# Семинар 7-8

## Оконные функции

* Ранжирующие  
  row\_number  
  rank
* Агрегатные  
  sum, max, min, avg, count
* Функции получение доступа  
  lag, lead

### Ранжирующие

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pno | Pname | Weight | Color |  |  |  |  |  |  |
| 1 | Гвоздь | 10 | К |  |  |  |  |  |  |
| 2 | Гвоздь | 12 | К |  |  |  |  |  |  |
| 3 | Винт | 13 | К |  |  |  |  |  |  |
| 4 | Шуруп | 14 | С |  |  |  |  |  |  |
| 5 | Гвоздь | 15 | С |  |  |  |  |  |  |
| 6 | Винт | 11 | З |  |  |  |  |  |  |
| 7 | Шуруп | 10 | З |  |  |  |  |  |  |

Таблица P - детали

select Pno, Pname, Weight, Color,  
row\_number() **over** (**partition by** Color **order by** weight) as r1,  
row\_number() over(… order by weight) as r2,  
row\_number() over(… partition by Pname order by weight) as r3

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pno | Pname | Weight | Color | r1 | r2 | r3 |  |  |  |
| 1 | **Гвоздь** | **10** | К | 1 | 1/2 | 1 |  |  |  |
| 2 | **Гвоздь** | **12** | К | 2 | 4 | 2 |  |  |  |
| 3 | *Винт* | *13* | К | 3 | 5 | 2 |  |  |  |
| 4 | Шуруп | 14 | С | 1 | 6 | 2 |  |  |  |
| 5 | **Гвоздь** | **15** | С | 2 | 7 | 3 |  |  |  |
| 6 | *Винт* | *11* | З | 2 | 3 | 1 |  |  |  |
| 7 | Шуруп | 10 | З | 1 | 2/1 | 1 |  |  |  |

select \*  
from P  
where (Color, Weight) in (select Color max(weight)  
 from P  
 group by Color)  
В итоге всё равно могут быть дубли  
  
with rnColor (Pno, Pname, Weight, Color, rn) as (  
 select \*, row\_number() over (partition by Color order by Weight)  
 from P  
)  
select \*  
from rnColor  
where m = 1

### Агрегатные функции

Select Pno, Pname, Weight, Color,  
 max(Weight) over (partition by Color) as m1,  
 max(Weight) over (partition by Color  
 order by Pno) as m2,  
 sum(Weight) over (order by Weight) as m3  
from P

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pno | Pname | Weight | Color | m1 | m2 | m3 |  |  |  |
| 1 | Гвоздь | 10 | К | **13** | 10 | 10/20 |  |  |  |
| 2 | Гвоздь | 12 | К | **13** | 12 | 43 |  |  |  |
| 3 | Винт | **13** | К | **13** | 13 | 56 |  |  |  |
| 4 | Шуруп | 14 | С | **15** | 14 | 70 |  |  |  |
| 5 | Гвоздь | **15** | С | **15** | 15 | 85 |  |  |  |
| 6 | Винт | **11** | З | **11** | 11 | 31 |  |  |  |
| 7 | Шуруп | 10 | З | **11** | 11 | 20/10 |  |  |  |

Пример из жизни: хожу в магазин, покупаю продукты

|  |  |  |
| --- | --- | --- |
| id | summa | date |
| 1 | 100 | 01-10-2021 |
| 1 | -20 | 01-10-2021 |
| 1 | -30 | 01-10-2021 |
| 1 | 150 | 02-10-2021 |
| 1 | -75 | 03-10-2021 |

Остаток на счету на конец дня?

