

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Administrace RDBMS Oracle

Jan Michek

© 2012 ČZU v Praze

!!!

**Místo této strany vložíte zadání bakalářské práce.
(Do jedné vazby originál a do druhé kopii)**

!!!

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Administrace RDBMS Oracle" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.3.2012

Poděkování

Rád bych touto cestou poděkoval Ing Václavu Vostrovskému, PhD. za vedení práce a kolegům ze společnosti ARC Consulting s.r.o za spolupráci a umožnění realizace.

Administrace RDBMS Oracle

Administration of RDBMS Oracle

Souhrn

Tato bakalářská práce se zabývá tématem správy databázového systému Oracle. Veškeré zdroje a zkušenosti vycházejí z absolvování kurzu Oracle Administration Workshop a následného působení jako Junior DBA.

Práci je možné rozdělit do několika úrovní. Úvodní část zcela jasně popisuje pouze teorii vztahující se k systému Oracle. Nalezneme zde zestručněný popis architektury klíčových částí nebo principů, jejichž obdobu lze často najít i napříč ostatními databázovými platformami. Následující praktičtější část se vztahuje ke dvěma podtématům. Část je věnována praktickým ukázkám běžné činnosti v rámci systému a následuje část implementace v reálném podnikovém prostředí, kde systém Oracle Databáze slouží jako zdroj pro ERP aplikaci PeopleSoft Enterprise.

Během zpracování práce byl zaujat postoj ke zpracování takovým způsobem, aby byl obsah přístupný pro širší publikum. Ačkoliv některé pojmy pokládané za samozřejmost nejspíš nebudou vysvětleny do nejmenších detailů, jako celek je práce cílená pro kohokoliv s vědomím o informačních technologiích.

Summary

This thesis is focused on the administration of the Oracle database system. All knowledge and experience is based on absolving an Oracle Administration Workshop and the consequent work as a Junior DBA.

The first part of this thesis is concerned only with theoretical aspects of Oracle Database including a brief description of the system architecture for key elements that can also be found in other relational database platforms. A more practical approach follows, outlining common tasks with examples and resulting in a specific

implementation in an enterprise environment where Oracle Database 11g serves as the source system for the ERP application bundle PeopleSoft Enterprise.

Klíčová slova: Relační databázová technologie, Oracle, RDBMS, migrace dat, upgrade databázového systému, Oracle PeopleSoft Enterprise

Keywords: Relational database technology, Oracle, RDBMS, data migration, database system upgrade, Oracle PeopleSoft Enterprise

Obsah

1	Úvod	5
2	Cíl práce a metodologie	6
3	Teoretické principy	7
3.1	Relační databáze	7
3.2	Transakční zpracování	8
4	Oracle 11g	9
4.1	Architektura a pojmy	9
4.2	Paměťové struktury	10
4.2.1	SGA	10
4.2.2	PGA	10
4.2.3	Procesy	10
4.2.4	Uživatelské procesy	11
4.2.5	Databázový proces	11
4.3	Oracle Database Instance	11
4.3.1	Spuštění instance	12
4.4	UNDO, Redolog, Archivelog	14
4.5	Oracle komunikační protokol	15
4.6	Backup [1] [2]	16
5	Praktické postupy	18
5.1	Restore a Recover	19
5.2	Klonování	20
5.3	Ukázková Restore & Recover procedura	21
5.3.1	Předpoklady	22
5.3.2	Nalezení částí zálohy	23
5.3.3	Identifikace relevantních částí	23
5.3.4	Proměnné prostředí	24
5.3.5	Listener	25
5.3.6	Vytvoření nové instance	25
5.3.7	RMAN příkazy obnovy	25
5.3.8	Další možnosti a rozšíření	28
5.3.9	Využití scénáře a dopad	28
6	Účast v rámci upgrade projektu	30
6.1	Výchozí situace	30
6.2	Identifikace a definice problému - Oracle certifikace	30
6.3	Rozložení DB serverů	31
6.4	Operační systém/Hardware/Software	31
6.5	Požadavky	31
6.6	Aplikace PeopleSoft – databázová vrstva	32
6.7	Migrace	34
6.7.1	Datová pumpa	34
6.7.2	Transportable tablespace	35
6.7.3	Export a Import	35
6.8	Kroky postupu	36
6.9	Popis prvků nové implementace	36
6.9.1	Data Guard	37
6.9.2	Recovery Catalog	38

6.9.3	Enterprise Manager a Grid Control 11g.....	38
6.9.4	Implementace pravidelných záloh.....	38
6.9.5	Vytváření záloh - RMAN.....	39
6.9.6	Pravidelné ověření restore procedury.....	40
7	Závěr	41
8	Seznam použitých zdrojů	42
9	Přílohy	43
9.1	Příkaz list backup.....	43
9.2	Recovery Catalog pohledy.....	45

1 Úvod

Hlavním cílem a zaměřením této bakalářské práce je nejprve seznámit se a porozumět principům relačního databázového systému Oracle a následně využít získaných znalostí k podpoře podnikového prostředí. Daným úkolem je vyřešit problém necertifikovaného databázového prostředí a hrozby ztráty oficiální podpory licencovaných produktů Oracle Database. Kromě kroků implementace certifikovaného prostředí a objasnění architektury systému Oracle Database se práce věnuje i praktickým dovednostem v rámci následné podpory zákazníka.

2 Cíl práce a metodologie

Bakalářská práce je tematicky zaměřena na seznámení s databázovým systémem Oracle a následnou dlouhodobější správou podnikového prostředí. Cílem je vymezení teoretických principů relačně databázového systému Oracle, identifikovat problémy v souvislosti k daným vstupním požadavkům a přechodu k vyšší verzi. Práce se také zaměří na způsoby provedení migrace dat v rámci Oracle, na navržení rozložení databázových instancí a nastavení konkrétního prostředí, nebo na implementaci řešení a ověření kritických částí.

Použitá metodika řešené bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů. Navíc budou použity základní i pokročilejší metody a techniky relačně databázové technologie a především systému Oracle Database 11g. navrhované řešení bude realizováno formou praktického řešení, které bude respektovat identifikované požadavky na tato řešení a ověří navrhované záležitosti. Na podkladě poznatků a výsledků budou formulovány závěry a následně zobecněny pro další možná použití.

3 Teoretické principy

Prvním krokem je vymezení pojmů a důležitých aspektů architektury využívaného databázového systému. Předmětem není poskytnutí kompletní dokumentace produktu, nýbrž představení kritických částí, které třeba chápat před praktickými úkony.

3.1 Relační databáze

Databáze je soubor dat se kterým se zachází jako s jednotkou, jejím základním účelem je ukládání a poskytování souvisejících informací. Oracle RDBMS (relation database management system) spolehlivě udržuje velké objemy dat v multi-user prostředí, to znamená, že uživatelé mohou současně přistupovat nejen k databázi jako celku, ale i ke stejným detailům zároveň. Společně s poskytováním dat je základním požadavkem na databázový systém transakcionalita rychlost, zabezpečení proti neautorizovaným zásahům a spolehlivé možnosti zálohování. Úkolem databázového administrátora tedy není správa databáze jakožto dat, ale celého databázového systému. V této práci se budeme zabývat pouze strukturou a možnostmi Oracle Database Server 11g, nikoliv ostatních databázových systému jako je MS SQL, Postgresql nebo DB2.

Oracle Database je v principu tzv. relační databáze, to znamená že k propojení dat jednotlivých tabulek se využívá redundantních záznamů namísto ukazatelů. Atomická a skalární data jsou uspořádána ve dvourozměrných tabulkách a výsledná informace je relací mezi řádkem a sloupcem. Důležitým principem je právě užití redundantní informace pro propojení záznamů a vytvoření závislostí. Obrázek 1 zobrazuje tři tabulky a jejich následné propojení ve výsledné tabulce Transactions. Schéma zobrazuje skutečnost využití cizích klíčů k propojení tabulek. Například hodnota '103' v tabulce Transactions je cizí klíč, odkazující na řádkové hodnoty tabulky Account.

Customer

<u>cust_id</u>	fname	lname
1	George	Blake
2	Sue	Smith

Account

<u>account_id</u>	<u>product_cd</u>	<u>cust_id</u>	balance
103	CHK	1	\$75.00
104	SAV	1	\$250.00
105	CHK	2	\$783.64
106	MM	2	\$500.00
107	LOC	2	0

Product

<u>product_cd</u>	name
CHK	Checking
SAV	Savings
MM	Money market
LOC	Line of credit

Transaction

txn_id	txn_type_cd	<u>account_id</u>	amount	date
978	DBT	103	\$100.00	2004-01-22
979	CDT	103	\$25.00	2004-02-05
980	DBT	104	\$250.00	2004-03-09
981	DBT	105	\$1000.00	2004-03-25
982	CDT	105	\$138.50	2004-04-02
983	CDT	105	\$77.86	2004-04-04
984	DBT	106	\$500.00	2004-03-27

Obrázek 1. Tabulky relační databáze a jejich propojení (Zdroj: Learning SQL [9])

Ačkoliv se Oracle řadí mezi relační databáze a veškerý přístup a správa dat se drží relačních principů, v jádru využívá systém moderních objektových přístupů bez striktně redundantních klíčů a záznamů. Z historických nebo dokonce z finančních důvodů se stále používá stejný přístup k databázi pomocí jazyka SQL. Srovnáním relačních a ryze objektových databází se tato práce nebude zabývat. [4] [5]

3.2 Transakční zpracování

Konzistence dat v existujících tabulkách je nejen v databázovém systému Oracle zajištěna transakčním zpracováním. Jde o přístupu k datům takovým způsobem, kdy nedojde k nechtěnému poškození tabulek nebo vzájemnému přepsání dat uživateli. Jednoduše transakce převádí databázi z jednoho konzistentního stavu do druhého,

neexistuje žádný mezistav. Pokud by došlo v průběhu zpracování k chybě, databáze se navrátí do předchozího konzistentního stavu bez poškození struktury nebo obsahu. Během transakce v rámci uživatelské session je třeba rozlišovat dva pojmy: *commit* a *rollback*. Commit je v překladu potvrzení operace, jedná se o potvrzení sady SQL příkazů a zavedení nového konzistentního stavu databáze. Naopak použitím rollbacku se přechází k předchozímu konzistentnímu stavu, tedy k předchozímu commit stavu. Commit se použije jako závěrečný příkaz úspěšně provedeného bloku, naopak rollback poslouží pokud v průběhu zpracování nastane nějaká chyba. [4] [5]

Konkrétní implementace transakčního zpracování bude systém od systému odlišná, způsob implementace v systému Oracle 11g pomocí redo logu a undo prostoru je popsán v později v kapitole 4.2.5.

4 Oracle 11g

Databáze Oracle se postupně stává jedním ze standardů podnikových prostředí a v současnosti stojí v čele databázových technologií po boku MS SQL nebo IBM DB/2. Nejnovější verze v době psaní této práce je Oracle Database 11g Release 2 s možností updatu 11.2.0.3.

4.1 Architektura a pojmy

Relační databázový systém Oracle se skládá ze tří hlavních bloků: paměťové struktury, procesů a datové struktury. Obecnější seskupení rozdělí systém na instanci a databázi. Instance představuje tu část zavedenou v paměti včetně procesů, databáze pak označuje pouze data jako například diskové bloky. V korektní Oracle terminologii pojem databáze označuje striktně jen datové struktury, především datové soubory, tzv. *data files*¹. Datové soubory mohou být logicky seskupeny do tzv. *tablespace*, tedy například soubory *users1.dbs* a *users2.dbf* budou součástí jednoho tablespace *users*. Tablespace je logická struktura, tedy zavedena po startu databáze a ačkoliv zastupuje datové soubory, jedná se o část instance.

