



Datové a procesní modely

[Relační databáze]

Přednáška analytické funkce

Marian Kamenický, Jiří Měska

marian.kamenicky@syntea.cz

jiri.meska@silicium.cz

MFFUK Praha

2020



SQL dotazy



```
SELECT ...  
  From ...  
  Where ...  
  Order By
```

- detailní
- z každé [vybrané] řádky tabulky - **1 řádka výsledku**



```
SELECT ...  
  From ...  
  ....  
Group By a,b,c
```

```
SELECT Sum(...),..  
  From  
  .....
```

Group By ()

- skupinové - agregační
řádka výsledku -
1 za každou skupinu





- původně pouze 5 agregačních funkcí
- možnost agregovat pouze za jedno skupinové hledisko
- Group By
- postupem doby - desítky nových agregačních funkcí
- nové možnosti RollUp, Cube, agregační skupiny
- analytické funkce
- proč takový rozsáhlý rozvoj "skupinování"
- ? ? ? ? ? [Obchod peníze]



Agregační funkce



- count

- sum

- max

- min

- min

- Group_Concat

- Bit_And

- Bit_Xor

- Bit_And

- StdDev

- StdDev_Pop

- StdDev_Samp

- Var_Pop

- Var_Samp

- Variance

- Corr

- Covar_Pop

- Covar_Samp

- Stats_f_test

- Stats_mode

- Stats_ks_test

-

Agregační funkce v PostgreSQL



■ count

■ sum

■ max

■ min

■ min

Name	Description
<u>CUME_DIST</u>	Return the relative rank of the current row.
<u>DENSE_RANK</u>	Rank the current row within its partition without gaps.
<u>FIRST_VALUE</u>	Return a value evaluated against the first row within its partition.
<u>LAG</u>	Return a value evaluated at the row that is at a specified physical offset row before the current row within the partition.
<u>LAST_VALUE</u>	Return a value evaluated against the last row within its partition.
<u>LEAD</u>	Return a value evaluated at the row that is offset rows after the current row within the partition.
<u>NTILE</u>	Divide rows in a partition as equally as possible and assign each row an integer starting from 1 to the argument value.
<u>NTH_VALUE</u>	Return a value evaluated against the nth row in an ordered partition.
<u>PERCENT_RANK</u>	Return the relative rank of the current row $(rank-1) / (total\ rows - 1)$
<u>RANK</u>	Rank the current row within its partition with gaps.
<u>ROW_NUMBER</u>	Number the current row within its partition starting from 1.



AFunction (arg1,..., argn)

OVER ([PARTITION BY <...>]

[ORDER BY <.....>]

[<window_clause>])

Afunction

umožňuje získat agregaci na úrovni sloupce jakékoliv řádky

spolu s jakýmikoli sloupci řádky

doplňuje řádky tabulky o agregační údaje

Statistiky

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	?
Bill	M	100	20 000	?
Bob	M	300	60 000	?
Dan	M	300	20 000	?
Hana	Z	100	10 000	?
Jana	Z	100	30 000	?
Pepa	M	100	20 000	?
Tere	Z	150	30 000	?
Tom	M		10 000	?
Zuzi	Z	400	40 000	?

```
SELECT    Jm, Poh, Odd, Plat,
```

```
(Select  Count(*)  
  From OurZam Z2  
  Where Z1.Odd = Z2.Odd) as PocOdd
```

```
From      OurZam Z1
```

```
Order by  Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Tom	M		10 000	0
Zuzi	Z	400	40 000	1

Statistiky

AF funkci

si mohu sám vytvořit pomocí SUB-SELECTu

a to korelovanou i nekorelovanou

```
SELECT    Jm, Poh, Odd, Plat,
```

```
(Select  Count(*)
```

```
  From OurZam Z2
```

```
  Where Z1.Odd = Z2.Odd) as PocOdd
```

```
From      OurZam Z1
```

```
Order by  Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	?
Bill	M	100	20 000	?
Bob	M	300	60 000	?
Dan	M	300	20 000	?
Hana	Z	100	10 000	?
Jana	Z	100	30 000	?
Pepa	M	100	20 000	?
Tere	Z	150	30 000	?
Tom	M		10 000	?
Zuzi	Z	400	40 000	?

