



Oracle datu vārdnīca

Gints Plivna
gints.plivna@gmail.com

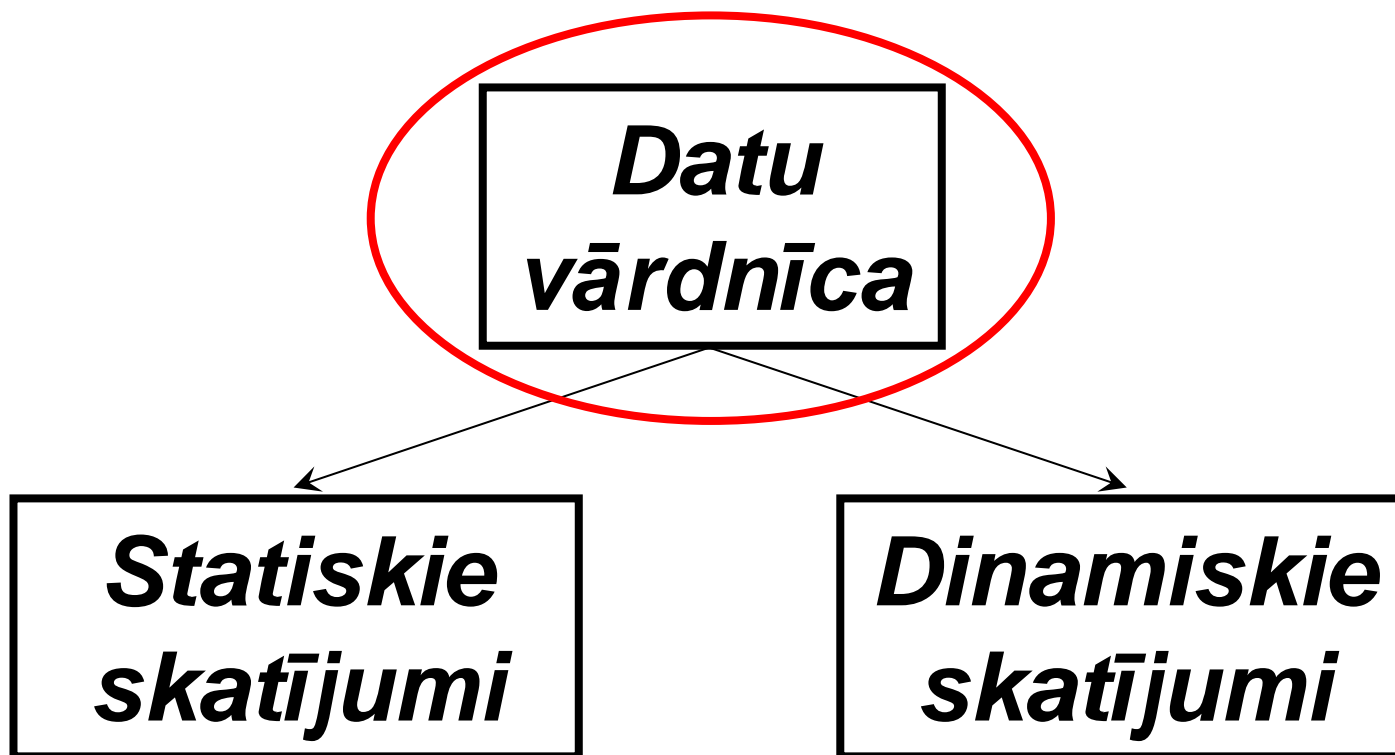
Kas es esmu?

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

Datu vārdnīca – definīcija

Tabulas un skatījumi, no kuriem iegūt informāciju par datubāzes objektiem un tās aktivitāti

Datu vārdnīcas uzbūve



Ar ko sākt?

