

Praktikum 2: Andmebaasi juhtimine, hoidla struktuurid

I Andmebaasi juhtimine

Andmebaasi instantsi käivitamine ja peatamine.

Listeneri käivitamine ja peatamine.

Instantsi informatsiooni vaatamine failist alert.log.

Andmebaasi kohta informatsiooni vaatamine:

- *Initsialiseerimise parameetrite vaatamine.*
- *Andmebaasi versiooni, staatuse jm. vaatamine.*

Pärast selle praktikumi läbimist üliõpilane teab, kuidas:

- peatada ja käivitada andmebaasi ja listeneri;
- otsida informatsiooni andmebaasi ja instantsi kohta;

Andmebaasi instantsi käivitamine ja peatamine

Vaatame, kuidas toimub utiliidi SQL*Plus abil andmebaasi instantsi käivitamine ja peatamine.

Selleks tuleb sisse logida kasutajana *sys* ning käivitada vastavalt käsk *startup* või *shutdown*.

Võib kasutada kataloogis *\$ORACLE_HOME/config/scripts* asuvad skripte *startdb.sh* ja *stopdb.sh*, kuid kasutaja peab olema grupis *dba*.

Meil *\$ORACLE_HOME* tähendab kausta

```
/usr/lib/oracle/xe/app/oracle/produkt/10.2.0/server/
```

Juhul, kui Oracle on konfigureeritud selliselt, et ta käivitub alglaadimisel, saavad Linux'i kasutajad kasutada järgmisi käsk:

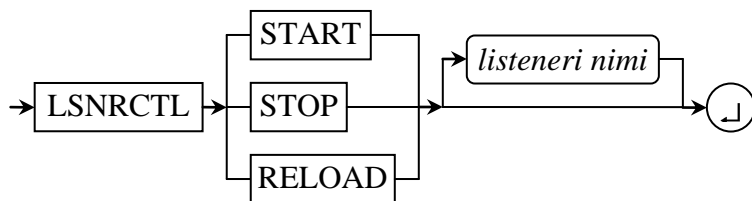
```
$> /etc/init.d/oracle-xe start  
$> /etc/init.d/oracle-xe stop  
$> /etc/init.d/oracle-xe restart
```

NB! Kasutajal peab olema root õigused.

Andmebaasi listeneri käivitamine ja peatamine

Listeneri saab juhtida utiliidi *lsnrctl* abil, mida käivitatakse tavalisest terminaali aknast.

Utiliidi *lsnrctl* käivitamise süntaks:



START – käivitab listeneri;

STOP – peatab listeneri;

RELOAD – loeb sisse uued listeneri konfiguratsiooni muudatused, selleks et poleks vaja peatada ja taaskäivitada listeneri.

listeneri nimi – mittekohustuslik parameeter, mis tähistab listeneri, millega soovitakse töötada.

NB! Utiliidi *lsnrctl* kasutamiseks peab kasutaja olema grupis *dba*.

Instantsi informatsiooni vaatamine failist *alert.log*

Fail, kuhu kronoloogiliselt salvestatakse andmebaasi teateid ja vigu, mis tekivad igapäevases kasutamises, kannab nime *alert.log*. Need teated sisaldavad järgmist informatsiooni:

- info instantsi käivitamise ja peatamise kohta;
- sõnumid operaatori konsoolile;
- andmebaasile, tabeliruumidele ja tagasikerimise segmentidele rakendatud CREATE;
- ALTER ja DROP SQL laused;
- *Trace* failide tekkimise vead;
- ORA-00600 (sisemised) vead;
- ORA-00060 vead (lukustused);
- ...

alert.log on tekstifail, mida saab avada iga tekstiredaktoriga. Kataloog, kus asub *alert.log*, on määratud initsialiseerimise parameetri *background_dump_dest* abil. Meil asub *alert_XE.log* kaustas */usr/lib/oracle/xe/app/oracle/admin/XE/bdump*.

Andmebasi informatsiooni vaatamine: initsialiseerimise parameetrid

Andmebaasi vaated, mis on seotud initsialiseerimise parameetritega:

- *v\$parameter* – väljastab informatsiooni initsialiseerimise parameetrite kohta, mis kehtivad antud sessiooni jaoks. Veerge *isses_modifiable* ja *issys_modifiable* saab

kasutada selleks, et määrata kas parameetrit on võimalik muuta sessiooni tasemel ALTER SESSION lausendiga või süsteemi tasemel ALTER SYSTEM lausendiga.