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Tom	M		10 000	0
Zuzi	Z	400	40 000	1



- to lze jednoduše zadat pomocí analytických funkcí

```
SELECT Jm, Poh, Odd, Plat,  
       Count(*) Over (Partition by Odd) PocOdd  
From   OurZam  
Order by Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Tom	M		10 000	1
Zuzi	Z	400	40 000	1



- to lze jednoduše zadat pomocí analytických funkcí

```
SELECT Jm, Poh, Odd, Plat,  
       Count(*) Over (Partition by Odd) PocOdd  
From   OurZam  
Order by Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Tom	M		10 000	1
Zuzi	Z	400	40 000	1



- vypsát jednoduché informace o zaměstnancích
 - jm, poh, odd, plat

```
SELECT  Jm, Poh, Odd, Plat
From    OurZam
Order by Jm;
```

- doplněné počtem
- zaměstnanců
- v jeho oddělení

OurZam			
Jm	Poh	Odd	Plat
Adam	M	100	20 000
Bill	M	100	20 000
Bob	M	300	60 000
Dan	M	300	20 000
Hana	Z	100	10 000
Jana	Z	100	30 000
Pepa	M	100	20 000
Tere	Z	150	30 000
Tom	M		10 000
Zuzi	Z	400	40 000

Statistiky

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	?
Bill	M	100	20 000	?
Bob	M	300	60 000	?
Dan	M	300	20 000	?
Hana	Z	100	10 000	?
Jana	Z	100	30 000	?
Pepa	M	100	20 000	?
Tere	Z	150	30 000	?
Tom	M		10 000	?
Zuzi	Z	400	40 000	?

```
SELECT    Jm, Poh, Odd, Plat,
```

```
(Select  Count(*)  
  From OurZam Z2  
  Where Z1.Odd = Z2.Odd) as PocOdd
```

```
From      OurZam Z1
```

```
Order by  Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Tom	M		10 000	0
Zuzi	Z	400	40 000	1



- to lze jednoduše zadat pomocí analytických funkcí

```
SELECT Jm, Poh, Odd, Plat,  
       Count(*) Over (Partition by Odd) PocOdd  
From   OurZam  
Order by Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Tom	M		10 000	1
Zuzi	Z	400	40 000	1



- mírně to přetřídíme

```
SELECT Jm, Poh, Odd, Plat,  
       Count(*) Over (Partition by Odd) PocOdd  
From   OurZam  
Order by Odd, Jm;
```

OurZam				
Jm	Poh	Odd	Plat	PocOdd
Adam	M	100	20 000	5
Bill	M	100	20 000	5
Hana	Z	100	10 000	5
Jana	Z	100	30 000	5
Pepa	M	100	20 000	5
Tere	Z	150	30 000	1
Bob	M	300	60 000	2
Dan	M	300	20 000	2
Zuzi	Z	400	40 000	1
Tom	M		10 000	1



■ a něco přidáme

```
SELECT Jm, Poh, Odd, Plat,  
       Count(*) Over (Partition by Odd) PocOdd,  
       Count(*) Over (Partition by Poh) PocPoh  
From   OurZam  
Order by Odd, Jm;
```

OurZam					
Jm	Poh	Odd	Plat	PocOdd	PocPoh
Adam	M	100	20 000	5	6
Bill	M	100	20 000	5	6
Hana	Z	100	10 000	5	4
Jana	Z	100	30 000	5	4
Pepa	M	100	20 000	5	6
Tere	Z	150	30 000	1	4
Bob	M	300	60 000	2	6
Dan	M	300	20 000	2	6
Zuzi	Z	400	40 000	1	4
Tom	M		10 000	1	6



■ a něco přidáme

```
SELECT Jm, Poh, Odd, Plat,  
       Count(*) Over (Partition by Odd) PocOdd,  
       Count(*) Over (Partition by Poh) PocPoh  
From   OurZam  
Order by Odd, Jm;
```

