Administriranje baza podataka

Predavanje 02

Tema: Sigurnost na SQL Serveru

Sigurnost na SQL Serveru

1. Osnovno o sigurnosti na SQL Serveru
2. Sigurnost na razini instance
3. Sigurnost na razini baze

Osnovno o sigurnosti na SQL Serveru

- Entiteti kod upravljanja sigurnošću
- GRANT, REVOKE, DENY

Entiteti kod upravljanja sigurnošću (1)

- Korisnici (engl. *principals*)
  - Trebaju koristiti objekte unutar DBMS-a
- Štićeni objekti (engl. *securables*)
  - Objekti do kojih se regulira pristup
- Prava (engl. *permissions*)
  - Koriste se za kontrolu pristupa štićenim objektima

Entiteti kod upravljanja sigurnošću (2)

- Korisnici na SQL Serveru:

Razina	Tipovi objekata
Windows	• Lokalni Windows korisnički računi ili grupe korisnika • Domenski Windows korisnički računi ili grupe korisnika
SQL Server instanca	• *Logini* • Serverske role (engl. *server roles*)
Baza	• *Useri* (hrv. *korisnici baze*) • Bazne role (engl. *database roles*)

Entiteti kod upravljanja sigurnošću (3)

- Primjeri štićenih objekata na SQL Serveru:

Doseg	Štićeni objekti
Instanca	• Baze
Baza	• Sheme
Shema	• Tablice • Pogledi • Funkcije • Procedure

Shema

- **Shema je objekt u SQL Server bazi**
- **Služi za logičko grupiranje srodnih objekata**
  - Tablice, pogledi, funkcije, procedure
- **Služi kao centralna točka upravljanja svim objektima koje sadržava**
  - Primjerice, ako damo korisniku pravo na shemu, dali smo mu pravo na sve objekte unutar te sheme
- **Shema je logički slična mapi na datotečnom sustavu**

Referenciranje objekata

- **Objekti se obično referenciraju koristeći dvodijelne oznake**

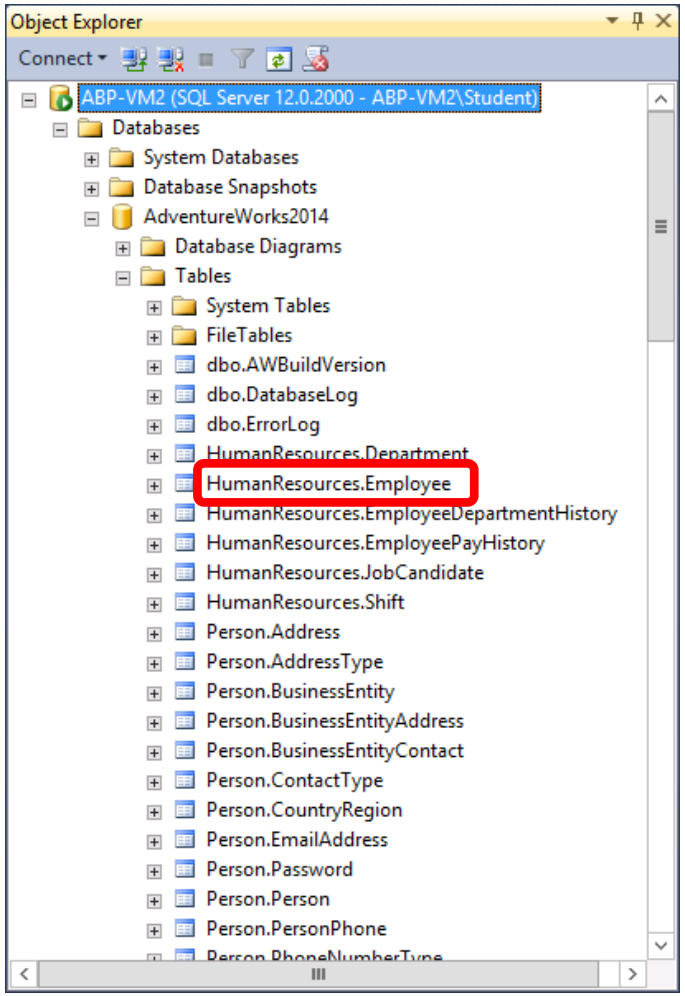
  [schema_name].object_name

  – Naziv sheme se može izostaviti, pri čemu se podrazumijeva korisnikova *defaultna* shema
- **Ponekad nam trebaju trodijelne ili četverodijelne oznake**

