



# Analītiskās funkcijas un SQL Model klauza

Gints Plivna  
gints.plivna@gmail.com

# Kas es esmu?

- Pieredze darbā ar Oracle kopš 1997
- Oficiālais “amats” – sistēmanalītiķis Rix Technologies
- Oracle sertificēto kursu pasniedzējs Affecto Latvija (Mebius IT)
- Autors vienīgajam pastāvīgajam online informācijas avotam par datubāzēm latviešu valodā  
(<http://datubazes.wordpress.com>)
- LVOUG biedrs

## Fakti

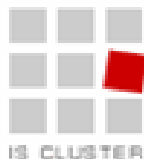
- Dibināts 2001.gada 1.jūnijā
- Darbības joma – IS izstrāde
- Microsoft Gold Certified Partner
- Oracle partneris
- LIKTA un IS klastera biedrs
- ISO 9001:2000 sertificēts uzņēmums



ISV/Software Solutions  
Information Worker Solutions  
Business Process and Integration



**LIKTA**  
Latvijas Informācijas  
un komunikācijas  
tehnoloģijas asociācija



## Projekti

- Kultūra
  - Digitālo kultūrkaršu projekts
  - Nacionālo muzeju krājuma kopkataloga IS
  - Digitālo objektu pārvaldības sistēmas LNB
  - Valsts vienotā arhīvu IS
- E-pārvalde
  - Elektronisko pakalpojumu izstrādes pilotprojekts
  - Latvijas valsts portāls
  - Elektroniskā iepirkumu sistēma
- Valsts nozīmes reģistri
  - Vienotās migrācijas IS
  - Latvijas nacionālā vīzu IS
- Integrēti dokumentu aprites risinājumi
  - Sistēma DAUKS
  - Pašvaldību e-pārvaldes centrālo lietojumu izstrāde
  - ĀM diplomātisko pārstāvniecību dokumentu apmaiņas sistēma
- Citas sistēmas
  - Juridiskās palīdzības administrēšanas IS
  - Jelgavas Domes Projektu pārvaldības risinājums

## Produkti

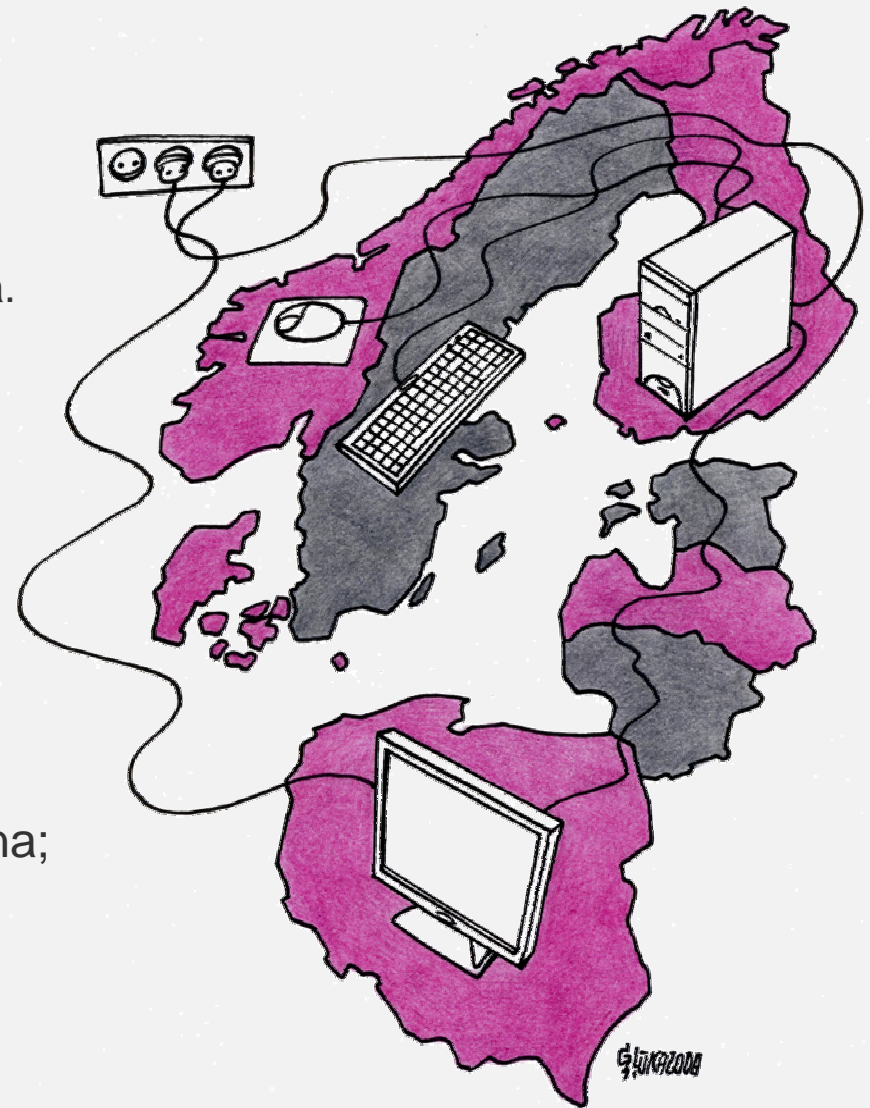
- Dokumentu aprites risinājumi
  - ePortfelis
  - Namejs
- MS Project Server projektu pārvaldības risinājums
- Elektroniskā iepirkumu sistēma