OurZam					
Jm	Poh	Odd	Plat	PocOdd	PocPoh
Adam	M	100	20 000	5	6
Bill	M	100	20 000	5	6
Hana	Z	100	10 000	5	4
Jana	Z	100	30 000	5	4
Pepa	M	100	20 000	5	6
Tere	Z	150	30 000	1	4
Bob	M	300	60 000	2	6
Dan	M	300	20 000	2	6
Zuzi	Z	400	40 000	1	4
Tom	M		10 000	1	6



pro zamestnance OurZam zjistěte

- pocet zam v odd, soucet platů v oddělení

```
SELECT  Jm, Odd, Plat,  
        Count(*) Over ( Partition by Odd ) Lidi,  
        Sum(Plat) Over ( Partition by Odd ) Celkem  
From    OurZam  
Order by Odd, Jm;
```

Jm	Odd	Plat	Lidi	Celkem
Adam	100	20 000	5	100 000
Bill	100	20 000	5	100 000
Hana	100	10 000	5	100 000
Jana	100	30 000	5	100 000
Pepa	100	20 000	5	100 000
Tere	150	30 000	1	30 000
Bob	300	60 000	2	80 000
Dan	300	20 000	2	80 000
Zuzi	400	40 000	1	40 000
Tom		10 000	1	10 000



- přidejte počet zam v podniku, součet platů v podniku

```
SELECT Jm, Odd, Plat,  
       Count(*) Over ( Partition by Odd ) Lidi,  
       Sum(Plat) Over ( Partition by Odd ) Celk,  
       Count(*) Over (           ) Lidi2,  
       Sum(Plat) Over (           ) Celk2  
From   OurZam  
Order by Odd, Jm;
```

Jm	Odd	Plat	Lidi	Celk	Lidi2	Celk2
Adam	100	20 000	5	100 000	10	260 000
Bill	100	20 000	5	100 000	10	260 000
Hana	100	10 000	5	100 000	10	260 000
Jana	100	30 000	5	100 000	10	260 000
Pepa	100	20 000	5	100 000	10	260 000
Tere	150	30 000	1	30 000	10	260 000
Bob	300	60 000	2	80 000	10	260 000
Dan	300	20 000	2	80 000	10	260 000
Zuzi	400	40 000	1	40 000	10	260 000
Tom		10 000	1	10 000	10	260 000



a zas něco

dalšího

Sportovní evidence hodů [dvě hry]



Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody	
Jm	
Cim	
Body	



- potřebujeme označit výsledky
- kdo je první, druhý, třetí,...
- pokud nejlepších jsou tři
- další je až čtvrtý
- vypište tabulku Hody
- s určením výsledku -
 - tedy s pořadím sportovců v každé hře

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody		
Jm	Cim	Body

Sportovní evidence hodů [dvě hry]



- potřebujeme označit
- kdo je první, druhý
- pokud nejlepší
- další je až čtvrtý
- vypište tabulku
- s určením výsledku
- tedy s pořadím

Hody			
Jm	Cim	Body	Umístění
Joe	Noha	15	1
Bob	Noha	11	2
Tim	Noha	11	2
Han	Noha	11	2
Kid	Noha	7	5
Sam	Noha	4	6
Cic	Noha	4	6
Emi	Noha	3	8
Fif	Ruka	5	1
Grg	Ruka	5	1
Vil	Ruka	4	3
Lik	Ruka	4	3
Zoe	Ruka	4	3

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody	
Jm	
Cim	
Body	

hře

Jak na to ???



- potřebujeme pořadové funkce
- v dané "partition" přiřadí každé řádce
- pořadí - které si zaslouží

- funkce pro přiřazení pořadí
- jsou zvláštní funkce v okně "partition"
- existuje jich několik

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody	
Jm	
Cim	
Body	



každé řádce INTEGER hodnotu pořadí v PARTITION

■ Row_Number

■ Rank

■ Dense_Rank

- seriové pořadí řádky v Partition
- pořadí řádky v partition dle hodnoty
 - s "dírami" - přeskok hodnot při opakování
- pořadí řádky v partition dle hodnoty
 - stísněně bez "děr,, při opakování

Function (arg1,..., argn)

OVER ([PARTITION BY <...>]