  database_name.[schema_name].object_name

  server_name.[database_name].[schema_name].object_name

Primjer sheme



SHEMA

[ABP-VM2].AdventureWorks2014.HumanResources.Employee

Entiteti kod upravljanja sigurnošću (4)

- Primjeri prava na SQL Serveru:

Tip štićenog objekta	Prava
Instanca	• ALTER ANY DATABASE, ALTER ANY LOGIN • SHUTDOWN
Baza	• CREATE TABLE, CREATE PROCEDURE • ALTER ANY USER
Tablica	• SELECT, INSERT, UPDATE, DELETE, ALTER
Pohranjena procedura	• ALTER • EXECUTE

GRANT, DENY, REVOKE (1)

- Prava se *loginima* i *userima* dodjeljuju preko T-SQL naredbi:
  - **GRANT** – davanje prava

```
GRANT INSERT ON Artikli TO TOKYO\ana
```

  - **DENY** – eksplicitna zabrana

```
DENY INSERT ON Artikli To TOKYO\ana
```

  - **REVOKE** – poništavanje prethodnog statusa

```
REVOKE INSERT ON Artikli FROM TOKYO\ana
```

  - Ako korisniku nije eksplicitno dano pravo sa GRANT, to obično znači da korisnik ne može izvesti određenu akciju

GRANT, DENY, REVOKE (2)

- Nasljeđivanje prava
  - Ako korisnik ima pravo nad shemom, ima pravo na svim objektima u toj shemi
  - Pravo se može naslijediti preko pripadnosti grupi (roli)
- Efektivna prava
  - Korisnik može izvesti određenu akciju ako vrijede obje tvrdnje:
    - Pravo je eksplicitno dano tom korisniku ili nekoj njegovoj roli
    - Pravo nije eksplicitno zabranjeno tom korisniku niti bilo kojoj njegovoj roli
  - DENY je uvijek jači od GRANT

GRANT, DENY, REVOKE (3)

- Primjerice, postoji tablica Nabava.Artikl
- Zadajemo sljedeće naredbe:

```
GRANT INSERT ON schema::Nabava TO Skladistar
DENY INSERT ON Artikl To Skladistar
REVOKE INSERT ON Artikl FROM Skladistar
```

Objekt	Pravo	User	Dozvola
schema::Nabava	INSERT	Skladistar	GRANT
Artikl	INSERT	Skladistar	DENY

  - Nakon druge naredbe, Skladistar nema pravo INSERT u Artikl
  - Nakon treće naredbe, Skladistar ponovno ima pravo INSERT u Artikl

2. Sigurnost na razini instance

- Autentikacija u SQL Serveru
- Upravljanje *loginima*
- Serverske role

Autentikacija u SQL Serveru (1)

- **Autentikacija** – provjera identiteta korisnika
  - Odgovara na pitanje „tko si ti?”
  - Upisivanje korisničkog imena i zaporke
  - SQL Server se može osloniti na autentikaciju koju naprave Windowsi
  - Autentikacija preko digitalnih certifikata
- **Autorizacija** – davanje prava na izvođenje određene akcije
  - Odgovara na pitanje „što smiješ?”
  - SQL Server provjerava popis dopuštenja za dotičnog korisnika
  - Da bi se korisniku dala autorizacija, prvo mora biti autenticiran

Autentikacija u SQL Serveru (2)

- Dva načina autentikacije

- **Windows authentication mode**
  - Autentikaciju korisnika obavljaju Windowsi
  - Za pristup SQL Serveru, Windows korisnički račun mora biti povezan sa SQL Server *loginom*
    - Korisnikov token, kreiran kod prijave na Windowse, pokazuje se SQL Serveru
    - Ako SQL Server za taj token pronađe *login*, uspostavlja se konekcija

