UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan Harnik

NESREČE ORACLE PODATKOVNIH BAZ IN VARNOSTNE STRATEGIJE

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01628/2010
Datum: 15.01.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **BOŠTJAN HARNIK**

Naslov: **NESREČE ORACLE PODATKOVNIH BAZ IN VARNOSTNE STRATEGIJE**

ORACLE DATABASE DISASTERS AND SECURITY STRATEGIES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Nesreče računalniških sistemov so pogoste in če se želimo zavarovati pred neželenimi posledicami (npr. izguba podatkov), se moramo pravilno zavarovati.

V diplomski nalogi predstavite primere nesreč, ki so najpogostejše v sistemih za obvladovanje podatkovnih baz. Osredotočite se na sistem ORACLE ter predlagajte najprimernejše varnostne strategije, s katerimi se lahko zavarujemo pred nesrečami.

Mentor:

prof. dr. Marko Bajec

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Franc Solina

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan Harnik

NESREČE ORACLE PODATKOVNIH BAZ IN VARNOSTNE STRATEGIJE

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

MENTOR: dr. Marko Bajec

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01628/2010
Datum: 15.01.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **BOŠTJAN HARNIK**

Naslov: **NESREČE ORACLE PODATKOVNIH BAZ IN VARNOSTNE STRATEGIJE**

ORACLE DATABASE DISASTERS AND SECURITY STRATEGIES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Nesreče računalniških sistemov so pogoste in če se želimo zavarovati pred neželenimi posledicami (npr. izguba podatkov), se moramo pravilno zavarovati.

V diplomski nalogi predstavite primere nesreč, ki so najpogostejše v sistemih za obvladovanje podatkovnih baz. Osredotočite se na sistem ORACLE ter predlagajte najprimernejše varnostne strategije, s katerimi se lahko zavarujemo pred nesrečami.

Mentor:
prof. dr. Marko Bajec

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:
prof. dr. Franc Solina

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani/-a Boštjan Harnik,

z vpisno številko 63030089,

sem avtor/-ica diplomskega dela z naslovom:

Nesreče Oracle podatkovnih baz in varnostne strategije

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

  dr. Marko Bajec

  in somentorstvom (naziv, ime in priimek)

  _____

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja/-ice: _____

Zahvala

Za izkazano pomoč, usmerjanje in podporo pri izdelavi diplomskega dela se zahvaljujem mentorju dr. Marku Bajcu. Iskreno se zahvaljujem tudi svoji družini, še posebej staršema, ki me podpirata že od samega začetka študija. Hvala.

Kazalo

Kazalo slik

Seznam uporabljenih simbolov in kratic

Simboli:

#	- rezultati poizvedb znotraj Sqlplus urejevalnika
$	- ukaz se izvaja iz ukazne vrstice Linux-a
/ x /	- x diskovna enota
//	- poročilo alert.log datoteke
rman>	- ukaz se izvaja znotraj RMAN klienta
sql>	- ukaz se izvaja znotraj Sqlplus urejevalnika

Kratice:

ASCII	- American standard code for information interchange
b	- Bit
B	- Bajt
BS	- British standards
DBW0	- Database writer 0
DDL	- Data definition language
DML	- Data manipulation language
EXP	- Oracle export utility
FTP	- File transfer protocol
G	- Giga
IMP	- Oracle import utility
ISO	- International Organization for Standardization
K	- Kilo
M	- Mega
NTS	- Windows NT native authentication
PB	- Podatkovna baza
RMAN	- Recovery manager
SCN	- System change number
SQL	- Structured query language
SUPB	- Sistem za upravljanje s podatkovno bazo
TSPITR	- Tablespace point in time recovery

Povzetek

Diplomsko delo se osredotoča na podatke, jih tudi predstavi ter nam pove kaj ti pravzaprav so, hkrati pa opredeli njihovo vlogo pri pridobivanju informacij. Podrobno opiše postopek, skozi katerega izluščimo ustrezne, kvalitetne, natančne, pravočasne in varne informacije. Opisuje namen shranjevanja podatkov in prednosti, ki jih s tem pridobimo. Sprehodi se skozi zgodovino medijev za shranjevanje podatkov, vse od kamna in papirja do današnjih podatkovnih diskov, ki so sprožili pravo podatkovno revolucijo. Na kratko se ustavi pri podatkovnih bazah, ki nam omogočajo računalniško podprto strukturirano shranjevanje podatkov, ter pri sistemu za upravljanje s podatkovnimi bazami, ki nam je v veliko pomoč pri shranjevanju in administraciji podatkov. Vse to pa nas privede do osrednje teme diplomskega dela, to je varnost podatkov v sami podatkovni bazi. Naredi se pregled standardnih nesreč podatkovne baze, katere varnostno pokrijemo z našo strategijo, nad par nesrečami, pa tudi ocenimo stroške, ki jih le te prinašajo.

Ob vsem tem se poraja vprašanje kaj storiti, da bodo naši podatki znotraj podatkovnih baz varnejši? Velik poudarek je na podrobnem pregledu vrst in načinov varnostnih kopij, kot so komulativne, diferencialne, inkrementalne, tedenske, mesečne ... Ne izpusti se niti študija pogostosti oziroma frekvenc izdelovanja varnostnih kopij, ki je seveda odvisna od izvajanih aktivnosti podatkovne baze. Osrednji namen diplomskega dela pa je seznaniti bralca s tremi osnovnimi orodji, ki jih Oracle ponuja za zagotavljanje varnosti, ki so izvozno/uvozna funkcionalnost, razveljavitev sprememb in obnovitveni upravitelj. Sledi obsežna raziskava za kakšne probleme je kateri izmed njih najprimernejši in opis njegove vloge v naši varnostni strategiji.

Diplomsko delo se usmeri tudi na praktično področje in opiše konkretno študijo razporeditve samih fizičnih datotek Oracle podatkovne baze, med dva podatkovna diska na primarni lokaciji, ter enemu na oddaljeni sekundarni lokaciji. Naredi se sistematični pregled, kaj je v primeru nesreče ali odpovedi enega, dveh ali celo vseh podatkovnih diskov možno narediti, da izgubimo čim manjšo količino podatkov. Tako se lepo povežeta prvi teoretični in drugi praktični del naloge. Jasno uvrstimo vso našo teorijo varnostnih kopij v primer sistematizacije nesreč podatkovnih diskov. Postavimo razmeroma poceni in stabilno strukturo Oracle podatkovne baze, ki ustreza ISO 17799 standardu.

Ključne besede: podatkovna, baza, varnostna, strategija, Oracle.

Abstract

This diploma work focuses on data, their introduction and it describes what data really is. At the same time it also defines their role in gaining new information. It thoroughly represents the process, which helps us get the relevant, high-quality, accurate and secure information. It also defines the purpose and benefits of storing data. We also mention the history of media storage devices from stone to paper up to the present data discs, which caused real data revolution. It briefly stops at the databases, which allow us structured computer-based data storage, and management system for databases, which are very helpful in terms of data storage and administration of database. All this brings us to the safety of the database itself, the central theme of graduate work. We make a standard review of expected database disasters which should be covered by our security strategy and estimate cost for couple of these disasters.

What can we do to make our database, or data whitin, more secure? Much emphasis is put on detailed review of the types and methods for backup, such as cumulative, differential, incremental, weekly, monthly etc. Diploma work although includes the study of frequency for making backup copies, which is of course dependent on activity within the database itself. The main goal of the thesis is to acquaint the reader with three basic tools, which Oracle provides for ensuring the safety of database:
- export/import utility,
- flashback functionality,
- recovery manager.

Thesis is onward focused on extensive research for usability of these three tools in our security strategy.

After theory we finally move on to some practical work and describe a specific study for allocation of Oracle physical database files between two data disks, on the primary location, and one on the remote, secondary location. Systematic review of what could be done in the event of an accident or failure of one, two or even all data disks, to lose the least information possible. This connects the first theoretical and second practical part of the diploma work. Systemizing accidents of data discs and solving problems caused by accidents gives a coherent set of relatively inexpensive and very stably structured Oracle database itself. This structure is compliant with ISO 17799 standard.

Keywords: data, database, security, strategy, Oracle.

1 Podatki

Podatke opredeljujemo kot zapisana dejstva, ki jih zbiramo in s katerimi zbudimo v možganih določeno predstavo. Podatek je na primer temperatura 10 ºC. Kadar nas zanima le temperatura, nam ta podatek povsem zadošča. Kadar pa želimo vedeti, kakšno je vreme, potrebujemo več podatkov in nam ta podatek sam ne zadošča. Potrebujemo še podatke o padavinah, vetru, vlažnosti itd. Človek s pomočjo svojega znanja pripisuje podatkom pomen in s tem svoje znanje dopolni in nanj ustrezno odreagira. Računalnik podatkom ne pripisuje pomena, ampak jih samo predela v obliko, ki si jo zaželi človek, ali v obliko, primerno za krmiljenje procesov.

Definirajmo podatek [1]:

- podatek je poljubna predstavitev s pomočjo simbolov ali analognih veličin, ki je predpisana ali se ji lahko predpiše nek pomen,
- podatek je predstavitev dejstva, koncepta ali instrukcije na formaliziran način, ki je primeren za komunikacijo, interpretacijo ali obdelavo s strani človeka ali stroja.

Iz prve definicije lahko izluščimo, da je lahko podatek diskreten, če so pri predstavitvi uporabljeni simboli (npr. 25 ºC), ali pa je analogen, če se za predstavitev uporablja kakšna fizikalna veličina (npr. dolžina živosrebrnega stolpca).

Druga definicija pa nam pove, da mora biti predstavitev izvedena na formaliziran način, kar pomeni, da mora obstajati nek predpis – konvencija, po katerem simbole ali vrednosti analognih veličin zapisujemo oziroma beremo.

Informacija ima pomen in prejemniku pove nekaj novega, seveda pa mora biti razumljiva. S tem informacija poveča znanje prejemnika in vpliva na odločitve in ravnanje posameznika.

Obstaja več definicij termina informacija [2]:

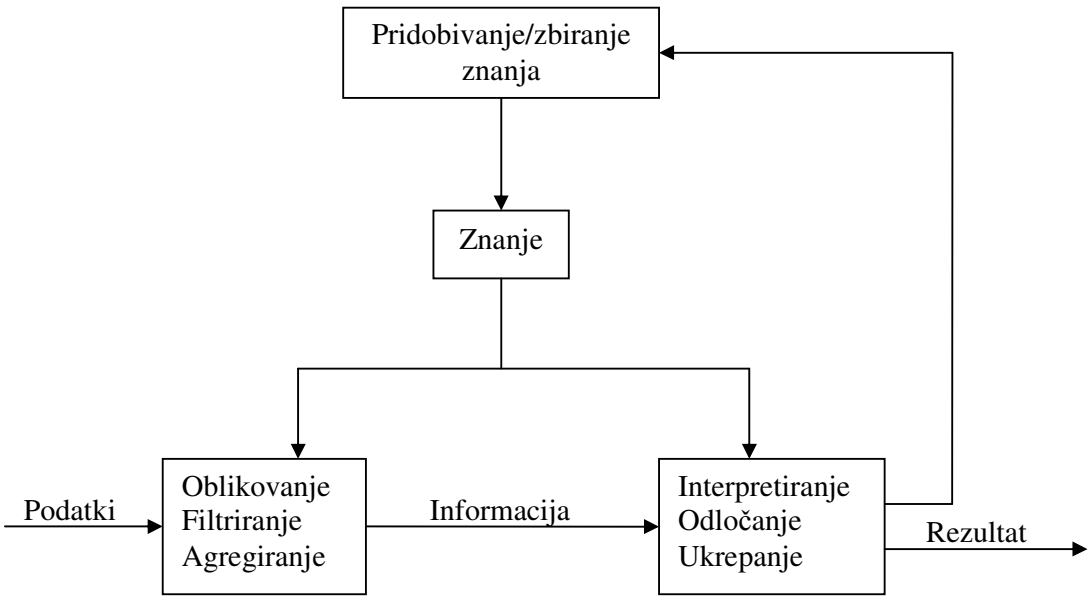
- informacija je pomen, ki ga človek pripiše podatkom s pomočjo znanih konvencij, ki so uporabljene pri njihovi predstavitvi,
- informacija so podatki, ki so obdelani tako, da so dobili pomen,
- informacija so podatki, ki so oblikovani tako, da imajo pomen in so koristni uporabnikom (Laudon & Laudon, 2000),
- informacija so podatki, katerih oblika in vsebina je primerna za določeno uporabo (Alter, 1999).

Količino informacije izražamo z merskimi enotami b, B, KB, MB, GB, TB ... Bit je osnovna merska enota za količino informacije in predstavlja odgovor na vprašanje, za katerega obstajata le dva enako verjetna odgovora. Zato sta možni vrednosti za bit le nič ali ena. Bajt je sestavljen iz osmih bitov. En bajt omogoča predstavitev ene od 256 možnih vrednosti. Večje merske enote so KB = 1024 bajtov; MB = 1024 KB; GB = 1024 MB; TB = 1024 GB; PB = 1024 TB.

V zvezi z informacijami pa seveda ne smemo zanemariti njihove kakovosti, ki je bistvena za uspešno delovanje posameznika ali organizacije. Da bi bila informacija kakovostna, mora zadoščati določenim pogojem oziroma mora imeti naslednje lastnosti:

- točnost - točna informacija je brez napak,
- popolnost - popolna informacija vsebuje vsa pomembna dejstva,
- relevantnost - informacija mora biti zanimiva in pomembna za uporabnika,
- dosegljivost - informacija je dostopna uporabnikom takrat, ko jo potrebujejo,
- preverljivost - preverljivo informacijo uporabniki lahko preverijo in se prepričajo o njeni točnosti,
- dostopnost - informacijo lahko pridobijo tisti uporabniki, ki jo potrebujejo,
- varnost - informacija je zaščitena pred nepooblaščenimi uporabniki.

Kot vidimo iz zgoraj opisanega sta podatek in informacija v tesni zvezi oziroma informacija celo izhaja iz podatkov. Da pridobimo ustrezno, kvalitetno, pravočasno, točno, popolno, varno informacijo iz zbranih podatkov, potrebujemo znanje, ki je tretja in ena pomembnejših sestavin našega procesa pridobivanja informacij. Le z znanjem lahko iz podatkov izluščimo informacijo [3] in le z njeno pravilno interpretacijo jo lahko koristno uporabimo. Povezavo med podatki, informacijo in znanjem ponazarja naslednji diagram.



Slika 1: Podatki, znanje in informacije

2 Shranjevanje podatkov

Zakaj bi ljudje shranjevali podatke in kaj bi s tem pridobili?
Če pogledamo ljudi, kot reševalce problemov oziroma možgane in njihove zmožnosti, lahko zelo hitro opazimo, da smo ustvarjeni le za reševanje problemov v sedanjosti. Nimamo ne dobre možnosti pomnjenja zgodovinskih podatkov in ne dobrega predvidevanja prihodnosti. Na splošno se lahko pri reševanju večine problemov, zelo veliko naučimo iz zgodovine in podobnih situacij iz preteklosti oziroma iz naših izkušenj, saj nas le-te lahko pripeljejo do zelo dobrih odločitev, katere sprejemamo v sedanjosti, ampak močno vplivajo na našo prihodnost. Ker nimamo možnosti dobrega pomnjenja, si pomagamo s shranjevanjem podatkov. Tako so ti dostopni, kadar jih potrebujemo, saj nikoli ne vemo, kdaj nam bodo prišli prav - bodisi pri reševanju aktualnih problemov bodisi le pri raziskovanju nekaterih tem iz radovednosti in želje po znanju.

Najbolj razširjen medij za shranjevanje podatkov je papir. S takšnim načinom shranjevanja podatkov so se ukvarjali najdlje in je zato tudi najbolj razširjeno, vendar pa še zdaleč ne edino, pred tako imenovano *podatkovno revolucijo*, kot označujemo računalniško podprto shranjevanje podatkov. Pred papirjem smo poznali še kamen, glino, les, kovino, papirus, pergament, blago, lubje in »papir« iz stržena riževih stebel.

Podatkovna revolucija

Eden izmed možnih načinov zapisovanja in shranjevanja podatkov je računalniško podprto shranjevanje - uporaba podatkovnega modela. Pri tem ne moremo mimo izraza *podatkovna zbirka* [4], ki je sklop zbirke dokumentov, medsebojnih sklicevanj na dokumente in sistema za razvrščanje, iskanje in urejanje podatkov v podatkovni bazi. Podatkovna baza je torej tudi klasična knjižnica, v vsakdanjem pogovoru pa baza pomeni računalniški sistem za hrambo podatkov.

Obstaja več definicij termina podatkovna baza [5]:

- podatkovna baza je posplošena združena zbirka podatkov skupaj z njenim opisom, ki jo uporabljamo tako, da zmore zadostiti vsem različnim potrebam uporabnikov,
- je zbirka med seboj povezanih podatkov o organiziranem delovno zaključenem sistemu, ki so namenjeni različnim uporabnikom,
- je zbirka povezanih podatkov, pri čemer so podatki dejstva, ki jih lahko zabeležimo in imajo nedvoumen pomen,
- je mehanizirana, večuporabniško formalno definirana in centralno nadzirana zbirka podatkov,
- je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni v računalniškem sistemu, dostop do njih je centraliziran in omogočen s pomočjo sistema za upravljanje s podatkovno bazo.

Prednosti:

- shranjevanje velikih količin podatkov s hitrim dostopom,
- hiter in natančen prenos podatkov,
- hitre in natančne obdelave ter preoblikovanje podatkov.

Slabosti:

- zanesljivost obstoja podatkovnih baz in z njimi povezane nesreče (uporabniške in sistemske).

3 Podatkovne baze

Podatkovne baze postajajo nepogrešljiv del kateregakoli računalniško podprtega poslovnega ali tehničnega sistema. Prvi začetki sistemov za upravljanje podatkovnih baz segajo v obdobje, v katerem so se kot poglaviten pomnilniški medij uporabljali še magnetni trakovi. Pravi razvoj sistemov za upravljanje podatkovnih baz pa se je pričel v začetku 70-ih let s predstavitvijo Coddovega relacijskega podatkovnega modela.