[**ORDER BY** <....>]

[<window_clause>])

ORDER BY VýrazSQL [Asc | Desc] NULLS [First | Last]

Sportovní evidence hodů [dvě hry]



- potřebujeme oznámit
- kdo je první, druhý
- pokud nejlepších
- další je až čtvrtý
- vypište tabulku
- s určením výsledků
- tedy s pořadím

Hody			
Jm	Cim	Body	Umístění
Joe	Noha	15	1
Bob	Noha	11	2
Tim	Noha	11	2
Han	Noha	11	2
Kid	Noha	7	5
Sam	Noha	4	6
Cic	Noha	4	6
Emi	Noha	3	8
Fif	Ruka	5	1
Grg	Ruka	5	1
Vil	Ruka	4	3
Lik	Ruka	4	3
Zoe	Ruka	4	3

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody	
Jm	Cim
Cim	Body

■ řešte

hře



Select Jm,

Cim,

Body,

Rank () Over (Partition By Cim) Por

From Hody

Order By Cim, Body Desc;

error

- pro funkci **Rank**
- musí být zadáno třídění
- v partition





Select Jm,

Cim,

Body,

Rank () Over (Partition By Cim **Order By** Body) Por

From Hody

Order By Cim, Body Desc;

OK

- pro funkci **Rank**
- musí být zadáno třídění
- v partition





Select Jm,

Cim,

Body,

Rank (

From Hody

Order By Cim

Hody			
Jm	Cim	Bod y	Umístění
Joe	Noha	15	8
Bob	Noha	11	5
Tim	Noha	11	5
Han	Noha	11	5
Kid	Noha	7	4
Sam	Noha	4	2
Cic	Noha	4	2
Emi	Noha	3	1
Fif	Ruka	5	4
Grg	Ruka	5	4
Vil	Ruka	4	1
Lik	Ruka	4	1
Zoe	Ruka	4	1

im Order By Body) Por

OK

funkci Rank

ýt zadáno třídění

■ v partition





- potřebujeme také určit
- v jakém pořadí půjdou na "platz"
- jeden po druhém
- jsou tam vrátka pouze pro jednoho
- určete pořadí "jde"

- k tomu je pořadová funkce Row_Number

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody	
Jm	
Cim	
Body	

■ řešte



Select Jm,

Cim,

Body,

Row_number ()

Over (Partition By Cim **Order By** Body **Desc**)

Jde

From Hody

Order By Cim, **Jde**;



Skupinové dotazy



Select Jm,

Cim,

Body,

Row_#

Over

Jde

From Hody

Order By Cim

Hody			
Jm	Cim	Body	Jde
Joe	Noha	15	1
Bob	Noha	11	2
Tim	Noha	11	3
Han	Noha	11	4
Kid	Noha	7	5
Sam	Noha	4	6
Cic	Noha	4	7
Emi	Noha	3	8
Fif	Ruka	5	1
Grg	Ruka	5	2
Vil	Ruka	4	3
Lik	Ruka	4	4
Zoe	Ruka	4	5

Order By Body Desc)





- po ceremonii odvezou sportovce auta
- postupně podle pořadí
- pokud je více se stejným pořadím jedou ve stejném autě

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

- k tomu je pořadová funkce Dence_Rank
- která stanoví do kolikátého auta nasadnout

Hody
Jm Cim Body

■ řešte



Select Jm,

Cim,

Body,

Dense_Rank ()

Over (Partition By Cim **Order By** Body **Desc**)

Vuz

From Hody

Order By Cim, **Vuz**;



Skupinové dotazy



Select Jm,

Cim,

Body,

Dense

Over

Vuz

From Hody

Order By Cim

Hody			
Jm	Cim	Body	Vuz
Joe	Noha	15	1
Bob	Noha	11	2
Tim	Noha	11	2
Han	Noha	11	2
Kid	Noha	7	3
Sam	Noha	4	4
Cic	Noha	4	4
Emi	Noha	3	5
Fif	Ruka	5	1
Grg	Ruka	5	1
Vil	Ruka	4	2
Lik	Ruka	4	2
Zoe	Ruka	4	2

Order By Body Desc)





- borcům by se měli dát peníze
- vítězi/vítězům 1000 Kč
- ostatním dle zásluh - adekvátně
- !!!!!