Databáze, tedy disková část systému, sestává jak z oddělené fyzické části tak z návazných logických struktur. Protože jsou tyto části odděleny, fyzické úložiště může

¹ pojem "data files" je oficiální termín pro Oracle datové soubory, typicky s koncovkou .dbs

být za předpokladu korektní synchronizace datového slovníku spravováno bez ovlivnění přístupu k logické úrovni. [1] [2]

4.2 Paměťové struktury

Oracle vytváří a používá paměťové struktury pro různé účely, například pro uložení programového kódu, sdílených dat nebo privátních prostorů jednotlivých uživatelů. K instanci se pojí dvě primární paměťové struktury, tzv. SGA a PGA. [1]

4.2.1 SGA

System Global Area (SGA) představuje skupinu sdílených struktur a komponent obsahujících data a kontrolní informace pro jednu Oracle instanci. SGA je sdílená pro všechny serverové² a background³ procesy (v rámci konkrétní instance). Pro ilustraci, v SGA najdeme například cache data a sdílené prostory pro parse SQL příkazů. [1]

4.2.2 PGA

Program Global Area (PGA) uchovává data a kontrolní informace pro konkrétní server nebo background process. Jedná se o privátní nesdílenou paměť alokovanou systémem Oracle Database během spouštění procesu. Práva pro přístup k PGA jsou udělena pouze samotnému server procesu, každý proces má tedy přidělenou vlastní PGA. [1]

Zkratky SGA nebo PGA se často chybně označují jako Shared (sdílená) a Private (soukromá) Global Area. Ačkoliv například v oficiálních Oracle certifikacích je to samozřejmě chybné, pro pochopení rozdílu je to vhodná mimo technická pomůcka.

4.2.3 Procesy

Úlohy běžící v počítačové paměti se označují jako procesy. Konkrétněji je proces definován jako tzv. *thread of control* nebo mechanismus operačního systému, běžící ve sledu kroků. [1]

² server proces je proces na straně serveru, například parsování SQL dotazu

³ background process je anglický pojem označující skryté procesy a programy běžící v pozadí, bez zásahu uživatele, například zápis do logu.

4.2.4 Uživatelské procesy

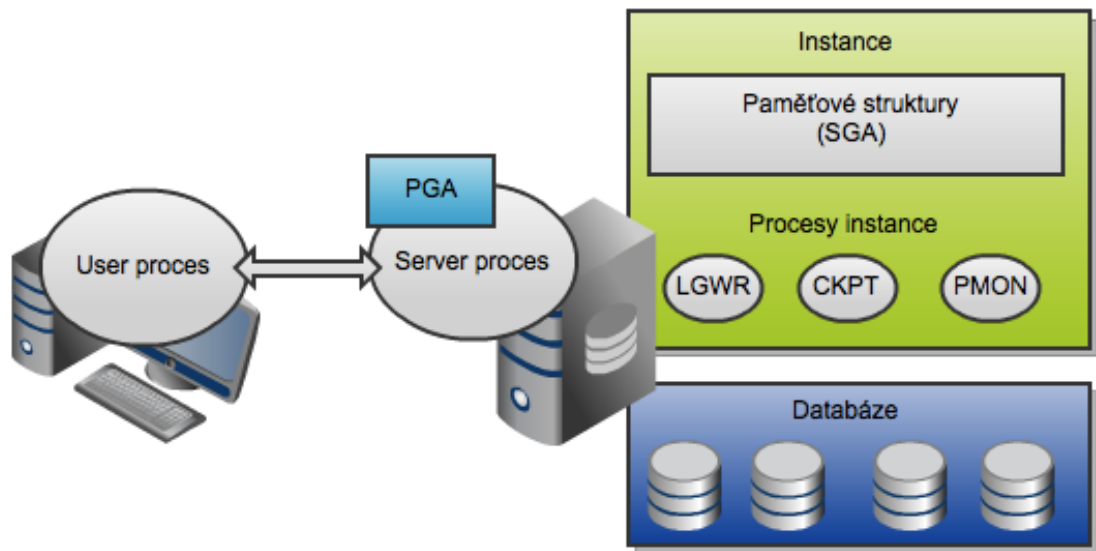
Aplikace nebo nástroj prostřednictvím kterého uživatel přistupuje k systému se řadí k uživatelským procesům Oracle. Jedná se tedy o běžící kód aplikací přistupujících k instanci, například program sqlplus. [1]

4.2.5 Databázový proces

Úlohy spouštějící kód systému Oracle jsou označovány jako databázové procesy. Jsou to serverové procesy zajišťující například spojení pro sezení (session) nebo background procesy spuštěné během startování instance, například proces log writer. [1]
[9]

4.3 Oracle Database Instance

Instance zahrnuje paměťové struktury a background procesy spojené s touto instancí. Pokaždé když se nainstaluje nějaká instance, alokuje si sdílenou část paměti zvanou SGA (system global area) a nainstaluje vlastní procesy. Až po nainstalování instance je přiřazena konkrétní databáze, tedy soubor dat. Hovoří se o tzv. *mountu* podobně jako je tomu u adresářové struktury unixového file systému. Po kontrole a synchronizaci datových souborů může následovat otevření a tedy zpřístupnění databáze uživatelům. Instance je sdružená s jednou a pouze s jednou databází. Pokud je tedy na jednom serveru více databází, musí být pro každou spuštěna separátní instance. Příloha č. X představuje schéma RDBMS Oracle a rozdělení na Instanci a Databázi. [1]



Obrázek 2. Struktura systému Oracle (Zdroj: Vlastní tvorba podle Administration Workshop [1])

4.3.1 Spuštění instance

Instance se může nacházet v jednom ze čtyř stavů, tedy *shutdown*, *nomount*, *mount* a *open*. Přejít mezi stavy je sekvencí a řídí se nástrojem *sqlplus* pouze uživatelem s oprávněním *sysdba*.

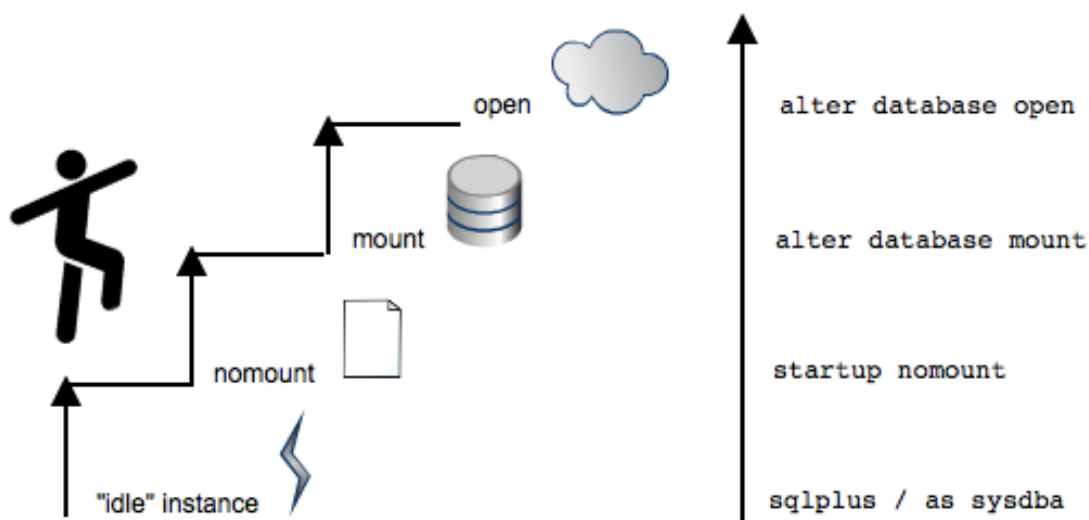
Stav *shutdown* představuje vypnutou instanci, tento stav nedovoluje nic jiného než připojení *sysdba* uživatele. Neběží žádné procesy a nejsou dostupné žádné datové soubory. Tento stav se používá pro vytvoření instance.

Pro přechod do stavu *nomount* je nutné identifikovat *server parameter (SP)* file, alokovat SGA paměť, spustit background procesy a otevřít soubor *alert.log* do kterého se zapisují veškeré informace ohledně běhu systému, převážně chybové výstupy. SP file se typicky nachází v *ORACLE_HOME/dbs* pod označením *spfile<sid>.ora*. Pokud Oracle nenalezne tento binární soubor, pokusí se automaticky najít ještě jeho textovou reprezentaci v podobě *init<sid>.ora*. Pokud jsou tyto soubory v jiném než defaultním umístění, je nutné soubory specifikovat zároveň s příkazem *startup*. Specifikace těchto souborů se využívá hlavně při ručním vytváření nové instance.

Přechod do stavu *mount* představuje asociaci instance, tedy paměťové struktury, s náležitými datovými soubory. Jednodušeji řečeno, přiřazení spuštěné instance k databázi. V této fázi se tedy otevře *control file* (specifikovaný v *sp* souboru) pro zjištění

umístění a stavu všech datových souborů a redo logů. Prozatím se však nekontroluje dostupnost a práva přístupu. Stav mount se používá pro maintenance operace záloh, obnovy, přesunu a přejmenování souborů, nebo změny parametrů.

Stav open znamená běžnou operaci systému, kdy se mohou přihlásit běžní uživatelé. Otevření databáze znamená otevření datových a redo souborů, včetně kontroly přístupu. Během otevření dochází také ke kontrole konzistence a případné automatické obnovy. Pokud byl databázový systém nekorektně vypnut, například při výpadku elektřiny, automaticky se zkontroluje konzistentní stav a data se nahrají z posledních archive a redo logů. Následující obrázek č. 3 představuje schéma spouštění instance, kdy každý schod odpovídá jedné fázi. Pravá část diagramu pak zobrazuje příkazy. [1][2][10]



Obrázek 3. Startup instance (Zdroj: Vlastní tvorba podle Administration Workshop [1])

Ve chvíli kdy je instance otevřená pro uživatele, může následovat přihlášení. Spojení (connection) a sezení (session) jsou úzce spjaty s uživatelskými procesy, ale jsou velice odlišné svým významem.

Connection je komunikační cesta mezi uživatelským procesem a Oracle Database instancí. Toto spojení může být navázáno různým protokolem buď lokálně na serveru nebo po síti ze vzdáleného počítače.

Session představuje spíše stav aktuálně přihlášeného uživatele pod specifickým login id, heslem, uživatelským nastavením nebo dalšími parametry. Oracle dovoluje spustit paralelní session jak pro více uživatelů, tak pro toho samého zároveň.

4.4 UNDO, Redolog, Archivelog

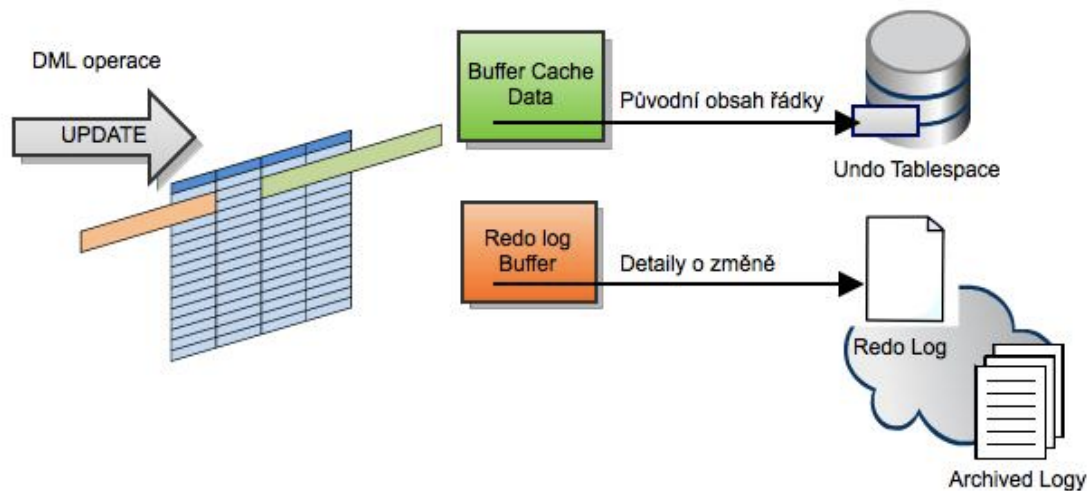
Undo data představují kopii původních záznamů před úpravou - tedy např. zásahem SQL příkazem (DDL⁴/DML⁵) - a je zachyceno pro každou transakci, která mění data. Doba udržení těchto původních záznamů musí být minimálně do ukončení transakce. Undo data jsou tedy nositelem možností transakčního zpracování v systému Oracle a jsou využívána pro možnosti rollbacku, čtení stejných dat více uživateli v průběhu různých transakcí, pro obnovení nezdařilých transakcí nebo pokročilé metody Oracle Flashback⁶. [2]

Redo log buffer je paměťová struktura, která živě v každém momentě udržuje informace o změnách provedené na databázi. Redo logy jsou zapisovány souvisle a sekvenčně a na disk je z bufferu zapisuje proces běžící v pozadí do tzv. aktivního redo souboru. Oracle poskytuje ještě možnost tzv. archivelogů. Jedná se jednoduše o historické redo logy. Ve chvíli přetečení nastavené velikosti redo logu existuje možnost před přepsáním nejdříve uložit shromážděná data. Tyto archivelogy jsou velice důležité jakožto část záloh, obsahují informace potřebné pro rekonstrukci změn provedených pomocí DDL, DML nebo interních operací. [1][2][4]

⁴ Data Definition Language (DDL) představuje sadu operací pro vytváření databázových objektů, jedná se o SQL výrazy CREATE TABLE, CREATE INDEX apod.

