

Histrie Oracle

- 1977 – Larry Ellison, Bob Miner, Ed Oates – Software Development Laboratories (SDL); projekt pro CIA – DBMS nazvaný “Oracle”. Přímo inspirováno teoretickou prací o relačním databázovém modelu. Poté začali s vývojem komerční verze.
- 1978 – změna firmy z SDL na Relational Software Inc. (RSI).
- 1979 – první komerční verze produktu Oracle.
- 1992 – Oracle7 – včetně deklarativní referenční integrity, triggerů a uložených procedur.
- 1997 – Oracle8 – koncepty objektového rozšíření, opuštění čistě relačního model – ORDBMS.
- 1998 – Oracle8i – otevření směrem k internetu – podpora Javy a HTTP protokolu.
- 2001 – Oracle9i – Real Application Cluster (RAC), optimalizace a větší možnosti konfigurace instance.
- 2003 – Oracle10g – Grid computing.
- 2004 – příklon k linuxové platformě.
- 2007 – Oracle11g

Rámcový přehled produktů Oracle

- databázový server:
 - ▶ enterprise edition
 - ▶ standard edition
 - ▶ express edition
- rozšíření databázového serveru (spatial, ...)
- aplikační server
- vývojová prostředí – SQL Developer, JDeveloper
- správcovské nástroje – Enterprise Manager
- enterprise aplikace a řešení (Financials, ...)
- specializované DB stroje (Berkeley DB)
- ...

- především komerční záležitost
 - ▶ variace (množství procesorů, současně pracujících uživatelé, ...)
 - ▶ ceny standard a enterprise se výrazně liší
- k dispozici pro různé platformy
- k vyzkoušení lze stáhnout libovolný produkt v “trial” verzi
- Express Edition – kompletně zdarma i pro komerční účely, limitováno velikostí databáze – max. 4GB

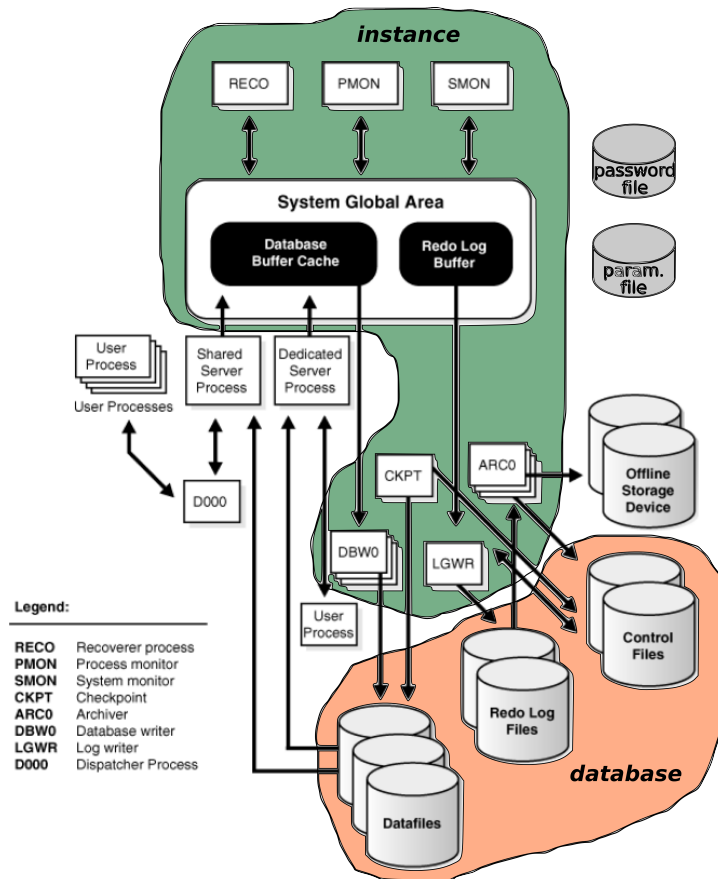
Dokumentace a jiné zdroje

- <http://technet.oracle.com>
 - ▶ dokumentace – online i ke stažení
 - ▶ produkty k vyzkoušení (“trial”)
 - ▶ nutno se registrovat
- <http://metalink.oracle.com>
 - ▶ znalostní báze s vyhledáváním
 - ▶ patche
 - ▶ vázané na koupenou podporu
- Linuxová komunita je velmi sdílná i mimo oficiální zdroje.

- ... je velký “bussiness”
- “2 Day DBA (od verze 9i)
- Administration Workshop I - 5 dnů
- Administration Workshop II - 5 dnů
- Performance And Tuning Workshop - 5 dnů
- a mnoho dalších kurzů
- certifikační program
- cena cca 50.000,-Kč za 5 dnů školení

Nástroje administrátora

- shell, textový editor
- SQL*Plus / iSQL*Plus (`sqlplus`)
- Listener Control (`lsnrctl`)
- `tnsping`
- Recovery Manager (`rman`)
- SQL Loader (`sqlldr`)
- Export / Import (`exp`, `imp`)
- DB Verify (`dbv`)
- Oracle Universal Installer (`oui`)
- Database Configuration Assistant (`dbca`)
- Network Configuration Assistant (`ntca`)
- Oracle Enterprise Manager (OEM), OEM Grid Control (`emctl`)



Základní pojmy

- **instance** = System Global Area (SGA) + background processes
- **SID** = jméno instance, instance může mít jiné jméno než databáze, ke které se připojuje
- **database** = data files + log files + control files
- 2 a n-vrstvá architektura ... z hlediska administrátora databáze vždy dvouvrstvá
- **SQL*Net** = proprietární nadstavba nad TCP/IP protokolem
- SGA = Database Buffer Cache + Redo Log Buffer + Shared Pool + ...
- Shared Pool = Data Dictionary Cache + Library Cache
- PGA = Program Area – část paměti alokovaná server procesem

parameter file (init.ora / spfile)

- buď textový nebo binární (spfile)
- <SID>.ora / spfile<SID>.ora (default; spfile má přednost)
- umístění \$ORACLE_HOME/dbs (default varianta)
- čte se při startu instance
- obsahuje parametry týkající se zejména:
 - ▶ umístění control file
 - ▶ velikosti sdílené paměti
 - ▶ dump files a alert log
 - ▶ shared server konfigurace
- Oracle definuje cca 300 parametrů
- většina má default hodnotu
- jen několik málo je povinných
- pohled `V$PARAMETER` nebo příkaz `show` v SQL*Plus