■ řešte

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody	
Jm	
Cim	
Body	

- k tomu účelu slouží funkce **Cume_Dist**
- funkce počítá podíl počtu závodníku pod ním včetně jeho samého, vítěz závodu Noha dostane 1 (všichni), poslední potom $1/8$ tj 0.125 , Sam a Cic tvoří skupinu $3 \times 1/8 = 0,375$



Select Jm,

Cim,

Body,

Cume_Dist ()

Over (Partition By Cim **Order By** Body **Asc**) Odm

From Hody

Order By Cim, Body Desc;





Select Jm,

Cim,

Body,

Cume

Over (

From Hody

Order By Cim

Hody			
Jm	Cim	Bod y	Umístění
Joe	Noha	15	1
Bob	Noha	11	0,875
Tim	Noha	11	0,875
Han	Noha	11	0,875
Kid	Noha	7	0,5
Sam	Noha	4	0,375
Cic	Noha	4	0,375
Emi	Noha	3	0,125
Fif	Ruka	5	1
Grg	Ruka	5	1
Vil	Ruka	4	0,6
Lik	Ruka	4	0,6
Zoe	Ruka	4	0,6

Asc) Odm





Select Jm,

Cim,

Body,

1000 * Cume_Dist ()

Over (Partition By Cim **Order By** Body **Desc**) Odm

From Hody

Order By Cim, Body Desc;





Select Jm,

Cim,

Body,

1000

Over (

From Hody

Order By Cim

Hody			
Jm	Cim	Bod y	Umístění
Joe	Noha	15	1000
Bob	Noha	11	875
Tim	Noha	11	875
Han	Noha	11	875
Kid	Noha	7	500
Sam	Noha	4	375
Cic	Noha	4	375
Emi	Noha	3	125
Fif	Ruka	5	1000
Grg	Ruka	5	1000
Vil	Ruka	4	600
Lik	Ruka	4	600
Zoe	Ruka	4	600

Desc) Odm





- bohužel těch aut na odvoz jsou pouze 3
- na každou soutěž
- rozdělte sportovce do přesně 3 aut
- dle výsledků

- k tomu účelu slouží funkce **NTile**

Hody		
Jm	Cim	Body
Vil	Ruka	4
Tim	Noha	11
Kid	Noha	7
Zoe	Ruka	4
Joe	Noha	15
Cic	Noha	4
Emi	Noha	3
Sam	Noha	4
Fif	Ruka	5
Bob	Noha	11
Grg	Ruka	5
Lik	Ruka	4
Han	Noha	11

Hody		
Jm	Cim	Body



NTILE

- funkce pořadí řádky
 - jako Rank(), Dense_Rank()
- dělení řádek partition do balíčků
- počet balíčků nutno zadat

NTILE (počet balíčků) OVER ...

■ řešte



Select Jm,

Cim,

Body,

NTile (3)

Over (Partition By Cim **Order By Body Desc**) Vuz

From Hody

Order By Cim, Body Desc;





Select Jm,

Cim,

Body,

NTile (3)

Over (Partition I

From Hody

Order By Cim, Body D

jm character varying (64)	cim character varying (64)	body integer	vuz integer
Joe	Noha	15	1
Tim	Noha	11	1
Han	Noha	11	1
Bob	Noha	11	2
Kid	Noha	7	2
Cic	Noha	4	2
Sam	Noha	4	3
Emi	Noha	3	3
Fif	Ruka	5	1
Grg	Ruka	5	1
Vil	Ruka	4	2
Lik	Ruka	4	2
Zoe	Ruka	4	3

|Z





- Oblast [partition]
- **Okno** [window]
- **Aktuální řádek** [current row]