Štirje, pogosto imenovani "veliki" sistemi, so mrežni, relacijski, hierarhični in objektni, ki se med seboj razlikujejo predvsem po uporabniku vidnem in dostopnem podatkovnem modelu in z njim povezanimi podatkovnimi jeziki, manj pa po interni implementaciji, ki pri vseh sloni na datotečnem sistemu in iz preteklosti že dobro znanih pristopnih metodah. Mrežni sistemi so tako kraljevali vse do sredine 80-ih let, ko so jih pričeli, zahvaljujoč povečani zmogljivosti in zniževanju cen strojne opreme, prehitevati danes prevladujoči relacijski sistemi.

4 Sistem za upravljanje s podatkovno bazo (SUPB)

Je množica programov, namenjena kreiranju, vzdrževanju in nadzoru dostopa do podatkov v podatkovni bazi.

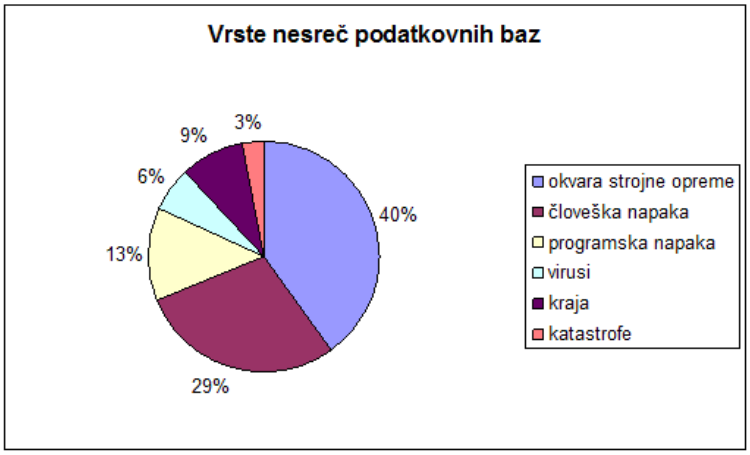
Glavne naloge sistemov za upravljanje s podatkovnimi bazami [6]:

1. **Kreiranje podatkovnih struktur**
   Kreiranje tabel in podatkovne baze omogoča modul SUPB imenovan DDL (data definition language). Omogoča kreiranje tabel, baz in spreminjanje, ter brisanje le-teh.
2. **Vzdrževanje podatkovne baze**
   DML (data manipulation language) omogoča črpanje, vstavljanje, brisanje in ažuriranje podatkov v podatkovni bazi. Kot zgled so ukazi insert, select, update in delete.
3. **Zaščita podatkov**
   SUPB mora vsebovati mehanizem za zaščito podatkov, s katerim omogoča vpogled podatkov le privilegiranim uporabnikom.
4. **Zagotavljanje integritete podatkov**
   Pred kakršnimkoli spreminjanjem (ali vstavljanjem) podatkov, se mora SUPB prepričati, da se relevantni podatki nahajajo v domeni vrednosti za dani tip. (Primer: pri vnosu v tabelo s poljem starost je vrednost 100039 primer neveljavne domene).
5. **Izvajanje transakcij**
   SUPB implementira transakcije, s katerimi omogoča izvedbo sklopa operacij na atomaren način (vse ali nič). Kot primer lahko vzamemo nakazilo denarja. Ko osebek A nakaže denar osebi B, mora mehanizem v SUPB "odvzeti" X enot denarja osebi A, ter prišteti X enot denarja osebi B. Nikakor se ne sme zgoditi, da bi se izvedla samo ena od navedenih operacij.

Seznam nekaterih podatkovnih baz: 4th Dimension, Btrieve, Centura, dBase, Fox, IBM DB2, Informix, Ingres, InterBase, MS SQL Server, MS Access, MySQL, Oracle, ter sistemov za upravljanje s podatkovnimi bazami Paradox, PostgreSQL, Progress, Sybase, Total, Ultra in Visual dBase.

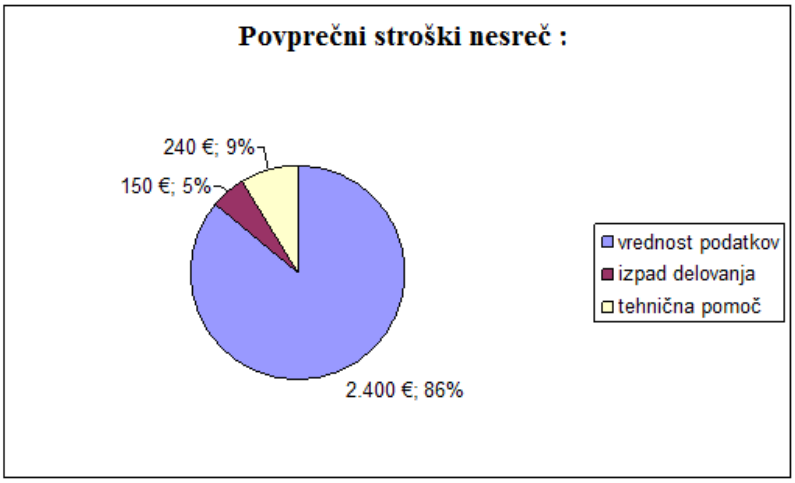
5 Nesreče podatkovnih baz

Vse prepogosto prihaja do nesreč podatkovnih baz, pri čemer lahko pride do izgube podatkov ali celotnih podatkovnih baz. Na grobo lahko ločimo šest vrst nesreč [7], in sicer uporabniške/človeške (uporabnik hote/nehote zbriše, prepiše določene podatke), sistemske/strojne (prekinitev napajanja, okvare diskov, itd.), programske (program nenadzorovano spreminja podatke), viruse, kraja podatkov, naravne katastrofe (vulkani, potresi, poplave, požari). Razmerje med njimi je prikazano na spodnjem grafu:



Slika 2: Vrste nesreč podatkovnih baz

Seveda ne moremo mimo povprečnega stroška izgubljenih podatkovnih baz [8], kjer moramo gledati obe skrajnosti obnove, in sicer popolna obnova (poceni) in popolna izguba (drago) podatkov in podatkovne baze.V povprečju nas to pripelje do naslednjega grafa:



Slika 3: Povprečni stroški nesreč

6 Kaj lahko storimo?

Najboljša rešitev bi bila popolnoma zanesljiva oprema, ki jo je nemogoče pokvariti, kar je seveda nerealno. Tudi če bi imeli opremo, ki predstavlja tehnološki višek današnjega časa, nas sama po sebi ne bi obvarovala pred samimi uporabniki, ki so zmožni marsičesa, tako namerno kot nenamerno. Zaradi vseh teh razlogov se je dandanes močno uveljavilo varnostno kopiranje, ki je zanesljivo, če mu posvečamo dovolj pozornosti in truda. Problem varnostnega kopiranja je v tem, da moramo nanj misliti v času, ko sistem deluje, vemo pa, da v tem času najmanj razmišljamo o varnosti podatkov in stabilnosti podatkovne baze. O varnostnem kopiranju začnemo ponavadi razmišljati, ko je prepozno oziroma ko je do nesreče že prišlo. Zato je potrebna disciplina in nenehno razmišljanje o zaščiti podatkovne baze in njene vsebine.

Glavni vzroki za izdelovanje varnostnih kopij so:

• zaščita pred okvaro opreme,
• zaščita v primeru nesreče (na primer potres, poplava, požar itd.),
• zaščita v primeru terorističnega napada,
• zaščita pred izgubo podatkov, zaradi napake v programski opremi,
• zaščita pred izgubo podatkov, zaradi napake uporabnika,
• poslovni razlogi za shranjevanje podatkov,
• pogodbene obveznosti do poslovnih partnerjev glede shranjevanja podatkov,
• morebitna zakonska obveznost shranjevanja podatkov.

S preučitvijo teh vzrokov in njihove pomembnosti za naš poslovni proces, lahko izdelamo ustrezno metodologijo za varnostno kopiranje.

Kratek pregled prednosti varnostnih kopij oziroma kaj nam le-te omogočajo:

• zaščito pred izgubo podatkov,
• možnost obnove (tako imenovan restore) namerno ali nenamerno spremenjenih podatkov, ter podatkovnih baz (strukture, vzpostavitvenih parametrov ...),
• stalno dosegljivost podatkov (podatke lahko obnovimo na drug sistem, po potrebi tudi na drugi lokaciji),
• možnost dodatnega arhivskega shranjevanja podatkov, ki jih sicer ne potrebujemo za tekoče delo, vendar jih moramo zaradi kakršnegakoli razloga še vedno hraniti.

Prednosti rednega izdelovanja varnostnih kopij so torej nesporne, saj so podatki na računalniških sistemih lahko zelo ranljivi. Trenutno je to eden izmed najučinkovitejših načinov varovanja podatkovnih baz.

Ugotovimo lahko, da takšen ali drugačen sistem varnostnega kopiranja potrebuje praktično vsako podjetje. Poslovanje večine podjetij je dandanes vsaj posredno odvisno od podatkov, shranjenih na računalniških pomnilniških medijih ali obdelanih s pomočjo računalnikov. Lahko gre le za podatke podpornih služb (npr. računovodstvo, kadrovska služba itd.), v vse več podjetjih pa je celotno poslovanje odvisno od računalnikov in podatkov na njih, ter tako vsaj posredno tudi od njihove varnosti.

7 Frekvenca izdelovanja varnostnih kopij

Za določitev frekvence izdelovanja varnostnih kopij je najprej potrebno določiti vse parametre, ki vplivajo na izdelovanje varnostnih kopij, predvsem:

• Ugotoviti količino, pomembnost in pogostost spreminjanja podatkov.
• Identificiranje vseh sistemov na posameznem omrežju in identificiranje vseh povezanih omrežij je pomembno za skrbništvo omrežja. Če so povezave med povezanimi omrežji dovolj hitre, je smiselno vzpostaviti centralni sistem za izdelavo varnostnih kopij. Tako imamo poenostavljen stalni fizični nadzor nad programsko opremo in enotami za shranjevanje podatkov. Pred izdelavo strategije je potrebno vedeti, kateri računalniki bodo vključeni v sistem, kakšna bo predvidena količina podatkov in kakšen bo tip podatkov (npr. podatkovna baza, datotečni strežnik, internetni strežnik…).
• Potrebno je oceniti potrebe po obnovi podatkov in ugotoviti, kolikšen je najdaljši še dopusten čas od uporabnikove zahteve do obnove podatkov. Za orientacijo lahko pogledamo rezultate raziskave, opravljene leta 2005 v ZDA s strani podjetja Terian [10], v kateri so svoje stranke spraševali, kakšen je najdaljši čas, v katerem morajo biti podatki obnovljeni, ne da bi to vplivalo na preživetje podjetja:

  • 40 % - največ 72 ur,
  • 21 % - največ 48 ur,
  • 15 % - največ 24 ur,
  • 8 % - največ 8 ur,
  • 9 % - največ 4 ure,
  • 3 % - največ 1 uro,
  • 4 % - manj kot 1 uro.

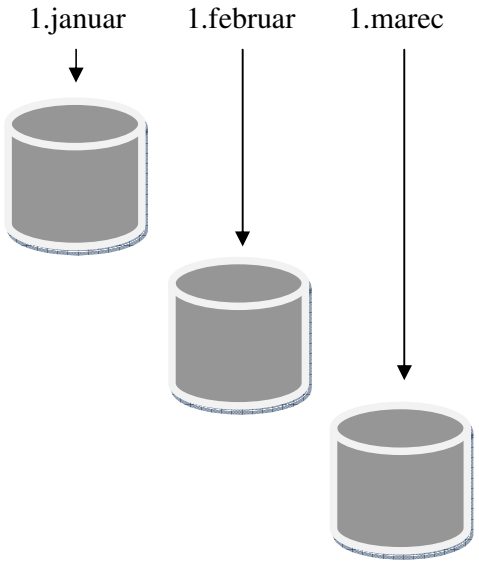
• Tudi v okviru podjetja lahko obstajajo razlike med potrebami posameznih organizacijskih enot, saj navadno nimajo enakih (časovnih, količinskih …) potreb po varnostnem kopiranju. Temu se je potrebno prilagoditi oziroma najti zadovoljiv kompromis.
• Nujno je ugotoviti potrebe po dosegljivosti podatkov. Za kopiranje nekaterih podatkov je potrebno, da ima sistem za izdelovanje varnostnih kopij v času kopiranja edini dostop do teh podatkov oziroma podatkovne baze - ta način izdelave varnostnih kopij imenujemo "cold backup". Baza v času izdelave kopije ni dostopna aplikaciji oziroma uporabniku, kar v večini primerov ne predstavlja problema. Pri aplikacijah, ki morajo biti neprekinjeno dosegljive, pa je to lahko resna težava - v tem primeru izdelujemo "hot backup", ki pa ne odraža konsistentnega stanja baze v nekem trenutku, saj je le-ta še zmeraj v uporabi in se spreminja s strani aplikacije oziroma uporabnika.
• Izvesti je potrebno lociranje vseh morebitnih obstoječih enot za izdelavo varnostnih kopij. Če so bile kakšne enote že v uporabi, jih je smiselno identificirati, pregledati in če je možno, uporabiti. Nesmiselno je še uporabno opremo zavreči in namesto nje kupovati novo. V okviru tega je potrebno ugotoviti, na kakšen način bomo v primeru potrebe obnovili že shranjene podatke po dosedanjem sistemu.
• Ugotoviti največjo še dopustno količino izgubljenih podatkov, saj sta od tega odvisna pogostost izdelovanja varnostnih kopij, ter nastavitve arhiviranja dnevnikov sprememb.
• Varnostno kopiranje dodatno obremeni sisteme, predvsem diske in vhodno/izhodne enote, saj se pretok podatkov lahko poveča preko limita posameznih komponent, kar lahko povzroči njihovo preobremenjenost ali odpoved.

8 Vrste varnostnih kopij

O vrstah varnostnih kopij [12] je nekaj malega omenjenega že na prejšnjih straneh, kjer omenjamo "cold" in "hot backup", delimo pa jih tudi glede na izbrano tehniko varnostnega kopiranja, in sicer imamo:

- popolno varnostno kopiranje celotne podatkovne baze,
- inkrementalno varnostno kopiranje, ki ga dalje delimo na diferencialno in komulativno.

8.1 Popolno varnostno kopiranje celotne podatkovne baze



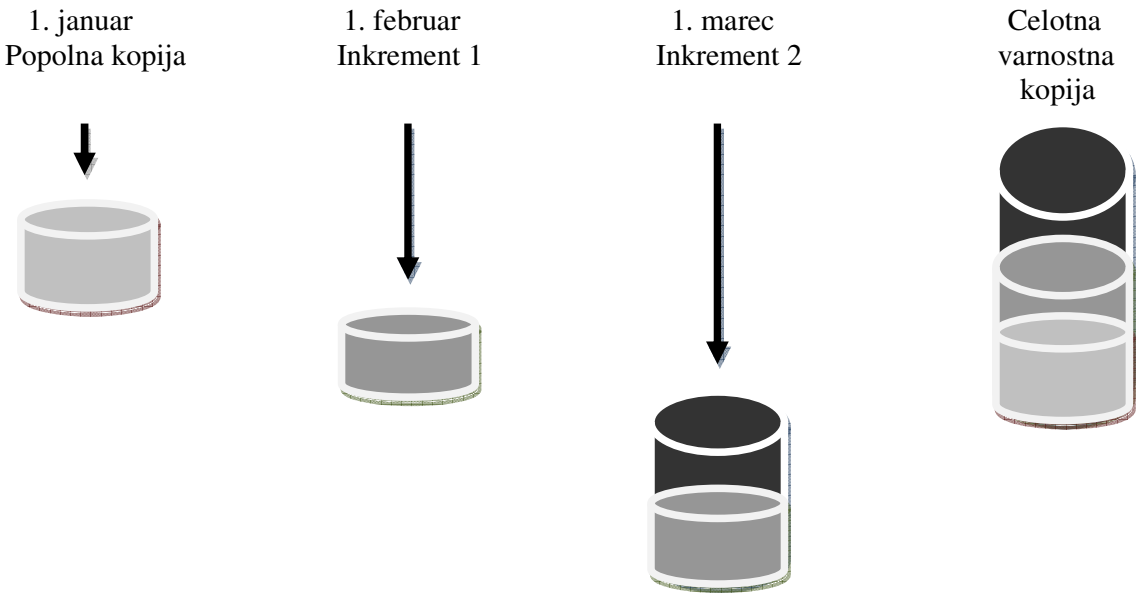
Pri popolnem varnostnem kopiranju izdelujemo kopije celotne podatkovne baze. Dobra stran tega je, da so vsi podatki fizično locirani skupaj, na enem ali več zaporednih podatkovnih medijih. Tako je morebitna obnova podatkov hitra in enostavna, ter možna za različna časovna obdobja, odvisno od tega, kdaj so bile narejene. Potrebujemo le popolno varnostno kopijo, narejeno v času, za katerega želimo obnoviti podatke. Slaba stran pa je, da vedno kopiramo tudi podatke, ki se ne spreminjajo in imamo tako več kopij istih podatkov. Na ta način se po nepotrebnem poveča poraba podatkovnih medijev in čas, potreben za zapisovanje podatkov.