- *v\$pparameter* – väljastab informatsiooni serveri parameetrifaili sisu kohta.

Initsialiseerimise parameetreid saab vaadata ka

```
SQL> SHOW PARAMETER parameetri nimi
```

käsu abil.

Vaate *v\$parameter* kasutamine parameetri *background_dump_dest* väärtuse vaatamiseks:

```
SQL> SELECT value
1 FROM v$parameter
2 WHERE lower(name) = lower('background_dump_dest');
```

Andmebaasi informatsiooni vaatamine: Andmebaasi versioon ja staatus

Andmebaasi vaated, mis on seotud informatsiooniga andmebaasi kohta

- *v\$databse* – väljastab informatsiooni andmebaasi kohta, kontrollfailist;
- *v\$instance* – väljastab informatsiooni jooksva andmebaasi instantsi seisundi kohta;
- *v\$version* – väljastab jooksva andmebaasi versiooni;
- *v\$sga* – näitab, kui palju mälu kasutab *shared global area* (SGA).

SGA vaatamiseks saab kasutada ka

```
SQL> SHOW SGA
```

käsku.

Näidispäring andmebaasi ARCHIVELOG seisundi määramiseks :

```
SQL> SELECT log_mode
1 FROM v$databse;
```

Ülesanded

1. Logige sisse andmebaasi kasutajana:

- a) *HR* (parool: *HR*) (või hr/hr)
- b) *sys*

Pange kirja, kas sisse logimine õnnestus või ebaõnnestus, miks?

2. Kasutajana *sys*:

Käivitada sql- lause:

```
alter user hr account unlock;
```

3. Millised tabelid on kasutajal *HR*?

```
select table_name, owner from dba_tables where owner = 'HR';
```

4. Logida uuesti kasutajana *HR*.

5. Peatada listener
6. Teatada Oracle instants.
7. Käivitage Oracle instants ja listener.
8. Missugune on andmebaasi versioon ?

```
select * from v$version;
```

II osa Hoidla struktuurid

Teemad

Andmefailid.

Tabeliruumid, extendid, segmendid, andmeblokid.

Tabeliruumi kvoodid.

Päringud mis on seotud andmefailidega ja tabeliruumidega

Pärast selle praktikumi läbimist üliõpilane teab kuidas:

- luua, redigeerida ja kustutada tabeliruumi;
- muuta andmefaili suurust;
- saada infot andmebaasi andmefailide ja tabeliruumide kohta.

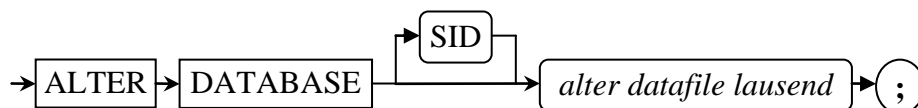
Andmefailid

Andmefailid on füüsilised struktuurid, millele vastavad operatsioonisüsteemi failid, kus asub Oracle andmebaas. Andmefailid võivad olla seotud ainult ühe andmebaasiga ning ainult ühe tabeliruumiga. Andmeid tabeliruumiga assotsieeritud skeemi objektidega hoitakse füüsiliselt ühes või mitmes andmefailis, mis kuuluvad antud tabeliruumi. Andmefaili loomisel hõivatakse spetsifitseeritud kettaruumi kogus faili ja faili päise jaoks.

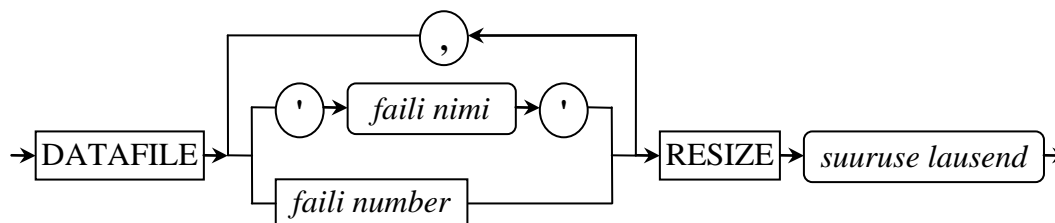
Andmebaasi vaated, mis on seotud andmefailidega:

- *dba_data_files* – kirjeldab kõiki andmebaasi andmefailide;
- *v\$datafile* – sisaldab infot andmefaili(de) kohta kontrollfailist;
- *v\$datafile_header* – sisaldab infot andmefaili päise kohta.