⁵ Data Modification Language (DML) zahrnuje ty operace, které vkládají a upravují data v již existujících strukturách, v SQL se jedná o výrazy SELECT, UPDATE a INSERT.

⁶ Flashback technologie umožňují například navrácení tabulky nebo celé databáze do předchozího stavu, nebo jen provést dotaz podle časové známky.



Obrázek 4. Cache při změně tabulkových dat (Zdroj: Vlastní tvorba podle Administration Workshop [1][2])

4.5 Oracle komunikační protokol

Pro komunikaci jak mezi komponenty systému Oracle Database se využívá Oracle Net. Jedná se o softwarový balík umožňující navázání komunikace mezi klientskou aplikací a databázovou platformou Oracle Database. Oracle Net je nezávislá na operačním systému a k jednomu databázovému serveru je možné připojit se z libovolné platformy. Server a klient části v některých případech ani nevyžadují stejnou verzi, je například možné použít 10g klienta k navázání komunikace s 11g databází.

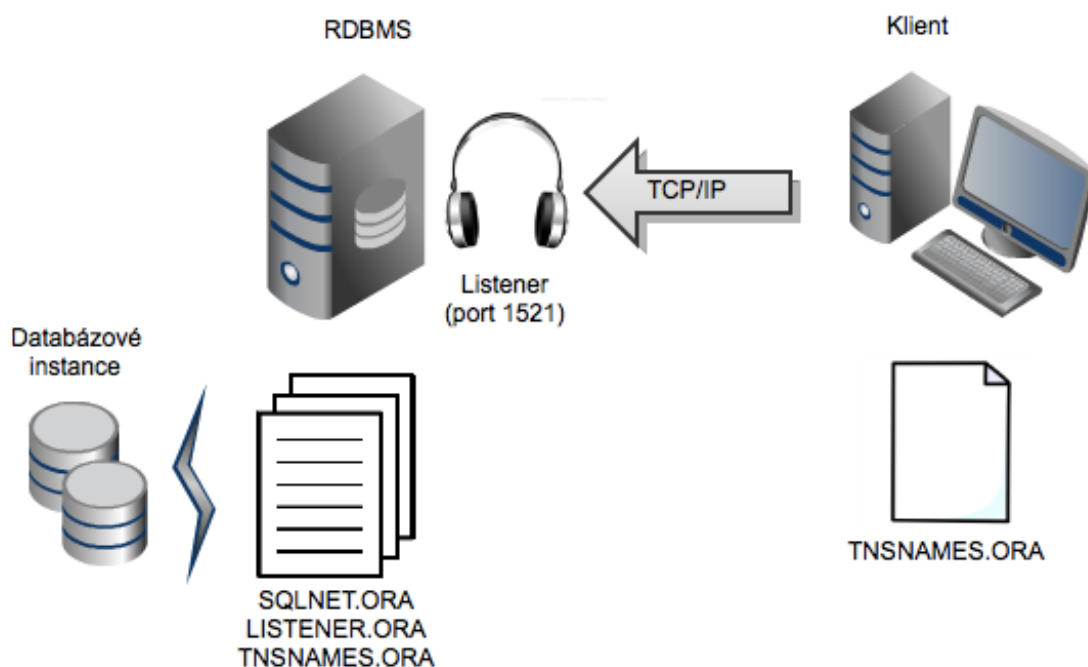
Softwarový balík Oracle Net Services obsahuje Oracle Net, Listener, Net Configuration a Connection Manager.[1]

Oracle Net je proprietární síťový protokol, který prostřednictvím TCP/IP navazuje spojení a zajišťuje předávání zpráv mezi serverem a klientem.

Listener je komponenta běžící pouze na serveru. Jejím účelem je detekce, nebo-li naslouchání, přichozím požadavkům o spojení a následné zajištění uživatelské session. Listener implementuje konfigurační soubor listener.ora, ve kterém jsou specifikované databáze, hosté a poskytující služby. [1] [5]

Oracle Net Configuration Assistant (*netca*) je gui nástroj pro zavedení a nastavení Listeneru na lokálním serveru. Programy *sqlplus* nebo *tnsping* nejsou součástí Oracle Net balíku.

Pro nastavení protokolu slouží na strane serveru tři klíčové textové soubory: sqlnet.ora, listener.ora a tnsnames.ora. Listener.ora již byl zmíněn jako konfigurační soubor pro Oracle Net Listener, definuje umístění listeneru a seznam poskytovaných služeb. Soubor sqlner.ora poskytuje nastavení pro Oracle Net jako je způsob rozeznání služeb, šifrování nebo trasování. Třetí konfigurační soubor tnsnames.ora poskytuje na bázi aliasů umístění hledané služby, tedy Oracle databáze. Obecně může být alias libovolný, nicméně pro PeopleSoft (viz dále) databázi musí odpovídat skutečnému jménu databáze - parametru db_name. Následující obrázek č. 5 představuje schéma komunikace mezi serverem a klientem pomocí Oracle Net. [2]



Obrázek 5. Komunikace mezi Oracle serverem a klientem (Zdroj: Vlastní tvorba podle Administration workshop [1])

4.6 Backup [1] [2]

Jedním z hlavních předmětů práce v rámci správy databázového systému je zabezpečení před ztrátou dat v podobě záloh neboli backupu. V kontextu této práce není prostor pro identifikaci všech havarijních scénářů a postupů jejich napravení, vyžadují podrobnější znalost architektury systému. Nicméně, ještě před prozkoumáním specifických možností backupu, konkrétní implementace v tomto prostředí a jejím

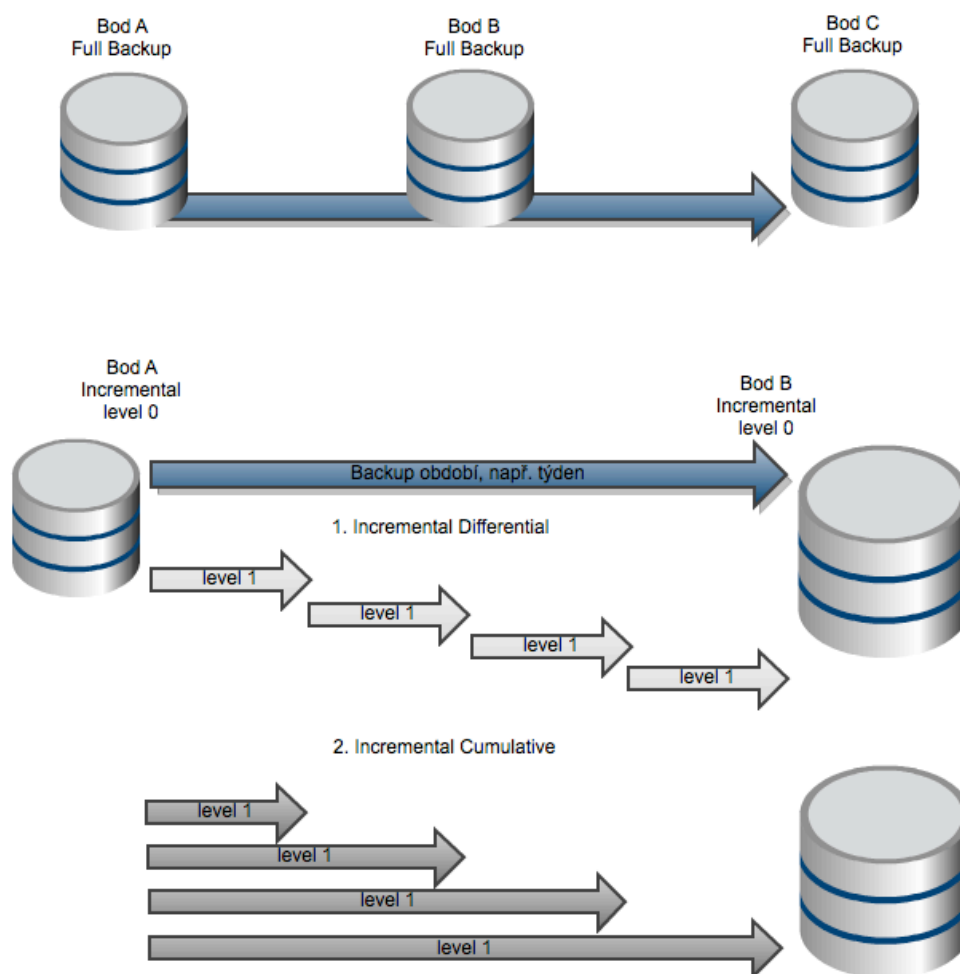
nastavením je potřeba pochopit dva režimy chodu databáze Oracle 11g - *noarchivelog* a *archivelog*.

Režim *noarchivelog* představuje takový stav systému kdy nejsou průběžně zaznamenávány prováděné změny databáze, tedy nejsou zapisovány informace o provedených transakcích. Záloha takového systému v podobě kopie má smysl jen ve vypnutém stavu, jedině tak získáme jistotu konzistentního stavu. Hovoří se o tzv. studené záloze. Ponechání databáze v režimu *noarchivelog* po delší dobu je potenciálně nebezpečné a dá se považovat za hazard.

Oproti tomu režim *archivelog* využívá *redo logy* zmíněné 4.2.5, které jsou zavedeny právě za účelem zaznamenávání prováděných transakcí tak, aby bylo možné zrekonstruovat veškeré změny za dané období od posledního definovaného konzistentního stavu, zpravidla bod full backupu. Vzhledem k tomu že data jsou do redologu zapisovány souběžně s prováděnou transakcí, jedná se o živou zálohu během chodu databáze. V režimu *archivelog* je možné provést i full backup bez změny stavu instance - bez vypnutí.

Full backup představuje zálohu kompletního systému, ať už v komprimované podobě nebo čisté kopie, v daném fixním okamžiku. Princip zálohy spočívá pouze v překopírování datových souborů v konzistentním stavu. Takový typ zálohy vyžaduje vypnutí instance a mluví se o tzv. studené záloze (z angl. Cold Backup). Nevýhodou full backupu je nezachycení transakcí v období mezi jednotlivými zálohami. Pro zálohování všech transakcí s možností návratu do libovolného bodu v čase slouží inkrementální zálohy využívající již zmíněný koncept redo logů, respektive archive logů.

Pro utilizaci inkrementálních záloh je nejprve nutné stanovit výchozí bod, tento stav, označovaný jako inkrementální nultá úroveň (level 0), může pak sloužit jako východisko inkrementálních záloh první úrovně (level 1) *kumulativních*, zachycujících veškeré změny od nulté úrovně, nebo *diferenciálních*, zachycujících změny pouze od předchozí zálohy úrovně jedna. Inkrementální záloha nulté úrovně představuje vlastně full backup, ale vzhledem k tomu, že vychází kromě datových souborů také z transakčních redo logů, je vytvářena za běhu databáze bez zamezení přístupu uživatelů. Rozdíl mezi kumulativní a diferencíální zálohou je zobrazen také na obrázku č. 6 společně s porovnáním oproti full backupu.



Obrázek 6. Oracle zálohy databáze (Zdroj: Vlastní tvorba)

5 Praktické postupy

Tato část práce je zaměřena na seznámení a aplikaci s některých praktických procesů a úkolů, které představují běžnou práci v rámci Oracle databázového prostředí. Zmíněné skripty a nastavení vychází z know-how autora a vznikaly během práce jako Junior DBA v rámci soukromých nebo klientských testování a ostatních činností. Velká část sestavení například zálohovacích skriptů vychází z konzultace s kolegy a prací s portálem My Oracle Support, diskuzních fór a blogů v rámci dlouhodobého působení v podnikovém prostředí. Spolupráce v rámci Oracle Database podpory a maintenance nadále přetrvává.

5.1 Restore a Recover

Nejspíš nejkritičtějším prvkem podpory databázového prostředí je udržování a především zaručení záloh a schopnost obnovy po havárii.

