

# Databázové systémy a SQL

## Lekce 7

Daniel Klimeš, Monika Kratochvílová

Vypište pacienty, kteří nejsou zařazeni do žádné studie.

Vypište pacienty, kteří nejsou zařazeni do žádné studie.

```
SELECT * FROM patients p
WHERE NOT EXISTS (SELECT * FROM patient_study ps
                  WHERE ps.patient_id=p.patient_id);
```

Agregujte podle pohlaví seznam pacientů, kteří nejsou zařazeni do žádné studie.

Agregujte podle pohlaví seznam pacientů, kteří nejsou zařazeni do žádné studie.

```
SELECT sex, COUNT(*) FROM patients p
WHERE NOT EXISTS (SELECT * FROM patient_study ps
                  WHERE ps.patient_id=p.patient_id)
GROUP BY sex;
```

Napište dotaz, který vrátí počet pacientek zařazených do studií, kde celkový počet pacientek není větší než 10.

Napište dotaz, který vrátí počet pacientek zařazených do studií, kde celkový počet pacientek není větší než 10.

```
SELECT SUM (pocet_pac)
FROM (
    SELECT study_id, COUNT(patient_id) pocet_pac
    FROM patient_study
    GROUP BY study_id
    HAVING COUNT(patient_id)<=10
) tab;
```

Napište dotaz, který vrátí všechny studie v jednotlivých letech s počty pacientů, kteří byli do studie zařazeni, a počet pracovišť, odkud data o pacientech pocházejí.



Napište dotaz, který vrátí všechny studie v jednotlivých letech s počty pacientů, kteří byli do studie zařazeni, a počet pracovišť, odkud data o pacientech pocházejí.

```
SELECT EXTRACT(YEAR FROM ps.date_of_enrollment), s.study_name,
       COUNT(ps.patient_id), COUNT(DISTINCT ps.study_site)
FROM patient_study ps
JOIN studies s
ON ps.study_id=s.study_id
GROUP BY EXTRACT(YEAR FROM ps.date_of_enrollment), s.study_name;
```

Napište dotaz, který vrátí seznam studentů, dále jejich věk a informaci o tom, kdy má student narozeniny (ještě letos/příští rok/neznámo).

Napište dotaz, který vrátí seznam studentů, dále jejich věk a informaci o tom, kdy má student narozeniny (ještě letos/příští rok/neznámo).

```
SELECT *, age(datum_narozeni),
        CASE WHEN EXTRACT(MONTH FROM datum_narozeni)=11
              OR EXTRACT(MONTH FROM datum_narozeni)=12
              THEN 'ještě letos'
              WHEN datum_narozeni IS NULL
              THEN 'neznámo'
              ELSE 'příští rok' END AS narozeniny
FROM student;
```

## Ranking function – číslování řádků

RANK, DENSE\_RANK, ROW\_NUMBER

	RANK	DENSE_RANK	ROW_NUMBER
100	1	1	1
200	2	2	2
200	2	2	3
300	4	3	4
400	5	4	5

- **RANK( ) OVER ([PARTITION BY sex] ORDER BY date\_of\_birth DESC)**
- RANK( ) OVER (ORDER BY date\_of\_birth DESC NULLS LAST)
- **Není možné používat za WHERE a HAVING - nutné zanoření**

## Příklad:

```
SELECT patient_id, sex, date_of_birth,  
RANK( ) OVER (PARTITION BY sex ORDER BY date_of_birth DESC NULLS LAST),  
DENSE_RANK( ) OVER (PARTITION BY sex ORDER BY date_of_birth DESC NULLS LAST),  
ROW_NUMBER( ) OVER (PARTITION BY sex ORDER BY date_of_birth DESC NULLS LAST)  
FROM patients  
ORDER BY sex, date_of_birth DESC NULLS LAST LIMIT 100
```

## Využití v sekci WHERE – nutné zapouzdření

```
SELECT * FROM (  
  SELECT patient_id, sex, date_of_birth,  
  RANK( ) OVER (PARTITION BY sex ORDER BY date_of_birth DESC NULLS LAST) poradi  
  FROM patients) x  
WHERE poradi < 10
```

- LAG (value\_expression [,offset] [,default]) OVER ([query\_partition\_clause] order\_by\_clause)
  - LEAD (value\_expression [,offset] [,default]) OVER ([query\_partition\_clause] order\_by\_clause)
- 
- LAG = hodnota z předchozího řádku
  - LEAD = hodnota z následujícího řádku

```
SELECT study_id, TO_CHAR (date_of_enrollment, 'yyyy'), COUNT(*) letos,  
LAG(COUNT(*),1, '0') OVER(PARTITION BY study_id  
ORDER BY TO_CHAR (date_of_enrollment, 'yyyy') ) loni  
FROM patient_study  
GROUP BY study_id, TO_CHAR (date_of_enrollment, 'yyyy')  
ORDER BY study_id, TO_CHAR (date_of_enrollment, 'yyyy')
```

*Pozn. POSTGRESQL 9.1: LAG(COUNT(\*),1, '0')*

- 1) Vytvořte sestavu: rok - měsíc, počet\_nově zařazených pacientů, kumulativní\_počet\_pacientů z tabulky patient\_study sloupec date\_of\_enrollment
- 2) Doplňte meziměsíční rozdíly v počtech?
- 3) Doplňte sloupec s pořadím velikosti meziměsíčního přírůstku

Období	Počet zařazených	Kumulativní	Rozdíl	Pořadí
2010 - 01	10	10		
2010 - 02	14	24	4	2
2010 - 03	40	64	26	1
2010 - 04	5	69	-35	3

```

SELECT obdobi, ted, minule, ted - minule rozdil, kum,
CASE WHEN minule is NOT NULL THEN
    RANK() OVER (ORDER BY ted-minule desc NULLS LAST)
END poradi
FROM (
    SELECT TO_CHAR (date_of_enrollment, 'yyyy-mm') obdobi, COUNT(*) ted,
    SUM(COUNT(*)) OVER (
        ORDER BY TO_CHAR (date_of_enrollment, 'yyyy-mm') ) kum,
    LAG(COUNT(*),1,NULL) OVER(
        ORDER BY TO_CHAR (date_of_enrollment, 'yyyy-mm')) minule
    FROM patient_study
    GROUP BY TO_CHAR (date_of_enrollment, 'yyyy-mm')
) a
ORDER BY obdobi

```