Andmefaili suuruse muutmise käsu süntaks:



Alter datafile lausend:



Faili numbri saab teada vaatest *dba_data_files*, veeru *FILE_ID* järgi.

Näidispäring andmefaili suuruse muutmiseks:

```
SQL> ALTER DATABASE DATAFILE 5 RESIZE 100M;
```

```
SQL> ALTER DATABASE DATAFILE 'andmefaili nimi.dbf' RESIZE 100M;
```

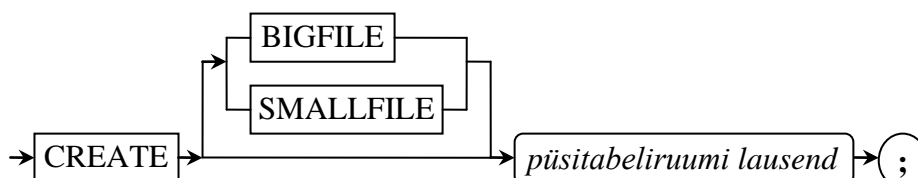
Tabeliruumid

Loogilisel tasemel hoiab Oracle andmeid tabeliruumides. Tabeliruumid võivad koosneda ühest või mitmest andmefailist, mis on jagatud loogilisteks hoidla ühikuteks – segmentideks. Segmentid koosnevad laienditest (extent), mis omakorda koosnevad fikseeritud arvust järjestikku asuvatest andmeblokkidest (data block). Andmebloki suurus on määratud parameetriga *db_block_size*.

Andmebaasi vaated, mis on seotud tabeliruumidega, segmentidega, laienditega ja andmeblokkidega:

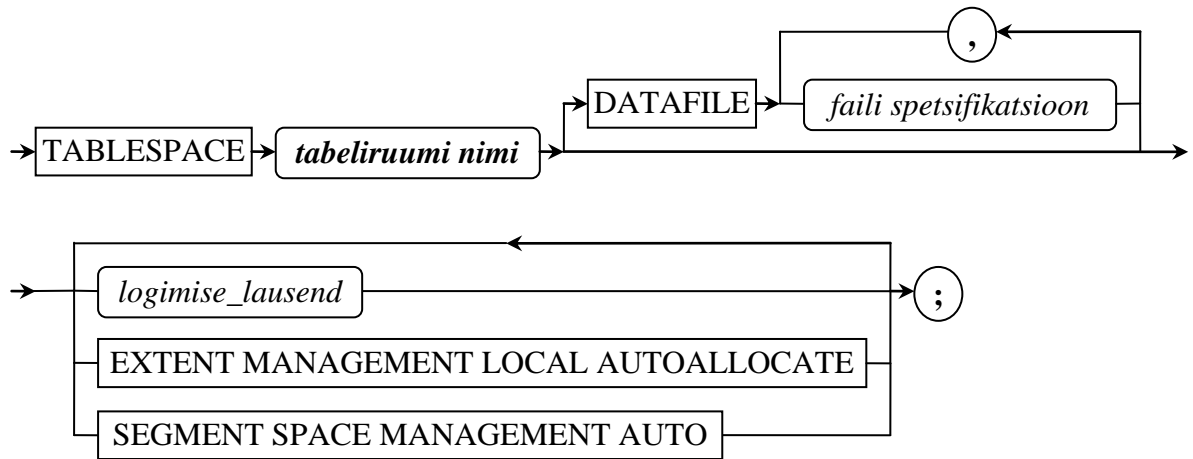
- *dba_tablespace* – kirjeldab kõiki tabelirume andmebaasis;
- *dba_segments* – kirjeldab andmebaasi kõikide segmentide jaoks eraldatud ruumi;
- *dba_extents* – kirjeldab laiendeid, millest koosnevad segmentid andmebaasi kõikides tabeliruumides;
- *dba_free_space* – kirjeldab vabu laiendeid kõikides andmebaasi tabeliruumides.

Tabeliruumi loomise käsu süntaks:

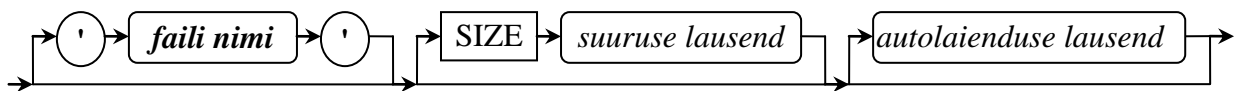


Tabeliruum *BIGFILE* koosneb ühest andmefailist, mis võib olla kuni 4GB suur (32-bitilise süsteemi puhul).

