

Databázové systémy a SQL

Lekce 6

Daniel Klimeš

- Určování pořadí záznamů
- Hodnoty předchozích a následujících řádků
- Rozšířené agregace
 - Výpočet procent
 - Parciální agregace
 - Kumulativní součet
 - Klouzavý průměr

Rozšíření SQL o

... OVER (PARTITION BY sloupec ORDER BY sloupec)

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 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')*

```
SELECT COUNT(*) FROM student
```

```
SELECT studium, COUNT(*) FROM student
GROUP BY studium
```

```
SELECT studium, COUNT(*) pocet , COUNT(*) *
100.0/(SELECT COUNT(*) FROM student) procento FROM
student
GROUP BY studium
```

```
SELECT studium, COUNT(*) pocet, COUNT(*) *100.0 /
SUM(COUNT(*) OVER ()) procento FROM student
GROUP BY studium
```

Procentické zastoupení – standardní SQL:

```
SELECT study_id, COUNT(*),  
COUNT(*) * 100.0 / (SELECT COUNT(*) FROM patient_study) procento  
FROM patient_study  
GROUP BY study_id
```

Analytická funkce

```
SELECT study_id, COUNT(*),  
COUNT(*) / SUM(COUNT(*) OVER ()) * 100 procento  
FROM patient_study  
GROUP BY study_id
```

```
UPDATE student SET ukonceni = 'Z'  
WHERE mod(uc0,2) = 1 – Rozdělení datového souboru
```

```
SELECT ukonceni, studium, count(*) pocet, COUNT(*) *100.0 /  
SUM(COUNT(*)) OVER () procento FROM student  
GROUP BY ukonceni, studium  
ORDER BY ukonceni
```

```
SELECT ukonceni, studium, count(*) pocet, COUNT(*) *100.0 /  
SUM(COUNT(*)) OVER () procento ,  
COUNT(*) *100.0 / SUM(COUNT(*)) OVER (PARTITION BY ukonceni)  
proc_podskupiny  
FROM student  
GROUP BY ukonceni, studium  
ORDER BY ukonceni
```


Procentické zastoupení pracovišť (počtu jejich pacientů) v jednotlivých studiích

```
SELECT study_id, study_site, COUNT(*),  
COUNT(*) / SUM(COUNT(*)) OVER (PARTITION BY study_id) * 100 procento  
FROM patient_study  
GROUP BY study_id, study_site
```

```
SELECT studium, COUNT(*) pocet FROM
student
GROUP BY studium
```

```
SELECT studium, COUNT(*) pocet,
SUM(COUNT(*)) OVER (ORDER BY studium)
FROM student
GROUP BY studium
```

```
SELECT pohlavi, studium, COUNT(*) pocet,
SUM(COUNT(*)) OVER (PARTITION BY pohlavi ORDER BY studium)
kumulace_skupina,
SUM(COUNT(*)) OVER (ORDER BY pohlavi, studium) kumulace_celkem
FROM student
GROUP BY pohlavi, studium
ORDER BY pohlavi, studium
```

AVG(sloupec) OVER
(ORDER BY sloupec ROWS BETWEEN x PRECEDING AND CURRENT ROW)

• **ROWS BETWEEN**



- **UNBOUNDED PRECEDING**
- **UNBOUNDED FOLLOWING**
- **CURRENT ROW**
- **počet řádků PRECEDING**
- **počet řádků FOLLOWING**

```
CREATE TABLE pocet_pacientu as
SELECT TO_CHAR(date_of_enrollment, 'yyyy-mm') mesic, COUNT(*)
pocet FROM patient_study
WHERE date_of_enrollment >= '2004-01-01'
GROUP BY TO_CHAR(date_of_enrollment, 'yyyy-mm')
ORDER BY TO_CHAR(date_of_enrollment, 'yyyy-mm')
```

```
SELECT * FROM pocet_pacientu
ORDER BY mesic
```

```
SELECT AVG(pocet) FROM
pocet_pacientu
```

```
SELECT mesic, pocet,
ROUND(AVG(pocet) OVER (ORDER BY mesic ROWS BETWEEN 3
PRECEDING AND CURRENT ROW),1) klouzavy_prumer
FROM pocet_pacientu
```

1) Spočítejte v tabulce pocet_pacientu kumulativní počet pacientů

```
SELECT mesic, pocet
FROM pocet_pacientu
ORDER BY mesic
```

```
SELECT mesic, pocet
FROM pocet_pacientu
ORDER BY mesic
```

```
SELECT mesic, pocet,
SUM(pocet) OVER (ORDER BY mesic)
FROM pocet_pacientu
ORDER BY mesic
```

- Přidejte ke kumulativnímu počtu kumulativní procento

- Přidejte ke kumulativnímu počtu kumulativní procento

```
SELECT mesic, pocet, SUM(pocet) OVER (ORDER BY mesic),
SUM(pocet) OVER(), SUM(pocet) OVER (ORDER BY mesic) * 100 /
SUM(pocet) OVER() FROM pocet_pacientu
ORDER BY mesic
```

- Zobrazte kumulativní procentické zastoupení pacientů podle věku
 - Věk, počet pacientů, kumulativní procento

- Zobrazte kumulativní procentické zastoupení pacientů podle věku
 - Věk, počet pacientů, kumulativní procento

```
SELECT EXTRACT (YEAR FROM AGE(date_of_birth))
FROM patients limit 100
```

```
SELECT vek, COUNT(*) FROM (
  SELECT EXTRACT (YEAR FROM AGE(date_of_birth)) vek
  FROM patients) a
WHERE vek > 0 and vek < 100
GROUP BY vek
ORDER BY vek
```

```

SELECT vek, pocet, kum_pocet * 100 / pocet_celkem kum_procento
FROM (
  SELECT vek, COUNT(*) pocet, SUM(COUNT(*)) OVER (ORDER BY VEK)
    kum_pocet, SUM(COUNT(*)) OVER () pocet_celkem
  FROM (
    SELECT EXTRACT (YEAR FROM AGE(date_of_birth)) vek
    FROM patients) a
  WHERE vek > 0 and vek < 100
  GROUP BY vek
  ORDER BY vek
) b

```