Proces obnovení databáze (ve smyslu datových souborů náležících instanci) ze záloh je nutné rozdělit na dvě části a pochopit rozdíl mezi dvou krokovým postupem obnovy.

Prvním krokem ať už kompletní nebo nekompletní obnovy⁷ je nalezení datových souborů a zjištění stavu těchto částí. Vychází se tak z full backupu nebo z inkrementální zálohy nulté úrovně a celý krok nahradí současné datové soubory. Proces nahrazení stávajících souborů staršími soubory nalezené v zálohách se označuje pojmem *restore*. Databázový systém se tak však ocitne v nekonzistentním stavu, datové části totiž neodpovídají stavu zaznamenaném v kontrolních souborech a nesouhlasí se stavem instance.

Pro uvedení celého systému do stejného konzistentního stavu následuje další krok nazývaný jako *recover*, kdy dochází ke zpětnému přehraní transakcí pro uvedené období. Rozsah znovu přehraných transakcí záleží na konkrétním nastavení obnovy. V systému Oracle Database 11g je defaultní obnova až do posledního známého konzistentního stavu, je však možné nastavit přesný čas do kterého se má celý systém obnovit. K tomu slouží RMAN příkaz `set until`, který přijímá jako parametr buď systémové číslo SCN, string ve formátu date nebo log sekvenci. Následující ukázka zobrazuje celý RMAN skript potřebný pro obnovu do určeného času 26. února 2012, 23:00 hodin. Klauzule `set until` není povinná.

```
RUN
{
SET UNTIL TIME "TO_DATE('2012-26-02:23:00:00','yyyy-dd-mm:hh24:mi:ss')";
RESTORE DATABASE;
RECOVER DATABASE;
ALTER DATABASE OPEN RESETLOGS;
}
```

⁷ Nekompletní obnova může být například obnovení jediné tabulky

5.2 Klonování

Během působení jako Oracle DBA může být poměrně častým úkolem naklonování produkční databáze pro účely vývoje nebo testování. Vytvoření odděleného testovacího a vývojového prostředí je dnes běžnou praxí.

Pro klonování databáze je nejprve nutné vytvořit prázdnou Oracle instanci. Nezáleží na způsobu vytvoření, můžeme například využít nástroj Oracle Database Configuration Assistant (dbca), přímočarý X11 “wizard” pro vytváření a mazání databáze a instance. Vytvoří jak systémové datové soubory, tak paměťové struktury - instanci. Configuration Assistant postupně zobrazí sled kroků a vytvoření jednoduché defaultní databáze nevyžaduje téměř žádnou znalost použité technologie nebo parametrů.

Po vytvoření instance je již možné spustit klonování pomocí nástroje Recover Manager (RMAN) jednoduchým skriptem, například jako je následující.

```
connect target sys/oracle@dbsand1;
connect auxiliary sys/oracle@dbsand2;
run{
allocate channel prmy1 type disk;
allocate channel prmy2 type disk;
allocate channel prmy3 type disk;
allocate auxiliary channel stby1 type disk;
duplicate target database to dbsand2 from active database;
}
```

Target označuje klonovanou instanci (například produkční systém) zatímco *auxiliary* novou instanci (klon). Názvy mohou být matoucí, vzhledem k tomu že *auxiliary* je pro nás cílová instance, kterou vytváříme jako kopii například produkce.

Po připojení obou instancí následuje blok `run` seskupující příkazy klonování. Tento blok není jen formální úpravou, příkazy musí být spouštěny v rámci bloku, protože operace jsou spolu úzce spjaty. Například při možném uvedení přejmenování souboru je tak jasně stanoveno, ke kterým souborům se přejmenování vztahuje.

V ukázce je využit příkaz s argumentem `from active database`. Tato klauzule byla představena ve verzi 11g a umožňuje kopírování aktuálního stavu zdrojové databáze. Bez uvedení parametru je možné vytvářet klon ze záloh zdroje. Záleží tak na možnostech zatížení primární instance. Se zatížením souvisí také množství alokovaných kanálů určujících stupeň paralelismu.

Stejný skript lze použít i pro klonování za účelem vytvoření Data Guard Standby databáze, vyžaduje jen drobnou změnu posledního řádku. Zavedení technologie Data Guard však vyžaduje další nastavení a zavedené parametry.

```
duplicate target database for standby from active database;
```

Vytvoření klonu se používá typicky například pro vytvoření vývojového nebo akceptačního prostředí pro navazující aplikaci, v případě této práce se jedná o PeopleSoft Enterprise, blíže popsány v pozdější kapitole. Klonování PeopleSoft databáze navíc ještě vyžaduje dodatečné změny souvisejícím s některými tabulkami obsahující záznamy o umístění webového serveru nebo hesel uživatelů. PeopleSoft totiž využívá pouze jednoho souhrnného databázového uživatele a bezpečnost pro uživatele aplikace je řešena šifrováním přístupových tabulek. Veškerá databázová spojení jsou tedy pod jedním uživatelem, označovaný jako access id, až následně je ověřeno pokud koncový PeopleSoft uživatel má pravomoc pro hledaný objekt [3].

Tento způsob klonování vytvoří přesnou kopii, klon tak může obsahovat citlivá data. Existují nástroje a procesy pro zašifrování kopírovaných dat, nebylo to však nikdy požadavkem a podobné procedury jsem tedy nezaváděl a nebude to ani předmětem bakalářské práce.

5.3 Ukázková Restore & Recover procedura

Tato kapitola je zaměřena na konkrétní případ selhání Oracle Database, respektive na obnovení systému po katastrofě, kdy jsou dostupné pouze zálohy, například po vyhoření serveru. Běžným scénářem obnovy dat se dnes vztahuje na instanci v provozu, kde došlo například k poškození datového úložiště nebo ztrátě tabulky a je potřeba provést neúplnou obnovu dat. Nekompletní obnovu, ve smyslu obnovení části databáze přičemž ostatní komponenty zůstanu nedotčeny, je možné i za chodu navrátit tabulku nebo datový soubor do téměř libovolného konzistentního stavu. Zde se však zaměříme na daleko extrémnější případ kdy dojde ke ztrátě celého databázového serveru a máme dostupné jen zálohy na externím úložišti, například na pásce. V tuto chvíli je tedy potřeba vytvořit novou instanci na základě dostupných záloh.

Celý proces obnovení databázového systému Oracle pouze ze záloh, respektive vytvoření nové instance nad samotnou zálohou je navazujícím textu rozčleněn a popsán ve sledu jednotlivých kroků. V následujících odstavcích je pojmem zdroj označena ta databáze, kterou je třeba obnovit, pojmem cíl pak nově vytvořená databáze a instance.

Postup zde naznačený je praktické využití záloh, vznikl v rámci testování vytvořených backupů. Obnova celého systému pro mne bylo novinkou, v rámci školení a běžné práce jsem se totiž setkal hlavně s teoretickými principy a krátkými cvičeními na osvojení těchto principů. Samozřejmě samotná implementace a naplánování záloh se nedá považovat za kompletní bez vyzkoušení. Výsledné rozhodnutí bylo pravidelně v definovaném intervalu obnovit celý systém jako v případě, kdy by došlo k selhání celého serveru. Postup jsem nejdřív vyzkoušel v malém měřítku v testovacím prostředí.

Téměř identický postup ukazuje i blogger pod názvem Arju na svých stránkách.[6]

5.3.1 Předpoklady

Primárním předpokladem obnovení databázového systému je platný a kompletní backup zdrojové databáze. Požadována je nultá úroveň⁸, ostatní inkrementální zálohy první úrovně⁹ nebo dodatečné archiveology¹⁰ jsou nutné jen na základě specifikace období. Například pokud máme k dispozici nedělní full backup a databázi je nutné obnovit až do stavu v úterý večer ještě než proběhl úterní incremental, je potřeba kromě nulté úrovně ještě první úroveň z pondělí a archiveology z úterý až do stavu havárie.

Kromě provedené a dostupné zálohy je ještě nutné zvolit způsob identifikace potřebných částí. Vzhledem k tomu, že se zálohování provádí průběžně a zálohy mohou být archivovány libovolně dlouho, je nutné správně určit pouze relevantní soubory pro obnovu do posledního žádoucího a konzistentního stavu. Tento stav může být hodinu zpátky, nebo pokud to záloha dovolí, také několik týdnů nebo i roků.

Zásadním předpokladem je také shoda verzí databázového systému.

⁸ *level 0 incremental backup* je zvláštní typ full backupu, který je však pořizován za chodu databáze včetně redo dat a který může být použit jako základ pro následující inkrementální úroveň [2]

⁹ *level 1 incremental backup* představuje změny od posledního level 0 backupu. [2]

¹⁰ *archiveolog* představuje soubor jakýchkoli a všech transakcí archivovaných za běhu systému. Jedná se o historické redo logy.

5.3.2 Nalezení částí zálohy

Nejprve je samozřejmě potřeba najít relevantní zálohové data a identifikovat pouze potřebné části. V Oracle terminologii se hovoří o tzv. *backup pieces*, které zahrnují vždy datové soubory a archiveology za období. Všechny tyto části je třeba zpřístupnit pro cílový systém.

5.3.3 Identifikace relevantních částí

Pod identifikací správných souborů si můžeme jednoduše představit jejich umístění na lokálním či síťovém úložišti. Podle konkrétní implementace zálohování se nabízí dvě možnosti nalezení relevantních částí backupu, použití recovery catalogu nebo backup logu. Pro pochopení principu je přímočařejší použití logu, nicméně seznam dostupných pohledů recovery catalogu je součástí přílohy.

Mějme následující extrakt logu zálohy z požadovaného období. Jedná se pouze o ilustrativní příklad. V ukázce jsou zvýrazněny relevantní části. Modře jsou vyznačeny datové soubory a redology jako *archivelog*, zeleně *server parameter(SP)* a *control file* a tučně černě dodatečné archiveology. Za zmínku stojí separátní backup část pro *control file*¹¹ a *SP file*¹². Tento soubor lze rozeznat i podle odlišného jména. Bez parametrických a stavových souborů by nebylo možné obnovit instanci do požadovaného stavu, záloha datových souborů je samostatně pro obnovu v čase nepoužitelná.

Tento výňatek je ze skutečného logu, opakující se pasáže jsou pouze vynechány pro přehlednost.

```
Starting backup at 15-OCT-11
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=284 device type=DISK
channel ORA_DISK_1: starting compressed archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=23313 RECID=40986 STAMP=743987456
input archived log thread=1 sequence=23314 RECID=40987 STAMP=743989410
input archived log thread=1 sequence=23315 RECID=40988 STAMP=743991150
channel ORA_DISK_1: starting piece 1 at 15-OCT-11
channel ORA_DISK_1: finished piece 1 at 15-OCT-11
```

¹¹ *Control file* představuje jednak repozitář pro záznam backup operací, jednak informace o stávajícím stavu.

¹² *Parameter file*, jak již název napovídá, obsahuje inicializaci veškerých proměnných.

piece

```
handle=E:\ORACLE\ORAFLASH\HPROD89\BACKUPSET\2011_10_15\O1_MF_ANNNN_TAG20111015T190048_79
MH43RV_.BKP tag=TAG20111015T190048 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:06
channel ORA_DISK_1: deleting archived log(s)
channel ORA_DISK_1: starting compressed archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=23377 RECID=41052 STAMP=744071662
input archived log thread=1 sequence=23378 RECID=41053 STAMP=744072647
input archived log thread=1 sequence=23379 RECID=41054 STAMP=744073014
channel ORA_DISK_1: starting piece 1 at 15-OCT-11
channel ORA_DISK_1: finished piece 1 at 15-OCT-11
```

piece

```
handle=E:\ORACLE\ORAFLASH\HPROD89\BACKUPSET\2011_10_15\O1_MF_ANNNN_TAG20111015T190048_79
MH8RC5_.BKP tag=TAG20111015T190048 comment=NONE
```

Starting Control File and SPFILE Autobackup at 27-AUG-11

piece

```
handle=E:\ORACLE\ORAFLASH\HPROD89\AUTOBACKUP\2011_08_27\O1_MF_S_760304656_75LC7LWD_.BKP
comment=NONE
Finished Control File and SPFILE Autobackup at 27-AUG-11
```

```
channel ORA_DISK_1: starting compressed archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=34169 RECID=77075 STAMP=760175765
input archived log thread=1 sequence=34170 RECID=77078 STAMP=760177514
input archived log thread=1 sequence=34171 RECID=77081 STAMP=760179440
channel ORA_DISK_1: starting piece 1 at 03-SEP-11
channel ORA_DISK_1: finished piece 1 at 03-SEP-11
```

piece

```
handle=E:\ORACLE\ORAFLASH\HPROD89\BACKUPSET\2011_09_03\O1_MF_ANNNN_TAG20110903T190041_76
4QVMBK_.BKP tag=TAG20110903T190041 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:25
```

5.3.4 Proměnné prostředí

Kromě nastavení proměnných ORACLE_HOME a ORACLE_SID jako identifikace instance a RDBMS instalace, které budou společné na všech operačních systémech, v prostředí Windows je ještě nutné definovat velikost alokované paměti proměnnou ORA_RMAN_MEMORY_TARGET.

