
Jaroslav	Cipak	840821/8027	500425
Alojz	Gazo	840410/6777	500426
Martin	Kluciar	791229/5431	501096
Martin	Kluciar	791229/5431	501103
Martin	Kluciar	791229/5431	501333
Frantisek	Murgas	870913/3326	501381
Frantisek	Murgas	900913/3326	550945
Juraj	Papun	890310/2145	501345
Marek	Ratroch	850130/3695	550545
Janos	Toth	860907/1259	501414

7.3.7 HAVING

HAVING - podmienky s agregáčnými funkciami

select cis_predm, count(*)	IE11	1
from zap_predmety	IP09	1
where skrok = 2007	IN05	4
group by cis_predm;	IN09	1
	IZ01	8
	SI28	6
SI31	BS07	2
IN10	IL02	1
BM11	IM09	1

IPM3	1
IPA3	3
IZ99	7
IPN1	1
SI36	1
IS07	2
IP05	9

```
SQL> select cis_predm, count(*)
2   from zap_predmety
3   where skrok = 2007
4   and count(*) >3
5   group by cis_predm;

and count(*) >3
*
ERROR at line 4:
ORA-00934: group function is not allowed here
```

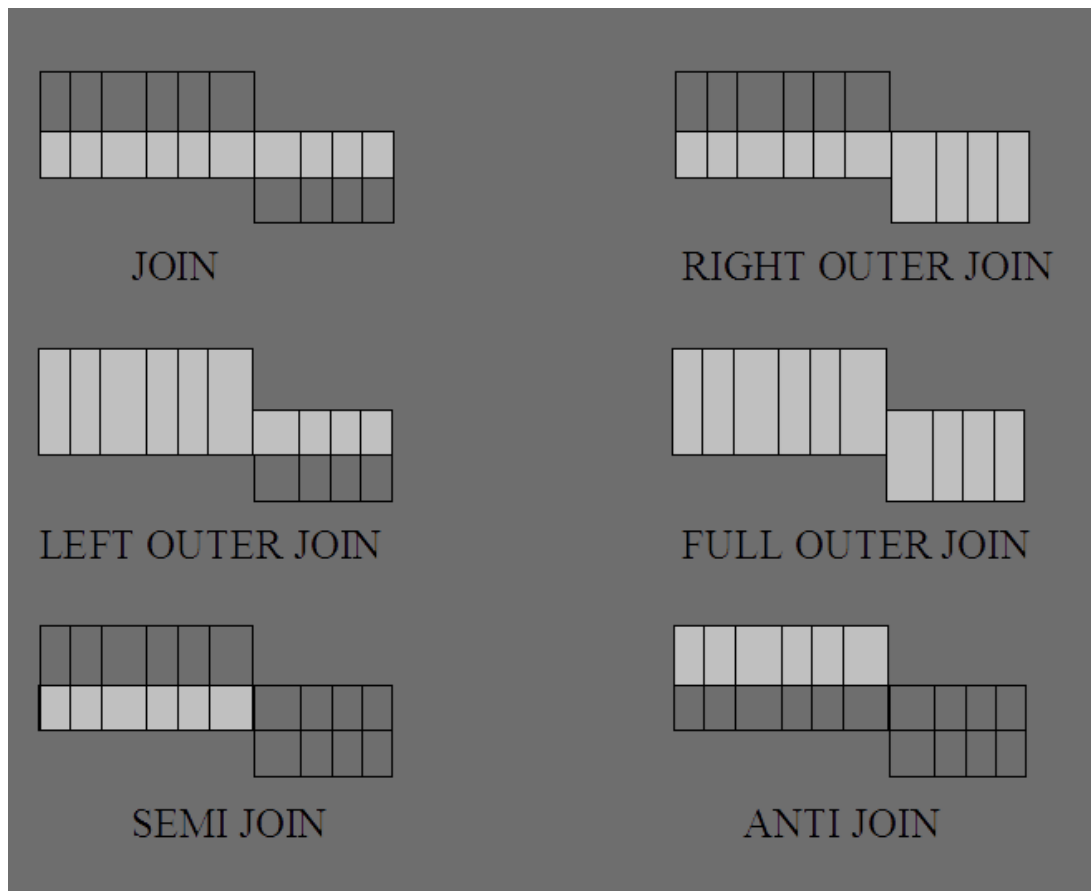
```
select cis_predm, count(*)
  from zap_predmety
 where skrok = 2007
  group by cis_predm
 having count(*) >3;
```

IN05	4
IZ01	8
SI28	6
IZ99	7
IP05	9

7.4 Spojenia

Typy spojení

- (INNER) JOIN - výsledkom sú len dáta, ktoré majú zhodnú hodnotu PK - FK.
- OUTER JOIN - okrem spojených riadkov budú vo výsledku aj riadky, u ktorých nedošlo k spojeniu.
 - LEFT outer join
 - RIGHT outer join
 - FULL outer join
- SEMI JOIN - z prvej tabuľky, kde by došlo ku spojeniu. (IN, EXISTS)
- ANTI JOIN - NOT IN, NOT EXISTS.