DICT

```
SQL> desc dict
```

Name	Null?	Type
-----	-----	-----
TABLE_NAME		VARCHAR2(30)
COMMENTS		VARCHAR2(4000)

Ar ko sākt? II

```
SQL> SELECT * FROM dict
      2  WHERE table_name LIKE '%INDEX%'
      3  ORDER BY table_name;
```

TABLE_NAME

COMMENTS

ALL_INDEXES

Descriptions of indexes on tables accessible to the user

ALL_INDEXTYPE_ARRAYTYPES

All array types specified by the indextype

...

25 rows selected.

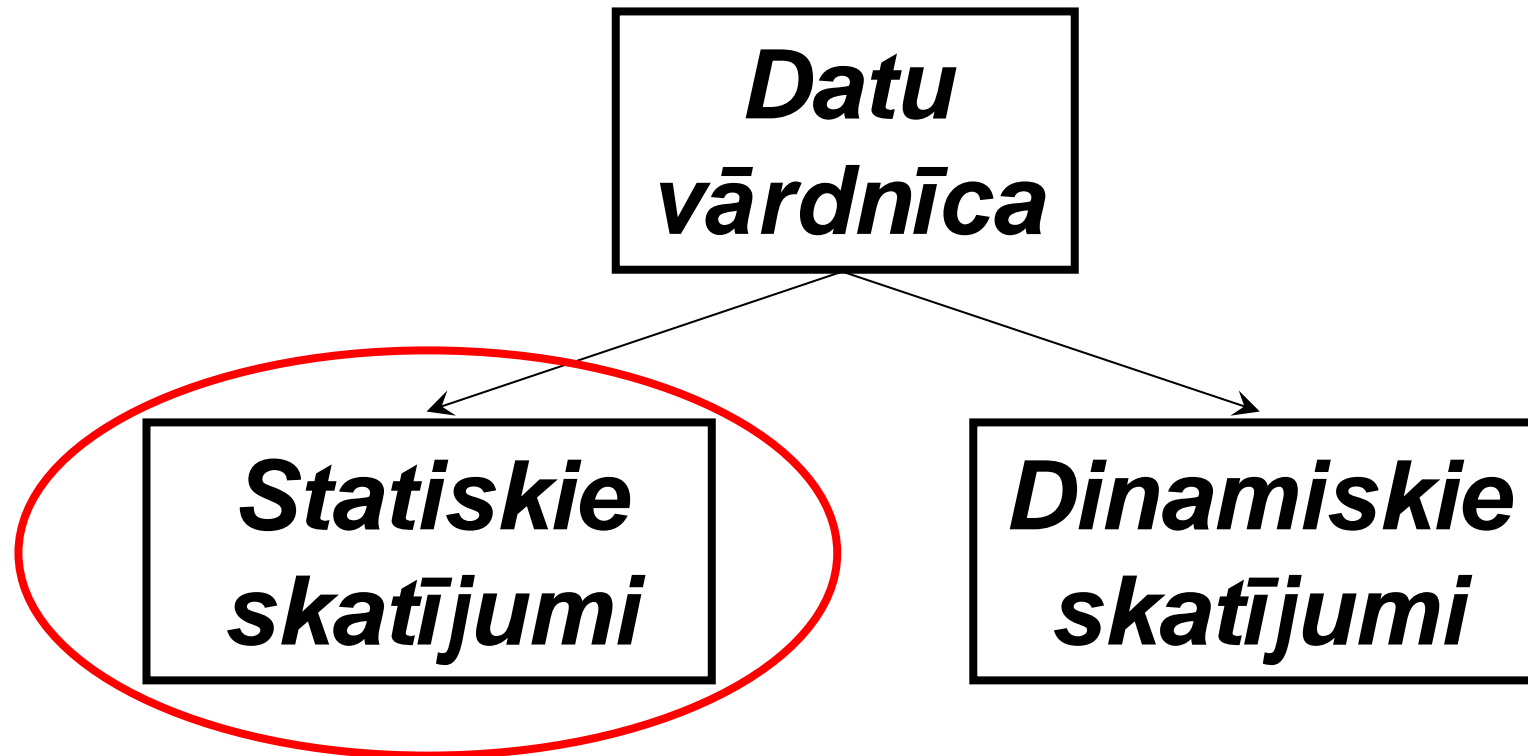
SELECT count(*) FROM dict;