5.3.5 Listener

Pro komunikaci a připojení k databázové instanci používá systém Oracle Database protokol Oracle Net, je tedy nutné nastavit identifikaci nově vznikající instance. K tomu slouží soubory listener.ora a tnsnames.ora. Podrobnosti o komunikaci viz kapitola 4.5

5.3.6 Vytvoření nové instance

Vyjdeme-li z architektury Oracle Database systému, pokud máme zálohovaná data a transakce s nimi v podobě backupsetů, zbývá pro kompletní systém ještě paměťová struktura a procesy - tedy instance. Instance je zavedena v paměti a s daty pouze pracuje, může tedy být vytvořena nová generická instance, je však důležité zachovat Oracle verzi. Konkrétní konfigurace instance jako je alokace paměti nebo definice recovery prostorů se obnoví ze záloh.

Vytvoření instance se bude lišit v závislosti na operačním systému. Zatímco pod Linuxem je možné nastartovat prázdnou instanci ihned pouze s ukazatelem na pfile, pod Windows je nezbytný ještě jeden krok navíc a to právě ruční vytvoření instance následujícím příkazem. Právě tato část byla pro mne poněkud překvapením, vzhledem k tomu, že veškeré předchozí zkušenosti jsem čerpal pod OS Linux. Nicméně funkcionality je dobře dokumentována a krok navíc není žádná velká komplikace. Nutnost kroku navíc souvisí i s implementací Oracle Database pod OS Windows, kde je možné jednotlivé instance startovat také pomocí Windows služeb.

```
oradim -NEW -SID <sid> -STARTMODE MANUAL -PFILE <pfile>
```

5.3.7 RMAN příkazy obnovy

Nyní se podívejme na samotný proces nastartování prázdné instance, obnovení parametrů, zavedení částí záloh a samotné obnovy dat. Následující RMAN příkazy vychází z dat v předchozí ukázce logu. V této první části je důležité nejprve nastartovat v *nomount* módu a připravit tak prázdnou instanci. Obnovením *server parameter file(spfile)* se získají potřebná nastavení a instance je tím připravena pro spuštění v mount režimu. Barevně jsou zapsány komentáře příkazů.

```
#1 rman> set dbid 763891066;
```

```

#2 rman> startup nomount

#3 rman> restore spfile to pfile '<pfile>' from 'O1_MF_S_760304656_75LC7LWD_.BKP';

#4 rman> startup force nomount pfile='<pfile>';

#5 rman> restore controlfile from 'O1_MF_S_760304656_75LC7LWD_.BKP';

#6 rman> alter database mount

#1 Nastavení DBID, jednoznačného a unikátního identifikátoru databáze. ID zdrojové
databáze se dá lehce dohledat v logu nebo v recovery catalogu.

#2 Přechod do nomount stavu, jelikož neexistují žádné parametry, Oracle zahlásí chybu
ohledně nenalezení souboru (LRM-00109: could not open parameter file), nainstuluje se
vsak právě pro provedení obnovy, nelze totiž pracovat s vypnutou instancí.

#3 Obnovení server parameter file, <pfile> označuje umístění, jméno backup souboru je
získáno z backup logu.

#4 Ve chvíli kdy už je obnoven parameter file, instance přejde do dalšího stavu.

#5 Dalším kritickým souborem je controlfile, uchovávající stav databáze a její
strukturu. Jméno backup souboru je získáno opět z logu.

#6 Nyní už je možné uvést databázi do mount stavu

```

V tuto chvíli máme připravenou instanci a backup, jsou však odděleny a instance nemá zavedeny ani systémové soubory jako je system.dbf, users.dbf nebo sysaux.dbf. V úvodu této sekce jsme sice identifikovali potřebné soubory, ale zbývá ještě vazba mezi instancí a zálohou. Ve chvíli kdy instance bude mít korektně dostupné zálohy, je možné provést obnovu dat. Pro vytvoření takové vazby slouží příkaz catalog. Katalogování souborů je analogické procesům při vytváření zálohy, kdy jsou však odkazy na backup části vytvářeny automaticky bez zásahu administrátora. Všechny potřebné soubory se tedy musí ručně přidat příkazem catalog. Pokračování předchozích RMAN příkazů tedy bude následující:

```

rman> catalog backuppiece 'O1_MF_ANNNN_TAG20111015T190048_79MH43RV_.BKP';
rman> catalog backuppiece 'O1_MF_ANNNN_TAG20111015T190048_79MH8RC5_.BKP';

```

Instance je tedy nastartována a zálohy datových souborů zapsány v katalogu, nezbývá než se pokusit o kompletní obnovu. Ještě před spuštěním příkazu pro obnovení je nutné stanovit přesně moment do kterého konzistentního stavu chceme databázi

uvést. Toto může být čas full backupu nebo jakýkoliv okamžik až do posledního dostupného archivelogu. V příloze je ukázka výpisu příkazu list backup, který slouží pro zobrazení podrobností o všech katalogovaných souborech, včetně jejich SCN¹³, což je hodnota určující každou commit verzi . Součástí přílohy je také vysvětlení určení správného SCN. Pro samotnou obnovu jsem sestavil a použil následující RMAN skript, poznámky k příkazům jsou opět uvedeny barevně pod textem.

```
#1 rman> run{
#2   set until scn 690770341;
#3   restore database;
#4   switch datafile all;
#5   recover database;
    }
```

```
#1 Skript je zapsán v run[] bloku, nastavení typu set totiž platí jen v rámci bloku.
```

```
#2 Nastavení SCN jako poslední transakce do které se bude obnovovat.
```

```
#3 Příkaz pro obnovení databáze, tedy datových souborů
```

```
#4 Přepnutí mezi zálohou a obnoveným souborem
```

```
#5 Fáze Recover "přehraje" veškeré transakce provedené v daném časovém intervalu
```

Následuje otevření databáze s argumentem `resetlogs`, který je v tomto scénáři nutný hlavně kvůli vytvoření souborů pro redo logy. Parameter `resetlogs` také nastaví novou inkarnaci systému.

```
rman> alter database open resetlogs
```

Zde popsany proces vznikl především jako scénář pro vyzkoušení a ověření vytvářených záloh. Jedná se o extrémní případ, například vyhoření serveru, a možnost obnovení předchozího konzistentního stavu. Je tedy zřejmá nutnost zálohování souborů nejen na úrovni databáze prostřednictvím nástroje RMAN, ale i udržování záloh na externím úložišti. V tomto případě se provádí záloha specifikovaných adresářů operačního systému na páskové úložiště.

¹³ System Change Number

5.3.8 Další možnosti a rozšíření

V reálném prostředí je vhodné využít ještě příkaz `set new name` v rámci restore skriptu. Příkazem je možné definovat cestu pro soubory odlišnou od záznamu v původním SP souboru. Zamezí se tak možnosti přepsání již existujících souborů, nebo jen pro přehlednost na disku.

Tento konkrétní postup byl použit pro obnovu produkčních dat na alternativním serveru, nehrozilo tedy pomíchání datových souborů na disku, nicméně i přesto je vždy vhodné nastavit nové názvy souborů právě pro přehlednost, protože alternativní server nebyl využíván pouze pro vyzkoušení produkčních záloh. Je vhodné použít cílovou složku pro soubory například se jménem podle parametru `db_unique_name`.

Nejvíce práce a pozornosti je zde vyžadováno pro nalezení a specifikaci správných souborů. Na začátek byl zvolen postup s použitím logů ponechávaných v rámci záloh, jako efektivnější řešení se nabízí využít již implementovaný *recovery catalog*, která slouží právě k uchování historických informací včetně aktuálních záloh.

Vzhledem k sekundárním zálohám na pásku (na úrovni operačního systému) se nabízí možnost využít přímo možnosti nástroje RMAN pro zápisem na páskové úložiště. RMAN umí definovat kanály pro čtení a zápis s definicí typu příkazem `allocate channel <název> type disk`. Přístup k páskovému úložišti však není v mých kompetencích a pravomocích, tuto možnost jsem nevyužil, nicméně je zde prostor pro dodatečné úpravy.

5.3.9 Využití scénáře a dopad

Scénář v tomto znění a sledu kroků, jak již bylo částečně naznačeno, představuje možnost obnovy celého databázového systému pokud dojde k úplné ztrátě produkčního serveru. Důvody pro ztrátu mohou být různé - selhání hardwaru, počítačový útok, požár v serverovně, přírodní katastrofa - ale vzhledem k důležitosti poskytovaných dat je nutné nejen znát možnosti ale i vyzkoušet si prakticky způsob obnovení ztracených dat. I proto je třetí hardwarový server umístěn v alternativní budově, udržovaný na stejné

úrovni verzí a patchů jako v produkci, aby bylo možné kdykoliv počítač využít pro účely záchrany dat.

Samozřejmě důvod pro obnovu může být i důsledkem lidské neopatrnosti, kdy si mezi sebou pracovníci nevědomě smažou data a přijdou i o měsíce rozdělané práce.

Způsob obnovy popsany v této kapitole je možné použít i pro zachránění původních dat před spuštěním klonovacích skriptů. Spuštěním příkazu `duplicate` se vytvoří datová kopie zdrojové databáze a resetují se všechny záznamy, nicméně až do příštího spuštění lokální zálohy nové instance mohou být na disku stále dostupné původní zálohy, byť ve jsou pro nově vzniklou instanci neznámé. Popsaným způsobem je však možné použít tyto staré zálohy vytvořit nad nimi instanci.

Obnovení ze záloh po nově vytvořeném klonu bylo v rámci podpory prostředí využito když si pracovník zapomněl zálohovat svá data a požádal o naklonování čerstvých dat z produkce.

Za produkční prostředí se považuje dvojice databázových instancí HPROD a FPROD pro PeopleSoft HRMS¹⁴ a Financials, které se používají například pro počítání výplat nebo správu objednávek závazků, ztráta dat má tak přímo vliv na všechny zaměstnance a chod společnosti. Je nutné být připraven na možnost selhání a mít zaběhlé a prakticky vyzkoušené procedury pro obnovení.

Doba potřebná pro obnovu se bude nejvíce odvíjet od času stráveného nad získáním relevantních částí z důvodu identifikace problémového bodu v čase od kdy do kdy je možné získat data. Následující kroky obnovy jsou z velké části automatické a s využitím `recovery catalog` se nabízí i možnost předpřipravených skriptů. Poloautomatické skripty budou v budoucnu dodatečně implementovány. V tuto chvíli je doba obnovy odhadována na dvě až šest hodin, také v závislosti na množství dat, které je potřeba dotáhnout z páskového úložiště. Čistý databázový čas z vlastních testů pro obnovení netrvá ani hodinu.

¹⁴ Human Resources Management System

6 Účast v rámci upgrade projektu

Účelem projektu o kterém zbytek práce pojednává je především migrace a upgrade databázového prostředí systému Oracle Database mezi verzemi 10g a 11g a následná podpora a údržba systému - tedy práce Oracle Database administrátora. Je dáno jak výchozí prostředí, tak cílové servery společně s požadavky migrace. Předmětem práce je navržení a provedení migrace prostředí mezi verzemi, včetně zavedení nových záloh a zdokumentování fail-safe scénářů.

6.1 Výchozí situace

Prostředí ve kterém se bude pracovat není primárně jen datovým zdrojem, na databázové servery jsou navázány další vrstvy celistvé aplikace PeopleSoft Enterprise. Jedná se o integrovaný ERP (enterprise resource planning¹⁵) softwarový balík poskytující škálu business aplikací pro denní provádění a plánování business procesů pro větší až nadnárodní společnosti [8]. Pro stanovené účely však postačí porozumět převážně způsobu organizace dat a přihlášení popsané v kapitole 6.6, není potřeba znát podrobnosti o dalších aplikačních vrstvách - zásahy do ostatních vrstev nebyly nikdy součástí práce ani následné podpory systému.