# Affecto


- Kompleksu IS izstrādes un ieviešanas kompānija.
- BI risinājumu līderis Skandināvijā.
- Pārstāvniecība 8 valstīs.
- 2007. gadā 97,5 mlj. EUR apgrozījums, 10,8 mlj. peļņa.
- Vairāk kā 1200 darbinieku.
- Publiska akciju sabiedrība, akcijas kotējas OMX Helsinki (AFE1V).

- 
- Vairāk kā 80 darbinieku Latvijā.
  - 2007. gadā 4,75 mlj. LVL apgrozījums.
  - Darbības virzieni:
    - Biznesa Inteligences risinājumi;
    - Kompleksu IS izstrāde, ieviešana, uzturēšana;
    - TIA ieviešana;
    - Oracle tehniskais atbalsts, ārpakalpojumi;
    - Oracle apmācības;
    - Licenču tirdzniecība.



Affecto

# Analītiskās funkcijas – daži fakti

- Bez maksas!!!
- Pieejamas visās redakcijās (*editions*) – XE, SE One, SE, EE, PE
- Nav nekā jauns – jau vismaz kopš 8.1.7.x versijas
- Sajūta, ka vairumam izstrādātāju diezgan svešas – kāpēc? 

# Analītiskās funkcijas – kāpēc?

*In my opinion, analytic functions are the coolest thing to happen to SQL since the SELECT keyword was introduced. [Thomas Kyte](#)*

- Parasti Select teikumā vienlaicīgi “redz” vienu ierakstu

1	Imants	Ogre
<b>2</b>	<b>Liene</b>	<b>Jelgava</b>
3	Santa	Jelgava

**Kā vienlaicīgi “redzēt” citus ierakstus?**

# Kā vienlaicīgi “redzēt” citus ierakstus? I

## 1. Veidojot pašsavienojumus (*self join*)

1	Imants	Ogre	2	Liene	Jelgava
<b>2</b>	<b>Liene</b>	<b>Jelgava</b>	<b>3</b>	<b>Santa</b>	<b>Jelgava</b>
3	Santa	Jelgava			

- Sarežģī vaicājumu
- Ne vienmēr iespējams

## 2. Grupējot

count(*)		1	Imants	?	Ogre
1	Ogre	2	Liene		<b>Jelgava</b>
<b>2</b>	<b>Jelgava</b>	3	Santa	?	<b>Jelgava</b>

- Zūd informācija no kurienes ieraksts cēlies
- Nav iespējams iegūt pārējās ieraksta kolonas

# Kā vienlaicīgi “redzēt” citus ierakstus? II

## 3. Pareizā atbilde – analītiskās funkcijas!

				count(*)
1	<b>Imants</b>	Ogre		1
<b>2</b>	<b>Liene</b>	<b>Jelgava</b>	<b>Imants</b>	<b>2</b>
3	Santa	Jelgava		2

The diagram illustrates the relationships between rows in the table. An arrow points from the word 'Imants' in the first row to the word 'Imants' in the second row. Another arrow points from the number '3' in the third row to the word 'Imants' in the second row. The second row is highlighted with a black border.