Startup / shutdown proces

- start / nomount
 - ▶ přečte se parametr file
 - ▶ dojde k alokaci SGA
 - ▶ nastartují se background procesy
- mount
 - ▶ přečte se control file
 - ▶ dojde ke kontrole konzistence databáze (datových souborů, žurnálů)
- open
 - ▶ instance je připojena k databázi
 - ▶ databáze je v konsistentním stavu
 - ▶ databáze je zpřístupněna uživatelům

Privilegovaný přístup do databáze

- dvě speciální systémová privilegia - **SYSDBA**, **SYSOPER**
- při použití tohoto privilegia je uživatel přihlášen do schématu **SYS**
- přihlášení privilegovaného uživatele:
 - ▶ ověření přes OS (`connect / as sysdba`)
 - ▶ ověření přes passwordfile (`connect username as sysdba`)
- SYSDBA nebo SYSOPER může:
 - ▶ přihlásit se k instanci v jiném stavu než OPEN
 - ▶ nastartovat/shodit databázi
 - ▶ provádět zálohu a běžnou obnovu systému
 - ▶ provádět administrátorské operace typu přidání/odebrání datafile, tablespace, logfile
- pouze SYSDBA může provést “incomplete” recovery

Data Dictionary

- (static) data dictionary views
 - ▶ k dispozici až ve stavu OPEN
 - ▶ pohledy s prefixy **USER_**, **ALL_** a **DBA_**
 - ▶ používá podtržítka a množné číslo (`DBA_TABLES`, `DBA_ROLES`, `ALL_USERS`, `DBA_TABLESPACES`,...)
- (dynamic, performance) data dictionary views
 - ▶ některé k dispozici již ve stavu STARTED (`V$INSTANCE`, `V$PARAMETER`, ...)
 - ▶ další (zejména fyzická struktura) ve stavu MOUNT (`V$LOG`, `V$DATAFILE`, `V$TABLESPACE`, ...)
 - ▶ začínají vždy prefixem **V\$**, nepoužívají podtržítka,

Příprava systému a instalace

- předinstalační příprava:
 - ▶ parametry OS (sdílená paměť, semaforey)
 - ▶ file system (velikost bloku)
 - ▶ patche a požadované knihovny
- Oracle Flexible Architecture (OFA)
 - ▶ pravidla a zvyklosti adresářové struktury
 - ▶ software:
 - ★ \$ORACLE_BASE/product/<version>/ (= **\$ORACLE_HOME**)
například /opt/oracle/product/11.0.3
 - ★ struktura uvnitř \$ORACLE_HOME je pevně daná
(./admin, ./dbs, ./rdbms/admin, ./network/admin, ...)
 - ▶ databáze:
 - ★ <mountpoint>/oracle/data/<database_name>
například /u01/oracle/data/orcl

Podstatná rozhodnutí

- obvykle OS user `oracle`, OS groups `sysoper` a `sysdba`
- instalace:
 - ▶ umístění instalace (\$ORACLE_HOME)
- **? jedna aplikace = samostatná databáze ?**
- vytváření databáze:
 - ▶ jméno databáze
 - ▶ jméno instance
 - ▶ znaková sada (**platí vždy pro celou databázi**)
 - ▶ CHARACTER SET
 - ★ pro datové typy CHAR, VARCHAR2, LONG, CLOB
 - ★ UTF8, EE8ISO8859P2, EE8MSWIN1250, ...
(AL16UTF16 použít nelze)
 - ▶ NATIONAL CHARACTER SET
 - ★ pro datové typy NCHAR, NVARCHAR2, NCLOB
 - ★ AL16UTF16 nebo UTF8
- rozdílné znakové sady klientských aplikací řeší vrstva v SQL*Net

Proces instalace software

- od verze 8i (implicitně) grafická (na všech platformách; Java, vlastní JRE)
- lze provést “tichou” instalaci pomocí tzv. response file
- logy instalace obvykle v `$ORACLE_BASE/oraInventory/logs`
- Standard Edition | Enterprise Edition | Custom (mnoho závislostí)
- velikost instalace serveru obvykle více než 3GB
- probíhá i kontrola nastavení OS a požadovaných balíčků
- lze instalovat samostatně pouze software (implicitně volá dbca)
- zároveň též instalace instalátoru (přidání | odebrání sw)
(`$ORACLE_HOME/oui/bin/runInstaller`)
- na stroji si Oracle snaží držet seznam instalací – Oracle Inventory
 - ▶ odkaz na něj je (obvykle) v `/etc/oraInst.loc`
 - ▶ implicitně `$ORACLE_BASE/oraInventory`

Nastavení prostředí

- během instalace je třeba spustit skript `orainstRoot.sh`
- implicitně v `$ORACLE_BASE/oraInventory`
 - ▶ `/etc/oraInst.loc` (umístění Oracle Inventory)
 - ▶ `/etc/oratab` (seznam instancí)
 - ▶ lokace adresáře, který je v `$PATH` (`/usr/local/bin/`)
pro skripty `coraenv`, `oraenv`, `dbhome`, které
zajišťují správné nastavení prostředí
- důležité proměnné prostředí:
 - ▶ `ORACLE_HOME`
 - ▶ `PATH = $PATH:$ORACLE_HOME/bin`
 - ▶ `LD_LIBRARY_PATH =`
`$LD_LIBRARY_PATH:$ORACLE_HOME/lib`
 - ▶ `ORACLE_SID` (případně `TWO_TASK`)
 - ▶ `NLS_LANG` (vhodné `american_america.UTF8`)

Instalace databáze – DBCA

- \$ORACLE_HOME/bin/dbca
- pracuje s “templates”
- umí též rozšířit stávající databázi (například podpora replikací)
- umí správně smazat databázi (důležité zejména na MS Windows)
- umí vygenerovat skripty, které lze prohlédnout / doplnit / spustit.