6.2 Identifikace a definice problému - Oracle certifikace

Firma Oracle poskytuje zákazníkům oficiální podporu přes portál My Oracle Support. Z důvodů ručení za kvalitu a schopnost podpory jsou udržovány podporované sestavy operačních systémů a balíčků, jakožto požadavky pro Oracle produkty. Za tzv. "certifikované" prostředí je považována pouze sestava odpovídající těmto požadavkům.

Pro příklad uveďme sestavu operačního systému HP Tru64 Unix, Oracle verze 10g Release 2 a PeopleSoft HR a Financials 8.9. Toto není certifikované prostředí a proto by v budoucnu byl problém s oficiální Oracle podporou. Přejít na certifikované prostředí je tedy hlavním účelem celého projektu a potažmo této práce. Skutečnost necertifikované platformy lze jednoduše ověřit nahlédnutím do Oracle webového portálu My Oracle Support v sekci Certifications. Podobný problém bude řešit dříve či později každá společnost využívající oficiální Oracle podporu k zakoupeným licencím.

¹⁵ Zkratka ERP se do češtiny překládá jako Efektivní Řízení Podniku

Princip certifikované platformy se nevztahuje pouze na databáze a vzhledem k obrovskému množství různých operačních systémů a kombinací produktu není v silách ani velké společnosti jako je Oracle udržovat záznamy o problémech pro menšinové operační systémy.

Nesplnění certifikovaného prostředí by mělo za následek ztrátu podpory k zakoupenému produktu a ohrožení chodu celého systému například z důvodu nalezeného bugu, pro která však není dostupné řešení. Pokud by taková chyba nastala v procesech zpracování mezd, jsou tak poškozeni stovky pracovníků. Proto vznikl nový projekt a nutnost návrhu nového prostředí.

6.3 Rozložení DB serverů

Ještě před započítím samotné migrace předcházelo krátké období seznámení se stávajícím systémem a zjištění požadavku na novou implementaci. Výchozí databázové prostředí pro Oracle PeopleSoft aplikace je rozloženo na dvou serverech. Zatímco jeden slouží jako testovací pro dvě skupiny kompletních demo a vývojových databází, druhý je striktně vyhrazen pro živou produkci. Na těchto serverech se rozkládá několik databázových instancí, které se mají přesunout na nové fyzické servery, kde už poběží na nové verzi Oracle Database 11g.

6.4 Operační systém/Hardware/Software

Výchozím operačním systémem je Tru64 Unix. Kombinace zdrojového (stávajícího) a cílového nově zvoleného operačního systému bude mít zásadní vliv na potřeby a možnosti migrace. Samotná volba nového hardwaru a operačního systému je předem dána.

6.5 Požadavky

Představme v bodech hlavní vstupní požadavky, od kterých se bude následující práce odvíjet. Návrh a implementace nového prostředí musí dodržovat a splňovat výchozí podmínky uvedené níže.

1. Failsafe scénář pro celé prostředí PeopleSoft databází.
2. Alternativní lokace pro zálohy produkční databáze.

3. Stanovená RTO¹⁶ a RPO¹⁷ na dvě hodiny
4. Co nejlépe využít výkon a zdroje serverů
5. Umožnit co největší FLASH area
6. Zajistit a poskytnout kompletní backup řešení
7. Všechny Oracle instance poběží ve verzi 11g

6.6 Aplikace PeopleSoft – databázová vrstva

Konkrétní produkty pro které jsou stavěné databáze v této práci jsou Financials a Human Resource Management ve verzích 8.9 a v rámci migrace databází mezi Oracle Database 10g a 11g je třeba zjistit vliv migrace na navazující business aplikace a přizpůsobit tomu upgrade proces. Je vhodné porozumět alespoň základní struktuře PeopleSoft databáze.

Jednou z klíčových vlastností ERP systému Oracle PeopleSoft je nezávislost na platformě. Na databázové úrovni je tato skutečnost zajištěna vlastním databázovým slovníkem (Data Dictionary). PeopleSoft tedy nevyužívá k žádným operacím datový slovník databáze Oracle. Jednou výjimkou je vytváření PeopleSoft databáze, kdy jsou DDL skripty částečně upraveny podle cílové platformy.

Vlastní datový slovník jde ruku v ruce s faktem, že systém PeopleSoft je zároveň vývojovým prostředím a je tedy nutné umožnit vytvářet definice objektů ještě než vzniknou “fyzické” záznamy v databázi.

PeopleSoft databáze má tři části: katalog Oracle objektů (Oracle Data Dictionary), PeopleSoft aplikační tabulky obsahující data PeopleSoft uživatelů a tzv. PeopleTools tabulky představující druhý datový slovník. PeopleTools část databáze obsahuje zdrojové kódy komponent a tabulky udržující strukturu PeopleSoft dat nebo indexů.

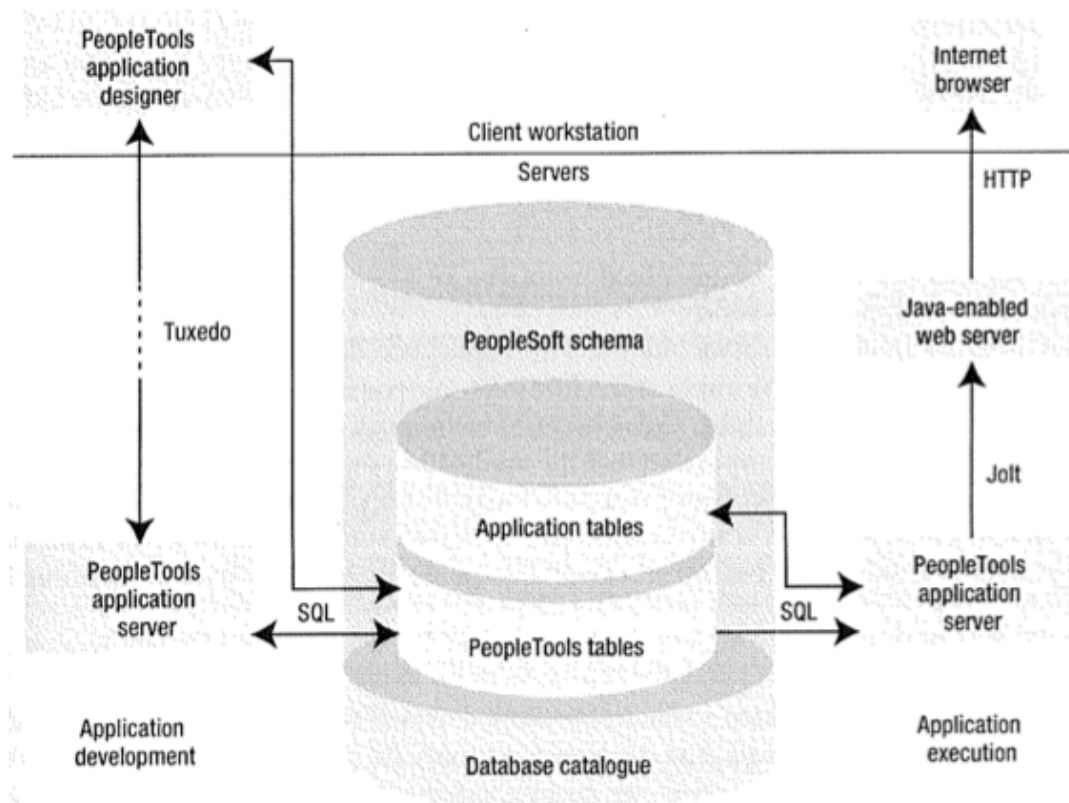
Možným problémem může být použití *temporary tablespace*. Tablespace je logická struktura seskupující stejné datové soubory, temporary tablespace se však vytváří až při startu, nemusí tedy existovat během exportování dat.

PeopleSoft instalace je kombinací mnoha technologií, které dohromady skládají a poskytují aplikaci prostřednictvím webového prohlížeče, PeopleSoft tento řetěz od

¹⁶ Recovery Time Objective - maximální doba trvání obnovy business procesů

¹⁷ Recovery Point Objective - maximální přípustná doba ztracených dat

databáze až po browser nazývá Pure Internet Architecture (PIA). Následující obrázek zobrazuje zjednodušené schéma PeopleSoft architektury.



Obrázek 7. PeopleSoft architektura (Zdroj: PeopleSoft for the Oracle DBA [3])

Souhrnně se veškeré nástroje a aplikace poskytující části PeopleSoft balíku nazývají PeopleTools, patří sem jak vývojový nástroj Application designer tak administrační nástroje pro aplikační server nebo deploy webové vrstvy.

Pro spolehlivou migraci databáze je nutné zjistit požadavky ostatních vrstev, zejména tedy způsoby přístupu k databázovému schématu.

Běžná Peoplesoft databáze obsahuje na nejnižší úrovni pouze dvě schémata s přístupem pro tři, respektive dva uživatele, a přístup k tabulkám a ostatním objektům je řízen na úrovni aplikačních tabulek, nikoli standardními databázovými pravomocemi.

Schéma vlastníka PeopleSoft databáze, běžněji označovaný pojmem Access ID, obsahuje většinu databázových objektů a právě tento uživatel je využíván PeopleSoftem pro přístup (tedy access) k databázi. Nejčastější pojmenování vlastníka PeopleSoft databáze je SYSADM a heslo tohoto uživatele představuje klíč ke království. [3]

Druhým uživatelem je uživatel označovaný jako Connect ID, nejčastěji pojmenován PEOPLE. Veškeré procesy se hlásí pouze pod tímto uživatelem jako první

krok zpracování. Účet má velmi omezené pravomoce, jediné přidělené právo je totiž CREATE SESSION a právo selektu nad trojicí tabulek PSSTATUS, PSOPRDEFN a PSACCESSPRFL definující převážně ostatní uživatele v rámci PeopleSoft aplikace, zvaní User ID, a jejich pravomoce v rámci systému. Až po User ID autentifikaci se proces přihlásí jako Access ID a pokračuje ve zpracování dostupných dat.

Třetí a poslední schéma, označované vždy jako PS, je používáno pro jedinou tabulku PSDBOWNER, která mapuje jméno PeopleSoft databáze a jejího vlastníka. Je to jediná tabulka v jiném schématu než všechny ostatní objekty. Uživatelé PS jsou hned po vytvoření tabulky navíc odebrána práva CREATE SESSION, zabrání se tak přístupu. [3]

6.7 Migrace

Přenesení dat mezi dvěma odlišnými systémy, v tomto případě mezi databázovým systémem různých verzí, bývá označováno pojmem migrace. Nejedná se o klonování instance popsané v kapitole, 5.2 kdy se zkopíruje celá databáze jedna ku jedné, ale o přesunutí pouze dat mezi zdrojem a cílem. Nejedná se o běžnější upgrade v rámci jednoho serveru, nicméně import ve verzi Oracle 11g je možné provést na základě exportovaných dat na nižší verzi 10g. Tento fakt je nutné ověřit, nejprve se provede několik testovacích běhů vývojových a testovacích databází a až po úspěšném průběhu a po domluvení tzv. downtime¹⁸ se stejný postup zopakuje v produkčním prostředí.

Migrace mezi servery se standardně provádí ve dvou krocích exportování a importování dat.

6.7.1 Datová pumpa

Doporučeným nástrojem pro export a import dat mezi systémem Oracle 10g a Oracle 11g je Datová Pumpa. Jedná se o následníka ve verzi 11g Release 2 dříve používaných nástrojů Import a Export. Oproti starším utilitám *exp* a *imp* má být datová pumpa až 45x rychlejší při importu [7] s podporou více možností jako jsou externí tabulky nebo dodatky pro použití PL/SQL skriptů. Právě použití datové pumpy bylo

¹⁸ Downtime představuje čas kdy je prostředí vypnuto nebo zamezen přístup běžných uživatelů za účelem údržby. Je to předem vymezený časový úsek, které je třeba dodržet.

součástí prvního návrhu provedení migrace, jedná se o plně podporovaný nástroj bez nutnosti separátního licencování, navíc je integrovaná v grafickém nástroji SQL Developer.