- Sarežģī vaicājumu – *bet mazāk*
- Ļoti elastīgas
- Var izdarīt to, ko principā ar savienojumiem un grupēšanu nevar panākt (vai arī tas prasa nesamērīgas pūles)



# Sintakse

**funkcija ([argumenti]) OVER (  
<partition by klauza>  
<order by klauza>  
<windowing klauza>)**

5	Juris	Ape
2	Liene	Jelgava
3	Santa	Jelgava
1	Imants	Ogre
4	Kaspars	Ogre
6	Kaspars	Ogre

**PARTITION BY pilseta  
ORDER BY vards, id**

- Funkcija kaut ko rēķina
- Partition by klauza sadala nodalījumos
- Order by klauza nosaka sakārtojumu nodalījuma ietvaros
- Windowing klauza tikai kopā ar Order by klauzu sīkāk nosaka f-  
jas darba apgabalu

# Funkcijas

- row\_number – numuri pēc kārtas (1, 2, 3, ..., n)
- rank, dense\_rank – vietu piešķiršana (1, 2, 2, ..., n, n)
- lag, lead – skats atpakaļ/uz priekšu
- first\_value, last\_value – pirmā/pēdējā vērtība pēc noteikta sakārtojuma
- min, max, avg, sum, count – grupēšanas funkciju analītiskie varianti
- ntile(N) – sadala ierakstu kopu N daļās
- ratio\_to\_report – daļa no kopsummas
- Kopā vairāk nekā 30 dažādas 10gR2
- Nepietiek? Veidojiet savas!

*Dokumentācijā Data Cartridge Developer's Guide -> Using User-Defined Aggregate Functions*

# Partition by klauza

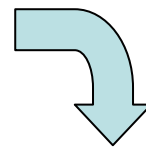
- Sadala ierakstu kopu nodalījumos
- Nav obligāta – ja nav, tad nodalījums ir visa ierakstu kopa
- Funkcijas darbojas tikai uz nodalījumu – tātad, ja nav, – tad uz visu ierakstu kopu
- Katrai funkcijai var būt sava



1	Imants	Ogre
5	Juris	Ape
4	Kaspars	Ogre
6	Kaspars	Ogre
2	Liene	Jelgava
3	Santa	Jelgava

~~PARTITION BY pilseta~~

ORDER BY vards, id



```
lag (vards) OVER (PARTITION BY pilseta ORDER BY id),  
count(*) OVER (PARTITION BY vards)
```

# Order by klauza

- Dažām funkcijām ir obligāta, piemēram, `row_number`, `rank`, `lag`, `ntile`
- Dažām funkcijām nedrīkst būt, piemēram, `ratio_to_report`
- Dažām (*windowing*) funkcijām var būt/var nebūt, piemēram, `first_value`, `count`
  - Jābūt uzmanīgam, jo *windowing* funkcijām izveido noklusēto logu no pirmās līdz pašreizējai rindai (skat. piemērus nākošā slaidā!)
  - Sakrītošās vērtības tiek uzskatītas par vienādām (skat. piemēru 3 slaidus tālāk)
  - NULL vērtības tiek uzskatītas par vienādām
  - `ORDER BY <kolona> [ASC|DESC] [NULLS FIRST|LAST]`

# Daži piemēri noklusētā loga ilustrēšanai

w -> count(\*) OVER ()

x -> count(\*) OVER (ORDER BY vards, id)

y -> count(\*) OVER (PARTITION BY pilseta)

z -> count(\*) OVER (PARTITION BY pilseta  
ORDER BY vards, id)

			w	x	y	z
1	Imants	Ogre	6	1	3	1
5	Juris	Ape	6	2	1	1
4	Kaspars	Ogre	6	3	3	2
6	Kaspars	Ogre	6	4	3	3
2	Liene	Jelgava	6	5	2	1
3	Santa	Jelgava	6	6	2	2