9.2.0.7 EE	1283
10.2.0.1 XE	1821
10.2.0.1 SE	1872
10.2.0.1 EE	1872
11.1.0.6 EE	2323
11.2.0.1 SE	2553
MySQL 5.1	21*
SQL Server 2008	20**

* INFORMATION_SCHEMA skatījumi – vēl ir ~40 show komandas, kas tos dublē un ~25 tabulas MySQL datubāzē

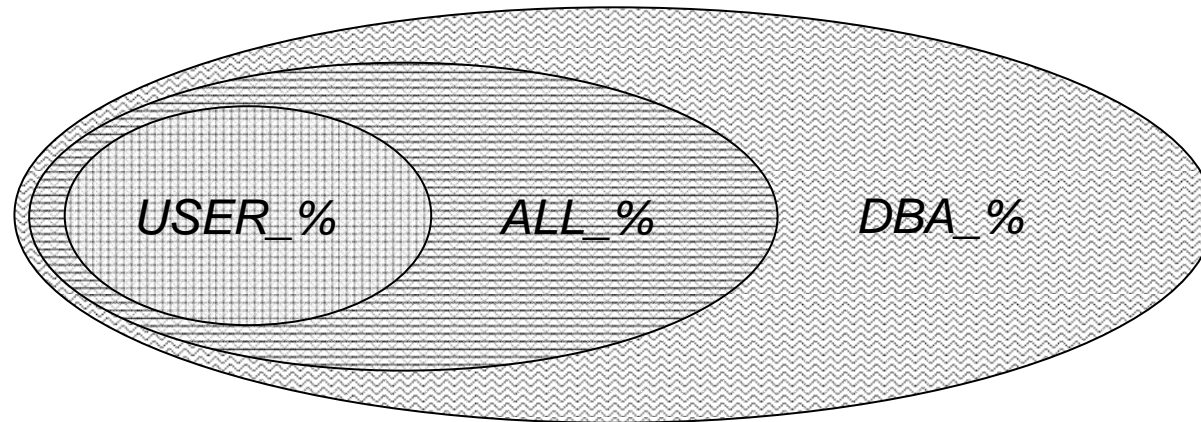
** INFORMATION_SCHEMA skatījumi – vēl ir ~350 sys shēmas tabulas, kam aptuveni atbilst Oracle 11.2.0.1. ~960 sys shēmas tabulas