8.2 Inkrementalno varnostno kopiranje

Pri tej tehniki gre za izdelavo ene popolne kopije, nakar pri vsaki naslednji kopiramo le spremenjene stvari. Poznamo dva načina, in sicer komulativno in diferencialno varnostno kopiranje:

a.) Inkrementalno - komulativno varnostno kopiranje

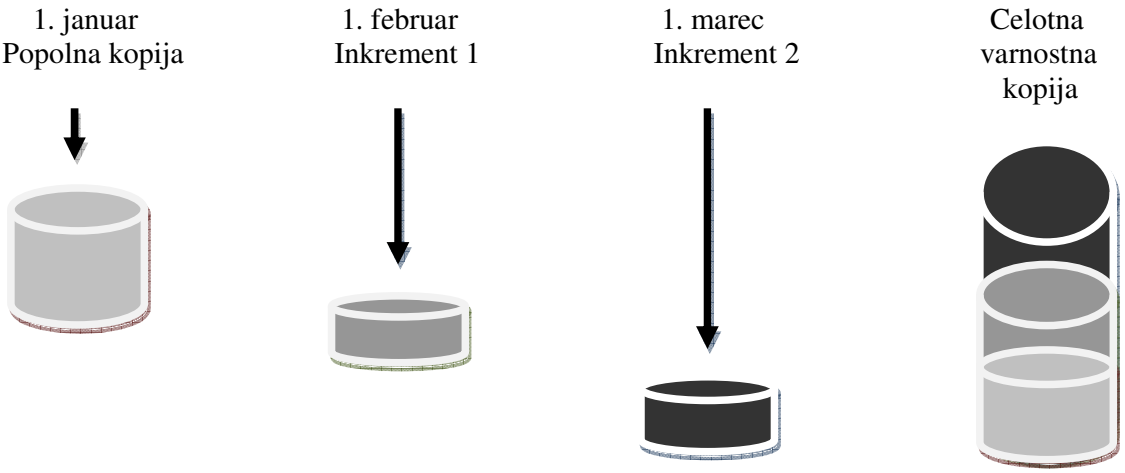
Ta zapisuje podatke, ki so se spremenili od zadnjega popolnega zapisa. Podatki, ki se v času med dvema popolnima zapisoma ne spremenijo, se tudi ne zapisujejo. Če pa se spremenijo le enkrat, se bodo zapisali vsakič, ko se bo izvajal zapis, vse do naslednjega popolnega zapisa. Količina zapisanih podatkov se povečuje, saj je praviloma podatkov, ki so se spremenili ali dodali od zadnjega popolnega zapisa, vedno več. Velika prednost pred popolnim zapisom je v manjši porabi prostora na medijih, ter krajšem času izdelave varnostne kopije. Slabost pa je daljši čas in več dela za obnovo podatkov. Za obnovo podatkov iz določenega obdobja potrebujemo zadnji popolni zapis in zadnji kumulativni inkrementalni zapis pred tem obdobjem. Iz popolnega zapisa obnovimo podatke, ki se kasneje niso več spremenili, iz inkrementalnega zapisa pa podatke, ki so se spremenili po popolnem zapisu.



1. januarja naredimo popolno varnostno kopijo, februarja pa naredimo inkrement 1, ki vsebuje vse spremembe od popolne kopije dalje. 1. marca naredimo inkrement 2, ki vsebuje vse spremembe od popolne kopije dalje.

b.) Inkrementalno - diferencialno varnostno kopiranje

Diferencialni inkrementalni zapis zapisuje le tiste podatke, ki so se spremenili od zadnjega zapisa, ne glede na to ali je bil ta popolni ali inkrementalni. S tem se, glede na kumulativni inkrementalni zapis, še dodatno zmanjša poraba prostora na medijih ter čas za izdelavo varnostne kopije. Nasprotno pa se še podaljša čas in poveča količina dela za morebitno obnovo podatkov. Za obnovo podatkov iz določenega obdobja potrebujemo zadnji popolni zapis in vse diferencialne inkrementalne zapise pred želenim obdobjem. Iz popolnega zapisa obnovimo podatke, ki se kasneje niso več spremenili, iz inkrementalnih zapisov pa po vrsti podatke, ki so se spremenili po vsakem prejšnjem zapisu.



1. januarja naredimo popolno varnostno kopijo, februarja pa naredimo inkrement 1, ki vsebuje vse spremembe od popolne kopije dalje. 1. marca naredimo inkrement 2, ki vsebuje vse spremembe od inkrementa 1 dalje.

9 Izvozno/Uvozna funkcionalnost Oracla "Oracle Import/Export Utilities"

Oraclova "export" (exp) ali izvozna in "import" (imp) ali uvozna funkcionalnost je obravnavana v našem sklopu kot dopolnilo varnostnemu kopiranju in restavriranju podatkovnih baz, saj nam omogoča varnost na logičnem nivoju strukture podatkovne baze in njenih podatkov. Nudi nam možnost izvoza oziroma uvoza posameznih logičnih komponent znotraj podatkovne baze in ne obravnava fizičnih datotek, ki sestavljajo podatkovno bazo. Z logičnimi komponentami označujemo elemente baze kot so tabele, indeksi, sprožilci in sheme. Ena izmed zelo velikih omejitev je ta, da izvoz v eni od verzij Oracla ni primeren za uvoz v nižje verzije Oracla, obratno iz nižjih verzij v višje pa to ne predstavlja problema.

Ta funkcionalnost se uporablja predvsem za sledeče naloge [11]:

- varnostno kopiranje in restavriranje manjših podatkovnih baz (do velikosti 50 GB), za večje ni primerno in se priporoča obnovitveni upravitelj,
- prenos podatkov med različnimi platformami (recimo Solaris in Windows),
- prenos shem, podatkov, uporabnikov in podatkovnih področij med Oracle podatkovnimi bazami,
- itd.

Uporaba je zelo preprosta, saj že pri namestitvi Oracle programske opreme le-ta poskrbi za definiranje globalnih spremenljivk imp in exp, ki nam skozi ukazno vrstico omogočata uporabo funkcionalnosti izvoza oziroma uvoza. Posebej je potrebno poudariti, da je potrebno imeti za izvoz posameznega logičnega elementa baze pravice izvoza tega elementa, kar nam le dodatno povečuje varnost naših podatkov in strukture pravic znotraj podatkovne baze.

Primer klicev exp in imp iz ukazne vrstice:

```
$exp username/password parameter=(value1, value2 ... valuen);
$imp username/password@instance as sysdba;
```

Seveda pa nam razna orodja, kot je na primer Toad, omogočajo uporabo raznih grafičnih predstavitev in čarovnikov, za izvoz in uvoz, ki bistveno olajšajo naše delo in izničijo potrebo po znanju ukazov ter njihovi sintaksi. Primer izpisa poročila Toad o izvozu sheme Bostjan lahko vidimo na desni strani, kjer je lepo opisan del postopka izvoza.

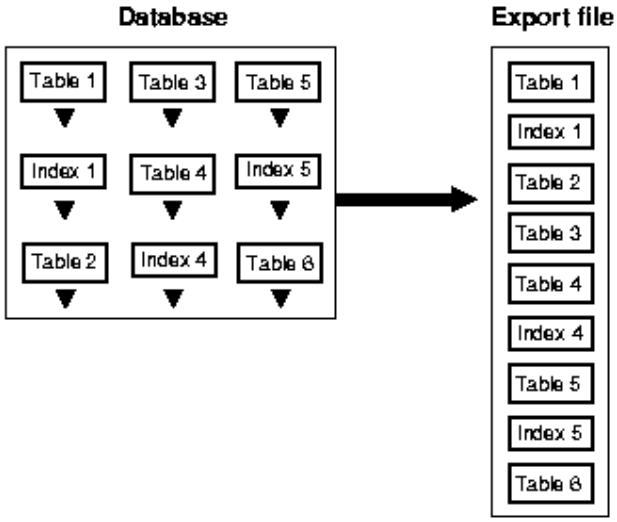
```
Export: Release 10.2.0.3.0 - Production on Cet Nov 5 09:56:13 2009

Copyright (c) 1982, 2005, Oracle.  All rights reserved.


Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
Export done in EE8MSWIN1250 character set and AL16UTF16 NCHAR character set

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user BOSTJAN
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user BOSTJAN
About to export BOSTJAN's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export BOSTJAN's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
```

Slika 4: Primer izpisa poročila orodja Toad

9.1 Izvozna funkcionalnost Oracla "Oracle Export Utility"

Pri uporabi izvoza Oracle baze se objekti, kot so tabele, pogledi, sprožilci ... izvozijo skupaj z njihovimi pripadajočimi stvarmi kot so indeksi, komentarji, pravicami nad objekti, seveda, če obstajajo. Vse skupaj se zapiše v eno datoteko s končnico dmp, ki jo znajo uvoziti le Oracle podatkovne baze. V tej izhodni datoteki so:

- definicije tipov,
- definicije tabel,
- podatki tabel,
- indeksi,
- tuji ključi,
- pogledi,
- procedure,
- sprožilci,
- sekvence.



Slika 5: Simbolična slika podatkovne baze in njena predstavitev v izvozni datoteki

Izvozimo lahko celotno podatkovno bazo, posamezne tabele, podatkovna področja, podatke tabel in uporabnike.

V vseh Oracle podatkovnih bazah moramo pred prvim izvozom poskrbeti še za kreiranje potrebnih izvoznih pogledov in podatkovnih slovarjev, ki so ključnega pomena za uspeh izvoza oziroma brez njih le-ta ni možen. To skripto "*catalog.sql*" priskrbi Oracle sam in jo lahko običajno najdemo v $oracle_home/rdbms/admin direktoriju.

Primer ukaza izvoza sheme sola, na bazi orcl, z izhodno datoteko sola.dmp, ter datoteko z dnevnikom sola.log:

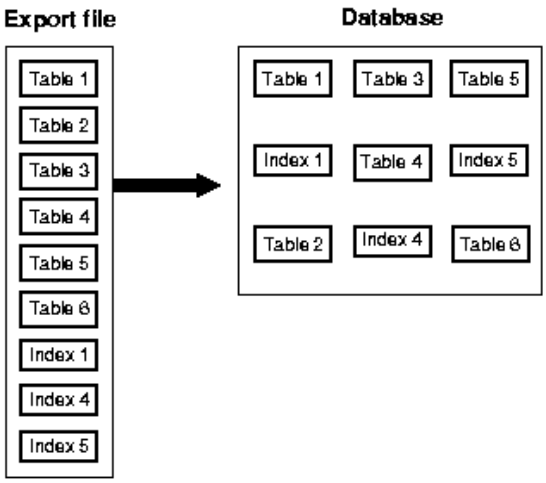
$exp system@orcl file=sola.dmp log=sola.log owner=('sola');

Po uspešno skreirani izhodni datoteki, nam le-ta predstavlja dobro varnostno kopijo podatkovne baze, ki pa je seveda ne smemo hraniti na istem disku, kjer se nahaja naša podatkovna baza. Pri procesu razvoja jo lahko uporabimo za varnost struktur elementov, saj tukaj podatki niso tako pomembni, v produkcijskem okolju pa le za podatkovno varnost, saj se tu ne, oziroma je zelo odsvetovano manipuliranje s strukturo podatkovne baze. Ta izhodna datoteka je zelo občutljiva za prenose po medmrežju, kjer se uporablja ftp protokol, saj se binarne datoteke pri prenosih s tem protokolom, delujočim v ascii načinu prenosa, poškodujejo in postanejo, vsaj v našem primeru, neuporabne. Potrebno je zagotoviti binaren način prenosa in s tem pravilno obravnavanje znaka na koncu vrstice znotraj datoteke.

9.2 Uvozna funkcionalnost Oracla "Oracle Import Utility"

Našo izhodno datoteko lahko uvozimo v novo oziroma drugo podatkovno bazo z ukazom imp, kjer je potrebno izbrati le kaj, kam in od kod uvažamo. Te izbire so potrebne pri uvozu logičnih elementov, ne pa pri uvozu celotne podatkovne baze. Vprašanje od kod se nanaša na uporabnika znotraj izvožene podatkovne baze, vprašanje kam pa na uporabnika nove podatkovne baze, ki mora seveda obstajati. Kaj uvažamo pa je seveda mišljeno za logične elemente uvoza. Spodaj je opisan enostaven primer uvoza tabele oziroma potek dogodkov pri uvozu, iz razloga, da se pobližje spoznamo z delovanjem te uvozne funkcionalnosti.

Osnovni scenarij poteka dogodkov pri uvozu tabel je sledeč. Zgenerira se nova tabela, vnesejo se podatki, nad njo se zgenerira indeks, nakar se postavijo sprožilci in omogočijo tuji ključi. Kot posebnost še omenimo, da uvoz podatkov v isto tabelo povzroči, seveda v primeru uspeha, samo dodajanje novih podatkovnih zapisov vanjo in ne prepisuje že obstoječih zapisov.



Slika 6: Simbolična slika vhodne datoteke in s pomočjo nje, kreirane podatkovne baze

Primer ukaza uvoza tabele predmet iz sheme bostjan (iz dmp datoteke) v shemo bostjan, na bazi, kjer želimo kreirati tabelo:

$imp bostjan/pwr file=sola.dmp fromuser= bostjan tables=(predmet);

10 Razveljavitev sprememb "Flashback"

Razveljavitev sprememb funkcionalnost je bila prvotno predstavljena v Oracle9i in vse od tedaj pridobiva na pomenu v vsaki novi verziji Oracla. V svojem bistvu je to mehanizem, ki razveljavlja spremembe, tako namerne kot nenamere. V procesu razvoja se pogosto zgodi, da zavijemo s prave poti razvoja in bi radi prišli nazaj do neke točke, kjer smo še bili prepričani o pravi smeri našega razvoja. Sedaj sta poti nazaj dve, in sicer lahko popravimo spremembe od te točke dalje, ali pa spremembe preprosto razveljavimo in nadaljujemo od te točke dalje. Temu pri Oraclu pravijo "točka v času restavriranja" oziroma "point in time recovery".

Seveda je potrebno nastaviti potrebne atribute in pridobiti pravice razveljavitve sprememb, da lahko s to funkcionalnstjo operiramo. Po vseh nastavitvah pa lahko naredimo nekaj tako pomembnega kot je vrnitev zbrisane tabele, ali pa samo razveljavitev sprememb, ki jih je povzročila zadnja transakcija.

Za razumevanje koncepta razveljavitev sprememb v Oracle podatkovni bazi je potrebno razložiti dve stvari, in sicer enolično številko spremembe imanovano SCN "System change number" in pa koš "Recycle bin". Ob vsaki spremembi baze se tej spremembi na bazi dodeli enolična številka spremembe. Tako je prehod na prejšnja stanja zelo enostaven, le poznati moramo številko spremembe do katere se želimo vrniti. Pri izbrisu tabele ti tudi ta številka ne pomaga, tu pa pride v uporabo koš, saj izbrisan objekt ne sprosti prostora takoj, ampak se shrani v koš, iz katerega ga lahko kasneje obnovimo. V primeru, da želimo prostor sprostiti takoj, uporabimo rezervirano besedo "purge", ki izbriše objekt iz koša za vedno.

Preden pa se zares začnemo ukvarjati z razveljavljanjem sprememb, je potrebno omeniti še, da ne deluje nad "system" podatkovnim področjem. Primer: če se prijavimo z sys uporabnikom, ki uporablja "system" podatkovno področje kot privzeto, kreiramo tabelo, jo zbrišemo in želimo razveljatviti brisanje, to ni mogoče.

Razveljavitev sprememb razdelimo na pet področij:

a.) razveljavitev sprememb na nivoju celotne podatkovne baze "Flashback database";
lahko si pomagamo z enolično številko sprememb, s časom, na katerega želimo vrniti podatkovno bazo ter z obnovitveno točko, ki jo moramo seveda narediti. Obnovitvene točke se običajno naredijo ob kakšnih večjih spremembah baze, kot je na primer nova verzija, če je le-ta verzionirana, ali katera koli druga večja sprememba, ki bi predstavljala nevarnost naši podatkovni bazi.

Primer:
sql> flashback database to scn 19513917;

b.) razveljavitev sprememb izbrisa "Flashback drop";
s tem lahko razveljavimo efekt izbrisa tabele. To naredimo s sintakso "flashback tabele *table_name* to before drop;", omejuje nas to, da mora biti tabela v košu. Posebej je potrebno poudariti, da se ob obnovitvi tabele ne obnovijo elementi, ki so nanjo vezani (indeksi ...).

Primer:
sql> flashback table *test* to before drop;

c.) razveljavitev verzij zapisov "Flashback versions query";
ta nam omogoča pregled nekega zapisa skozi čas; uporaben je na dva načina, s časovnim intervalom in intervalom enolične številke spremembe.

Primer:

```
sql>1 select id
     2 from test versions between scn 583074 and 583261;
```

d.) razveljavitev transakcije "Flashback transaction query";
imamo možnost enostavne rekonstrukcije "sql" izjave, ki je povzročila spremembe na bazi, seveda pa tudi njeno nasprotno "sql"' izjavo, ki jo lahko uporabimo za razveljavo vseh povzročenih sprememb na podatkovni bazi.

Primer :

```
sql> 1  select *
       2  from   flashback_transaction_query
       3  where logon_user = izbrani_uporabnik;
```

Znotraj te tabele je stolpec UNDO_SQL, ki ga lahko zaženemo za razveljavo sprememb.

e.) razveljavitev sprememb tabele "Flashback table";
s to funkcijo lahko razveljavimo spremembe nad določeno tabelo (le podatkovne), pomagamo si z enolično številko sprememb ali s časom. Ta funkcija ni podprta za "sys"' uporabnika.

Primer :

```
sql>flashback table test to scn 592081;
```

Najenostavnejši način vključitve in namestitve vseh potrebnih atributov za razveljavljanje sprememb, je že med kreiranjem same podatkovne baze. Za samo delovanje mora podatkovna baza delovati v načinu arhiviranja dnevnikov sprememb, kar pomeni, da se vse spremembe hranijo na disk in ne prepisujejo v vmesnem pomnilniku podatkovne baze. Potrebno je nastaviti čas, v katerem je še možno razveljaviti spremembe, mesto na disku, kamor se bodo spremembe hranile, ter velikost tega mesta oziroma datoteke.

```
sql> shutdown immediate;
sql> startup mount;
sql> alter database archivelog;
sql> alter system set db_flashback_retention_target=4320;
sql> alter system set db_recovery_file_dest_size=536870912;
sql> alter system set db_recovery_file_dest='/u02/fra';
sql> alter database flashback on;
sql> alter database open;
```

Zgoraj je opisan postopek priprave podatkovne baze za uporabo funkcionalnosti razveljavljenja sprememb. Postopek opisuje delo z sqlplus urejevalnikom, ki bazo zapre, jo zažene, postavi v način shranjevanja sprememb, definira 4320 minut (72 ur) za možnost razveljavitve, 512 MB za velikost direktorija, kamor se shranjujejo spremembe, ter lokacijo direktorija, omogoči razveljavitve, ter odpre podatkovno bazo.