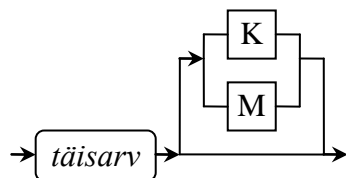
Püsitabeliruumi lausend:



Faili spetsifikatsioon:

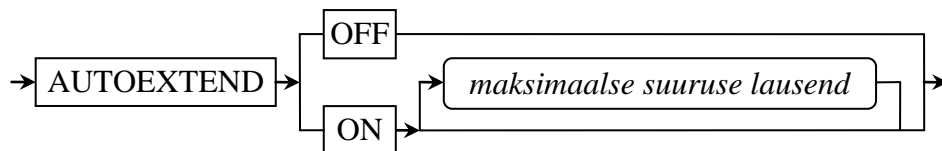


Suuruse lausend:

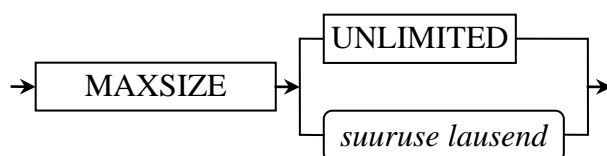


K – kilobaidid, M – megabaidid.

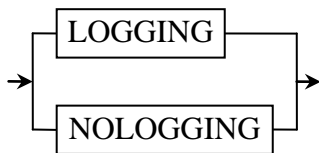
Automaatse suurendamise lausend:



Maksimaalse suuruse lausend:

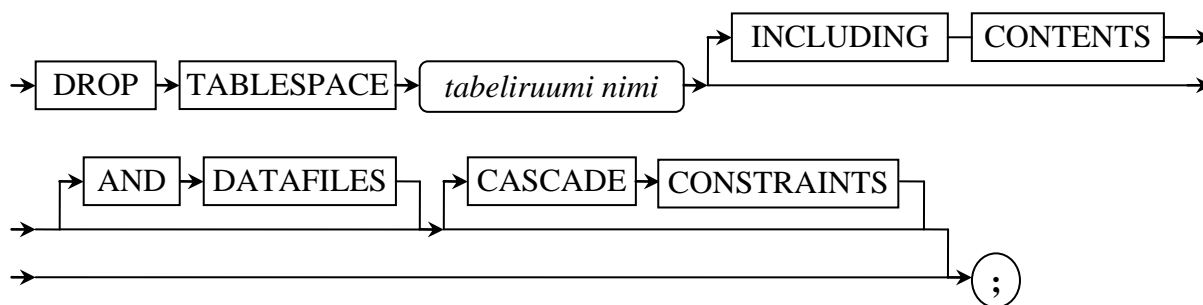


Logimise lausend:



Võtmesõna *LOGGING* kasutatakse siis, kui on soov salvestada tegevusi andmebaasi objektidega taastelogifaili.

Tabeliruumi kustutamise käsu süntaks



INCLUDING CONTENTS – on vajalik siis, kui soovitakse kustutada tabeliruum koos temas asuvate objektidega.

AND DATAFILES – ütleb andmebaasile, et on vaja kustutada ka tabeliruumiga assotsieeritud andmefailid. Oracle andmebaas kirjutab vastavad kustutamise teated faili *alert.log*.

CASCADE CONSTRAINTS – on vajalik siis, kui eksisteerivad võõrvõtmed tabeli(te)st, mis asuvad väljaspool kustutavat tabeliruumi, tabeli(te)sse, mis asuvad kustutavas tabeliruumis.

Tabeliruumi kvoodid

Tabeliruumidele võime määrata kvoodi – st. piirata kasutaja poolt kasutatavat tabeliruumi suurust.

Andmebaasi vaade, mis on seotud tabeliruumi kvootidega

- *dba_ts_quotas* – kirjeldab tabeliruumide kvoote kõikide kasutajate jaoks. Veergude *max_bytes* ja *max_blocks* väärtused „-1” tähendavad piiramatut kvoodi suurust (*unlimited*).