Datu vārdnīcas uzbūve



Statiskie skatījumi - definīcija

- Sastāv no 3 daļām

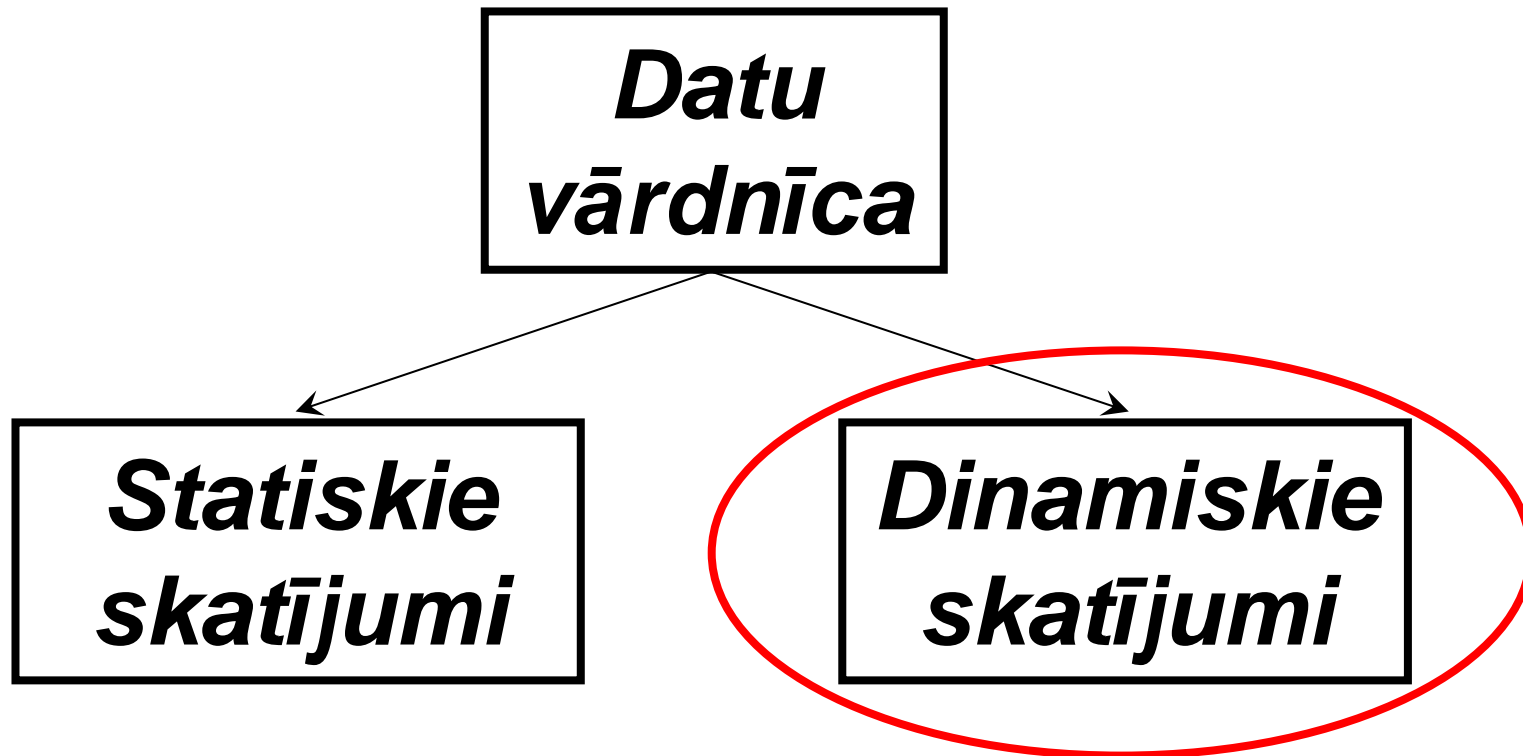


- USER_% - lietotāja paša shēmas objekti
- ALL_% lietotājam pieejamie objekti (izmantojot tiešās tiesības un lomas)
- DBA_% - visi datubāzes objekti (pieejami, piemēram, ja lomas DBA vai SELECT_CATALOG_ROLE)

Statiskie skatījumi II

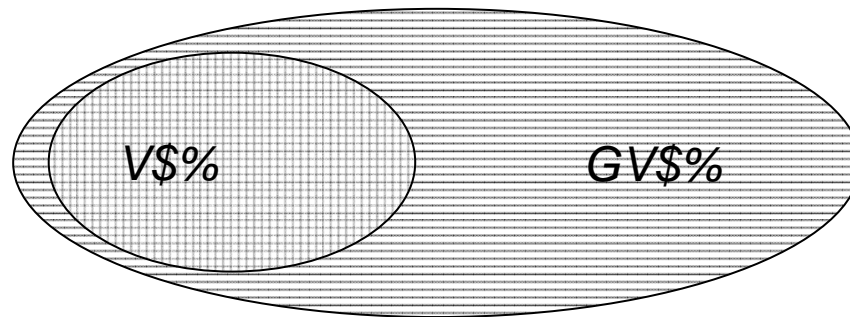
- USER un DBA skatījumos papildus OWNER kolona
- Ne vienmēr ir visi varianti – piemēram ir tikai DBA / ALL_DIRECTORIES
- Daži izņēmumi – piemēram AUDIT_ACTIONS, ROLE_ROLE_PRIVS, DUAL
- Piemēri
 - USER/ALL/DBA_OBJECTS
 - USER/ALL/DBA_TABLES
 - USER/ALL/DBA_SOURCE
 - USER/DBA_RECYCLEBIN
 - USER/ALL/DBA_TAB_COLUMNS