11 Obnovitveni upravitelj RMAN "Recovery manager"

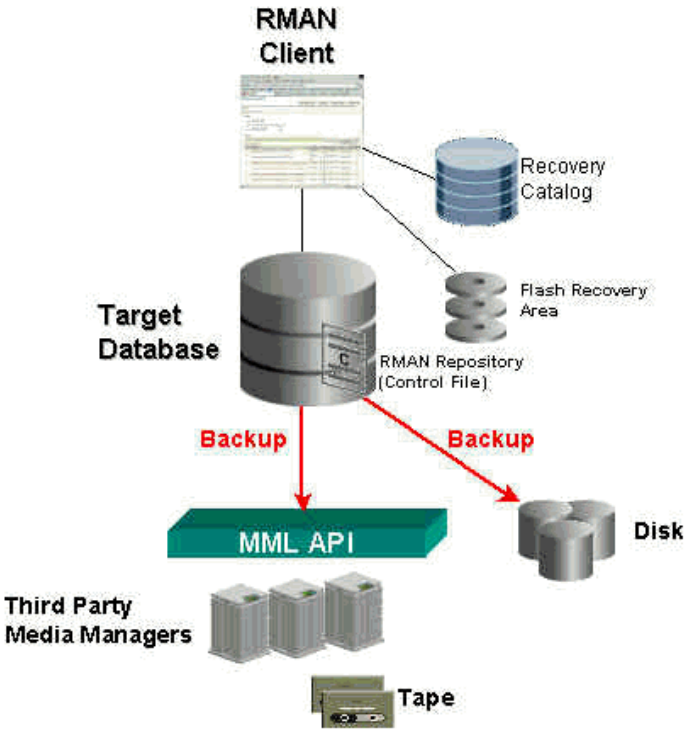
Je eden najpomembnejših načinov varnostnega kopiranja in restavriranja podatkovne baze. Njegovo delovanje opredeljujemo kot fizično, saj deluje nad fizičnimi datotekami, ki sestavljajo podatkovno bazo. Nudi nam pregled nad vso zgodovino varnostnih kopij, večprocesno varnostno kopiranje, varnostno kopiranje (v celoti ali inkrementalno), konsistentne in nekonsistentne varnostne kopije, varnostno kopiranje na več različnih mest in medijev (disk, trak), ter zagotavlja učinkovite tehnike za odkrivanje pokvarjenih podatkov.

Tehnike obnove zagotavljajo obnavljanje podatkov na nivoju posameznega bloka, obnavljanje podatkov (v celoti ali inkrementalno), obnavljanje do določenega časa (point in time recovery), podatkovne zbirke v pripravljenosti (standby databases) in obnavljanje na nivoju transakcij.

Te tehnike zagotavljajo obnavljanje podatkov, vendar lahko celoten proces traja dolgo, posebej v primeru obnavljanja s trakov. S preventivnim nadzorom nad podatki lahko prihranimo veliko časa, truda in stresa, ki jih povzročijo okvara ali izguba podatkov oziroma začasno neuporabna zbirka podatkov.

Najpomembnejši fizični elementi podatkovne baze, ki jih je možno z obnovitvenim upraviteljem varnostno kopirati so:

a.) control files,
b.) archive logs,
c.) datafiles,
d.) server parameter files,
e.) backup pieces.



Slika 7: Klient obnovitvenega upravitelja

11.1 Področje za obnovitvene datoteke "Flash recovery area"

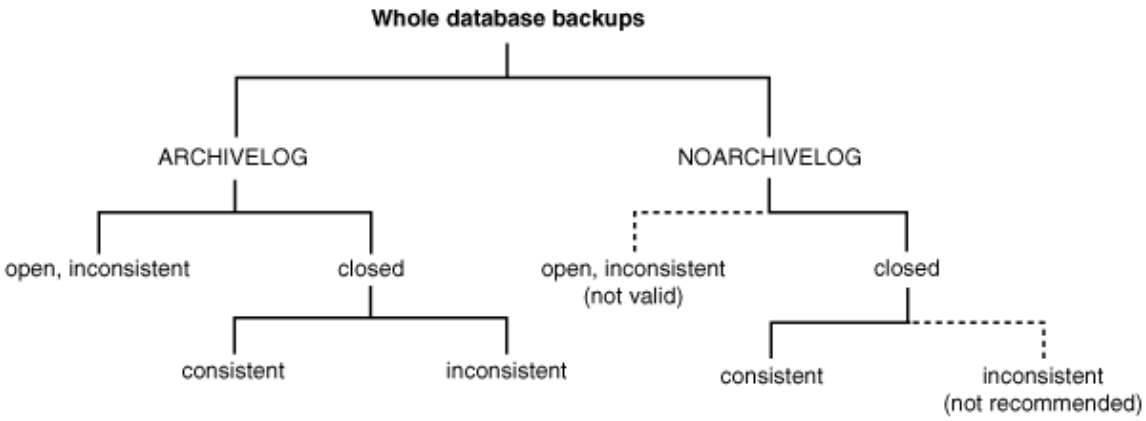
Mesto na disku, v katerega Oracle shanjuje varnostne kopije in katerega tudi sam upravlja. Tako nas obvešča ob posebnih dogodkih, kot je na primer premalo prostora za novo varnostno kopijo. Znotraj tega direktorija se lahko nahajajo arhivski dnevniki, dnevniki sprememb, varnostne kopije kontrolnih datotek, varnostne kopije podatkovne baze, seveda če smo tako nastavili potrebne atribute, kar je izjemno priporočljivo, saj nam sedaj ni potrebno skrbeti za ta prostor.

11.2 Obnovitveni katalog "Recovery catalog"

Posebna shema Oracla, v katero shranjuje meta podatke. Uporaba tega je opcijska, saj lahko te podatke shranjuje tudi v kontrolne datoteke. Moramo pa vedeti, da je potrebno v primeru uporabe te shemo le-to varnostno kopirati kot vse ostale.

11.3 Hladno in vroče varnostno kopiranje z obnovitvenim upraviteljem"Hot and cold backup with rman"

Obnovitveni upravitelj omogoča oboje, seveda pa je razlika v načinu delovanja baze, ki mora biti v primeru dela vročih varnostnih kopij v načinu shranjevanja arhivskih dnevnikov. V bistvu to pomeni, da baze ne zapiramo, ampak jo varnostno kopiramo med njenim delovanjem. To pa ima za posledico nekonsistentne varnostne kopije, saj je v trenutku varnostnega kopiranja podatkovna baza delujoča in spremembe na njej stalne. Ob obnovitvi podatkovne baze s te kopije, le-ta ni nujno enaka tisti ob kopiranju, saj so bile lahko med kopiranjem narejene nekatere nepotrjene spremembe, katerih pa nismo zajeli v varnostni kopiji.
Pri mrzlih varnostnih kopijah nimamo problema s konsistentnostjo, saj podatkovno bazo zapremo in upoštevamo vse spremembe do zaprtja. Tako točno vemo, kaj smo zajeli v sami varnostni kopiji. Ta način varnostnega kopiranja ne potrebuje podatkovne baze v načinu delovanja, kjer se shranjujejo arhivski dnevniki.

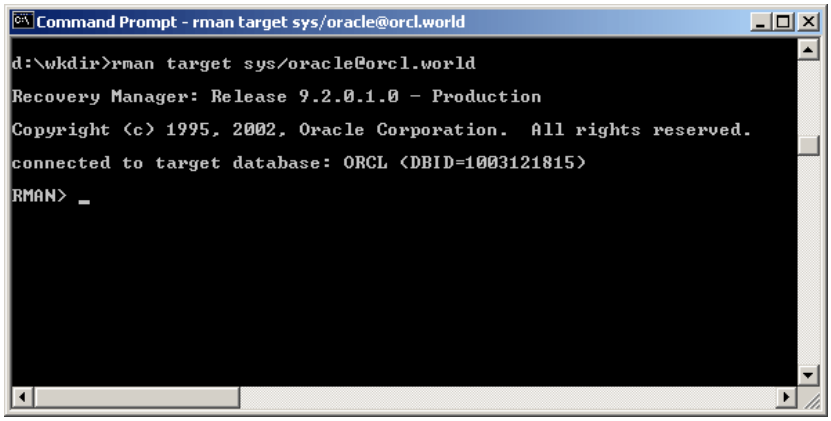


Slika 8: Podroben prikaz delitve konsistentnih in nekonsistentnih varnostnih kopij celotne podatkovne baze

11.4 Primeri uporabe obnovitvenega upravitelja:

a.) Prijava v klienta obnovitvenega upravitelja iz ukazne vrstice operacijskega sistema:

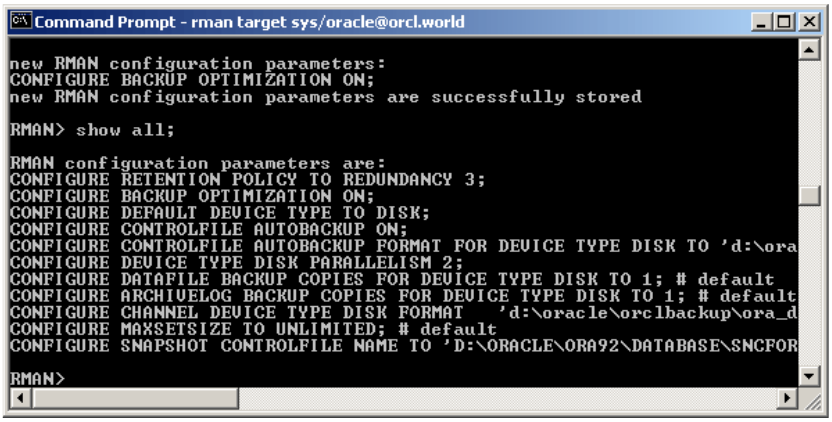
rman target sys/oracle@orcl.world;



Slika 9: Obnovitveni upravitelj skozi ukazno vrstico

b.) Nastavitve obnovitvenega upravitelja

Za pregled trenutnih nastavitev uporabimo ukaz "show all", kar nam prikaže pomembnejše atribute. Njihov posamezen pomen je do potankosti opisan v Oraclovi dokumentaciji.



Slika 10: Prikaz osnovnih atributov obnovitvenega upravitelja

c.) Varnostno kopiranje podatkovne baze

Po nastavitvi vseh atributov se povežemo na željeno podatkovno bazo in začnemo z varnostnim kopiranjem, kar izvedemo s pomočjo ukaza "backup database;", lahko kopiramo samo podatkovna področja s pomočjo ukaza "backup tablespace users;", ali posamezne fizične datoteke s pomočjo ukaza "backup datafile 5;".

d.) Inkrementalno varnostno kopiranje

Imamo možnost dela z različnimi inkrementalnimi opcijami kot sta komulativni in diferencialni inkrementi. V veliko pomoč pri tem nam je Oraclova zmožnost zapisovanja spremenjenih blokov od zadnje varnostne kopije. Vklop te možnosti pospeši inkrementalno varnostno kopiranje, saj obnovitveni upravitelj pozna spremenjene bloke v vsakem trenutku od zadnje varnostne kopije dalje, katere v naslednjem inkrementu varnostno kopira.

Možnost sledenja spremenjenih blokov vključimo z ukazom:

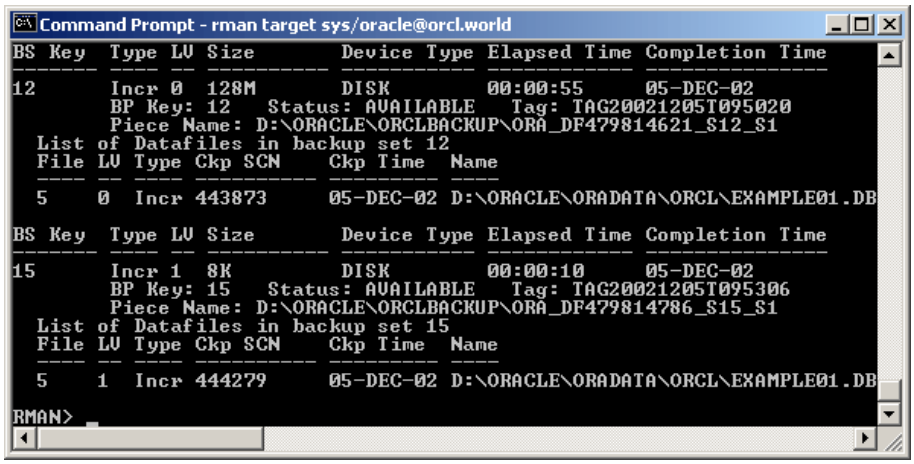
sql> alter database enable block change tracking [using file '<filename>'];

Inkrementalno lahko varnostno kopiramo celotno podatkovno bazo "backup incremental level 0 database;", kjer stopnja nič pomeni varnostno kopijo celotne podatkovne baze, kasneje za inkremente pa uporabljamo stopnjo ena.

e.) Administrativni ukazi

Po izvedenih varnostnih kopijah imamo številne administrativne ukaze za pregled kopij, kot so: report need backup,
list backup,
delete obsolete ...

list backup of datafile 5;



Slika 11: Prikaz varnostnih kopij izbrane datoteke

f.) Obnovitveni ukazi

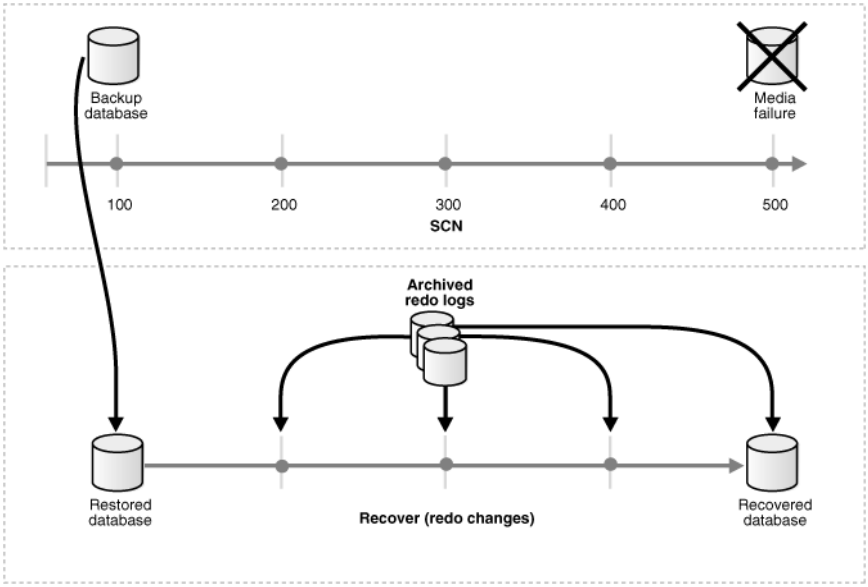
Dva glavna ukaza za obnovitev podatkovne baze iz varnostne kopije in potrjevanje sprememb od zadnje varnostne kopije dalje iz dnevnikov sprememb sta:

"restore database;",

ki prekopira pokvarjene ali izgubljene fizične datoteke iz varnostne kopije, ter

"recover database;",

ki poskrbi za realizacijo zabeleženih sprememb v samem dnevniku sprememb. Pri tem lahko s pomočjo enolične številke sprememb ali s časom določimo, do katere spremembe želimo imeti obnovljeno bazo.

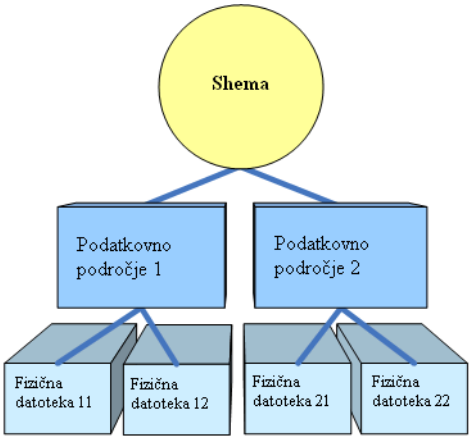


Slika 12: Ponazoritev "restore" in "recovery" ukazov, ter arhivskih dnevnikov sprememb

g.) Obnovitev podatkovnega področja

Po mojih izkušnjah je to najuporabnejši in najzahtevnejši postopek obnovitve, ki nam omogoča obnovitev posameznih podatkovnih področij med samim delovanjem podatkovne baze. V ta namen že pri sami izgradnji podatkovne baze pazljivo načrtujemo posamezna podatkovna področja, saj primer večine podatkovne baze v enem samem podatkovnem področju in večine podatkovnega področja v eni sami fizični datoteki na našem operacijskem sistemu, predstavlja veliko nevarnost. Podatkovno bazo načrtujemo tako, da posamezne sklope oziroma uporabnike vežemo na svoja podatkovna področja, le-te pa razdelimo med več fizičnih datotek, kar nam omogoča večji nadzor in varnost, saj v primeru nesreče enega, drugi delujejo neprekinjeno.

Podatkovna področja so logične enote, ki so sestavljene iz enega ali več fizičnih datotek na disku. Povezavo med podatkovnim področjem in fizičnimi datotekami na samem disku prikazuje desna skica.