Praxe – Debian “Etch and half” + KDE

- Oracle rozlišuje “certified” a “non-certified” linux
- při instalaci je třeba být v KDE přihlášen jako oracle (resp. vlastník instalace oracle)
`xhost + a následné su oracle v shell nestačí`
- ne všechny automatické kontroly dopadly úspěšně
- první pokus o instalaci skončil neúspěšně ve fázi kompilace
analýza log souboru + google
`⇒ aptitude install libaiol libaio-dev`
- používáte-li dynamické přidělování adresy přes DHCP
může mít `emctl` problémy se startem
`⇒ lze obejít přes /etc/hosts například:`
`127.0.1.1 valenta.felk.cvut.cz valenta`

Praxe – Gentoo 64bit

základní instalace z Gentoo Live CD

```
emerge dev-libs/libaio
```

```
ln -s /usr/lib64/gcc/x86_64-pc-linux-gnu/4.1.2/libstdc++.so.6  
/usr/lib64/libstdc++.so.6
```

```
${ORACLE_HOME}/lib32/stubs/libc.so: Code:  
- GROUP ( libc.so.6 /usr/lib/libc_nonshared.a )  
+ GROUP ( libc.so.6 /usr/lib32/libc_nonshared.a )  
${ORACLE_HOME}/lib32/stubs/libpthread.so:Code:  
- GROUP ( libpthread.so.0 /usr/lib/libpthread_nonshared.a )  
+ GROUP ( libpthread.so.0 /usr/lib32/libpthread_nonshared.a )  
$ORACLE_HOME/lib/sysliblist:Code:  
- -ldl -lm -lpthread -lnsl -lirc -lipgo  
+ -ldl -lm -lpthread -lnsl -lirc -lipgo -lrt
```

dotažení souboru libgatsh.so z 32-bitové instalace do \$ORACLE_HOME/lib32, zdroj:

database/stage/Components/oracle.rdbms.util/11.1.0.6.0/1/DataFiles/filegroup14.jar

```
export ORACLE_HOME=/opt/oracle/product/11.1.0/db_1  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib32:$ORACLE_HOME/lib  
$ORACLE_HOME/bin/genagtsh $ORACLE_HOME/lib32/libagtsh.so
```

chyby dohledány převážně pomocí \$ORACLE_HOME/install/make.log a [Google](#)

Ukázka – vytvoření databáze, strategie

- popíšeme kroky vytvoření databáze
přímo dle dokumentace [Database Administrator's Guide](#)
- komentovaný průchod DBCA
necháme asistenta vyrobit skripty.
- projdeme vygenerované skripty

Creating the Database - steps

- Step 1: Specify an Instance Identifier (SID)
- Step 2: Ensure That the Required Environment Variables Are Set
- Step 3: Choose a Database Administrator Authentication Method
- Step 4: Create the Initialization Parameter File
- Step 5: (Windows Only) Create an Instance
- Step 6: Connect to the Instance
- Step 7: Create a Server Parameter File
- Step 8: Start the Instance
- Step 9: Issue the CREATE DATABASE Statement
- Step 10: Create Additional Tablespaces
- Step 11: Run Scripts to Build Data Dictionary Views
- Step 12: Run Scripts to Install Additional Options (Optional)
- Step 13: Back Up the Database.
- Step 14: (Optional) Enable Automatic Instance Startup

CREATE DATABASE Statement

```
CREATE DATABASE mynewdb
USER SYS IDENTIFIED BY sys_password
USER SYSTEM IDENTIFIED BY system_password
LOGFILE GROUP 1 ('/u01/app/oracle/oradata/mynewdb/redo01.log') SIZE 100M,
GROUP 2 ('/u01/app/oracle/oradata/mynewdb/redo02.log') SIZE 100M,
GROUP 3 ('/u01/app/oracle/oradata/mynewdb/redo03.log') SIZE 100M
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
CHARACTER SET US7ASCII
NATIONAL CHARACTER SET AL16UTF16
EXTENT MANAGEMENT LOCAL
DATAFILE '/u01/app/oracle/oradata/mynewdb/system01.dbf' SIZE 325M
SYSAUX DATAFILE '/u01/app/oracle/oradata/mynewdb/sysaux01.dbf' SIZE 325M
DEFAULT TABLESPACE users
DATAFILE '/u01/app/oracle/oradata/mynewdb/users01.dbf'
        SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
DEFAULT TEMPORARY TABLESPACE tempst1
TEMPFILE '/u01/app/oracle/oradata/mynewdb/temp01.dbf' SIZE 20M REUSE
UNDO TABLESPACE undotbs
DATAFILE '/u01/app/oracle/oradata/mynewdb/undotbs01.dbf'
        SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

Co kde běží?

- Jaké instance existují?
`/etc/oratab`
- Jaké instance běží?
`ps -ef | grep pmon`
- dávky `db_start` a `db_shut` v `$ORACLE_HOME/bin`
- automatický start instance/instancí při bootování serveru
`/etc/init.d/ /etc/rc2.d /etc/rc0.d`
- Kde je parametr file?

Základní správa – přehled

- prostředí
`ORACLE_HOME ORACLE_SID NLS_LANG PATH`
- přihlášení
 - ▶ `sqlplus /nolog`
 - ▶ autentikace pomocí OS : `connect / as sysdba;`
 - ▶ autentikace pomocí password file :
`connect <username> as sysdba;`
- SQL*Plus příkaz `startup`
- SQL*Plus příkaz `shutdown`
- stav instance a práce s parametry
- příkaz `ALTER DATABASE`
- příkaz `ALTER SYSTEM`

SQL*Plus – STARTUP

STARTUP options | upgrade_options

options:

[FORCE] [RESTRICT] [PFILE=filename] [QUIET]
[NOMOUNT | MOUNT[dbname] | [OPEN[open_options][dbname]]]

open_options:

READ {ONLY | WRITE [RECOVER]} | RECOVER

upgrade_options:

[PFILE=filename] {UPGRADE | DOWNGRADE} [QUIET]

Typicky:

startup

startup nomount

startup mount

SQL*Plus – SHOW a SHUTDOWN

SHO[W] PAR[AMETER] <string>

- **například:** show parameter log
všechny parametry (instance) obsahující v názvu řetězec log
- mnoho dalších použití (show all, show errors,)

SHUTDOWN [ABORT | IMMEDIATE |
NORMAL | TRANSACTIONAL [LOCAL]]

- default varianta je NORMAL
- varianty NORMAL a TRANSACTIONAL mají časový limit 1 hodinu
- obvyklá praxe: shutdown immediate

define _editor=vi

vhodné pro editaci víceřádkových příkazů v SQL*Plus

ALTER DATABASE