Datu vārdnīcas uzbūve



Dinamiskie skatījumi - definīcija

- Sastāv no 2 daļām



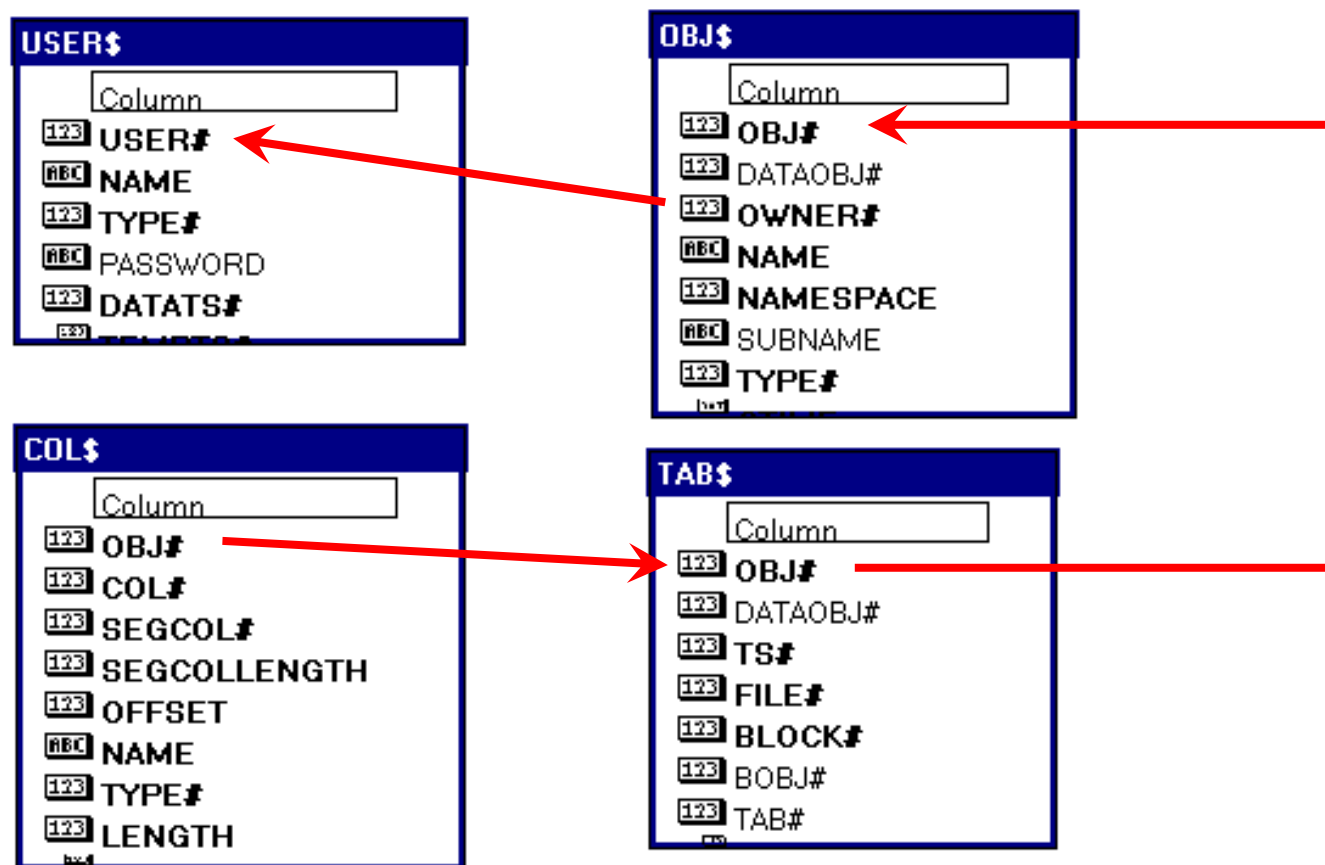
- V\$% - konkrētā pieslēguma instances informācija
- GV% - visu instanču informācija (RAC gadījumā)

Dinamiskie skatījumi II

- GV\$ skatījumos papildus INST_ID kolona
- Piemēri:
 - V\$INSTANCE
 - V\$MYSTAT, V\$STATNAME
 - V\$SQL, V\$SQL_PLAN
 - V\$TRANSACTION

Kas lācītim vēderā?