Slika 13: Povezava med shemo in fizičnimi datotekami znotraj Oracle podatkovnih baz

Osnovni skript klienta obnovitvenega upravitelja za obnovitev podatkovnega področja:

```
run{
sql'alter tablespace <podatkovno_podrocje> offline';
restore tablespace <podatkovno_podrocje>;
recover tablespace <podatkovno_podrocje>
 until scn <enolicno_stevilo_do_katerega_obnavljamo_bazo>
auxiliary destination='\u01\<destinacija_pomozne_baze>';
sql'alter tablespace <podatkovno_podrocje> online';
}
```

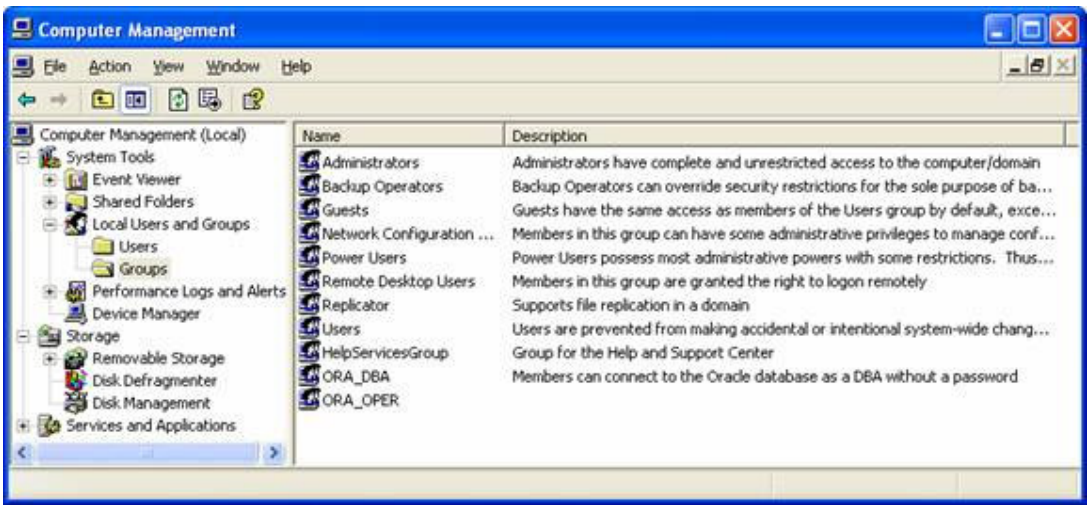
Opis dogajanja po zagnani skripti:

1. <podatkovno_podrocje> gre v stanje "offline", kar ga naredi neuporabnega oziroma nedosegljivega za uporabnike,
2. ukaz "restore" prekopira fizične datoteke, iz katerih je sestavljeno podatkovno področje,
3. ukaz "recover" realizira spremembe nastale od zadnje varnostne kopije do izbrane enolične številke sprememb, k temu spada še ukaz "auxiliary destination", ki določa prostor na našem disku, kjer se postavi nova instanca podatkovne baze in na tej se dejansko realizirajo spremembe do enolične številke sprememb, nakar se prepiše podatkovno področje na našo operacijsko podatkovno bazo,
4. po uspešnem obnavljanju podatkovnega področja je potrebno podatkovno področje omogočiti uporabnikom, za kar poskrbi zadnji ukaz.

Uspešno končana obnovitev sama za sabo pobriše avtomatsko generirano instanco, v primeru neuspešne obnovitve podatkovnega področja, pa je avtomatsko instanco potrebno pobrisati, kar storimo z ukazom:

```
exec dbms_backup_restore.manageauxinstance ('TSPITR',1);
```

Preden pa bo obnova mogoča, moramo poskrbeti še za določene avtentifikacijske probleme. S strani operacijskega sistema moramo dodati uporabnika, ki se lahko povezuje na podatkovno bazo brez uporabniškega imena in gesla, zaradi avtomatsko generirane instance.

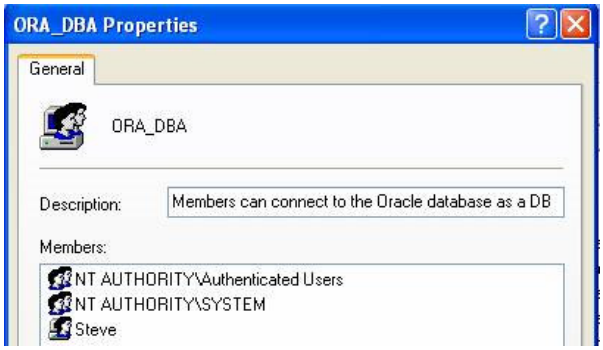


Slika 14: Windows - ov nadzornik

Izberemo skupino ORA_DBA,
ter dodamo uporabnika.

Slika 15: Izbira skupine ORA_DBA

"Authenticated Users",
kot novi uporabnik v skupini
ORA_DBA.

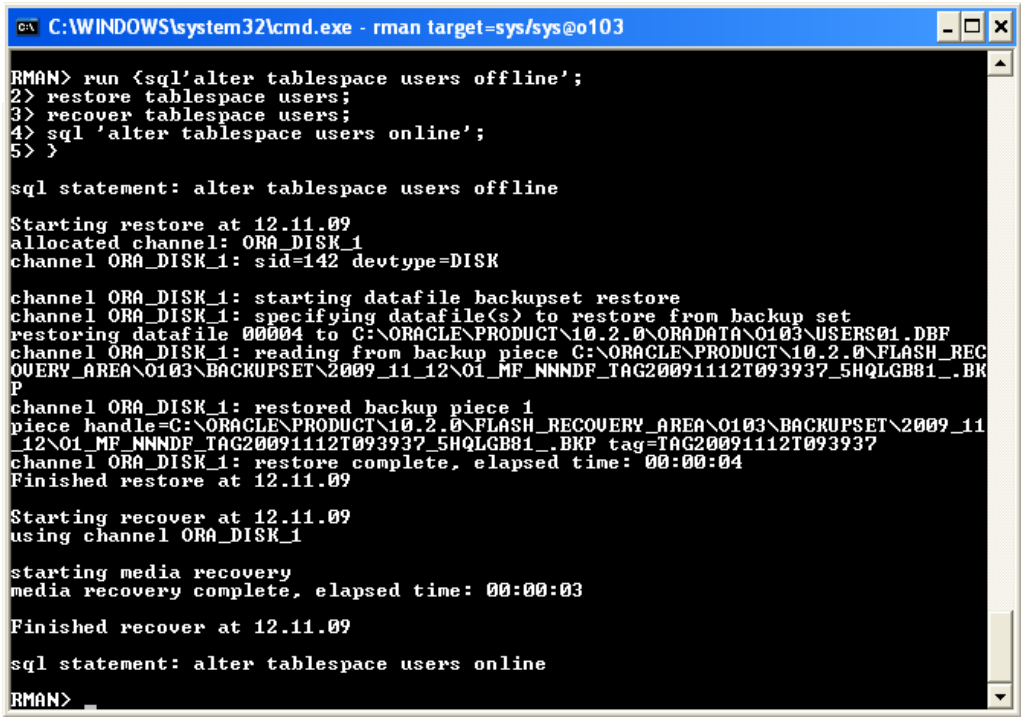


Slika 16: Dodan novi uporabnik v skupino ORA_DBA

S strani podatkovne baze pa moramo poskrbeti za avtentifikacijski protokol, ki ga najdemo znotraj datoteke sqlnet.ora. Dodani ukaz, ki bo omogočil obnovitev podatkovnega področja in enkratno avtentifikacijo uporabnika na Windows strežniku in podatkovni bazi, katero le-ta vsebuje, izgleda tako:

sqlnet.authentication_services = (nts[1])

[1] Windows NT native authentication



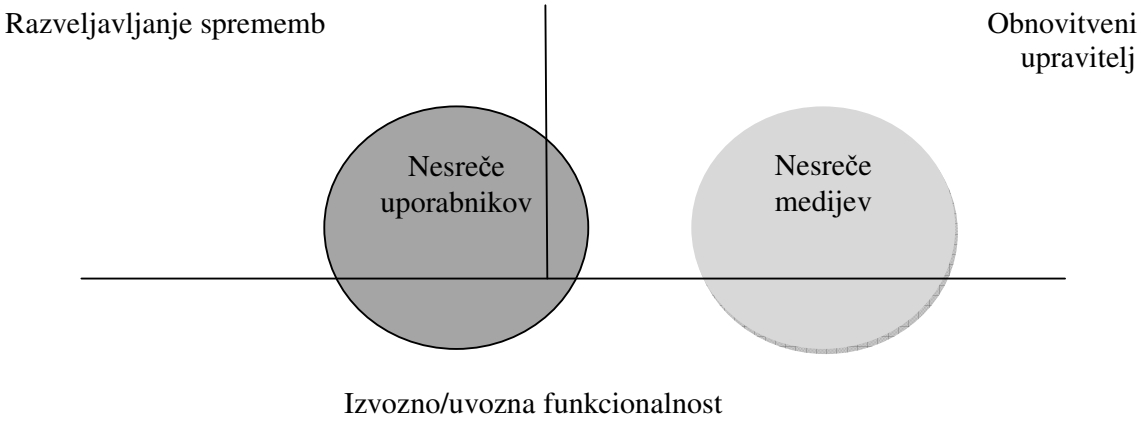
Slika 17: Prikaz izvajanja osnovne obnovitve podatkovnega področja iz ukazne vrstice klienta obnovitvenega upravitelja

12 Kako se orodja dopolnjujejo med sabo in kdaj so primerna za uporabo?

Orodja se dopolnjujejo med sabo glede na njihov način delovanja, saj obnovitveni upravitelj deluje nad fizičnimi datotekami in sestavnimi deli podatkovne baze, medtem, ko uvozno/izvozna funkcionalnost deluje nad logičnimi. Že ta delitev nam prikaže, da je za medijske okvare primernejši obnovitveni upravitelj, pri čemer pa se za napake uporabnikov zanašamo predvsem na možnost razveljavitve sprememb in pa uvozno/izvozne možnosti.

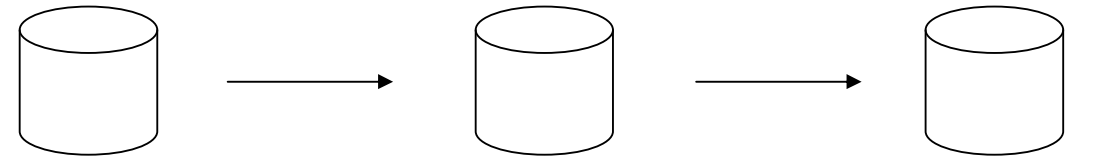
Vsako izmed orodij pa ima tako svoje dobre kot slabe lastnosti. Začnimo pri razveljavljanju sprememb, ki je odlično pri brisanju tabel ali zapisov znotraj tabel, postane pa neuporaben, kakor se spremeni definicija tega izbranega elementa. Uvozno/izvozna funkcionalnost nam lepo izvaža zaokrožene logične celote, tako je v primeru izgube oziroma podrtja kakšne strukture znotraj baze, ta več kot dovolj za njeno obnovitev. Moramo pa seveda pomisliti nanjo in izvoziti željene objekte pravočasno, ter paziti na kompatibilnost verzij Oracle programske opreme. Na koncu je tu še obnovitveni upravitelj, ki celo preverja pravilnost posameznih blokov v fizičnih datotekah. Nudi varnostno kopiranje na več alternativnih lokacij, ter na več različnih medijev. Delo z njim pa je izjemno nevarno, saj je zelo močno orodje in hitro lahko zaustavimo celotno bazo in izgubimo ali prepišemo še kakšno dodatno stvar pri obnovi.

Sedaj poznamo pozitivne lastnosti vsakega orodja, od nas pa je odvisno za kakšno kombinacijo orodij se bomo odločili oziroma kakšno vlogo v naši varnostni strategiji bo imelo posamezno orodje. Izbira strategije in zakaj je eno orodje primernejše za razvojno podatkovno bazo drugo pa za produkcijsko, bomo opisali v naslednjem poglavju.



Slika 18: Primerjava orodij oziroma primernost le-teh za posamezno vrsto nesreče.

13 Varnostne strategije podatkovnih baz v samem sistemu razvoja



Slika 19: Sistem razvoja podatkovnih baz

13.1 Produkcijska podatkovna baza

Produkcijska podatkovna baza je bila izbrana zaradi njene pomembnosti, saj je to podatkovna baza, ki se dejansko uporablja s strani uporabnikov sistema. Ima nekatere specifične zahteve, kot so neprekinjeno delovanje, ni tolerance izgube podatkov ...

Njena varnostna strategija bi bila sledeča:

- varnostno kopiranje celotne podatkovne baze mesečno,
- komulativne varnostne kopije tedensko,
- diferencialne varnostne kopije dnevno,
- vse varnostne kopije so vroče "hot", za kar mora podatkovna baza delovati v načinu shranjevanja dnevnikov sprememb,
- pred vsako novo verzijo celotna varnostna kopija, ter celoten izvoz podatkovne baze,
- vključena funkcionalnost razveljavljanja sprememb "flashback".

Za hitrejše inkrementalne varnostne kopije bi vklopili funkcionalnost sledenja sprememb oziroma "*block change tracking*", ki doprinese 1-10 % performančne izgube, vendar je tu strojna oprema najzmoglivejša in s tem nimamo problemov. Na tej podatkovni bazi se naj ne bi spreminjala struktura elementov, razen ob novih verzijah, kjer pa poskrbimo za varnostne kopije in izvoze logičnih elementov. Pri varnostnem kopiranju seveda ne smemo imeti kopij na istem disku, kjer je sama podatkovna baza, ampak jih izdelujemo na alternativni lokaciji.

Struktura kot sama, nam naj ne bi predstavljala težav, problemi so v podatkih, ki jih uporabniki ali aplikacija pokvarijo namerno ali nenamerno. Za podatke pa smo že ugotovili, da je zelo dobra razveljavitev sprememb "flashback" funkcionalnost in pa tabela "flashback_transaction_query", s tem popravimo podatke oziroma odvrtimo spremembe nazaj, kar pa ne reši problema. Po ugotovljeni in popravljeni napaki, moramo zagotoviti, da do tega ne bo več prišlo. Za napake medijev poskrbimo z obnovitvenim upraviteljem, ta nam omogoča obnovitev vsakega podatkovnega področja posebej - le tako lahko zadostimo zahtevi po neprekinjenem delovanju baze. Izgubljena ali pokvarjena datoteka se prvotno skopira iz varnostne kopije, nato se realizirajo spremembe do prekinitve delovanja oziroma točke ugotovitve nepravilnosti te datoteke.

Dandanes pa razpolagamo že s tehnologijo, ki lahko, v primeru kakšne okvare, sama preklopi med diski, kar nam varnost še poveča. Za varnost naše baze ni dovolj le dobra varnostna strategija Oraclovih orodij, ampak tudi kombinacija teh s strojno opremo, na kateri naša podakovna baza deluje.

13.2 Testna produkcijska podatkovna baza

Testna produkcijska podatkovna baza je bila izbrana zaradi njene uvrščenosti med razvojno in produkcijsko podatkovno bazo. Njena glavna zahteva je nenehno delovanje, vendar s toleranco izgube podatkov nekaj ur oziroma dni.

Njena varnostna strategija bi bila sledeča:

- varnostno kopiranje celotne podatkovne baze mesečno,
- komulativne varnostne kopije tedensko,
- vse varnostne kopije so vroče "hot", za kar mora podatkovna baza delovati v načinu shranjevanja dnevnikov sprememb,
- pred vsako novo verzijo celotna varnostna kopija, ter celoten izvoz podatkovne baze.

Na tej podatkovni bazi je tudi odsvetovano spreminjanje strukture elementov, razen ob novih verzijah, kjer pa poskrbimo za varnostne kopije in izvoze logičnih elementov. Pri varnostnem kopiranju ne smemo imeti kopij na istem disku, kjer je sama podatkovna baza, ampak jih izdelujemo na alternativni lokaciji.

Na to podatkovno bazo se prenašajo stari produkcijski podatki in testira funkcionalnosti našega razvojnega sistem z realnimi podatki. Za nesreče uporabnikov bi poskrbeli s pomočjo uvozno/izvozne funkcionalnosti, medtem ko kakšni pokvarjeni podatki niso takšen problem. Potrebno je ugotoviti, zakaj je do napake v podatkih prišlo in seveda popraviti aplikacijo. Tu se vse testira preko aplikacije in se ne zapisuje podatkov več direktno v samo podatkovno bazo. Za napake medijev poskrbimo ponovno z obnovitvenim upraviteljem, s katerim lahko zagotovimo nenehno delovanje podatkovne baze med samo obnovitvijo.

13.3 Razvojna podatkovna baza

Razvojna podatkovna baza je baza, na kateri poteka ves razvoj, posledično se njena struktura spreminja najpogosteje. Nima zahteve po neprekinjenem delovanju, kar nam omogoča izdelavo kosistentnih kopij. Podatki v njej so zgolj testne narave in kot takšni nepomembni, tudi nezanesljivi.

Njena varnostna strategija bi bila sledeča:

- varnostno kopiranje celotne podatkovne baze mesečno,
- pred vsako novo verzijo celotna varnostna kopija, ter celoten izvoz podatkovne baze.

Pri varnostnem kopiranju ne smemo imeti kopij na istem disku, kjer je sama podatkovna baza, ampak jih izdelujemo na alternativni lokaciji. Postopek obnovitve je podoben kot pri testni produkciji, pri uporabniških in aplikacijskih nesrečah se zanašamo na uvozno/izvozno funkcionalnost, ter razveljavljanje sprememb. Medijske nesreče bi reševali z zaustavitvijo baze in obnavljanjem podatkovnega področja ali celotne podatkovne baze.

14 Praktični primer:

V nadaljevanju bomo predstavili primer postavitve strukture Oracle programske opreme, načine varnostnega kopiranja, scenarije nesreč in ustrezne akcije za rešitev podatkovne baze po teh nesrečah.