7.4.1 INNER JOIN

INNER JOIN - všetky riadky u ktorých dôjde k spojeniu Syntax

`tabulkaA JOIN tabulkaB USING (zoznam_stlpcov)`

`tabulkaA JOIN tabulkaB ON (podmienky)`

tabA	
idA	
1	
2	
3	
4	

tabB	
idB	idA
1	2
2	2
3	3
4	3
5	9
6	9

```
select idA, idB
  from tabA JOIN tabB USING ( idA );
```

```
select idA, idB
  from tabA JOIN tabB ON ( tabA.idA = tabB.idA );
```

	idA	idB
	2	1
Výsledok	2	2
	3	3
	3	4

7.4.2 OUTER JOIN

Syntax

```
tabulkaA { LEFT | RIGHT | FULL } [OUTER] JOIN tabulkaB
        { USING ( zoznam_stlpcov ) | ON ( podmienky ) }
```

1. tabulkaA LEFT OUTER JOIN tabulkaB USING (zoznam_stlpcov)
 tabulkaA LEFT OUTER JOIN tabulkaB ON (zoznam_podmienok)
2. tabulkaA RIGHT OUTER JOIN tabulkaB USING (zoznam_stlpcov)
 tabulkaA RIGHT OUTER JOIN tabulkaB ON (zoznam_podmienok)
3. tabulkaA FULL OUTER JOIN tabulkaB USING (zoznam_stlpcov)
 tabulkaA FULL OUTER JOIN tabulkaB ON (zoznam_podmienok)

Poznámka: Kľúčové slovo OUTER je možné vynechať.

7.4.3 LEFT OUTER JOIN

tabA

idA
1
2
3
4

tabB

idB	idA
1	2
2	2
3	3
4	3
5	9
6	9

```
select idA, B.idB
  from tabA A LEFT JOIN tabB B USING ( idA );
```

```
select A.idA, B.idB
  from tabA A LEFT JOIN tabB B ON ( tabA.idA = tabB.idA );
```

Výsledok

idA	idB
2	1
2	2
3	3
3	4
1	
4	

7.4.4 RIGHT OUTER JOIN

tabA

idA
1
2
3
4

tabB

idB	idA
1	2
2	2
3	3
4	3
5	9
6	9

```
select idA, idB
  from tabA RIGHT JOIN tabB USING ( idA );
```

```
select A.idA, B.idB
  from tabA A RIGHT JOIN tabB B ON ( tabA.idA = tabB.idA );
```

Výsledok

idA	idB
2	1
2	2
3	3
3	4
	5
	5

7.4.5 FULL OUTER JOIN

tabA

idA
1
2
3
4

tabB

idB	idA
1	2
2	2
3	3
4	3
5	9
6	9

```
select idA, idB
  from tabA FULL JOIN tabB USING ( idA );
```

```
select A.idA, B.idB
  from tabA A FULL JOIN tabB B ON ( tabA.idA = tabB.idA );
```

	idA	idB
	2	1
	2	2
	3	3
Výsledok	3	4
	1	
	4	
		5
		5

7.4.6 SEMI JOIN

```
select ...
  from tabulkaA A
 where A.pk_fk IN
       (
         select B.pk_fk
           from tabulka B
         ...
       )
```

```
select ...
  from tabulkaA A
 where EXISTS
       (
         select 'x'
           from tabulka B
          where A.pk_fk = B.pk_fk
         ...
       )
```

Príklad použitia:

```
SELECT st.os_cislo, ou.meno, ou.priezvisko
  FROM os_udaje ou JOIN student st USING ( rod_cislo )
 WHERE EXISTS ( SELECT 'x'
                 FROM zap_predmety zp
                WHERE st.os_cislo = zp.os_cislo
              );
```

ANTI JOIN

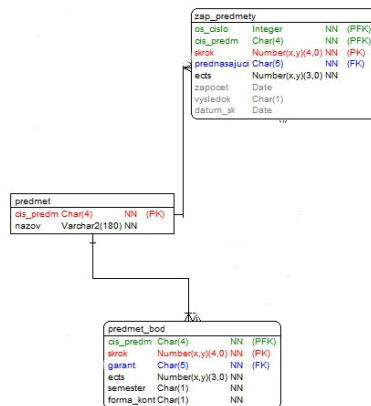
```
select ...
  from tabulkaA A
 where A.pk_fk NOT IN
       (
         select B.pk_fk
           from tabulka B
         ...
       )
```

```
select ...
  from tabulkaA A
 where NOT EXISTS
       (
         select 'x'
           from tabulka B
          where A.pk_fk = B.pk_fk
         ...
       )
```

Príklad použitia:

```
SELECT ou.*
  FROM os_udaje ou
 WHERE NOT EXISTS ( SELECT 'x'
                    FROM kontakty kont
                   WHERE kont.rod_cislo = ou.rod_cislo );
```

7.4.7 Spojenia s minimálnym počtom stĺpcov.



1. Spojenia presne podľa dátového modelu.

```

select zp.os_cislo, cis_predm,
       zp.ects as zp_ects, pb.ects as pb_ects
from zap_predmety zp join predmet pr USING ( cis_
       join predmet_bod pb USING ( cis_predm )
where zp.skrok = pb.skrok
       and zp.ects <> pb.ects;
  
```

2. Keďže nepotrebuje dáta z tabuľky predmet, môžeme ju vynechať.

```

select zp.os_cislo, cis_predm,
       zp.ects as zp_ects, pb.ects as pb_ects
from zap_predmety zp join predmet_bod pb
       USING ( cis_predm )
where zp.skrok = pb.skrok
       and zp.ects <> pb.ects;
  
```

```

select zp.os_cislo, cis_predm,
       zp.ects as zp_ects, pb.ects as pb_ects
from zap_predmety zp join predmet_bod pb
       USING ( cis_predm, skrok )
where zp.ects <> pb.ects;
  
```