Näited päringutest, mis on seotud andmefailidega ja tabeliruumidega

1. Info tabeliruumide kohta:

```
SQL> SELECT tablespace_name AS "Tabeliruumi Nimi",
1 dts.contents AS "Tüüp",
2 ROUND(SUM(ddf.bytes/1024/1024),2) AS "Suurus (MB)",
3 ROUND(100-(SUM(NVL(dfs.bytes,0))/SUM(ddf.bytes)*100),2) AS "%Hõivat."
4 FROM dba_data_files ddf
5 JOIN dba_tablespaces dts USING (tablespace_name)
6 LEFT OUTER JOIN dba_free_space dfs USING (tablespace_name)
7 GROUP BY tablespace_name, contents;
```

2. Info andmefailide kohta:

```
SQL> SELECT file_name AS "File Name",
1 tablespace_name AS "Tablespace Name",
2 round(bytes/1024/1024,2) AS "Current Size (MB)",
3 autoextensible AS "Autoextend?",
4 round(maxbytes/1024/1024,2) AS "Maximum Size (MB)"
5 FROM dba_data_files;
```

3. Uute failide vaikimisi kataloogi määramine:

```
SQL> ALTER SYSTEM
1 SET set db_create_file_dest = '$ORACLE_HOME/';
```

4. Näidis uue tabeliruumi tekitamise kohta:

```
SQL> CREATE SMALLFILE TABLESPACE example
1 DATAFILE 'example01.dbf'
2 SIZE 1088K
3 AUTOEXTEND ON
4 MAXSIZE 2112K
5 LOGGING
6 EXTENT MANAGEMENT LOCAL AUTOALLOCATE
7 SEGMENT SPACE MANAGEMENT AUTO;
```

Päringu tulemuseks on uus tabeliruum “*example*”, andmefailiks on “*example01.dbf*” järgmiste parameetritega: algsuurus – 1MB + 64kB, automaatselt laienev kuni 2MB + 64 kB. Struktuuri haldamisega tegeleb andmebaas ise. Tabeliruumi muudatused kirjutatakse taastelogsisse.

Ülesanded

1. Logige sisse kasutajana *sys*.
2. Koostage kõikide andmebaasis olemasolevate tabeliruumide kohta järgmine tabel:

Tabeliruumi nimi	Tüüp	Suurus (MB)	Kasutuses(used)%

3. Koostage kõikide andmebaasis olemasolevate andmefailide kohta järgmine tabel:

Faili nimi	Tabeliruumi nimi	Hetksuurus (MB)	Kas laiendatav?	Maksimaalne suurus (MB)

4. Lisage uus tabeliruum nimega *katsetamine*, milles on järgmised parameetrid:

- andmefaili nimi: *katsetamine01.dbf*;
- algsuurus: 1088kB;
- automaatselt laiendatav;
- maksimaalne suurus: 2112kB;
- taastelogeide kirjutamiseta;
- automaatse haldamisega.

5. Käivitage skript: *pr_2.sql*, mis

- tekitab tabeliruumis *katsetamine* tabeli *katse_tabel*;
- Loob protseduuri *insert_katse*.

6. Kutsuge välja protseduur *insert_katse()*:

```
SQL> CALL insert_katse();
```

7. Mitu rida on tabelis "katse_tabel" ?

8. Proovige lisada uus rida tabelisse *katse_tabel*. Kas õnnestus?

```
SQL> INSERT INTO katse_tabel VALUES(1, 'a');
```

9. Kustutage andmed tabelist *katse_tabel*:

```
SQL> DELETE FROM katse_tabel;
```

10. Mitu rida on tabelis *katse_tabel*?

- Kas saate lisada uut rida tabelisse *katse_tabel*?
- Mitu % tabeliruumist *katsetamine* on vaba?

11. Tühjendame tabeli *katse_tabel*:

```
SQL> TRUNCATE TABLE katse_tabel;
```

- Mitu % tabeliruumist *katsetamine* on vaba?
- Kas saate lisada uut rida tabelisse *katse_tabel*?

12. Lisada tabeliruumi *katsetamine* veel üks andmefail järgmiste parameetritega:

- andmefaili nimi: *katsetamine02.dbf*;
- algsuurus: 2112kB.

13. Muutke andmefaili *katsetamine02.dbf* suurst: 5184K

14. Tekitage uus tabeliruum *sybase*, millel on kaks andmefaili:

- Andmefaili nimi: *sybase01.dbf*:

- i. algsuurus: 30784kB;
 - ii. mitte laiendatav;
 - iii. logimisega;
 - iv. automaatselt hallatav.
- b) Andmefaili nimi: *sybase02.dbf*:
- i. algsuurus: 20544kB;
 - ii. mitte laiendatav;
 - iii. logimisega;
 - iv. automaatselt hallatav.

15. Kustutage tabeliruum *katsetamine* koos sisuga ja andmefailidega.