- Kā visvieglāk glabāt datus datubāzē?
 - Tabulās, protams.



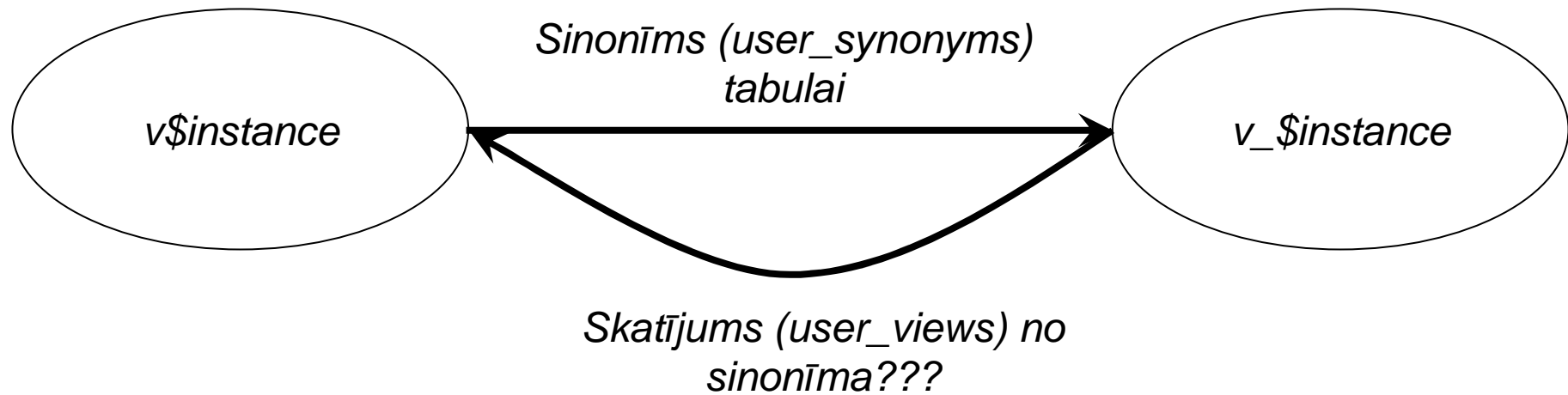
USER_TABLES definīcija

```
SQL> SELECT text
      2 FROM dba_views
      3 WHERE view_name = 'USER_TABLES';
```