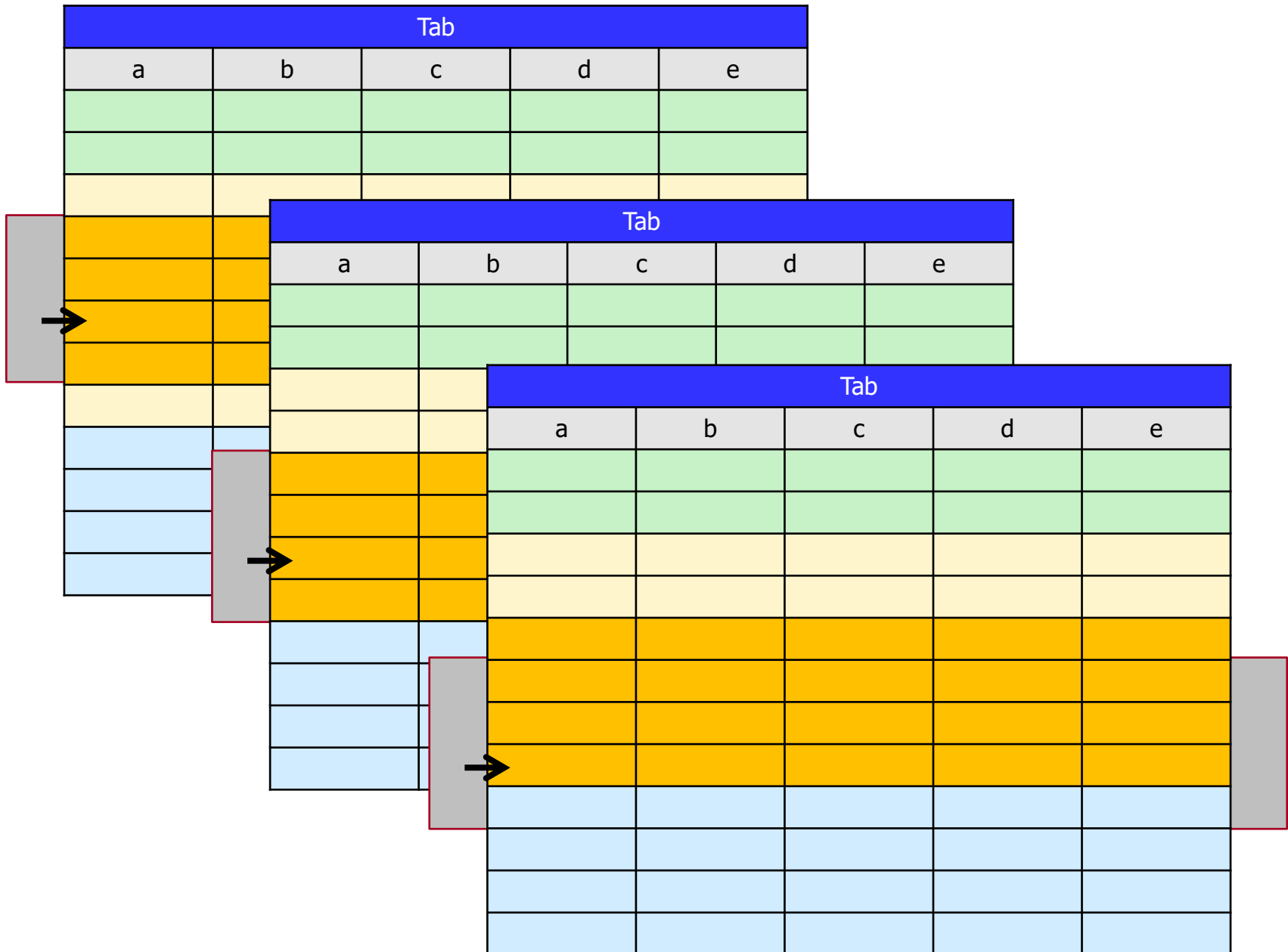
Okno

- (pracovní) **náhled** na skupinu řádků oblasti [partition]
- **velikost** ?
 - dána počátečním a koncovým řádkem
- **statické** okno
 - lze nastavit na celou oblast
- **dynamické** okno [okno s dynamikou]
 - mění se pohybem aktuálního řádku [v oblasti]
 - př: 3 řádky před, 3 řádky za
 - výpočet pohyblivých agregací