Za postavitev strukture se moramo bolje spoznati s sestavnimi deli Oracla. Najpomembnejši deli podatkovne baze [12, 13], katere smo že omenili v poglavju o obnovitvenem upravitelju, so:

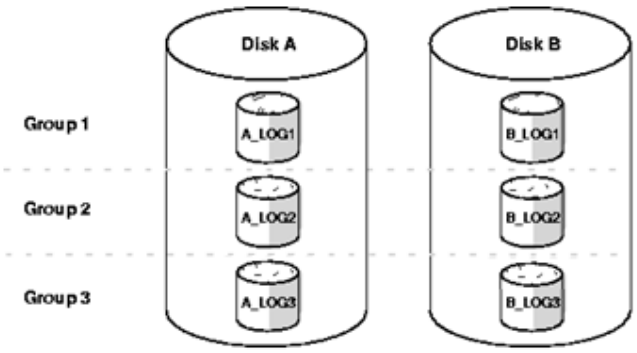
a.) control files:
- vsebujejo strukturo baze,
- Oracle zgenerira dva oziroma tri identične datoteke, iz katerih razbere strukturo baze,
- močno priporočljiva je ločitev teh datotek na dva fizično ločena podatkovna diska.

b.) archive logs:
- arhivirani dnevniki sprememb.

c.) datafiles:
- datoteke opredeljene kot fizične shrambene enote,
- v njih se dejansko shranjujejo podatki tabel podatkovne baze.

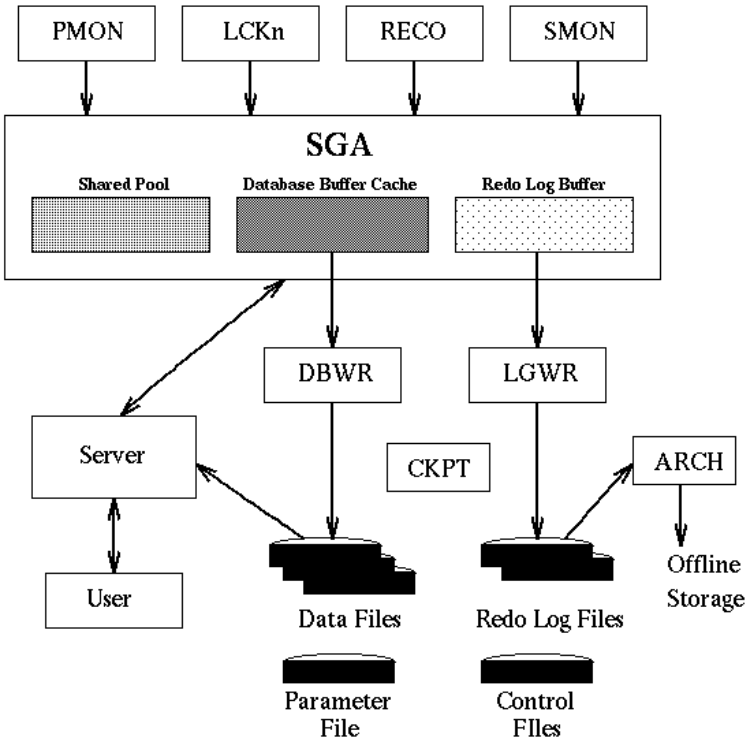
d.) server parameter files:
- vzpostavitveni parametri baze.

e.) redo logs:
- dnevniki, ki vsebujejo spremembe narejene nad podatkovno bazo,
- uporabljajo se ciklično, kar pomeni, da se po napolnitvi začno prepisovati,
- če se pred prepisovanjem arhivirajo, podatkovna baza deluje v načinu arhiviranja dnevnikov sprememb,
- vsaka skupina naj vsebuje najmanj dva člana v katera se vzporedno zapisujejo spremembe,
- obstajati morata najmanj dve skupini dnevnikov sprememb.



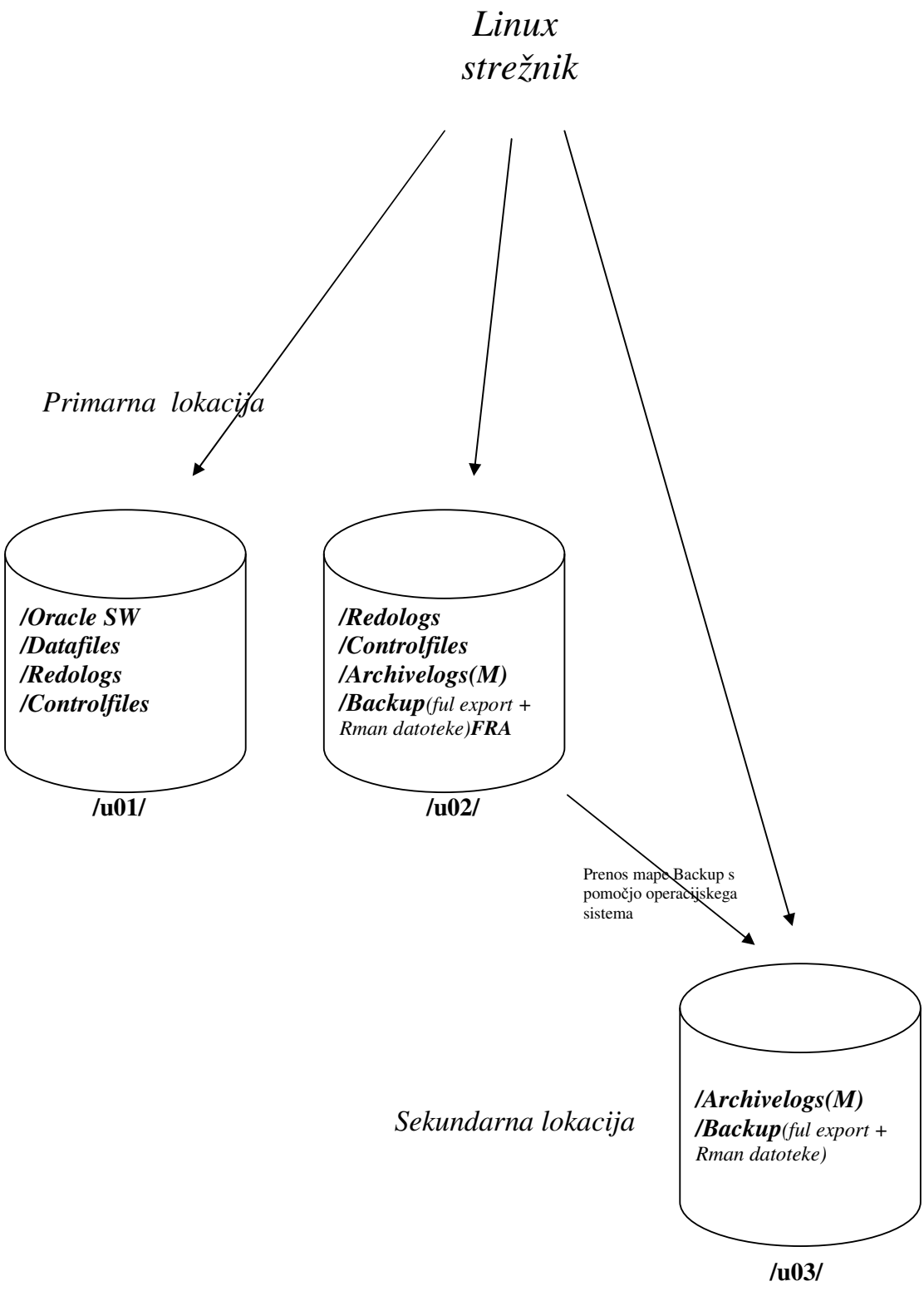
Slika 20: Skica dnevnikov sprememb

Umeščenost posameznih delov v sami strukturi podatkovne baze Oracle nakazuje naslednja skica:



Slika 21: Struktura Oracle podatkovne baze

Naš sistem teče na Linux (Ret Hat Enterprise 5) strežniku, kjer je nameščena naša Oracle programska oprema na disku z oznako /u01/ in še nadaljno porazdeljena na dva podatkovna diska /u02/ in /u03/. Podatkovni disk /u01/ vsebuje mape datafiles, redologs in controlfiles, disk /u02/ mape redologs, archivelogs, controlfiles in backup. Sama imena map ponazarjajo njihovo vsebino. Podatkovni disk z oznako /u03/ je naša sekundarna lokacija, ki se fizično nahaja na drugi strojni opremi. S sistemom je povezan preko internetnega ali intranetnega komunikacijskega omrežja. Vsebuje mapi poimenovani archivelogs in backup.



Slika 22: Skica postavitve strukture Oracle programske opreme

14.1 Postavitev strukture

Namestitvama Linux strežnika in Oracle programske opreme se ne bom posebej posvečal, več pozornosti bom namenil konfuguraciji Oracle okolja in prerazporeditvi datotek.

Začnemo z konfiguracijo Oracle okolja in izberemo željeno bazo na sistemu:

$. oraenv
ORACLE_SID = [] ? <*Oracle system identifier*>

Nakar vstopimo v sqlplus urejevalnik z ukazom sqlplus/nolog znotraj ukazne vrstice. Z uporabnikom, ki ima pravice sysdba, se povežemo na ustrezno bazo in začnemo z razporejanjem datotek. Prva in najpomembnejša je premestitev ene kontrolne datoteke na podatkovni disk /u02/, saj Oracle namesti vse na en podatkovni disk:

sql> select * from v$controlfile;

#/u01/datafiles /control01.ctl
#/u01/oracle/flash_recovery_area/control02.ctl

Poženemo spodnji ukaz za konfiguracijo novih poti:

sql> alter system set control_files='/u01/controlfiles/control01.ctl','/u02/controlfiles/control02.ctl' scope=spfile;

Podatkovno bazo ustavimo:

sql> shutdown immediate;

V ukazni vrstici se pomaknemo na podatkovni disk /u02/, ter z ukazom:

$ mkdir controlfiles, kreiramo direktorij z imenom controlfiles. Za tem je potrebno skopirati datoteko control02.ctl na lokacijo /u02/controlfiles:

$ cp control02.ctl /u02/controlfiles

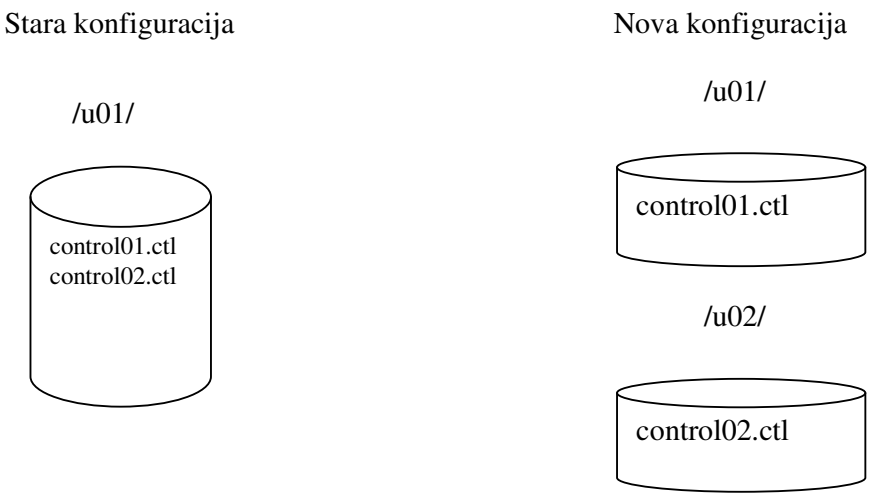
Podatkovno bazo ponovno zaženemo in odpremo uporabnikom:

sql> startup mount;
sql> alter database open;

Pregled destinacij kontrolnih datotek:

sql> select * from v$controlfile;

#/u01/controlfiles/control01.ctl
#/u02/controlfiles/control02.ctl



Slika 23: Stara in nova konfiguracija kontrolnih datotek

Naslednji so na vrsti dnevniki sprememb. Po namestitvi Oracle programske opreme nam le ta izdela tri skupine, vsako z dvema članoma. Izbrišemo lahko le skupine, ki so neaktivne oziroma v statusu "inactive". Njihov status izvemo s pomočjo poizvedbe spuščene v sqlplus urejevalniku:

```
sql> select * from v$log;
```

Primer, v katerem je skupina 2 v stanju 'inactive' :

```
sql> alter database drop logfile group 2;
```

Kreiramo nov dnevnik sprememb, na fizično ločen podatkovni disk:

```
sql> alter database add logfile group 2 ('/u01/redologs/redo_log02a.log','/u02/redologs/redo_l
og02b.log') size 200m;
```

Status posameznih skupin spreminjamo s spodnjima ukazoma, ter ponovimo za vse skupine, ki jih želimo kreirati:

```
sql> alter system switch logfile;
sql> alter system checkpoint global;
```

Pregledamo velikost in člane posameznih skupin:

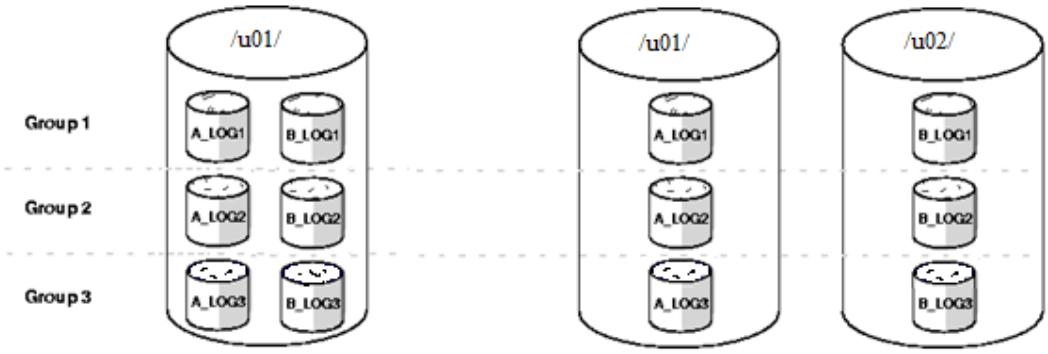
```
sql> select * from v$log;

# GROUP#    THREAD#  SEQUENCE#     BYTES  BLOCKSIZE    MEMBERS ARC STATUS
# ---------- ---------- ---------- ---------- ---------- ---------- ---
#      1          1          0  209715200        512          2   YES UNUSED
#      2          1         23  209715200        512          2   NO INACTIVE
#      3          1         24  209715200        512          2   NO CURRENT
```

Pregledamo lokacije posameznih članov znotraj skupin:

```
sql> select * from v$logfile;

#   GROUP#  STATUS  LOCATION
# ----------------------------------------------------------------
#   3           ONLINE  /u01/redologs/redo_log03a.log
#   2           ONLINE  /u01/redologs/redo_log02a.log
#   1           ONLINE  /u01/redologs/redo_log01a.log
#   2           ONLINE  /u02/redologs/redo_log02b.log
#   3           ONLINE  /u02/redologs/redo_log03b.log
#   1           ONLINE  /u02/redologs/redo_log01b.log
```

V zgornjem primeru smo kreirali tri skupine z dvema članoma, kjer je vsak član velikosti 200 MB. Iz lokacije njihovega nahajanja je lepo razvidno, da so razdeljeni med dva fizično ločena diska.

Stara namestitev dnevnikov sprememb: Nova namestitev dnevnikov sprememb:



Slika 24: Skica prerazporeditve dnevnikov sprememb

Direktno iz dnevnikov sprememb sledijo nastavitve za arhiviranje dnevnikov sprememb, ker mora podatkovna baza delovati v načinu arhiviranja dnevnikov sprememb. Podatkovna baza preide v tak način delovanja z zaporedjem ukazov:

```
sql> shutdown;
sql> startup mount;
sql> alter database archivelog;
sql> alter database open;
```

Oracle je zmožen arhivirati dnevnike sprememb na največ deset različnih mest naenkrat, kar nam določajo spremenljivke log_archive_dest_n, kjer je n celo število od ena do deset. Preglejmo, kaj Oracle naredi ob sami namestitvi:

```
sql>  1  select  dest_name, status, binding, destination
      2  from   v$archive_dest
      3  where destination is not null;

#DEST_NAME                STATUS     BINDING         DESTINATION
#-------------------------------------------------------------------------
#LOG_ARCHIVE_DEST_1  VALID    MANDATORY  USE_DB_RECOVERY_FILE_DEST
```

```
sql> show parameter db_recovery_file_dest;

#NAME                           TYPE        VALUE
#-------------------------------- ----------- ------------------------------
#db_recovery_file_dest           string      /u01/oracle/flash_recovery_area
```

Iz zgornjih poizvedb izvemo, da je lokacija arhiviranja dnevnikov nastavljena na disk /u01/. Lokacija je obvezna, kar pomeni, če arhiviranje ni možno, se baza ustavi in delo na njej ni mogoče, dokler te napake ne odpravimo. V našem načrtu je ustvariti dve fizično ločeni lokaciji arhiviranja, za kar si na diskih /u02/ in /u03/ kreiramo datoteke z pomočjo ukaza:

$ mkdir archivelogs, kateri kreira datoteko z imenom archivelogs.