```
ALTER DATABASE [ database ]
{ startup_clauses| recovery_clauses|
database_file_clauses| logfile_clauses|
controlfile_clauses| standby_database_clauses|
default_settings_clauses| instance_clauses|
security_clause } ;
```

startup_clauses:

```
{ MOUNT [ { STANDBY | CLONE } DATABASE ]
| OPEN { [ READ WRITE ][ RESETLOGS | NORESETLOGS ]
[ UPGRADE | DOWNGRADE ]
| READ ONLY } }
```

database_file_clause:

```
{ RENAME FILE 'filename' [, 'filename' ]...
TO 'filename'
| create_datafile_clause | alter_datafile_clause
```

ALTER SYSTEM

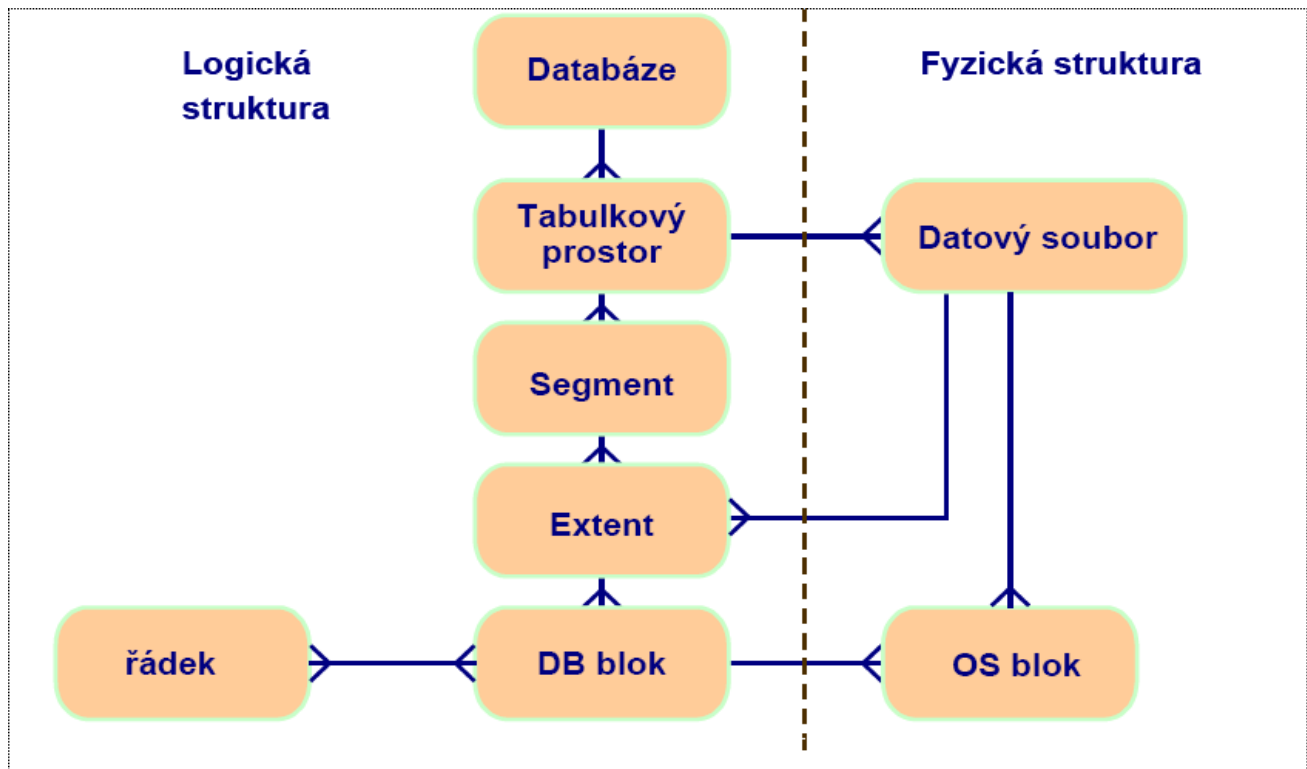
ALTER SYSTEM

```
{ archive_log_clause| checkpoint_clause
| check_datafiles_cl| distributed_recov_cl
| FLUSH { SHARED_POOL | BUFFER_CACHE }
| end_session_cl| SWITCH LOGFILE| { SUSPEND | RESUME }
| quiesce_cl | rolling_migration_cl
| alter_system_security_cl| shutdown_dispatcher_cl
| REGISTER
| SET alter_system_set_cl[ alter_system_set_cl ]...
| RESET alter_system_reset_cl[alter_system_reset_cl]};
```

alter_systemset_cl:

```
parameter_name =
parameter_value [, parameter_value ]...
[ COMMENT = string ][ DEFERRED ]
[ { SCOPE = { MEMORY | SPFILE | BOTH } }
```

Fyzická a logická struktura databáze



Fyzická struktura databáze

- control files
- log files
- data files
- parametr file
- password file
- archived log file

Žurnály (LOG FILES) – úvod

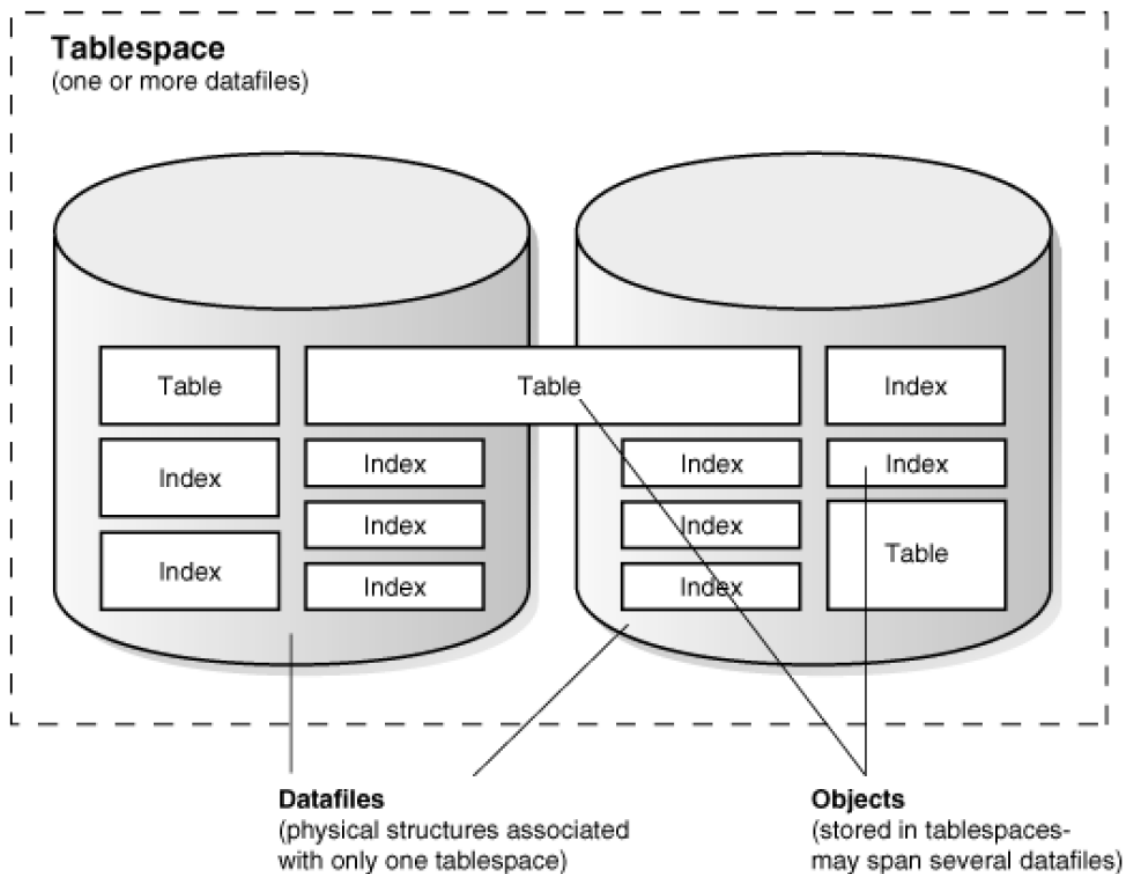
- archivni / nearchivni mód databáze
- konsistence v databázi (otevřená, zavřená databáze)
- synchtonizace (checkpoint, log sequence number)
- nekonsistentní databáze
 - instance recovery
 - media recovery
- SQL*Plus: archive log list
- redolog groups
- pohledy V\$LOG a V\$LOGFILE

ALTER DATABASE – logfile

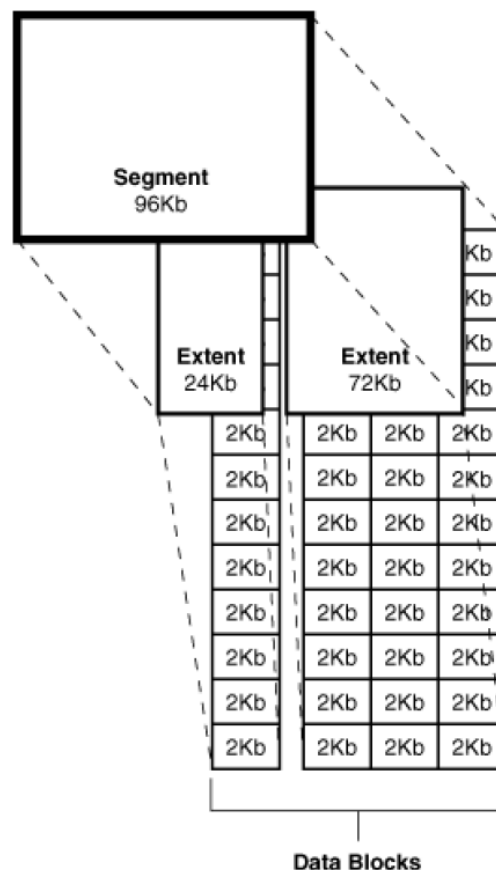
```
ALTER DATABASE
{ { ARCHIVELOG [ MANUAL ] | NOARCHIVELOG }
| [ NO ] FORCE LOGGING
| RENAME FILE 'filename' [, 'filename' ]...
TO 'filename'
| CLEAR [ UNARCHIVED ]
LOGFILE logfile_descriptor [, logfile_descriptor ]...
[ UNRECOVERABLE DATAFILE ]
| add_logfile_clauses | drop_logfile_clauses
| supplemental_db_logging};
```

```
ALTER DATABASE ADD [ STANDBY ] LOGFILE
{ [ INSTANCE 'instance_name' ]
[ GROUP integer ] redo_log_file_spec
[, [ GROUP integer ] redo_log_file_spec ]...
| MEMBER 'filename' [ REUSE ] [, 'filename' [ REUSE ] ]...
TO logfile_descriptor [, logfile_descriptor ]...};
```


Tablespace a segmenty



Segment, extent, datový block



Správa tablespace – přehled

- členění tablespace:
 - ▶ SYSTEM
 - ▶ SYSAUX
 - ▶ UNDO
 - ▶ TEMP
 - ▶ další jsou volitelné: (DATA, INDEX, USERS,)
- informace v data dictionary:
DBA_TABLESPACES DBA_DATA_FILES
V\$TABLESPACE V\$DATAFILE
- příkaz CREATE / ALTER TABLESPACE
- stav tablespace: online/offline readwrite/read
- extent management (MINEXTENS, MAXEXTENS, INITIAL, NEXT, PCTINCREASE)
- space management (locally/dictionary managed)

CREATE TABLESPACE

```
CREATE [ BIGFILE | SMALLFILE ]  
{ permanent_tablespace_clause  
| temporary_tablespace_clause  
| undo_tablespace_clause } ;
```