Autentikacija u SQL Serveru (3)

- **SQL Server and Windows authentication mode**
  - Mixed mode: pored Windows autentikacije, moguća je i autentikacija od strane samog SQL Servera
  - SQL Server provjerava korisničko ime (naziv *logina*) i lozinku u svom internom katalogu

Autentikacija u SQL Serveru (4)

- Prednosti **Windows autentikacije**:
  - Olakšana administracija jer se *logini* mogu povezati na Windows grupe
  - Preko mreže ne prenosi se lozinka, nego token
- Kada koristiti **Mixed mode**:
  - Za podršku starim aplikacijama
  - Kad SQL Serveru trebaju pristupiti korisnici bez Windows korisničkih računa
    - Vrlo često se koristi kod web aplikacija
  - Nema domene

Upravljanje loginima

- Tipovi *logina*:
  - Windows *logini* – mapirani na Windows korisničke račune
  - SQL *logini* – autenticira ih SQL Server
    - sa – SQL login s najvišim privilegijama
    - Ako instaliramo SQL Server u Windows authentication načinu i poslije se prebacimo u Mixed način, sa je onemogućen
      - Treba ga omogućiti i postaviti mu lozinku

Serverske role (1)

- Svrha je olakšati dodjelu prava loginima
- Predefinirani skupovi prava na razini instance
  - Primjerice, ako loginu želimo omogućiti da kreira/mijenja/briše/restaurira baze, onda umjesto da kažemo "dajem pravo create", "dajem pravo alter", "dajem pravo drop", itd. kažemo samo "učlani ga u dbcreatore"
- Vrste serverskih rola:
  - Fiksne (njihova prava se ne mogu mijenjati)
  - Korisnički definirane
    - Mogu se po volji kreirati
    - Mogu im se dodijeliti proizvoljna prava

Serverske role (2)

- Fiksne serverske role:

Rola	Opis
sysadmin	Može izvesti bilo koju akciju; rola s najvišim privilegijama na SQL Serveru
dbcreator	Može kreirati, mijenjati i brisati baze
diskadmin	Upravlja datotekama baza
serveradmin	Može konfigurirati instance SQL Servera
securityadmin	Upravlja loginima
processadmin	Upravlja procesima na SQL Serveru
bulkadmin	Može izvršavati BULK INSERT naredbe
setupadmin	Može konfigurirati replikaciju i povezane servere

Fiksne serverske role i prava

- Da bismo saznali koja prava ima fiksna rola, koristimo proceduru sp_srvrolepermission:

```
EXEC sp_srvrolepermission @srvrolename = 'securityadmin'
```

	ServerRole	Permission
1	securityadmin	Add member to securityadmin
2	securityadmin	Grant/deny/revoke CREATE DATABASE
3	securityadmin	Read the error log
4	securityadmin	sp_addlinkedsrvlogin
5	securityadmin	sp_addlogin
6	securityadmin	sp_defaultdb
7	securityadmin	sp_defaultlanguage
8	securityadmin	sp_denylogin
9	securityadmin	sp_droplinkedsrvlogin
10	securityadmin	sp_droplogin
11	securityadmin	sp_dropremotelogin
12	securityadmin	sp_grantlogin
13	securityadmin	sp_helplogins
14	securityadmin	sp_password
15	securityadmin	sp_remoteoption (update)
16	securityadmin	sp_revokelogin

Serverske role (3)

- Svaka instanca sadrži serversku rolu **public**
  - Svi su *logini* članovi te role
  - Ima pravo VIEW ANY DATABASE
    - Vide samo da baza postoji, nikakva prava nemaju na bazama
  - Njoj se mogu dodijeliti i neka druga prava
    - Oprezno!