Nastavimo vrednosti destinacijskih parametrov:

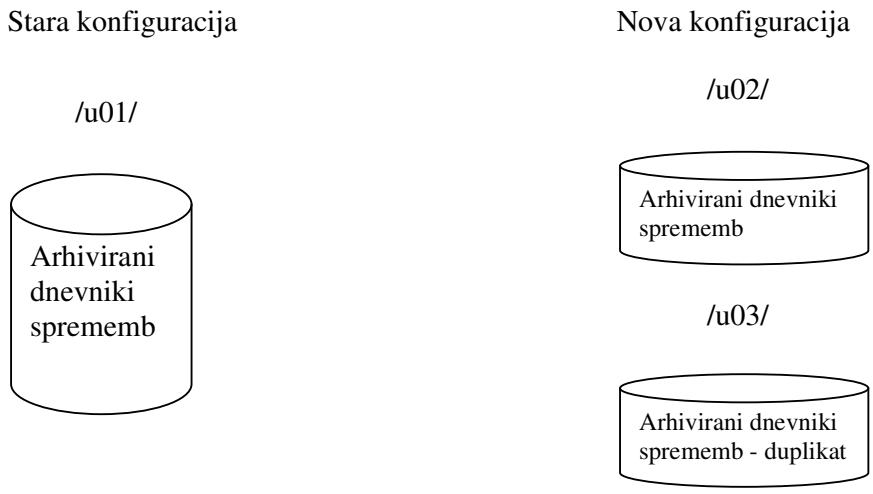
```
sql> alter system set log_archive_dest_1='location=/u02/archivelogs mandatory';

sql> alter system set log_archive_dest_2='location=/u03/archivelogs reopen=5 max_failure=3
mandatory';
```

Ukaz za izpisan format arhiviranih dnevnikov:

```
sql> alter system set log_archive_fomat='%t_TEST_%s_%r.arc' scope=spfile;
```

Pregledamo novo razporeditev arhiviranih dnevnikov:

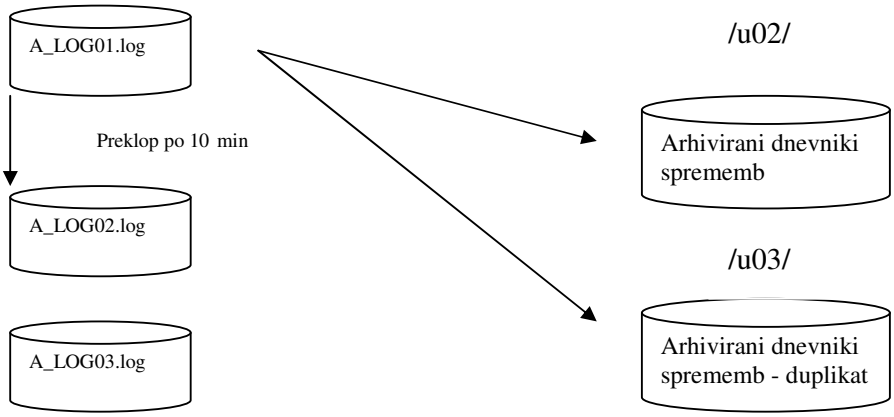
```
sql>  1 select dest_name,status,binding,destination
      2 from v$archive_dest
      3 where destination is not null;

#DEST_NAME                 STATUS    BINDING         DESTINATION
#-------------------------------------------------------------------------------------
#LOG_ARCHIVE_DEST_1  VALID    MANDATORY  /u02/archivelogs
#LOG_ARCHIVE_DEST_2  VALID    MANDATORY  /u03/archivelogs
```



Slika 25: Nova razporeditev arhiviranih dnevnikov sprememb

Uredimo še dodatne nastavitve podatkovne baze, katere nam omogočajo nadzor nad relacijo dnevnikov in arhiviranih dnevnikov. Parameter archive_lag_target nam omogoča določitev minut, v katerih se bo aktualni dnevnik arhiviral. Tu se skriva lastnost, ki jo pogosto opazimo na raznih zahtevah za postavitev sistema, in sicer določitev dovoljene izgube podatkov v primeru nesreče strežnika. Primer zahteve 10 minut oziroma 600 sekund:

```
sql> alter system set archive_lag_target=600 scope=spfile;
```



Slika 26: Preklop med dnevniki sprememb, ter njihovo podvojeno arhiviranje

14.2 Klasifikacija nesreč podatkovnih diskov naše namestitve

14.2.1 Nesreča diska /u01/

→ Podatkovna baza se ustavi

Najobčutljivejša stvar tega podatkovnega diska je nameščena Oracle programska oprema sama, takoj za njo so kontrolne datoteke, katerih pomembnosti ni potrebno posebej poudarjati, ter dnevniki sprememb, ki ne povzročajo posebnih težav, če le niso zadnji člani svojih skupin. Razred zase pa so podatkovne datoteke, ki so ključnega pomena za delovanje podatkovne baze in jih tudi varnostno kopiramo na podatkovni disk /u02/. Sistematično se lotimo pregleda postopkov obnove, po odpovedi vsake stvari posebej, kombinaciji teh in podatkovnega diska v celoti.

14.2.1.1 Podatkovna baza se ustavi zaradi nedostopnosti kontrolne datoteke

Pregled alert.log datoteke, ki se nahaja na $oracle_home\admin\<ime_podatkovne_baze>\ bdump nam pokaže sledeče:

```
\\...
\\ORA-00202: Message 202 not found; No message file for product=RDBMS, facility=ORA;
\\arguments: [\u01\CONTROLFILES\CONTROL01.CTL]
\\...
```

Iz poročila je lepo razviden problem naše podatkovne baze, in sicer kontrolna datoteka z imenom control01.ctl. Pomembno je omeniti tudi zaustavitev baze, kar zahteva poseben postopek ponovnega zagona.

Za rešitev nastale situacije je potrebno kopirati obstoječo kontrolno datoteko iz diska /u02/ na disk /u01/, ter jo preimenovati v control01.ctl. To deluje, ker so datoteke identične in že na začetku podvojene le iz varnostnih razlogov. Po kopiranju datoteke je potrebno podatkovno bazo ponovno zagnati in odpreti za uporabnike:

```
sql> startup mount;
sql> alter database open;
```

14.2.1.2 Podatkovna baza pri izgubi enega člana skupine dnevnikov sprememb

Iz vsake skupine dnevnikov je en član na kritičnem disku, kar pa nam ne predstavlja preveč problemov, saj izguba enega člana ne povzroči ne zaustavitve ne obesitve podatkovne baze. Potrebno je zaznati napako, ki ne povzroči drastičnih sprememb v delovanju baze in jo tudi razrešiti. Po uspešni razrešitvi problemov s podatkovnim diskom podatkovna baza sama poišče poti do dnevnikov in jih ponovno postavi v aktivno stanje. Pregled alert.log datoteke nam javi sledeče:

```
\\ORA-00345: Message 345 not found; No message file for product=RDBMS, facility=ORA; arguments: [2] [3]
\\ORA-00312: Message 312 not found; No message file for product=RDBMS, facility=ORA; arguments: [3]
\\\u01\REDOLOGS\REDO_LOG01A.LOG]
\\ORA-27072: Message 27072 not found; No message file for product=RDBMS, facility=ORA
\\OSD-04008: WriteFile() failure, unable to write to file
\\O/S-Error: (OS 21) The device is not ready.
```

Poraja se vprašanje, kaj se zgodi, če imamo znotraj skupine dnevnikov sprememb le enega člana in tega izgubimo?

→ Podatkovna baza se ustavi

Pregled alert.log datoteke nam javi sledeče:

\\ Mon Dec 21 10:23:52 2009
\\ Instance terminated by LGWR, pid = 5792

Rešitev problema poteka po sledečem scenariju. Uredi se destinacija, skreira več kot en član vsake skupine dnevnikov sprememb, ponovno se zažene še podatkovna baza z že znanima ukazoma:

sql> startup mount;
sql> alter database open;

14.2.1.3 Podatkovna baza pri izgubi podatkovnih datotek

Za to nesrečo smo pripravljeni v smislu izdelave varnostnih kopij posameznih podatkovnih datotek. Prva stvar je identificiranje poškodovane podatkovne datoteke, kar nam razkrije alert.log, ki nam pošlje sporočilo, podobno spodnjemu:

\\Mon Jan 11 11:17:05 2010
\\KCF: write/open error block=0x2 online=1
\\file=9 u01\podatkovna_datoteka.DBF
\\O/S-Error: (OS 21) The device is not ready.'
\\Automatic datafile offline due to write error on
\\file 9: u01\podatkovna_datoteka.DBF

Za razrešitev problema in postavitev ponovno delujoče podatkovne datoteke je potrebno izvršiti naslednje ukaze iz klienta obnovitvenega upravljalca, ob predpostavki obstoječe varnostne kopije, ki bi morala biti na podatkovnem disku /u02/:

rman> restore datafile podatkovna_datoteka.dbf;

rman> recover datafile podatkovna_datoteka.dbf;

rman> sql' alter tablespace podatkovna_datoteka.dbf online';

Obstaja pa pomembnejša podatkovna datoteka, ki omeji delovanje celotne podatkovne baze in kljub temu, da je le podatkovna datoteka, je ne moremo obnoviti brez ustavitve podatkovne baze in ponovnega zagone le-te. Podatkovna datoteka je system01.dbf in alert.log nam pokaže sporočilo podobno temu:

\\Mon Jan 11 14:22:52 2010
\\Errors in file u01\oracle\product\10.2.0\admin\o103\bdump\o103_reco_908.trc:
\\ORA-01243: Message 1243 not found; No message file for product=RDBMS, facility=ORA
\\Mon Jan 11 14:23:09 2010
\\Instance terminated by DBW0, pid = 5712

Žal iz sporočila ne moremo jasno razbrati, kaj je vzrok napake, ampak zapisovalec podatkovne baze DBW0 nakaže na resen problem in moramo pregledati ključne stvari podatkovne baze, med katerimi je seveda tudi sistemska podatkovna datoteka, ki jo obnovimo na sledeči način, po ponovnem zagonu baze:

rman> startup mount;

rman> restore datafile system01.dbf;

rman> recover datafile system01.dbf;

rman> alter database open;

14.2.1.4 Podatkovna baza pri izgubi programske opreme Oracle

O prenehanju delovanja ni smiselno govoriti, saj podatkovna baza niti ne obstaja več. Potrebno je poiskati napako, ki je to povzročila, mogoče okvara diska, in jo odpraviti, ter ponovno namestiti Oracle programsko opremo. Nameščeno opremo je potrebno ponovno konfigurirati in iz varnostnih kopij postaviti predhodni sistem, kar zahteva malo daljšo pot obnovitve. Za začetek je potrebno obnoviti strežnikovo vzpostavitveno datoteko, ki vsebuje vse namestitve strežnika:

rman> startup nomount;

rman> restore spfile from autobackup;

rman> shutdown;

Do tega koraka smo vzpostavili konfiguracijo podatkovne baze, ki seveda deluje kot strežnik in ima vse stare nastavitve. Vse poti so iste kot pred izgubo programske opreme, le podatkovne datoteke bomo morali obnoviti z obnovitvenim upravljalcem. Da pa ne obnavljamo vsake podatkovne datoteke posebej, uporabimo ukaze znotraj klienta obnovitvenega upravljalca v naslednjem vrstnem redu:

rman> startup mount;

rman> restore database;

rman> recover database;

rman> alter database open;

14.2.1.5 Podatkovna baza pri izgubi podatkovne in kontrolne datoteke

→ Podatkovna baza obvisi, zaradi nedostopnosti kontrolne datoteke

Pregledamo alert.log in po ugotovitvi te napake je postopek obnove sledeč. Sporočilo napake je podobno sporočilu iz točke 14.2.1.1, tudi tu je potrebno prekopirati obstoječo kontrolno datoteko iz diska /u02/ na /u01/, ter jo preimenovati. Po ponovnem zagonu baze je v tem primeru potrebna še obnova podatkovnih datotek, ter izvršitev sprememb zapisanih v arhiviranih dnevnikih sprememb, kar je ponazorjeno v točki 14.2.1.3. Podatkovna baza je po obnovitvi in odprtju pripravljena za delo z uporabniki.

14.2.1.6 Podatkovna baza pri izgubi celotnega /u01/ podatkovnega diska

→ Podatkovna baza obvisi, zaradi nedostopnosti kontrolne datoteke

Prikaz poročila žal ni možen, saj ga izgubimo skupaj s podatkovnim diskom. Poti rešitve podatkov so podobne zgoraj naštetim, z majhno razliko. Podatkovni disk je potrebno popraviti, nanj na novo namestiti Oracle programsko opremo, prav tako je potrebno prekopirati kontrolno datoteko control02.ctl iz podatkovnega diska /u02/ ter jo preimenovati v control01.ctl. Tudi postopek obnove je zelo podoben postopku iz točke 14.2.1.3. V prvem koraku obnovimo strežnikovo vzpostavitveno datoteko, skopiramo kontrolno datoteko in jo preimenujemo. Nato sledi obnovitev podatkovnih področij, čemur sledijo spremembe iz arhiviranih dnevnikov sprememb in nazadnje še odprtje baze za delo z uporabniki.

14.2.2 Nesreča diska /u02/

→ Podatkovna baza obvisi

Pri tej situaciji se srečamo s problemom hierarhije pomembnosti posameznih delov, saj vemo, da podatkovna baza obvisi ob nedostopnosti destinacije arhiviranja dnevnikov, ter se zaustavi ob nedostopnosti kontrolnih datotek. Izgubimo tudi enega člana v skupini dnevnikov - kar ne sproži drastičnih ukrepov - ter mapo z imenom backup. V tej mapi so varnostne kopije sistema, ki jih nadomestimo takoj po ponovni postavitvi podatkovnega diska /u02/. Za sistematičen pregled pa poglejmo, kaj je potrebno popraviti v primeru izgube katerekoli datoteke iz tega podatkovnega diska, ter pri izgubi kombinacij le-teh. Za razkritje napake se ponovno obrnemo na lokacijo:

$oracle_home\admin\<ime_podatkovne_baze>\bdump v datoteko alert.log.

14.2.2.1 Podatkovna baza obvisi zaradi nedostopnosti destinacije arhiviranja dnevnikov

Vse destinacije arhiviranja dnevnikov so obvezne. Ob ugotovitvi problema v destinaciji arhiviranja dnevnikov, je potrebno le-to pregledati ter popraviti. Po urejeneni destinaciji se lotimo popravka nad podatkovno bazo, kjer popravimo status te poti arhiviranja:

sql> alter system set log_archive_dest_state_1='enable';

Po popravku baza normalno nadaljuje s svojim delom in ne izgubimo nobenih podatkov.

V pomoč nam nemalokrat pride tudi sintaksa izbrisa ene od lokacij arhiviranja:

sql> alter system set log_archive_dest_1 ='';

14.2.2.2 Podatkovna baza se ustavi zaradi nedostopnosti kontrolne datoteke

Pregled alert.log datoteke, ki se nahaja na $oracle_home\admin\<ime_podatkovne_baze>\ bdump, nam pokaže sledeče:

\\...
\\ORA-00202: Message 202 not found; No message file for product=RDBMS, facility=ORA;
\\arguments: [\u02\CONTROLFILES\CONTROL02.CTL]
\\...

Za rešitev tega problema je potrebno kopirati obstoječo kontrolno datoteko iz diska /u01/ na disk /u02/, ter jo preimenovati v control02.ctl. Po kopiranju datoteke je potrebnno podatkovno bazo ponovno zagnati in odpreti za uporabnike:

sql> startup mount;
sql> alter database open;

14.2.2.3 Podatkovna baza pri izgubi enega člana skupine dnevnikov sprememb

Izgubili smo enega člana skupine, kar ne povzroči ne zaustavitve ne obesitve podatkovne baze. Potrebno je zaznati takšno napako, ki ne povzroči drastičnih sprememb v delovanju baze in jo tudi razrešiti. Po uspešni razrešitvi problemov s podatkovnim diskom podatkovna baza sama poišče poti do dnevnikov in jih ponovno postavi v aktivno stanje. Pregled alert.log datoteke nam javi sledeče:

\\ORA-00345: Message 345 not found; No message file for product=RDBMS, facility=ORA; arguments: [2] [3]
\\ORA-00312: Message 312 not found; No message file for product=RDBMS, facility=ORA; arguments: [3] [1]
\\ [\u02\REDOLOGS\REDO_LOG01b.LOG]
\\ORA-27072: Message 27072 not found; No message file for product=RDBMS, facility=ORA
\\OSD-04008: WriteFile() failure, unable to write to file
\\O/S-Error: (OS 21) The device is not ready.

Postopek rešitve problema je identičen postopku v točki 14.2.1.2.

14.2.2.4 Na podatkovnem disku imamo kontrolno datoteko ter edinega člana skupine dnevnikov sprememb, kaj zaustavi podatkovno bazo?

Zaustavi jo izguba kontrolne datoteke, stvar uredimo s prekopiranjem in preimenovanjem kontrolne datoteke kot v točki 14.2.2.2. Seveda se spremeni status člana dnevnika sprememb v nedosegljiv oziroma neveljaven, a preklop na tega člana in seveda najdena pot do njega, ga povrne v status veljaven.

Poročilo alert.log-a:

\\ORA-00206: Message 206 not found; No message file for product=RDBMS, facility=ORA; arguments: [3] [1]
\\ORA-00202: Message 202 not found; No message file for product=RDBMS, facility=ORA; arguments:
\\[\u02\CONTROLFILES\CONTROL02.CTL]
\\ORA-27072: Message 27072 not found; No message file for product=RDBMS, facility=ORA

14.2.2.5 Na disku imamo edinega člana skupine dnevnikov sprememb ter obvezno destinacijo arhiviranja dnevnikov sprememb, zakaj podatkovna baza obvisi?

Poročilo alert.log-a:

\\ORA-00345: Message 345 not found; No message file for product=RDBMS, facility=ORA;
\\ORA-00312: Message 312 not found; No message file for product=RDBMS, facility=ORA; arguments: [3]
\\[u02\REDOLOGS\REDO_LOG01b.LOG]
\\ORA-27070: Message 27070 not found; No message file for product=RDBMS, facility=ORA
\\OSD-04016: Error queuing an asynchronous I/O request.
\\O/S-Error: (OS 2) The system cannot find the file specified.

Podatkovna baza obvisi zaradi nedostopnosti dnevnika sprememb. Po ureditvi problemov s podatkovnim diskom mora imeti ta enako ime in strukturo kot pred nesrečo, da lahko Oracle sam poišče poti destinacij dnevnikov sprememb in arhiviranih dnevnikov, ter nadaljuje s svojim delom.

14.2.2.6 Na disku imamo kontrolno datoteko ter obvezno destinacijo arhiviranja dnevnikov sprememb, kaj ustavi podatkovno bazo?

Poročilo alert.log-a:

\\ORA-00206: Message 206 not found; No message file for product=RDBMS, facility=ORA;
\\ORA-00202: Message 202 not found; No message file for product=RDBMS, facility=ORA; arguments:
\\[\u02\CONTROLFILES\CONTROL02.CTL]
\\ORA-27072: Message 27072 not found; No message file for product=RDBMS, facility=ORA
\\OSD-04008: WriteFile() failure, unable to write to file
\\O/S-Error: (OS 21) The device is not ready.