# Windowing klauza

- Tikai viena nodalījuma ietvaros
- Iespējams tikai tad, ja ir ORDER BY klauza
- Definē slīdošu logu ar vienu vai abām mainīgām malām
- Noklusēti no sākuma līdz pašreizējai rindai
- Loģisks intervāls *RANGE* (atkarībā no kolonas vērtības) vai fiziska nobīde *ROWS* (ieraksti uz augšu un/vai leju)
- Var būt

	Fiksēts sākumā		Fiksēts beigās		Slīdošs fiksēta platuma		Slīdošs mainīga platuma
1	Imants	1	Imants	1	Imants	1	Imants
5	Juris	5	Juris	5	Juris	5	Juris
4	Kaspars	4	Kaspars	4	Kaspars	4	Kaspars
6	Kaspars	6	Kaspars	6	Kaspars	6	Kaspars

# Vēl daži piemēri

```
SELECT id, vards, pilseta,  
row_number() OVER (ORDER BY vards, id) rn,  
lag(vards) OVER (PARTITION BY pilseta  
ORDER BY vards, id) iepr,  
lead(pilseta, 2) OVER (ORDER BY vards, id) nak2,  
sum(id) OVER (ORDER BY vards, id ROWS BETWEEN  
1 PRECEDING AND 1 FOLLOWING) sum,  
sum(id) OVER (ORDER BY vards) sum#
```

id	vards	pilseta	rn	iepr	nak2	sum	sum#
1	Imants	Ogre	1		Ogre	6	1
5	Juris	Ape	2		Ogre	10	6
4	Kaspars	Ogre	3	Imants	Jelgava	15	16
6	Kaspars	Ogre	4	Kaspars	Jelgava	12	16
2	Liene	Jelgava	5			11	18
3	Santa	Jelgava	6	Liene		5	21

# Ko nozīmē nesamērīgas pūles?

## Slīdošā summa (running total)

```
SQL> SELECT x1.id, x1.vards, sum(x2.id) summa
  2 FROM (
  3   SELECT rownum rn, id, vards
  4   FROM (
  5     SELECT id, vards
  6     FROM prs
  7     ORDER BY vards, id
  8   )
  9 ) x1 INNER JOIN (
10   SELECT rownum rn, id
11   FROM (
12     SELECT id
13     FROM prs
14     ORDER BY vards, id
15   )
16 ) x2
17 ON (x1.rn>=x2.rn)
18 GROUP BY x1.id, x1.vards
19 ORDER BY x1.vards, x1.id;
```

ID	VARDS	SUMMA
1	Imants	1
5	Juris	6
4	Kaspars	10
6	Kaspars	16
2	Liene	18
3	Santa	21

```
SQL> SELECT id, vards,
  2   sum(id) OVER (ORDER BY vards, id) summa
  3 FROM prs
  4 ORDER BY vards, id;
```


ID	VARDS	SUMMA
1	Imants	1
5	Juris	6
4	Kaspars	10
6	Kaspars	16
2	Liene	18
3	Santa	21



# Ātrdarbība

- Katra atšķirīgā partition by un order by klauza var prasīt savu kārtošanu (*sort*) atmiņā vai uz diska
- Var paralelizēt (kas nepietiekamu resursu gadījumā tikai padarīs visu vēl lēnāku)
- Atkarīga no versijas – salīdzinot “vienkāršus” (ar pašsavienojumiem) SQL teikumus ar to analītiskajiem variantiem (piemēram, lai atrastu katram ierakstam atbilstošo maksimālo vērtību):
  - 9.2 nedaudz ātrāks analītiskais variants
  - 10.2 ātrāks parastais variants (jo nāca klāt hash group by un hash unique operācijas), bet tomēr arī ātrāks kārtošanas algoritms...
- **Tātad jātestē!**

# SQL Model klauza – daži fakti

- Bez maksas!!
- Pieejamas visās redakcijās (*editions*) – XE, SE One, SE, EE, PE
- Nav nekas *loti* jauns – jau kopš 10.1 versijas
- Sajūta, ka vairums izstrādātāju vispār par tādu neko nezin un nav dzirdējuši. Es pat nejautāšu – kāpēc 

# Kas tas ir?

- Elektroniskā tabula datubāzē – formulas, aprēķini
- Tīrs SQL – nav lietotāja definētu funkciju, masīvu, objektu
- Noder
  - Prognozēšanai (*Forecasting*)
  - Iteratīviem aprēķiniem
  - Gadījumos, kad Jūs datus no Oracle eksportējat uz Microsoft Excel