3. Sigurnost na razini baze

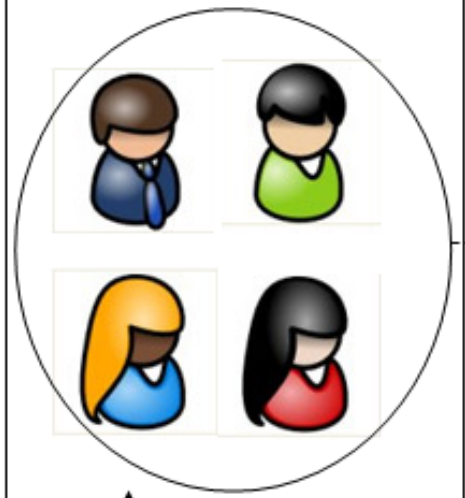
- Upravljanje *userima*
- Bazne role

Upravljanje userima (1)

- Za pristup instanci SQL Servera potreban je ***login***
- Za pristup bazi potreban je ***user***
- *Useri* se povezuju (mapiraju) s *loginima*
  - Da bi netko mogao raditi s određenom bazom, mora imati *login* povezan s nekim *userom* u toj bazi
  - *Useri* iz različitih baza mogu biti mapirani na isti *login*
    - Nazivi ne moraju biti jednaki
  - Jedan *login* ne može biti mapiran na dva različita *usera* u istoj bazi

Upravljanje userima (2)

- Primjer: desetak Windows korisnika treba imati ista prava na nekoj bazi
  - Windows korisnike stavimo u Windows grupu WG1
  - Kreiramo (Windows) *login* SL1 i mapiramo ga na WG1
  - Kreiramo *database usera* DU1 i mapiramo ga na SL1
  - Prava dodijelimo DU1

WINDOWS	SQL SERVER	BAZA ADVENTUREWORKS
 Domena1\Prodavaci (*Windows grupa*)	 Domena1\Prodavaci (*login*)	 Prodavac (*database user*)
 Domena1\Ana (*Domenski račun*)		

Upravljanje userima (3)

- Specijalni *useri* koji postoje u svakoj bazi:
  - dbo
    - *User* koji ima sva prava na bazi
    - *Login* **sa** i članovi role **sysadmin** automatski se mapiraju na *usera* dbo u svakoj bazi
    - Ne može se obrisati
  - guest
    - Ako neki *login* nije mapiran niti na jednog *usera* u nekoj bazi, pokušat će joj pristupiti pod identitetom *usera* guest
    - Onemogućen po *defaultu*
    - Ako ga omogućimo i damo mu neka prava, i *logini* bez povezanih *usera* će moći pristupati bazi

Bazne role (1)

- Bazne role – grupiranje prava na razini baze
  - Fiksne bazne role
    - Ugrađene role, ne mogu se mijenjati
  - Korisnički definirane bazne role
    - Omogućuju definiranje vlastitih rola s potrebnim skupom prava
    - CREATE ROLE, ALTER ROLE, DROP ROLE
      - ALTER ROLE <rola> ADD MEMBER <user>
      - ALTER ROLE <rola> DROP MEMBER <user>

Bazne role (2)

- Fiksne bazne role:

Naziv fiksne bazne role	Opis
db_accessadmin	Dodaje i briše *usere* i role
db_backupoperator	Izrađuje sigurnosne kopije baze
db_datareader	Može čitati podatke iz bilo koje tablice u bazi
db_datawriter	Može dodavati, brisati i mijenjati podatke u bilo kojoj tablici
db_ddladmin	Može stvarati, brisati i mijenjati objekte unutar baze
db_denydatareader	Ne može čitati podatke niti iz jedne tablice u bazi
db_denydatawriter	Ne može promijeniti podatke niti u jednoj tablici u bazi
db_owner	Može izvesti bilo koju aktivnost u bazi
db_securityadmin	Može mijenjati bazne i aplikacijske role te stvarati scheme
public	Služi za grupiranje *defaultnih* prava

Bazne role (3)

- Rola *public*
  - Ugrađena rola, svaki *user* je njezin član
  - Potreban poseban oprez pri modificiranju njezinih prava

Primjer 1

- Omogućite Evi dohvaćanje podataka iz tablice Proizvod.