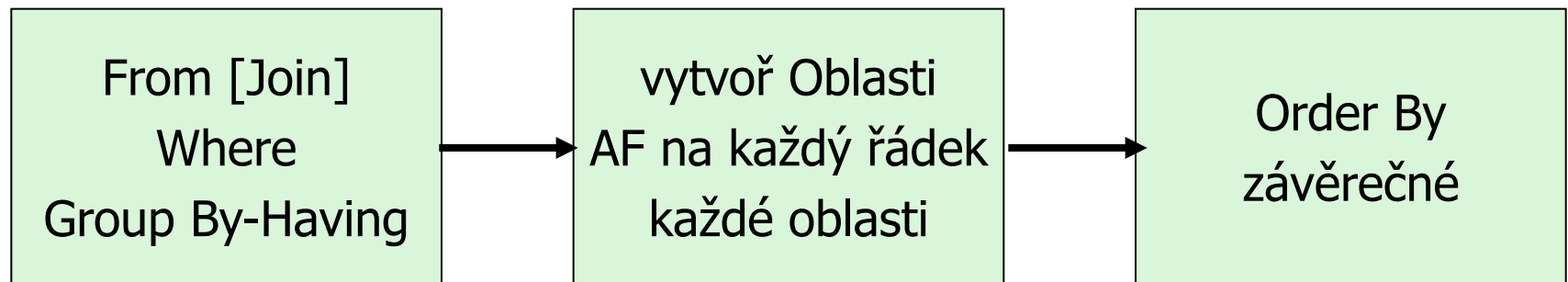
Tab				
a	b	c	d	e

Analytické funkce – Okno dynamické [od -2 do +1]





- dotaz obsahuje AF
 - specifické zpracování dotazu
- From [všechna join]
- Where
- Group By [vytváření skupin]
- Having [eliminace skupin]
- dle AF [vytváření oblastí]
- AF aplikovány na každý řádek [další seskupování]
- Order By [závěrečné]
 - již neovlivní AF pořadí oken a oblastí





- funkce pořadí [Ranking function]
- agregační funkce oken [Windowing ag.f]
- Lag/Led []
- First/Last []
- rozšíření funkcionality Group By []
- další skupiny funkcí



- relativní přístup k řádkům [dynamický]
 - Lag
 - přístup k předchozímu řádku oblasti
 - Lead
 - přístup k následnému řádku oblasti
 - k jednomu z následných
 - First, First_value
 - přístup k prvnímu řádku oblasti
 - Last, Last_value
 - přístup k prvnímu řádku oblasti

```
Function (arg1,..., argn)  
  OVER ( [PARTITION BY <...>]  
        [ ORDER BY <....> ]  
        [ <window_clause> ] )
```



Lag (co, offset, default) OVER (over klauzule)

- co výraz Sql, který chceme vyčíslit
sloupec předchozí řádky
- offset relativní index do řádky **PŘED** řádkou běžnou
pozitivní integer, defalut 1

Lead (co, offset, default) OVER (over klauzule)

- co výraz Sql, který chceme vyčíslit
- offset relativní index do řádky **ZA** řádkou běžnou
pozitivní integer, defalut 1
- default hodnota pro řádky mimo okno

Lag, Lead - funkce přístupu k řádkům



```
Select  Jmeno,Plat
From    Zam8
Order By Plat Desc, Jmeno;
```

```
Select  Jmeno,Plat,
        Rank() Over (Order by Plat Desc) Rank
From    Zam8
Order By Plat Desc, Jmeno;
```

Jmeno	Plat
Bob	60 000
Zuzi	40 000
Jana	30 000
Tere	30 000
Adam	20 000
Bill	20 000
Dan	20 000
Pepa	20 000
Hana	10 000
Tom	10 000

JMENO	PLAT	RANK
Bob	60000	1
Zuzi	40000	2
Jana	30000	3
Tere	30000	3
Adam	20000	5
Bill	20000	5
Dan	20000	5
Pepa	20000	5
Hana	10000	9
Tom	10000	9