```
CREATE TABLESPACE tbs_02  
DATAFILE '/u01/oradata/MyDB/tbs_02.dfb' SIZE 500K REUSE  
AUTOEXTEND ON NEXT 500K MAXSIZE 100M;
```

```
CREATE TABLESPACE tbs_04  
DATAFILE '/u01/oradata/MyDB/tbs_02.dfb' SIZE 10M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

- systémová a objektová privilegia
- uživatel
- profil
- role
- resource manager

Systémová privilegia

- definují právo na provedení akce
- CREATE SESSION, CREATE PROCEDURE, CREATE TABLE, CREATE VIEW,...
- DROP ANY TABLE, ALTER ANY USER, DROP ANY SESSION, CREATE ANY VIEW, ...
- SYSDBA, SYSOPER, ALTER SYSTEM, ALTER DATABASE, ...
- O7_DICTIONARY_ACCESSIBILITY parametr
- několik desítek systémových privilegií
- grant select any table to karkulka;
- grant create session to machal with admin option;

Objektová privilegia

- vztahují se ke konkrétnímu objektu a konkrétní akci
- `grant select, insert on MyTable to machal with grant option;`
- `grant update (jmeno, vek) on Osoby to karkulka;`
- odebrání privilegia (systémového i objektového) – REVOKE
objektová privilegia reagují kaskádně
systémová nekaskádně
- `revoke select any table from machal;`
- `revoke insert on MyTable from karkulka;`

příkaz GRANT

```
GRANT { grant_system_privileges
| grant_object_privileges } ;

{ system_privilege| role| ALL PRIVILEGES}
[, { system_privilege| role| ALL PRIVILEGES}]...
TO grantee_clause
[ WITH ADMIN OPTION ]

{ object_privilege | ALL [ PRIVILEGES ] }
[ (column [, column ]...) ]
[, { object_privilege | ALL [ PRIVILEGES ] }
[ (column [, column ]...) ]]...
on_object_clause
TO grantee_clause
[ WITH HIERARCHY OPTION ]
[ WITH GRANT OPTION ]
```

Uživatelé

- definují se pro každou databázi samostatně
- uživatelské jméno definuje schéma (logický kontejner pro segmenty)
- autentikace uživatele
 - ▶ heslo uložené v data dictionary (kryptovaně)
 - ▶ přes operační systém
 - ▶ Oracle SSO (Single Sign On) – vyžaduje aplikační server
 - ▶ specializované nástroje 3. stran (Kerberos, VeriSign, ...)
- každý uživatel má přiřazen:
 - default tablespace
 - temporary tablespace
 - profil
- uživatel může mít:
 - více rolí
 - limitovaný přístup k tablespace (quotes)

CREATE USER Statement

```
CREATE USER user
IDENTIFIED { BY password | EXTERNALLY [ AS 'certificate_DN' ]
| GLOBALLY [ AS '[ directory_DN ]' ] }
[ DEFAULT TABLESPACE tablespace
| TEMPORARY TABLESPACE { tablespace | tablespace_group_name }
| { QUOTA { size_clause | UNLIMITED } ON tablespace }...
| PROFILE profile
| PASSWORD EXPIRE
| ACCOUNT { LOCK | UNLOCK } ] ;
```

Například:

```
CREATE USER sidney IDENTIFIED BY out_standing1
DEFAULT TABLESPACE example QUOTA 10M ON example
TEMPORARY TABLESPACE temp
QUOTA 5M ON system
PROFILE app_user PASSWORD EXPIRE;
```

Profil

- každý uživatel má právě jeden (DEFAULT)
- jednotlivé parametry mohou mít hodnotu DEFAULT
- **resource limit**
vázaný na instanční parametr RESOURCE_LIMIT (true/false)

```
{ SESSIONS_PER_USER| CPU_PER_SESSION| CPU_PER_CALL| CONNECT_TIME  
| IDLE_TIME| LOGICAL_READS_PER_SESSION| LOGICAL_READS_PER_CALL  
| COMPOSITE_LIMIT} { integer | UNLIMITED | DEFAULT }  
| PRIVATE_SGA { size_clause | UNLIMITED | DEFAULT }}
```

- **password limit** – aktivní vždy

```
{ { FAILED_LOGIN_ATTEMPTS| PASSWORD_LIFE_TIME| PASSWORD_REUSE_TIME  
| PASSWORD_REUSE_MAX| PASSWORD_LOCK_TIME  
| PASSWORD_GRACE_TIME}{ expr | UNLIMITED | DEFAULT }  
| PASSWORD_VERIFY_FUNCTION{ function | NULL | DEFAULT }}
```

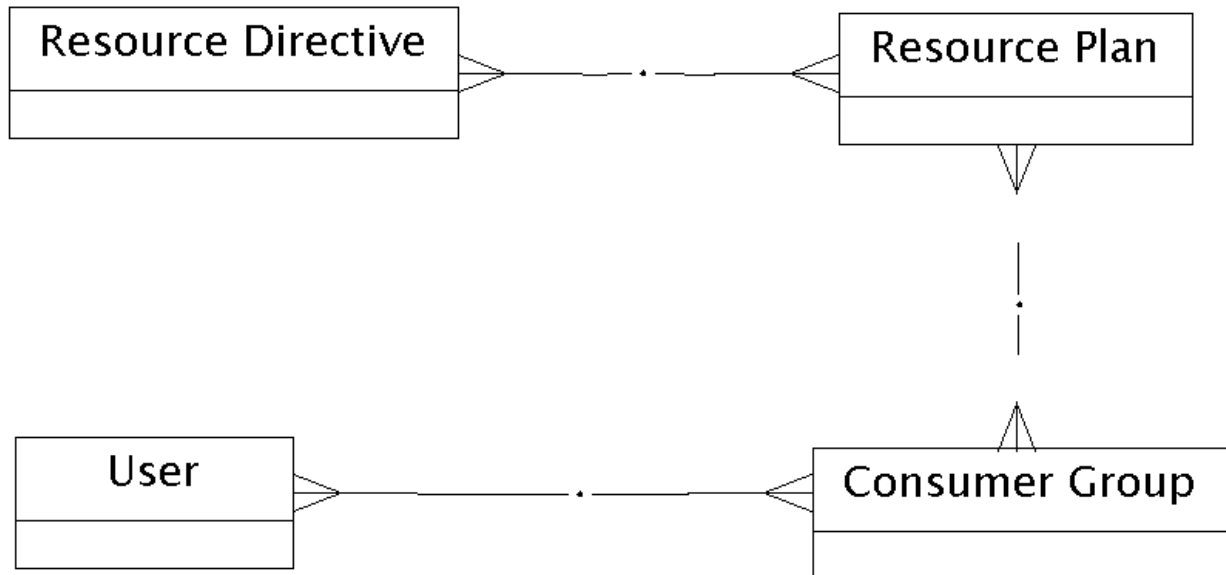
- CREATE / ALTER / DROP PROFILE
- CREATE / ALTER USER