Ожидаемый результат:

|  |  |  |
| --- | --- | --- |
| 1 | 01-10-2021 | 50 |
| 1 | 02-10-2021 | 200 |
| 1 | 03-10-2021 | 125 |

Как сделать?

with delta (id, date, s) as (  
 select id, date, sum(summa)  
 from t  
 group by id, date  
)  
select id, date, sum (s) over (partition by id order by date)  
from delta

### Функции для получения доступа

select \*,  
 lag(Weight) over (partition by color order by weight) l1,  
 lead(Weight) over (partition by color order by weight) l2

from P

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pno | Pname | Weight | Color | l1 | l2 |  |  |  |  |
| 1 | Гвоздь | 10 | К | null | 12 |  |  |  |  |
| 2 | Гвоздь | 12 | К | 10 | 13 |  |  |  |  |
| 3 | Винт | 13 | К | 12 | null |  |  |  |  |
| 4 | Шуруп | 14 | С | null | 15 |  |  |  |  |
| 5 | Гвоздь | 15 | С | 14 | null |  |  |  |  |
| 6 | Винт | 11 | З | 10 | null |  |  |  |  |
| 7 | Шуруп | 10 | З | null | 11 |  |  |  |  |

Где применяется?

scd1

|  |  |  |
| --- | --- | --- |
| id | dt | type |
| 1 | 9:00 | 1 |
| 1 | 9:10 | 2 |
| 1 | 9:20 | 1 |
| 1 | 19:00 | 2 |
| 2 | 9:00 | 1 |
| 2 | 19:00 | 2 |

OLTP – online transaction processiong

Нужна другая таблица:

scd2

|  |  |  |  |
| --- | --- | --- | --- |
| id | dfrom | dto | type |
| 1 | 9:00 | 9:10 | 1 |
| 1 | 9:10 | 9:20 | 2 |
| 1 | 9:20 | 19:00 | 1 |
| 2 | 9:00 | 19:00 | 1 |

OLAP– online analytic processiong

Для получение второй таблицы используем lag, lead к 1ой таблице, удалив строчки с type == 2, (2 - когда не на работе, 1 – на работе)

## Тута

Пусть R(A, B, C, D, E, F) – переменная отношения (*Таблица* – табличная переменная)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F |
| 1 | 2 | 5 | 6 | a | 4 |

– множество функциональных зависимостей.  
Найти замыкание -?

|  |  |  |
| --- | --- | --- |
|  | *A* | *A,B,C,D,E,F* |
|  | *A,B,C* | *A,B,C,D,E,F* |
|  | *A,B,C,D,E* | *A,B,C,D,E,F* |
|  | *A,B,C,D,E,F* | *A,B,C,D,E,F* |
|  | *A,B,C,D,E,F* | *A,B,C,D,E,F* |

*=>* A – ключ => потенциальный (слева куда меньше атрибутов?) но НЕ супер-ключ

- супер-ключ

Выводимая или не выводимая функциональная зависимость?

?

=> не выводима

Вывести

1. (декомпозиция)
2. (декомпозиция)
3. (дополнение)
4. (?Самоопределение?)
5. (Транзитивность)

Вариант 1.

1. SQL:  
   select count(\*)  
   from Toirists T join City C  
    on T.CityID=C.CityID  
   where Name = ‘Moskva’  
   РА:  
   (SUMMARIZE (T join C)where name = ‘Moskva’ PER CityID Add count as cnt)[cnt]  
   ИК:  
   Range of Cx is City where = ‘Moskva’  
   Range of Tx is T  
   count ((Tx.Id) where (exists Cx where (Cx.CityID=Tx.CityID)))
2. Города в которых было 5 или больше туристов  
   ИК:  
   Range of Cx is City  
   Range of Tx is T  
   (Cx.name, count(Tx.id where Tx.CityID=Cx.CityID) as cnt) where cnt = 5

## Функции

См. Презентация (если пришлют :D)

* Детерминированные (сложение, синус) – можно не пересчитывать
* Недетерминированные (рандомайзер, выборка) – всегда нужно считать

Бывают:

* Скалярные функции (возвращает скаляр – int, массивы, пользовательские…)
* Подставляемая табличная функция (возвращает множество кортежей)
* Многоператорная функция (возращение данных делается через табличные переменные - таблицу)

## Хранимая процедура

Функция возвращает набор данных. Если не нужно вернуть набор данных, то используйте процедуру.

См. Презентация