Done

Lag, Lead - funkce přístupu k řádkům



```
Select  Jmeno,Plat,  
        Rank() Over (Order by Plat Desc) Rank,  
        Lead(Plat,1,0) Over (Order by Plat Desc) Dalsi,  
        Lag (Plat,1,0) Over (Order by Plat Desc) Pred  
From    Zam8  
Order By Plat Desc, Jmeno;
```

JMENO	PLAT	RANK	DALSI	PRED
Bob	60000	1	40000	0
Zuzi	40000	2	30000	60000
Jana	30000	3	30000	40000
Tere	30000	3	20000	30000
Adam	20000	5	20000	30000
Bill	20000	5	20000	20000
Dan	20000	5	20000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	10000	20000
Tom	10000	9	0	10000

Done

Lag, Lead - funkce přístupu k řádkům



```
Select  Jmeno,Plat,  
        Rank() Over (Order by Plat Desc) Rank,  
        Lead(Plat,1,NULL) Over (Order by Plat Desc) Dalsi,  
        Lag (Plat,1,NULL) Over (Order by Plat Desc) Pred  
From    Zam8  
Order By Plat Desc, Jmeno;
```

Default - není-li hodnota

JMENO	PLAT	RANK	DALSI	PRED
Bob	60000	1	40000	0
Zuzi	40000	2	30000	60000
Jana	30000	3	30000	40000
Tere	30000	3	20000	30000
Adam	20000	5	20000	30000
Bill	20000	5	20000	20000
Dan	20000	5	20000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	10000	20000
Tom	10000	9	0	10000

JMENO	PLAT	RANK	DALSI	PRED
Bob	60000	1	40000	-
Zuzi	40000	2	30000	60000
Jana	30000	3	30000	40000
Tere	30000	3	20000	30000
Adam	20000	5	20000	30000
Bill	20000	5	20000	20000
Dan	20000	5	20000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	10000	20000
Tom	10000	9	-	10000

Done

Done

Lag, Lead - funkce přístupu k řádkům



```
Select  Jmeno,Plat,  
        Rank() Over (Order by Plat Desc) Rank,  
        Lead(Plat,2,NULL) Over (Order by Plat Desc) Dalsi2,  
        Lag (Plat,2,NULL) Over (Order by Plat Desc) Pred2  
From    Zam8  
Order By Plat Desc, Jmeno;
```

JMENO	PLAT	RANK	DALSI	PRED
Bob	60000	1	40000	-
Zuzi	40000	2	30000	60000
Jana	30000	3	30000	40000
Tere	30000	3	20000	30000
Adam	20000	5	20000	30000
Bill	20000	5	20000	20000
Dan	20000	5	20000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	10000	20000
Tom	10000	9	-	10000

JMENO	PLAT	RANK	DALSI2	PRED2
Bob	60000	1	30000	-
Zuzi	40000	2	30000	-
Jana	30000	3	30000	60000
Tere	30000	3	20000	40000
Adam	20000	5	20000	30000
Bill	20000	5	20000	30000
Dan	20000	5	10000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	-	20000
Tom	10000	9	-	20000

Lag, Lead - funkce přístupu k řádkům



```
Select  Jmeno,Plat,  
        Rank() Over (Order by Plat Desc) Rank,  
        Lead(Plat,2,NULL) Over (Order by Plat Desc) Dalsi2,  
        Lag (Plat,2,NULL) Over (Order by Plat Desc) Pred2  
From    Zam8  
Order By Plat Desc, Jmeno;
```

JMENO	PLAT	RANK	DALSI	PRED
Bob	60000	1	40000	-
Zuzi	40000	2	30000	60000
Jana	30000	3	30000	40000
Tere	30000	3	20000	30000
Adam	20000	5	20000	30000
Bill	20000	5	20000	20000
Dan	20000	5	20000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	10000	20000
Tom	10000	9	-	10000

JMENO	PLAT	RANK	DALSI2	PRED2
Bob	60000	1	30000	-
Zuzi	40000	2	30000	-
Jana	30000	3	20000	60000
Tere	30000	3	20000	40000
Adam	20000	5	20000	30000
Bill	20000	5	20000	30000
Dan	20000	5	10000	20000
Pepa	20000	5	10000	20000
Hana	10000	9	-	20000
Tom	10000	9	-	20000