# Vienkāršota sintakse

**SELECT <kolonas>**

**MODEL**

**<partition by klauza>**

**<dimension by klauza>**

**<measures klauza>**

**<rules klauza>**

- Partition by – līdzīga kā analītiskajām funkcijām
- Dimension by – nosaka dimensijas (x, y, z, ..., asis)
- Measures – definē aprēķināmās vienības (KO rēķināt)
- Rules – definē KĀ rēķināt

# Model *modelis*

Dimensijas

Nodalījums

Aprēķināmā kolona

Dimensijas	Nodalījums	Aprēķināmā kolona	
5	Juris	Ape	400
<b>10</b>	<b>Uvis</b>	<b>Ape</b>	<b>200</b>
2	Liene	Jelgava	450
3	Santa	Jelgava	350
<b>10</b>	<b>Uvis</b>	<b>Jelgava</b>	<b>200</b>
1	Imants	Ogre	500
4	Kaspars	Ogre	610
6	Kaspars	Ogre	230
<b>10</b>	<b>Uvis</b>	<b>Ogre</b>	<b>200</b>

MODEL

PARTITION BY (pilseta)

DIMENSION BY (varda, id)

MEASURES (alga)

**RULES (alga['Uvis', '10']=200)**

# any – norāda jebkuru dimensijas vērtību

1	Imants	Ogre	500
4	Kaspars	Ogre	610
6	Kaspars	Ogre	230
		<b>Ogre</b>	<b>1340</b>
2	Liene	Jelgava	450
3	Santa	Jelgava	350
		<b>Jelgava</b>	<b>800</b>
5	Juris	Ape	400
		<b>Ape</b>	<b>400</b>

MODEL

PARTITION BY (pilseta)

DIMENSION BY (0 flag, vards, id)

MEASURES (alga)

RULES (alga[1, null, null]=sum(alga)[0, **any**, **any**]);

# Kredīta kalkulators – rekursīvie aprēķini

Sāk summa			1000
Procenti			10
Ilgums mēnešos			5
Mēneša maksājums	200,00		
<b>Atlik. summa</b>	<b>Men. maksa</b>	<b>Proc</b>	<b>Kopā</b>
1000,00	200,00	8,33	208,33
800,00	200,00	6,67	206,67
600,00	200,00	5,00	205,00
400,00	200,00	3,33	203,33
200,00	200,00	1,67	201,67

Sāk summa			1000
Procenti			10
Ilgums mēnešos			5
Mēneša maksājums			=D1/D3
<b>Atlik. summa</b>	<b>Men. maksa</b>	<b>Proc</b>	<b>Kopā</b>
=D1	=\$D\$4	=ROUND(A7*\$D\$2/100/12;2)	=B7+C7
=A7-B7	=\$D\$4	=ROUND(A8*\$D\$2/100/12;2)	=B8+C8
=A8-B8	=\$D\$4	=ROUND(A9*\$D\$2/100/12;2)	=B9+C9
=A9-B9	=\$D\$4	=ROUND(A10*\$D\$2/100/12;2)	=B10+C10
=A10-B10	=\$D\$4	=ROUND(A11*\$D\$2/100/12;2)	=B11+C11

- **RULES ITERATE (N)** – cik reizes veikt iterāciju
- **ITERATION\_NUMBER** – iterācijas numurs

```

SELECT m menesis, to_char(summa, '999999.00') summa,
       to_char(men_maks, '999999.00') men_maks,
       to_char(proc, '999999.00') proc,
       to_char(kop_maks, '999999.00') kop_maks
FROM dual
MODEL DIMENSION BY (-1 m)
MEASURES (1000 summa,
          round(1000 * 10/100/12,2) proc,
          ceil(1000 / 5 * 100)/100 men_maks,
          ceil(1000 / 5 * 100)/100 + round(1000 * 10/100/12,2) kop_maks)
RULES ITERATE (4) (
summa[iteration_number] = summa[iteration_number-1] - men_maks[iteration_number-1],
proc[iteration_number] = round(summa[iteration_number] * 10/100/12,2),
men_maks[iteration_number] = least(ceil(1000/5*100)/100, summa[iteration_number]),
kop_maks[iteration_number] = men_maks[iteration_number] + proc[iteration_number]);