TEXT

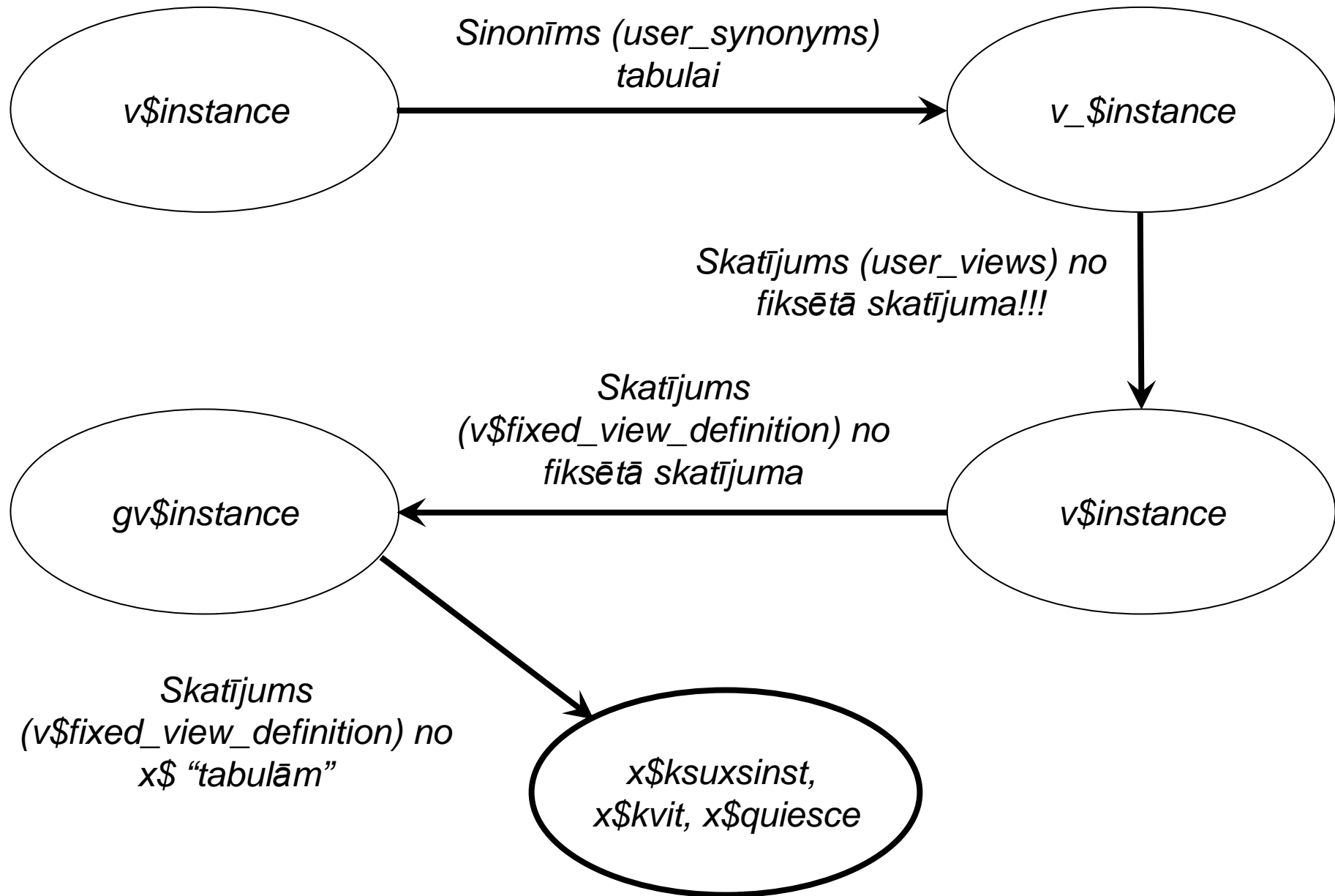
```
-----
select o.name,
[.]
from sys.ts$ ts, sys.seg$ s, sys.obj$ co, sys.tab$ t,
     sys.obj$ o, sys.obj$ cx, sys.user$ cu, x$ksppcv ksppcv,
     x$ksppi ksppi
where o.owner# = userenv('SCHEMAID')
     and o.obj# = t.obj#
[.]
```

Dinamisko skatījumu izcelsme



Cikls???

Dinamisko skatījumu izcelsme



x\$ “tabulas”

- Atmiņas struktūras, kas pieejamas kā tabulas; tām var izpildīt SQL vaicājumus;
- Saraksts skatījumā v\$fixed_table;
- Nosaukumi kā abreviatūras, piemēram, x\$bh (*buffer header*), x\$ksuse (*kernel service user session*)
- Lasīšana var bloķēt atmiņas struktūras izmaiņas!

Kāds no tā visa labums?

- Interesanti ;-)
- Dažādu problēmu, piemēram, ātrdarbības trūkumu identificēšana;
- Izmanto *GUI* rīki, bet, ja tie nav pieejami?
- Monotonu darbu automatizēšana;
- Dinamisku un iepriekš nezināmu struktūru apstrāde.