Role

- pojmenovaná množina systémových a objektových privilegií
- CREATE ROLE <role>;
- DROP ROLE <role>;
- může obsahovat další role (hierarchická struktura)
- role může být chráněna heslem
- uživatel může mít přiřazeno více rolí
- GRANT <role> TO <user>;
- neaktivní role lze selektivně aktivovat
ALTER SESSION SET ROLE = <role>
[IDENTIFIED BY <password>;]
- implicitní zapnutí rolí při přihlášení uživatele:
ALTER USER <user> DEFAULT ROLE <role> [, <role>...]
- aktivní role (v session): V\$ROLES

Resource Manager

- komplexní řešení přidělování zdrojů
- resource plan, consumer group, directive
- nastavení pomocí packages
- grafické nastavení v DBA Consoli



Správa schématu

- segment
 - ▶ = objekt uložený ve schématu
 - ▶ každý segment patří právě do jednoho schématu
 - ▶ segmenty si alokují prostor po extentech
 - ▶ přidělování extentů (segment, tablespace, default)
- schéma a uživatel
 - ▶ schéma = logický container na objekty
 - ▶ název schématu je totožný se jménem uživatele
 - ▶ úplné určení objektu: `user_name.segment_name`
- s objekty v DB lze manipulovat příkazy
CREATE / ALTER / DROP
- DD views: DBA / ALL / USER_OBJECTS
_SEGMENTS, _TABLES, _VIEWS, _CONSTRAINTS,
_SOURCE, _ROLES, PACKAGES,

Objekty ve schématu

- Clusters
- Constraints
- Database links
- Database triggers
- Dimensions
- External procedure libraries
- Indexes and indextypes
- Java classes, Java resources, and Java sources
- Materialized views and materialized view logs
- Object tables, object types, and object views
- Operators
- Sequences
- Stored functions, procedures, and packages
- Synonyms
- Tables and index-organized tables
- Views

Objekty, které nepatří do schématu

- Contexts
- Directories
- Parameter files (PFILES) and server parameter files (SPFILES)
- Profiles
- Roles
- Rollback segments
- Tablespaces
- Users

Možnosti:

- alert log
- background processes trace files
- sys audit trail
- database audit
- user (server proces) dump files
- value based audit

alert log, sys audit, trace files

alert log

- instanční parametr – `background_dump_dest`
- `alert<SID>.log`
- start/stop instance, nedefaultní parametry, recovery info, změna struktury DB
- problémy, jsou nejprve reporovány zde

background processes trace files

- instanční parametr – `core_dump_dest`
- `<SID>_<proces>_<PID>.trc`
- spíše pro support než pro administrátora

sys audit trail

- instanční parametr – `audit_file_dest`
- `ora_<PID>.aud`
- pro každé přihlášení do schématu SYS

user dump

- instanční parametr – `user_dump_dest`
- samostatně pro každý server proces
- slouží zejména jako podklad pro ladění aplikace
- lze jej zapnout:
 - na úrovni instance (velmi se nedoporučuje) – `SQL_TRACE`
 - na úrovni session (obvyklé):

```
ALTER SESSION SET SQL_TRACE = TRUE | FALSE
```

pro vybranou session pomocí package `DBMS_SESSION`
- dump file může velmi rychle narůstat
- nástroje pro reporty nad user dump souborem:
 - utilita `tkprof`
 - enterprise manager
- další instanční parametry: `TIMED_STATISTICS`,
`MAX_DUMP_FILE_SIZE`, `STATISTICS_LEVEL`

database audit, value based audit

database audit

- nutno nastavit `AUDIT_TRAIL` (OS file | databáze)
- příkaz `AUDIT` pro vypnutí `NOAUDIT`
- lze sledovat:
 - příkaz
 - použití privilegia
 - přístup k objektu (tabulce)
- možnosti sledování:
 - by session | by access
 - successful | unsuccessful | all

```
AUDIT INSERT ON MACHAL.SALARY  
  BY ACCESS WHENEVER SUCCESSFUL;
```

value based audit

- nutno zajistit pomocí triggerů

- Znakové sady
 - ▶ jednotné pro celou databázi
 - ▶ při vytváření databáze, nelze je změnit později
 - ▶ CHARACTER_SET (CHAR, VARCHAR2, LONG, CLOB)
 - ▶ NATIONAL_CHARACTER_SET (NCHAR, NVARCHAR2, NCLOB)
- NLS (National Language Settings)
 - ▶ úroveň instance, session, volání SQL příkazu
 - ▶ NLS_LANGUAGE
 - ▶ NLS_TERRITORY
- NLS_LANG parametr
- datové typy TIMESTAMP a INTERVAL

NLS_LANGUAGE

- Language for server messages
- Language for day and month names and their abbreviations (specified in the SQL functions TO_CHAR and TO_DATE)
- Symbols for equivalents of AM, PM, AD, and BC. (A.M., P.M., A.D., and B.C. are valid only if NLS_LANGUAGE is set to AMERICAN).
- Default sorting sequence for character data when ORDER BY is specified. (GROUP BY uses a binary sort unless ORDER BY is specified.)
- Writing direction.
- Affirmative and negative response strings (for example, YES and NO).