6.7.2 Transportable tablespace

Jednou z alternativ je použití méně známe technologie Transportable Tablespaces, která je však dostupná již od verze Oracle 8i. Principem Transportable Tablespaces je uvedení logické struktury tablespace do výjimečného stavu pouze pro čtení (read-only) a následného překopírování, metoda je tedy vhodná pro takové systému kde je důležitým faktorem přístup k datům bez ohledu na možnost změn. Čas potřebný pro migraci je tedy pouze doba nutná pro datový přenos.

Metoda může vycházet i ze záloh vytvořených nástrojem RMAN kdy není nutný přístup k živým datům systému.

6.7.3 Export a Import

Ačkoliv se jedná o nejzastaralejší způsob přenosu dat, nakonec byl zvolen obyčejný export a import celého SYSADM schématu (viz architektura PeopleSoft kapitola 6.7) a to z jednoduchého důvodu: v minulosti již byl tento způsob použit bez komplikací v rámci přesunu dat na zdrojovém serveru mezi verzemi 9i a 10g. Pro první testy se tedy využila již známá procedura a vzhledem k tomu že i pro vyšší verzi 11g proběhl import bez chyb v dostatečně krátkém časovém intervalu, nebyl důvod pro změnu technologie. Použitý skript je v následujícím extraktu. Příkaz `exp` je přímo odkaz na program dodávaný v rámci instalace Oracle Database, jediné nutné nastavení pro správná chod exportu je tedy proměnná prostředí `ORACLE_HOME` a `ORACLE_SID` a doplnění proměnné `PATH`.

```
exp sysadm/<heslo>@<SID> file=<cílový soubor> log=<log> owner=sysadm  
consistent=y statistics=none
```

Jsou zde použity pouze dva přepínače ve smyslu změnění defaultní hodnoty - *consistent*, *direct* a *statistics*. Nastavením přepínače `consistent` se zajistí nastavení systému `SET TRANSACTION READ ONLY`, zaručující konzistentní stav exportovaných dat

zamezením provádění DML operací v rámci transakce. Přepínač statistics už jen specifikuje ignorování statistik pro tabulky, je totiž vhodnější provést po migraci výpočet nových, čerstvých statistik podle provozu.

6.8 Kroky postupu

Postup celého procesu přípravy nového prostředí migrace dat je dobré si předem rozmyslet a rozvrhnout. Následující seznam představuje přehled potřebných kroků ve vhodném pořadí. Celý project byl realizován jen jednou osobou, nebylo potřeba vymýšlet rozdělení práce nebo složité projektové řízení vzhledem k tomu, že nebyl stanoven termín nutného dokončení.

1. Definice a příprava filesystem struktury nových serverů (oradata¹⁹, flasharea²⁰)
2. Instalace RDBMS, nastavení parametrů
3. Vytvoření instancí a nastavení Oracle Net
4. Dodatečné parametry pro PeopleSoft
5. Data export na zdrojovém systému
6. Data import na cílovém systému
7. Zavedení backupů
8. Vyzkoušení obnovy ze záloh
9. Dodatečné implementace - Grid Control, Data Guard

6.9 Popis prvků nové implementace

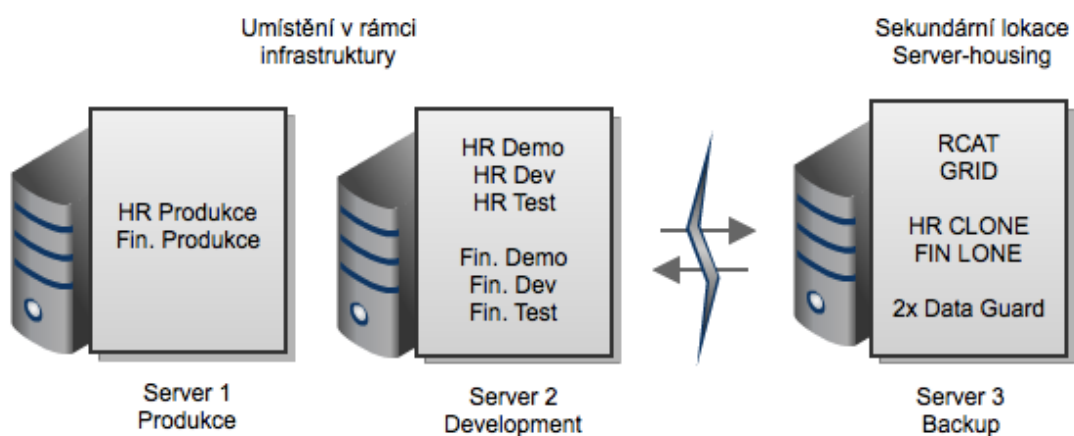
Dostupné jsou 3 fyzické servery s operačním systémem Windows Server 2008 SP2 64bit na které se má přesunout celkem 10 PeopleSoft databází: 2 produkční, 6 test a developemnt instancí a 2 klony produkce. Jedná se o databáze ERP systému Oracle PeopleSoft dvou produktových řad HRMS a Financials ve verzích 8.9. Tyto databáze musí být zachovány bez ztráty dat, navíc je ještě ponechán prostor pro další možnosti a zlepšení, například zavedení technologie Data Guard.

V rámci infrastruktury zůstanou umístěny dva hardwarově identické servery v podobné sestavě jako ve výchozím prostředí, jeden produkční a jeden vývojový. Třetí

¹⁹ Umístění datových souborů

²⁰ Fast Recovery Area - umístění lokálních záloh

server má být umístěn v jiné budově u nedalekého poskytovatele server-housing řešení a jeho primárním úkolem bude umístění tzv. Recovery Catalogu, Oracle Grid Control 11g společně s návrhem zavedení dvou Data Guard instancí pro případ selhání jedné z primárních databází. Nadále však zůstanou prvotním a kritickým zdrojem obnov klasické RMAN zálohy. Pro lepší utilizaci třetího serveru zde budou také zavedeny klonovací procedury pro případ potřeby čerstvých dat během vývoje nebo testování. Schéma navrženého rozvržení zachycuje následující obrázek.



Obrázek 8. Rozvržení databází (Zdroj: Vlastní tvorba)

6.9.1 Data Guard

Vzhledem k tomu že se používá verze Oracle Database 11g Enterprise Edition, pro doplnění pravidelných záloh se nabízí využití high availability technologie Data Guard, která je zahrnuta v Enterprise edici jako standartní služba.

Principem technologie je tzv. *Standby* databáze na jiném serveru, ne které se souběžně s primárním zdrojem přehrávají veškeré provedené transakce. Standby databáze prvotně vznikne naklonováním a následně se na ní přehrávají transakce odchycené již při zápisu do redo logu. Primární a Standby databáze jsou tedy naprosto identické.

Výsledkem implementace Data Guard je podvojný systém produkčních databází, které mezi sebou dovedou přepínat role primární a sekundární lokace. Pokud například dojde k selhání produkce na primárním serveru, Standby databáze je okamžitě schopná převzít úlohu primárního poskytovatele dat bez narušení navazujících procesů.

Bohužel v době psaní této práce ještě nebyla zavedena Data Guard implementace pro produkční servery a to z důvodu problémové síťové komunikace nebo bugů, technologie je totiž podle zvoleného režimu náchylná na ztrátu paketu apod.

6.9.2 Recovery Catalog

Databáze označovaná pojmem Recovery Catalog je separátní instance určená pro udržování historických dat o provedených zálohách na registrovaných instancích. Pro každou registrovanou instanci je v zde veden separátní katalog udržující informace například o umístění zálohových částí, nebo hodnoty nastavených parametrů. Pro účely zálohování zastupuje Recovery Catalog *control file* registrované instance.

6.9.3 Enterprise Manager a Grid Control 11g

Pro správu Oracle databáze je hlavním nástrojem *sqlplus* spouštěné pouze z příkazové řádky, nicméně existují i pokročilejší nástroje s grafickým prostředím, například Oracle Enterprise Manager. Jedná se o webovou aplikaci umožňující veškeré operace od nastavení, zálohování, ladění a performance metrik až po nastavení pokročilejších funkcí jako je Oracle Data Guard. Při použití Database Configuration Wizard pro vytvoření databáze je v rámci instalace nakonfigurován i Enterprise Manager.

Oproti Enterprise Managerovi, který je součástí distribuce Oracle Enterprise Edition a je vázán na jedinou instanci, Grid Control je zaměřen na správu libovolného počtu systémů a lze jej použít kromě správy databází i pro správu hostujícího operačního systému, aplikačních serverů nebo web serverů. Pokrývá tak celé spektrum čtyřvrstvé aplikační architektury, kterou využívá i PeopleSoft.

6.9.4 Implementace pravidelných záloh

Zálohování na nových serverech se bude opakovat v týdenních intervalech nulté úrovně inkrementálního backupu společně s každodenními zálohami úrovně první. Zálohovací skripty jsou vždy spouštěny v období minimálního provozu. Celý

PeopleSoft systém je využíván v několika časových pásmech a okno bez provozu je stanoveno v období od půlnoci do čtyř rána v GMT+1 pásmu.

Způsob zálohování je stejný jak pro produkční tak pro vývojové servery s výjimkou nastavení retenční doby. V produkčním prostředí je žádoucí ponechat dostupné starší zálohy a retence ponechání souborů tedy byla stanovena na dva týdny, zatímco na vývojových serverech je ponechána defaultní RMAN konfigurace ponechání pouze jedné kopie souborů nebo logů. Ponechání kratšího intervalu záloh ve vývojovém prostředí je mimo jiné dáno kapacitou diskového pole.

6.9.5 Vytváření záloh - RMAN

Pro vytváření a správu záloh je nejvhodnější použít nástroj RMAN dodávaný se systémem Oracle Database. Teoretický proces obnovy již byl popsán v kapitole 4.2.7 bez uvedení příkladů. Používaný skript pro zálohování databází napříč prostředím je následující.

```
run{
    backup incremental level 0 database;
    delete noprompt obsolete;
    crosscheck archivelog all;
    delete expired archivelog all;
}
```

Zálohovací skript sestává ze dvou částí - provedení zálohy a odmazání starých nepotřebných souborů. V rámci implementace byl zvolen týdenní interval záloh s každodenním inkrementem, zde zobrazený skript představuje zálohu úrovně nula, který se od navazujících inkrementálních částí liší pouze definicí úrovně v úvodu skriptu.

Po skončení zálohy je vhodné odmazat již nepotřebné, zvané *obsolete*, soubory. Všechny části záloh jsou ponechávány v závislosti na nastavené retenční době, která byla na produkčním serveru stanovena jako interval dvou týdnů.

Nastavení retenční doby není řešeno separátně s každým spuštěním skriptu, RMAN udržuje vlastní konfigurační soubory s definicí jak retenční doby, tak například způsob zálohování, typy zařízení nebo způsob šifrování. Toto vnitřní nastavení se provádí příkazem přímo z příkazové řádky nástroje příkazem `configure`, tedy

například nastavení pro udržení záloh po dobu dvou týdnů příkazem `configure retention policy to recovery window of 7 days.`

6.9.6 Pravidelné ověření restore procedury

V rámci nasazení pravidelných záloh v týdenních intervalech je vhodné zavést proces pravidelné kontroly jednak korektního průběhu a vytvoření například zálohy z předešlé noci, ale i důslednější vyzkoušení vytvořených backup částí. Kapitola 5.4 Ukázková Restore & Recover procedura popisuje podrobně způsob obnovy ze záloh. Jedná se o nestandardní testovací běh, který je v rámci podpory proveden jednou za tři měsíce.

7 Závěr

Bakalářská práce pokryla nejdůležitější teoretické principy od architektury databázového systému Oracle až po teoretické podklady pro běžné činnosti jako je zálohování. Na základě teoretických principů a znalostí získaných absolvováním Oracle Administration Workshop pak byly popsány některé praktické postupy využívané v rámci práce jako databázový administrátor. Postupy a metody popsány v této práci lze aplikovat v každém prostředí založeném na Oracle 11g.

Nabyté know-how v oblasti správy databázového systému Oracle je dále rozšiřováno během poskytování podpory podnikového prostředí a údržby zavedených procesů.

Kromě ukázání praktických postupů je důležitou částí práce úspěšná migrace dat mezi verzemi a vyřešení necertifikovaného prostředí za účelem zabránění možným budoucím problémům. V principu se jedná o podobnou záležitost jako aktualizace verze programu, jen ve větším měřítku. Zde navržené a postupně realizované řešení navíc plně vyhovuje požadavkům, ačkoliv ještě nabízí prostor pro další zlepšení jako je například definitivní zavedení technologie Data Guard.