Monotonu darbu automatizēšana

```
SQL> SELECT 'DROP TABLE ' || table_name || ';'
      2 FROM user_tables;
```

```
'DROPTABLE' || TABLE_NAME || ';'
-----
```

```
DROP TABLE TAB1;
DROP TABLE KLIENTI;
```

```
SQL> DROP TABLE TAB1;
```

```
Table dropped.
```

```
SQL> DROP TABLE KLIENTI;
```

```
Table dropped.
```

Monotonu darbu automatizēšana II

```
SQL> SELECT 'ALTER SYSTEM KILL SESSION ''' || sid || ',' ||  
  serial# || ''';'  
 2 FROM v$session WHERE username = 'GINTS';
```

```
'ALTERSYSTEMKILLSESSION''' || SID || ',' || SERIAL# || ''';'
```

```
ALTER SYSTEM KILL SESSION '145,50636';  
ALTER SYSTEM KILL SESSION '159,10087';
```

```
SQL> ALTER SYSTEM KILL SESSION '145,50636';  
ALTER SYSTEM KILL SESSION '145,50636'  
*
```

```
ERROR at line 1:  
ORA-00027: cannot kill current session
```

```
SQL> ALTER SYSTEM KILL SESSION '159,10087';
```

```
System altered.
```

Dinamisku struktūru apstrāde

- Uzdevums - arhivēšana
 - Periodiski pārkopēt datus no vienas shēmas otrā.
 - Laika gaitā tabulu un to kolonu skaits augs;
 - Dati diemžēl var būt jau priekšā – tad tie jākorģē.
- Risinājums – dinamisks MERGE sql teikums, ko ģenerē no datu vārdnīcas

Arhivēšana - kursori

```
CURSOR c_cdf_tables IS
SELECT column_name, table_name
FROM all_tab_columns
WHERE owner = v_TEMP_user
      AND (column_name LIKE '____ID');
```

```
CURSOR c_cdf_columns (pi_table_name IN VARCHAR2) IS
SELECT column_name, data_type
FROM all_tab_columns
WHERE owner = v_TEMP_user
      AND table_name = pi_table_name
ORDER BY column_id;
```

Arhivēšana – kolonu klauzas definēšana

```
FOR j IN c_cdf_columns (i.table_name) LOOP
  -- update klauzā matching kolonu nevajag
  IF i.column_name <> j.column_name THEN
    v_upd_clause := v_upd_clause || v_upd_comma ||
      'trg.' || j.column_name || ' = src.' || j.column_name;
    v_upd_comma := ', ';
  END IF;
  v_ins_trg_clause := v_ins_trg_clause || v_ins_comma ||
    j.column_name;
  v_ins_src_clause := v_ins_src_clause || v_ins_comma ||
    'src.' || j.column_name;
  v_ins_comma := ', ';
END LOOP;
```


Arhivēšana – MERGE sql teikums

```
FOR i IN c_cdf_tables LOOP
    v_upd_comma := NULL; <tas pats ar citiem mainīgajiem>
    <skat kursoru iepriekšējā lapā>
    -- lipinām sqlu kopā
    v_sql :=
'MERGE INTO ' || v_DB_user || '.' || i.table_name || ' trg
USING ' || v_TEMP_user || '.' || i.table_name || ' src
ON (trg.' || i.column_name || ' = src.' || i.column_name ||
')
WHEN MATCHED THEN UPDATE SET ' || v_upd_clause ||
' WHEN NOT MATCHED THEN INSERT (' || v_ins_trg_clause || ')
VALUES (' || v_ins_src_clause || ');
EXECUTE IMMEDIATE v_sql;
END LOOP;
```

Arhivēšana - kopsavilkums

- Tas kļūst sliktāk, kad jākopē tabulas, kuru kolonas ir tabulas (*nested tables*)
- Izmantotie skatījumi:
 - all_tables
 - all_tab_columns
 - all_nested_tables
 - all_coll_types
 - all_type_attrs

Ja Jūs to visu jau zinājāt..

..vai zinājāt pat vairāk nekā šeit
stāstīts,..

..tad ziniet, ka Rix Technologies
meklē PL/SQL programmētāju ;-)

Jautājumi?