NLS_TERRITORY

NLS_TERRITORY specifies the conventions for the following default date and numeric formatting characteristics:

- Date format
- Decimal character and group separator
- Local currency symbol
- ISO currency symbol
- Dual currency symbol
- First day of the week
- Credit and debit symbols
- ISO week flag
- List separator

NLS Data Dictionary Views

Applications can check the session, instance, and database NLS parameters by querying the following data dictionary views:

- **NLS_SESSION_PARAMETERS** shows the NLS parameters and their values for the session that is querying the view. It does not show information about the character set.
- **NLS_INSTANCE_PARAMETERS** shows the current NLS instance parameters that have been explicitly set and the values of the NLS instance parameters.
- **NLS_DATABASE_PARAMETERS** shows the values of the NLS parameters for the database. The values are stored in the database.

NLS Dynamic Performance Views

Applications can check the following NLS dynamic performance views:

- **V\$NLS_VALID_VALUES** lists values for the following NLS parameters: NLS_LANGUAGE, NLS_SORT, NLS_TERRITORY, NLS_CHARACTERSET
- **V\$NLS_PARAMETERS** shows current values of the following NLS parameters: NLS_CALENDAR, NLS_CHARACTERSET, NLS_CURRENCY, NLS_DATE_FORMAT, NLS_DATE_LANGUAGE, NLS_ISO_CURRENCY, NLS_LANGUAGE, NLS_NUMERIC_CHARACTERS, NLS_SORT, NLS_TERRITORY, NLS_NCHAR_CHARACTERSET, NLS_COMP, NLS_LENGTH_SEMANTICS, NLS_NCHAR_CONV_EXP, NLS_TIMESTAMP_FORMAT, NLS_TIMESTAMP_TZ_FORMAT, NLS_TIME_FORMAT, NLS_TIME_TZ_FORMAT

NLS - praxe

- obvykle NLS_LANG na úrovni klienta
environment proměnná na úrovni shell (linux)
registry na MS Windows
- NLS_LANG =
NLS_LANGUAGE_NLS_TERRITORY.CHARACTER_SET
NLS_LANG = american_america.utf8
NLS_LANG = "czech_czech republic.ee8iso8859p2"
- jemnější nastavení lze také na úrovni klienta
(NLS_DATE_FORMAT, NLS_DATE_LANGUAGE, NLS_SORT, ...)
- dále lze na úrovni session
`ALTER SESSION SET NLS_SORT = 'XCZECH';`
- pohled NLS_SESSION_PARAMETERS
- funkce TO_CHAR, TO_DATE mají nepovinný parameter NLS
`select to_char(sysdate, 'DD. Month YYYY',
'nls_language="czech"') from dual;`

Speciální datové typy pro datum

TIMESTAMP

TIMESTAMP WITH TIME ZONE

TIMESTAMP WITH LOCAL TIME ZONE

normalizováno vzhledem k DB_TIME_ZONE

INTERVAL YEAR TO MONTH

```
SELECT TO_DATE('29-FEB-2004', 'DD-MON-YYYY')  
      + TO_YMINTERVAL('4-0')  
FROM DUAL;
```

INTERVAL DAY TO SECOND

```
SELECT TO_DATE('14:23:00', 'HH:MI:SS')  
      + TO_DSINTERVAL('00 2:34:00')  
FROM DUAL;
```

Přenos dat - přehled

- **export/import** utility
pro přenos Oracle – Oracle
lze použít i jako zálohu
utility `exp` a `imp` na úrovni OS novější varianta – **data pumps**
- **SQL*Loader**
zpracovává tzv. **flat files**
bohatá možnost konfigurace řídicího souboru
import dat z ne-oracle zdrojů
- **external tables**
na ne-oracle zdroje lze pohlížet jako na tabulky
- **export dat do ne-oracle zdrojů**
`sqlplus` - parametr `spool`
nástroje 3. stran – SQL Developer, ...

- **ovládání:** dialog / dávkový soubor
- `exp/imp help=yes`
- **úroveň:** tabulka / schéma / databáze
- **pozor na NLS_LANG**

SQL*Loader – diskuze/ukázka

Koncept:

http://oraserv.felk.cvut.cz/11gdoc/server.111/b28319/ldr_concepts.htm#i1007641

Ukázka 1:

http://oraserv.felk.cvut.cz/11gdoc/server.111/b28319/ldr_concepts.htm#i1004697

Ukázka 2:

http://oraserv.felk.cvut.cz/11gdoc/server.111/b28319/ldr_control_file.htm#i1004663

External tables – ukázka/diskuse

http://oraserv.felk.cvut.cz/11gdoc/server.111/b28319/et_concepts.htm#i1009391

Export pro ne-oracle zdroje – ukázka

- ❶ sqlplus export dat – anketa
- ❷ sqlplus export výstupu skriptu do html
<http://service.felk.cvut.cz/courses/X36DBS/xml/test/test.xml>
- ❸ SQL Developer – ukázka exportů dat ankety.

- architektura
- konfigurace SQL*Net
`listener.ora`, `tnsnames.ora`, `sqlnet.ora`
- ukázka – včetně services
- Network Manager (`netmgr`)

Architektura

Možnosti:

HTTP, SQL*Net, JDBC, ...

<http://oraserv.felk.cvut.cz/11gdoc/network.111/b28316/intro.htm#i453694>

SQL*Net:

dedicated server, shared server

<http://oraserv.felk.cvut.cz/11gdoc/network.111/b28316/architecture.htm#i1046119>

Konfigurace SQL*Net – popis

listener.ora

na straně serveru

definuje porty, protokoly, služby

může být pro více listener procesů

sqlnet.ora

na straně klienta

definuje základní přístupy vyhodnocení

NAMES.DIRECTORY_PATH

TNSNAMES, EZCONNECT, HOSTNAME, LDAP, CDS, NIS

log file, logging

trace file, trace level, ...

tnsnames.ora

na straně klienta

pokud TNSNAMES (v sqlnet.ora), slouží pro překlad adresy

často používaná metoda u menších sítí

Konfigurace SQL*Net – ukázka

- ukázka konfiguračního nástroje `netmgr`
- ukázka přidání service na úrovni databáze:
 - ▶ do `service_names` parametru na úrovni instance registrujeme service **test**
 - ▶ ověříme (automatickou) registraci service v listeneru (pomocí `lsnrctl`)
 - ▶ do `tnsnames.ora` přidáme service `muj-test`
 - ▶ ověříme funkčnost pomocí `tnsping` a `sqlplus`