Podatkovno bazo ustavi nedosegljivost kontrolne datoteke, kar uredimo na podoben način kot v točki 14.2.2.2, seveda pa ob ugotovitvi napake katere druge datoteke, ustrezno ukrepamo.

14.2.2.7 Na disku imamo kontrolno datoteko, edinega člana skupine dnevnikov sprememb ter obvezno destinacijo arhiviranja dnevnikov sprememb, kaj ustavi podatkovno bazo?

Poročilo alert.log-a:

\\ORA-00206: Message 206 not found; No message file for product=RDBMS, facility=ORA; arguments: [3] [1]
\\ORA-00202: Message 202 not found; No message file for product=RDBMS, facility=ORA; arguments:
\\[\u02\CONTROLFILES\CONTROL02.CTL]
\\ORA-27072: Message 27072 not found; No message file for product=RDBMS, facility=ORA
\\OSD-04008: WriteFile() failure, unable to write to file
\\O/S-Error: (OS 21) The device is not ready.

Podatkovno bazo ustavi nedosegljivost kontrolne datoteke, kar uredimo na podoben način kot v točki 14.2.2.2, seveda pa ob ugotovitvi napake katere druge datoteke, ustrezno ukrepamo.

14.2.2.8 Datoteka backup

Po vsej tej delitvi in testih nad diskom /u02/ pa ne smemo preskočiti mape backup, kamor se shranjujejo datoteke naše varnostne strategije, ki se seveda razlikuje glede na podatkovno bazo in njen namen. Primer za produkcijsko podatkovno bazo,bi, kot je že omenjeno v poglavju 13.1, izgledal nekako tako:

- varnostno kopiranje celotne podatkovne baze mesečno,
- komulativne varnostne kopije tedensko,
- diferencialne varnostne kopije dnevno,
- pred vsako novo verzijo celotna varnostna kopija ter celoten izvoz podatkovne baze.

V datoteki backup bi imeli varnostne kopije sistema ter polne izvoze sistema, kar ne bi bilo kritično ob izpadu tega podatkovnega diska, le popolna varnostna kopija po popravilu diska bi bila nujna. Lahko pa bi si pomagali celo s podatkovnim diskom /u03/, saj vse narejene varnostne kopije prenašamo s pomočjo operacijskega sistema na ta disk z namenom, da jih v primeru nesreče na podatkovnem disku /u02/ lahko prenesemo nazaj.

14.2.3 Nesreča diska /u03/

→ Podatkovna baza obvisi

Pregled alert.log datoteke, ki se nahaja na $oracle_home\admin\<ime_podatkovne_baze>
\bdump.

Vse destinacije arhiviranja dnevnikov so obvezne, kar je edina stvar, ki onemogoči normalno delovanje podatkovne baze iz tega diska. Na disku je še datoteka backup, ki je kopija backup datoteke iz diska /u02/ in jo le ponovno prekopiramo. Ob ugotovitvi problema v destinaciji arhiviranja dnevnikov, je potrebno destinacijo pregledati, ter popraviti. Po urejeneni destinaciji se lotimo popravka nad podatkovno bazo, kjer popravimo status te poti arhiviranja dnevnikov sprememb:

sql> alter system set log_archive_dest_state_2='enable';

Po teh popravkih baza normalno nadaljuje s svojim delom in ne izgubimo nobenih podatkov.

Naj omenim le alternativno možnost nastavitve destinacije arhiviranja dnevnikov, saj nam, zaradi oddaljenosti podatkovnega diska, le-ta lahko povzroči težavo ali dve. Uporabimo ukaz:

sql> alter system set log_archive_dest_2 ='location=\u03\archivelogs reopen=5
max_failure=3 mandatory';

S tem dosežemo, da tudi ob manjšem izpadu povezave, Oracle poizkuša še 3-krat vsakih 5 sekund izvršiti prenos arhiviranih dnevnikov sprememb na podatkovni disk /u03/, če po teh poizkusih ne pride v stik z oddaljeno lokacijo, se zgodba ponovi.

14.2.4 Nesreča diskov /u01/ in /u02/

Ta nesreča simulira izpad celotne primarne lokacije, kar povzroči izgubo podatkov zadnjih deset minut in ogromno dela za ponovno vzpostavitev sistema. Začnemo s ponovno vzpostavitvijo celotne primarne lokacije in njenih podatkovnih diskov. Namestimo Oracle programsko opremo ter iz podatkovnega diska /u03/ prenesemo mapo backup, iz katere bomo obnovili podatkovno bazo.

Po nameščeni programski opremi je potrebno obnoviti strežnikovo vzpostavitveno datoteko, kar je opisano v točki 14.2.1.4. Za tem je potrebno obnoviti kontrolne datoteke, saj smo obe izgubili. Obnovitev kontrolnih datotek vedno povzroči izbris vsebine dnevnikov sprememb, kar pa za nas v trenutni situaciji ni pomembno, saj smo prav tako izgubili vse dnevnike sprememb. Iz prenesene backup datoteke ponovno obnovimo podatkovne datoteke, katerim sledijo spremembe iz arhiviranih dnevnikov sprememb in nazadnje še odprtje baze za delo z uporabniki.

14.2.5 Nesreča diskov /u01/ in /u03/

Izguba podatkovnega diska /u01/ povzroči novo namestitev in konfiguracijo Oracle programske opreme. V tem primeru moramo ponovno obnoviti strežnikovo vzpostavitveno datoteko, kar je opisano v točki 14.2.1.4, sledi kopiranje kontrolne datoteke iz podatkovnega diska /u02/ na /u01/ ter njeno preimenovanje. Tako je kritčni del podatkovne baze rešen, saj ni bilo potrebe po obnovi kontrolnih datotek iz varnostne kopije, kar pomeni, da se vsebina dnevnikov sprememb ohrani. Obnovimo še podatkovne datoteke, katerim sledijo spremembe iz arhiviranih dnevnikov sprememb in nazadnje še odprtje baze za delo z uporabniki. Sekundarno lokacijo uredimo s preprostim kopiranjem stvari iz podatkovnega diska /u02/, saj je vse kar vsebuje podatkovni disk /u03/ kopija oziroma duplikat kritičnih stvari na podatkovnem disku /u02/.

14.2.6 Nesreča diskov /u02/ in /u03/

Izpad podatkovnih diskov /u02/ in /u03/ za naše podatke ne pomeni nič kritičnega, ustavi pa sistem kot celoto, saj /u02/ vsebuje kontrolno datoteko. Za ponovno normalno delovanje podatkovne baze sledimo postopku obnove, opisanem v točki 14.2.1.1. Tako dobimo ponovno delujočo bazo, brez izgube podatkov. Ponovno uredimo vse poti dnevnikov sprememb in arhiviranih dnevnikov sprememb, da jih lahko Oracle avtomatsko zazna in nadaljuje z delom. Delujoč sistem obvezno takoj varnostno kopiramo in kopijo prenesemo na podatkovni disk /u03/ oziroma na sekundarno lokacijo.

14.2.7 Nesreča diskov /u01/, /u02/ in /u03/

Nesreča vseh treh podatkovnih diskov za naš sistem predstavlja katastrofo, saj smo izgubili celoten sistem, skupaj z vsemi podatki. Verjetnost tega pojava je zelo majhna, saj morata v istem trenutku zatajiti primarna in sekundarna lokacija, vendar pa sem osebno to tveganje pripravljen sprejeti.

15 Varnost informacij in priporočila standarda ISO 1779

Standard ISO 17799 [9] je zbirka pravil in metod nadzora za področje informacijske varnosti. Predstavlja priporočilo o varovanju informacij in informacijskih sistemov. Bistvo standarda ISO 17799/BS 7799 je v ocenjevanju in obvladovanju tveganj ter varnosti informacij, njegov cilj pa je varnost. Razvili so ga iz BS 7799 (British Standard for Informaton Security Management) in je kot takšen dobil mednaroden naziv ISO 17799.

Organiziran je v deset glavnih sekcij ali kategorij, od katerih vsaka pokriva specifično področje:

1. *Politika varovanja* - obravnava formalne usmeritve in podpore za varovanje informacij. Poudarja pomen sodelovanja vodstvenih struktur pri ustvarjanju in vpeljevanju sistema varovanja informacij v celotni organizaciji.
2. *Ocena tveganj in njihovo obvladovanje* – obravnava, razvršča, prioritizira, ocenjuje varnostna tveganja in podaja smernice za njihovo obvladovanje.
3. *Organiziranost varovanja* - obravnava organizacijsko infrastrukturo, ki mora biti ustvarjena za učinkovito zaščito podatkov v sami organizaciji in pri prenosu podatkov izven organizacije.
4. *Razvrstitev in kontrola sredstev* - se ukvarja z lastnino in odgovarjajočo zaščito sredstev informacijske tehnologije in informacij v organizaciji. Obravnava tudi pomen razvrstitve informacij glede na pomen in občutljivost.
5. *Varovanje v zvezi z osebjem* - natančno določi izbor, šolanje in odgovornosti zaposlenih. Poudarja postopke za zmanjšanje tveganja človeške napake, kraje, poneverbe in izrabe kapacitet. Zaposleni se morajo zavedati pomena in vrednosti informacij.
6. *Fizično in okolno delovanje* - se ukvarja s fizičnim varovanjem in z dostopom do opreme ter informacij. Pomen lokacije in varovanja glavnega strežnika oziroma delovne postaje za obdelavo podatkov.
7. *Ravnanje z računalniki in omrežjem* - svetuje pravilno upravljanje in varno delovanje omrežja oziroma vseh sistemov za obdelavo informacij. Določa postopke v primeru okvar ali vdora v sistem.
8. *Obvladovanje dostopov* - obravnava kontrolo dostopa do informacij glede na poslovne in varnostne potrebe.
9. *Razvoj in vzdrževanje sistemov* - se posveča načrtovanju varnih in sodobnih sistemov, ki preprečujejo izgubo in zlorabo informacij ter zagotavljajo njihovo avtentičnost, zaupnost in neoporečnost. Svetuje glede varnosti pri vključevanju novih aplikacij oziroma sprememb programske opreme.
10. *Usklajenost* - ta kategorija se posveča usklajenosti varovanja informacij oziroma delovanja z domačimi in mednarodnimi zakoni ter standardi.

Kategorije, ki se dotikajo našega področja so ocena tveganj in njihovo obvladovanje, fizično in okolno varovanje, ter ravnanje z računalniki in omrežji. To so tudi teme, katerim bomo namenili malo več pozornosti in pogledali, kaj priporočajo v zvezi s samo informacijsko varnostjo.

15.1 Ocena tveganj in njihovo obvladovanje

Ta kategorija standarda govori o tveganjih, ki jih je potrebno zaznati, jih zmanjšati in nenazadnje tudi obvladovati.

Skozi nalogo se ocenjuje tveganje pri izpadu vsakega podatkovnega diska in določi minute izgubljenih informacij za vsak primer posebej. Razvrstitev pomembnosti tveganj bi bila sledeča:

- izpad celotnega sistema, kjer izgubimo vse podatke,
- izpad primarne lokacije, kar pomeni izgubo podatkov zadnjih 10 min, od tu dalje več ne izgubljamo podatkov,
- izpad podatkovnega diska /u01/,
- nadaljne kombinacije tveganj izpada diskov so približno enake.

Pomembno je omeniti, da je sistem narejen s politiko minimiziranja tveganja, kar pomeni, da količina podatkov izgubljenih z nesrečo narašča s padanjem verjetnosti pojavitve nesreče. Najpomembnejše tveganje je izpad celotnega sistema, se pravi primarne in sekundarne lokacije skupaj, kar pa je praktično nemogoče.

Obnovitve po posameznih nesrečah so opisane v praktičnem delu naloge oziroma pod točko 14, kar nam nakazuje tudi način obvladovanja teh tveganj.

15.2 Fizično in okolno varovanje

Cilj tega sklopa je preprečiti nepooblaščen fizičen dostop, povzročitev škode, ter raznih motenj v organizacijskem prostoru in informacijah.

Kritične in občutljive lokacije strežnikov morajo biti dobro varovane, tako fizično (na primer ograja in preverjanje dostopov), kot omrežno (požarni zidovi, varne povezave, kodirane povzave ...).

Za našo strukturo pomemben del o varovanju podatkov pred zunanjimi in naravnimi nesrečami opisuje, kako je potrebno sistem fizično zavarovati pred požari, poplavami, potresi, raznimi protesti, ter eksplozijami. V tem sklopu moramo biti pozorni tudi na sosednje sobe in stavbe, saj lahko le-te predstavljajo velike nevarnosti, zato so potrebni naslednji ukrepi:

a) nevarne snovi (npr. eksplozivne) shranjujemo na varni razdalji,
b) medij z varnostnimi kopijami, naš podatkovni disk /u03/ mora biti shranjen na varni razdalji, da nesreča primarne lokacije ne vpliva tudi nanj,
c) primerna gasilska orodja je potrebno namestiti znotraj in v bližini kritičnega območja.

Razdelek podpornih dejavnosti nas opozori na zaščito sistema pred izpadom električne energije, primanjkljajem vode, izpadom klime, kar mora biti redno pregledano in testirano ter obvladljivo.

Pomembnost napajalnih in telekomunikacijskih poti ter njihovo zaščito pred poškodbami in prestrezanjem, ISO standard omenja v sklopu varnosti prenosnih poti. Za varnost le-teh sledimo naslednjim smernicam:

a) napajalne in telekomunikacijske poti naj bodo, kjer možno, pod zemljo,
b) telekomunikacijskih povezav, kjer možno, ne speljemo skozi javna omrežja,
c) napajalne in komunikacijske poti naj bodo ločene za preprečevanje motenj,
d) komponente komuniciranja in napajanja, naj bodo jasno označene, za lažje vzdrževanje.

15.3 Ravnanje z računalniki in omrežji

Cilj kategorije je zagotoviti pravilno in varno delovanje naprav za obdelavo informacij.

Naš ključni del tega poglavja je varnostno kopiranje in omrežni nadzor, zanimanje za prvi del je očitno, drugi del pa se nanaša na naš prenos podatkov preko omrežja komunikacijskih kanalov na sekundarno lokacijo.

Varnostno kopiranje izvajamo za ohranjanje celovitosti in razpoložljivosti informacij. Potrebno je uvesti rutinske postopke varnostnega kopiranja za izvajanje dogovorjene varnostne politike in strategije, ter kopije redno testirati. Pri varnostnem kopiranju je potrebno upoštevati še sledeče smernice:

a) definirati stvari potrebne varnostnega kopiranja,
b) natančne in celovite varnostne kopije,
c) dokumentiranje metod restavriranja,
d) frekvenca izdelovanja varnostnih kopij mora ustrezati zahtevam poslovanja, ter varnostnim zahtevam vključenih informacij,
e) varnostne kopije se shranjujejo na oddaljeno lokacijo,
f) mediji, na katerih so varnostne kopije, morajo biti redno testirani,
g) procedure obnavljanja morajo biti redno pregledane in testirane,
h) pomembnejše informacije kriptiramo.

Omrežni nadzor zagotavlja varnost informacij v omrežju ter infrastrukturi, katero povezuje. Omrežja razširjajo meje organizacij in so kot takšna zelo občutljiva na nepredvidljivih javnih mrežah. Potrebno jih je skrbno načrtovati, vzdrževati, nadzirati in zaščiti pred morebitnimi prestrežbami in motnjami zunanjih dejavnikov.

Skrbniki omrežja so zadolženi za implementacijo kontrole dostopov uporabnikov v omrežja, s katerim preprečujejo nepooblaščene dostope. Na splošno je potrebno upoštevati:

a) odgovornosti za zagotavljanje omrežne varnosti morajo biti ločene od odgovornosti za računalniško varnost,
b) odgovornosti in procedure za oddaljeno opremo morajo biti jasne in natančno dokumentirane,
c) vzpostavljene morajo biti posebne kontrole za zagotovitev zaupnosti in celovitosti prenosa podatkov skozi javna omrežja,
d) vzpostaviti je potrebno sistem prijavljanja uporabnikov v omrežje, za njihovo sledljivost.

16 Literatura

[1] (2009) Podatki. Dostopno na: http://www2.arnes.si/~bmohor3/Urejanje_Besedila/podatek.html

[2] (2009) Definicija informacije. Dostopno na: http://colos.fri.uni-lj.si/eri/racunalnistvo/informatika/podatke_informacija_znanje.html

[3] (2009) Opis razlike med podatki in informacijami. Dostopno na: http://files.gsobar.uni.cc/gradiva_informatika_omrezja_baze/colos/informatika/informatika/podatki_in_informacije/vloga_in_pomen_informacije.html

[4] (2009) Podatkovna revolucija. Dostopno na: http://sl.wikipedia.org/wiki/Podatkovna_baza

[5] (2009) Termin podatkovna baza. Dostopno na: http://sl.wikipedia.org/wiki/Podatkovna_baza

[6] (2009) Naloge sistemov za upravljanje s podatkovnimi bazami. Dostopno na: http://sl.wikipedia.org/wiki/SUPB

[7] (2009) Vrste nesreč podatkovnih baz. Dostopno na: http://gbr.pepperdine.edu/033/dataloss.html

[8] (2009) Stroški nesreč podatkovnih baz. Dostopno na: http://gbr.pepperdine.edu/033/dataloss.html

[9] (2009) Mednarodni standardi. Dostopno na: http://www.iso.org

[10] (2010) Podjetje Terian. Dostopno na: http://www.terian.com/

[11] Kevin Loney, Bob Bryla, *DBA Handbook*, str. 38-40, 2005.

[12] Robert Freeman, Matthew Hart, *Oracle9i RMAN Backup & Recovery,* str. 250-256, 2007.

[13] Thomas Kyte, *Expert Oracle Database Architecture*, str. 88-103, 2005.

Boštjan Harnik

DIPLOMSKO DELO