```

MENESIS	SUMMA	MEN_MAKS	PROC	KOP_MAKS
-1	1000.00	200.00	8.33	208.33
0	800.00	200.00	6.67	206.67
1	600.00	200.00	5.00	205.00
2	400.00	200.00	3.33	203.33
3	200.00	200.00	1.67	201.67



Sāk sumn		1000
Procenti		10
Ilgums mē		5
Mēneša m		=D1/D3

Atlik. summa	Men. maksā	Proc	Kopā
=D1	=\$D\$4	=ROUND(A7*\$D\$2/100/12;2)	=B7+C7
=A7-B7	=\$D\$4	=ROUND(A8*\$D\$2/100/12;2)	=B8+C8
=A8-B8	=\$D\$4	=ROUND(A9*\$D\$2/100/12;2)	=B9+C9
=A9-B9	=\$D\$4	=ROUND(A10*\$D\$2/100/12;2)	=B10+C10
=A10-B10	=\$D\$4	=ROUND(A11*\$D\$2/100/12;2)	=B11+C11

```
SELECT m menesis, to_char(summa, '999999.00') summa,
       to_char(men_maks, '999999.00') men_maks,
       to_char(proc, '999999.00') proc,
       to_char(kop_maks, '999999.00') kop_maks
```

```
FROM dual
```

```
MODEL DIMENSION BY (-1 m)
```

```
MEASURES (1000 summa,
```

```
         round(1000 * 10/100/12,2) proc,
```

```
         ceil(1000 / 5 * 100)/100 men_maks,
```

```
         ceil(1000 / 5 * 100)/100 + round(1000 * 10/100/12,2) kop_maks)
```

```
RULES ITERATE (4) (
```

```
summa[iteration_number] = summa[iteration_number-1] - men_maks[iteration_number-1],
```

```
proc[iteration_number] = round(summa[iteration_number] * 10/100/12,2),
```

```
men_maks[iteration_number] = least(ceil(1000/5*100)/100, summa[iteration_number]),
```

```
kop_maks[iteration_number] = men_maks[iteration_number] + proc[iteration_number]);
```

# cv() – atsauce uz pašreizējo vērtību

```
SELECT rn, pilseta, vards FROM prs
MODEL
PARTITION BY (pilseta)
DIMENSION BY (
    row_number() over (partition by pilseta order by id) rn)
MEASURES (cast (vards as varchar2(30)) vards)
RULES (
    vards[any] order by rn desc = vards[cv()] || ',' || vards[cv()+1]
);
```

RN	PILSETA	VARDS
1	Ogre	Imants,Kaspars,Kaspars,
2	Ogre	Kaspars,Kaspars,
3	Ogre	Kaspars,
1	Jelgava	Liene,Santa,
2	Jelgava	Santa,
1	Ape	Juris,

# Saistītie informācijas avoti

- Oficiālā dokumentācija (<http://tahiti.oracle.com>) attiecīgās versijas Oracle® Database Data Warehousing Guide nodaļas SQL for Analysis and Reporting un SQL for Modeling
- [http://www.quest-pipelines.com/newsletter-v7/0706\\_B.htm](http://www.quest-pipelines.com/newsletter-v7/0706_B.htm)
- <http://rwijk.blogspot.com/2007/10/sql-model-clause-tutorial-part-one.html>
- <http://volder-notes.blogspot.com>
- **Šī prezentācija** (un arī citas interesantas lietas) - <http://datubazes.wordpress.com>

# Jūs *(droši vien)* neatcerēsieties sintaksi, bet varat atcerēties idejas!

- Analītiskās funkcijas
  - Redz ierakstus uz priekšu un atpakaļ ierakstu kopā
  - Grupēšanas funkcijas saglabājot detaļas
  - Numurēšana, vietu piešķiršana, pirmā un pēdējā, lielākā un mazākā vērtība saglabājot detaļas
- SQL model klauza
  - Rekursīvi rēķini
  - MS Excel SQLā
  - Prognozēšana

**Paldies!**