Problém necertifikovaného prostředí a hrozby ztráty oficiální podpory se nevztahuje pouze k jednomu zákazníkovi, stejné úvahy a postup bude využívat každý kdo v tuto chvíli stále využívá starší verze a dříve nebo později bude nucen podstoupit upgrade databázového systému ať už podobným postupem migrace mezi fyzickými servery nebo zachování původních hardwarových zdrojů.

Z dlouhodobého projektu ať už samotné migrace nebo následné podpory prostředí lze stanovit několik jednoduchých doporučení. Vzhledem k tomu že Oracle platforma je rozšířená po celém světě, je velice pravděpodobné že na podobný problém je již nalezeno řešení. Nesmírně se vyplatí pečlivá práce s oficiální knowledge base v podobě My Oracle Support, stejně jako studium dokumentace, pochopení teoretických principů nebo i běžný podvojný způsob implementace nejprve v menším testovacím prostředí a až následně v produkci.

8 Seznam použitých zdrojů

[1] Deirdre Matishak a Mark Fuller: Oracle Administration Workshop I. Study Guide, Edition 2.0, Oracle srpen 2010, D50102GC20

[2] Maria Billings: Oracle Administration Workshop II. Study Guide, Edition 2.0, Oracle září 2010, D50079GC20

[3] David Kurtz: PeopleSoft for the Oracle DBA, Apress 2005, ISBN: 1-59059-422-3

[4] Kreines, D.C.: Oracle DBA - kapesní průvodce, GRADA 2006, ISBN 80-247-1669-0

[5] Procházka, D.: Oracle - průvodce správou, využitím a programováním, GRADA 2009, ISBN 978-80-247-2762-2

[6] <http://arjudba.blogspot.com/2008/05/restore-and-recover-database-to-new.html>

[7] <http://www.oracle.com/technetwork/database/enterprise-edition/index-093639.html>

[8] <http://arc-consulting.cz>

[9] Alan Beaulieu: Learning SQL, O'Reilly srpen 2005, ISBN 0-596-00727-2

[10] Iggy Fernandez: Beginning Oracle 11g Database Administration, Apress 2009, ISBN-10 1-59059-968-3

9 Přílohy

9.1 Příkaz list backup

rman> list backup;

List of Backup Sets

=====

BS Key	Size	Device Type	Elapsed Time	Completion Time
--------	------	-------------	--------------	-----------------

5520	396.79M	DISK	00:02:38	15-OCT-11
------	---------	------	----------	-----------

BP Key: 5541 Status: AVAILABLE Compressed: YES Tag: TAG20111015T190048

Piece Name:

E:\KPZ\PROD_BACKUP\2011_10_15\O1_MF_ANNNN_TAG20111015T190048_79MKSH03_BKP

List of Archived Logs in backup set 5520

Thrd Seq	Low SCN	Low Time	Next SCN	Next Time
----------	---------	----------	----------	-----------

1	37170	690296258	14-OCT-11	690301863	14-OCT-11
---	-------	-----------	-----------	-----------	-----------

1	37171	690301863	14-OCT-11	690306944	14-OCT-11
---	-------	-----------	-----------	-----------	-----------

1	37172	690306944	14-OCT-11	690312517	14-OCT-11
---	-------	-----------	-----------	-----------	-----------

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
--------	------	---------	-------------	--------------	-----------------

5521	Incr 0	494.68M	DISK	00:03:24	15-OCT-11
------	--------	---------	------	----------	-----------

BP Key: 5542 Status: AVAILABLE Compressed: YES Tag: TAG20111015T195017

Piece Name:

E:\KPZ\PROD_BACKUP\2011_10_15\O1_MF_NNND0_TAG20111015T195017_79MKZP6W_BKP

List of Datafiles in backup set 5521

File	LV	Type	Ckp SCN	Ckp Time	Name
------	----	------	---------	----------	------

14	0	Incr	690768806	15-OCT-11	E:\ORACLE\ORADATA\KPZ\DATAFILES\CCAPP.DBF
----	---	------	-----------	-----------	---

126	0	Incr	690768806	15-OCT-11	E:\ORACLE\ORADATA\KPZ\DATAFILES\WAAPP.DBF
-----	---	------	-----------	-----------	---

131	0	Incr	690768806	15-OCT-11	E:\ORACLE\ORADATA\KPZ\DATAFILES\PSDEFAULT02.DBF
-----	---	------	-----------	-----------	---

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
--------	------	---------	-------------	--------------	-----------------

5522 Incr 0 193.66M DISK 00:01:38 15-OCT-11

BP Key: 5543 Status: AVAILABLE Compressed: YES Tag: TAG20111015T195017

Piece Name:

E:\KPZ\PROD_BACKUP\2011_10_15\O1_MF_NNND0_TAG20111015T195017_79ML6DXJ_BKP

List of Datafiles in backup set 5522

File LV Type Ckp SCN Ckp Time Name

3 0 Incr 690769344 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\UNDOTBS01.DBF
4 0 Incr 690769344 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\USERS01.DBF
128 0 Incr 690769344 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\SYSAUX02.DBF
132 0 Incr 690769344 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\UNDO3.DBF

BS Key Type LV Size Device Type Elapsed Time Completion Time

5523 Incr 0 644.30M DISK 00:04:11 15-OCT-11

BP Key: 5544 Status: AVAILABLE Compressed: YES Tag: TAG20111015T195017

Piece Name:

E:\KPZ\PROD_BACKUP\2011_10_15\O1_MF_NNND0_TAG20111015T195017_79ML9NMK_BKP

List of Datafiles in backup set 5523

File LV Type Ckp SCN Ckp Time Name

1 0 Incr **690770341** 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\SYSTEM01.DBF
2 0 Incr **690770341** 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\SYSAUX01.DBF
127 0 Incr **690770341** 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\SYSTEM02.DBF
129 0 Incr **690770341** 15-OCT-11 E:\ORACLE\ORADATA\KPZP\DATAFILES\SYSAUX03.DBF

BS Key Size Device Type Elapsed Time Completion Time

5524 18.86M DISK 00:00:10 15-OCT-11

BP Key: 5545 Status: AVAILABLE Compressed: YES Tag: TAG20111015T200023

Piece Name:

E:\KPZ\PROD_BACKUP\2011_10_15\O1_MF_ANNNN_TAG20111015T200023_79MLKTVM_BKP

List of Archived Logs in backup set 5524

Thrd Seq Low SCN Low Time Next SCN Next Time

```

1 37233 690759044 15-OCT-11 690764137 15-OCT-11
1 37234 690764137 15-OCT-11 690768526 15-OCT-11
1 37235 690768526 15-OCT-11 690770962 15-OCT-11

```

RMAN>

V tomto případě je bod full backupu na SCN **690770341**. Pro obnovení až do stavu posledního archiveologu, je vhodné SCN **690770962**.

9.2 Recovery Catalog pohledy

Recovery Catalog View	Catalog View Describes ...
<u>RC_ARCHIVED_LOG</u>	Archived and unarchived redo logs
<u>RC_BACKUP_ARCHIVELOG_DETAILS</u>	Details about archived redo log backups for Enterprise Manager
<u>RC_BACKUP_ARCHIVELOG_SUMMARY</u>	Summary of information about archived redo log backups for Enterprise Manager
<u>RC_BACKUP_CONTROLFILE</u>	Control files backed up in backup sets
<u>RC_BACKUP_CONTROLFILE_DETAILS</u>	Details about control file backups for Enterprise Manager
<u>RC_BACKUP_CONTROLFILE_SUMMARY</u>	Summary of information about control file backups for Enterprise Manager
<u>RC_BACKUP_COPY_DETAILS</u>	Details about datafile image copy backups for Enterprise Manager
<u>RC_BACKUP_COPY_SUMMARY</u>	Summary of information about datafile image copy backups for Enterprise Manager
<u>RC_BACKUP_CORRUPTION</u>	Corrupt block ranges in datafile backups
<u>RC_BACKUP_DATAFILE</u>	Datafiles in backup sets
<u>RC_BACKUP_DATAFILE_DETAILS</u>	Details about datafile backups for Enterprise Manager

<u>RC_BACKUP_DATAFILE_SUMMARY</u>	Summary of information about datafile backups for Enterprise Manager
<u>RC_BACKUP_FILES</u>	RMAN backups and copies known to the repository.
<u>RC_BACKUP_PIECE</u>	Backup pieces
<u>RC_BACKUP_PIECE_DETAILS</u>	Details about backup pieces for Enterprise Manager
<u>RC_BACKUP_REDOLOG</u>	Archived redo logs in backup sets
<u>RC_BACKUP_SET</u>	Backup sets for all incarnations of databases registered in the catalog
<u>RC_BACKUP_SET_DETAILS</u>	Details about backup sets for Enterprise Manager
<u>RC_BACKUP_SET_SUMMARY</u>	Summary of information about backup sets for Enterprise Manager
<u>RC_BACKUP_SPFILE</u>	Server parameter files in backups
<u>RC_BACKUP_SPFILE_DETAILS</u>	Details about server parameter file backups for Enterprise Manager
<u>RC_BACKUP_SPFILE_SUMMARY</u>	Summary of information about server parameter file backups for Enterprise Manager
<u>RC_CHECKPOINT</u>	Deprecated in favor of <u>RC_RESYNC</u>
<u>RC_CONTROLFILE_COPY</u>	Control file copies on disk
<u>RC_COPY_CORRUPTION</u>	Corrupt block ranges in datafile copies
<u>RC_DATABASE</u>	Databases registered in the recovery catalog
<u>RC_DATABASE_BLOCK_CORRUPTION</u>	Database blocks marked as corrupted in the most recent RMAN backup or copy
<u>RC_DATABASE_INCARNATION</u>	Database incarnations registered in the recovery catalog

<u>RC_DATAFILE</u>	Datafiles registered in the recovery catalog
<u>RC_DATAFILE_COPY</u>	Datafile copies on disk
<u>RC_LOG_HISTORY</u>	Online redo log history indicating when log switches occurred
<u>RC_OFFLINE_RANGE</u>	Offline ranges for datafiles
<u>RC_PROXY_ARCHIVEDLOG</u>	Archived log backups taken with the proxy copy functionality
<u>RC_PROXY_ARCHIVELOG_DETAILS</u>	Details about proxy archived redo logs for Enterprise Manager
<u>RC_PROXY_ARCHIVELOG_SUMMARY</u>	Summary of information about proxy archived redo logs for Enterprise Manager
<u>RC_PROXY_CONTROLFILE</u>	Control file backups taken with the proxy copy functionality
<u>RC_PROXY_COPY_DETAILS</u>	Details about datafile proxy copies for Enterprise Manager
<u>RC_PROXY_COPY_SUMMARY</u>	Summary of information about datafile proxy copies for Enterprise Manager
<u>RC_PROXY_DATAFILE</u>	Datafile backups that were taken using the proxy copy functionality
<u>RC_REDO_LOG</u>	Online redo logs for all incarnations of the database since the last catalog resynchronization
<u>RC_REDO_THREAD</u>	All redo threads for all incarnations of the database since the last catalog resynchronization
<u>RC_RESTORE_POINT</u>	All restore points for all incarnations of the database since the last catalog resynchronization
<u>RC_RESYNC</u>	Recovery catalog resynchronizations
<u>RC_RMAN_BACKUP_JOB_DETAILS</u>	Details about backup jobs for Enterprise Manager

<u>RC_RMAN_BACKUP_SUBJOB_DETAILS</u>	Details about backup subjobs for Enterprise Manager
<u>RC_RMAN_BACKUP_TYPE</u>	Used internally by Enterprise Manager
<u>RC_RMAN_CONFIGURATION</u>	RMAN configuration settings
<u>RC_RMAN_OUTPUT</u>	Output from RMAN commands for use in Enterprise Manager
<u>RC_RMAN_STATUS</u>	Historical status information about RMAN operations.
<u>RC_SITE</u>	Databases in a Data Guard environment
<u>RC_STORED_SCRIPT</u>	Names of scripts stored in the recovery catalog
<u>RC_STORED_SCRIPT_LINE</u>	Contents of the scripts stored in the recovery catalog
<u>RC_TABLESPACE</u>	All tablespaces registered in the recovery catalog, all dropped tablespaces, and tablespaces that belong to old incarnations
<u>RC_TEMPFILE</u>	All tempfiles registered in the recovery catalog
<u>RC_UNUSABLE_BACKUPFILE_DETAILS</u>	Unusable backup files registered in the recovery catalog