1. Kreiramo *login* DOMENA\Eva i vežemo ga uz domenski račun
2. Mapiramo *login* na *usera* DOMENA\Eva u bazi
3. *Useru* damo prava SELECT na tablicu Proizvod

Primjer 2

- Odjel Marketinga ima 25 zaposlenika. Omogućite im pristup tablici Promocija tako da mogu s njom raditi sve DML operacije.

1. Napravimo Windows grupu Marketing
2. Dodamo zaposlenike u tu grupu
3. Kreiramo *login* DOMENA\Marketing za Windows grupu Marketing
4. Mapiramo *login* na *usera* DOMENA\Marketing u bazi
5. *Useru* damo prava SELECT, INSERT, UPDATE, DELETE na tablicu Promocija

Primjer 3

- **Nastavno na prethodni primjer, 10 zaposlenika Uprave također želi raditi jednake akcije s tablicom Promocija kao i Marketing**

1. **Napravimo Windows grupu Marketing**
    1. Dodamo zaposlenike u tu grupu
    2. Kreiramo *login* DOMENA\Marketing
    3. Mapiramo *login* na *usera* u bazi
2. **Isto ponovimo za Upravu**
3. **Napravimo baznu rolu**
    1. Roli damo prava na tablicu Promocija
    2. U rolu dodamo *usere* Marketing i Uprava

4. Smještaj datoteka na disku

- Tipovi datoteka od kojih se baze sastoje
- Raspored datoteka po diskovima

Tipovi datoteka od kojih se baze sastoje

- Tablice i indeksi smještaju se u standardne datoteke operativnog sustava

- Tipovi datoteka koje čine bazu:

  – Podatkovne datoteke (engl. *data files*)
    - Sadrže korisničke i sistemske tablice i indekse

  – Log datoteke (engl. *log files*, *transaction logs*)
    - Osiguravaju transaktibilnost operacija nad bazom
    - Omogućavaju oporavljivost baze

Raspored datoteka po diskovima

- Tablice i indekse treba dobro rasporediti u datoteke
  - Datoteke treba dobro razmjestiti po diskovima
- Raspored može značajno utjecati na performanse, pa vrijede preporuke:
  - Ne držati podatkovne i log datoteke na istom disku
  - Odvojiti tablice od njihovih indeksa
  - Odvojiti tablice koje se često spajaju
  - Odvojiti arhivske podatke
- Bolje je imati više manjih diskova nego jedan veliki

5. Upravljanje bazama i datotekama na SQL Serveru

- Tipovi datoteka u SQL Server bazama
- Sistemske baze
- Raspored datoteka na disku
- Grupe datoteka

Tipovi datoteka u SQL Server bazama (1)

Tip datoteke	Ekstenzija	Opis
Primarna datoteka	.mdf	Sadrži *startup* informacije za bazu te sistemske tablice. U njoj se nalaze i korisničke tablice. Svaka baza ima točno jednu primarnu datoteku.
Sekundarne datoteke	.ndf	One su opcionalne. Koriste se ako postoji potreba da se podaci razdvoje na više diskova.
Transakcijski logovi	.ldf	Svaka baza mora imati barem jedan transakcijski log. U njemu se nalaze podaci nužni za oporavak baze.

Sistemske baze u SQL Serveru

- **Master**
  - Sadrži konfiguracijske postavke instance
  - Sadrži informacije o korisničkim bazama
- **MSDB**
  - Sadrži informacije o automatiziranim zadacima, operatorima i *alertima*
- **Model**
  - Služi kao predložak za kreiranje novih baza
- **Tempdb**
  - Sadrži privremene tablice, radne tablice koje SQL Server koristi interno, međurezultate sortiranja, …

Raspored datoteka na disku

- **Općenite preporuke vrijede i za SQL Server baze**

- **Dodatna preporuka:**
  - Bazu tempdb odvojiti na poseban disk

Rast datoteka

- **Podatkovnim i transakcijskim datotekama možemo definirati sljedeća svojstva:**
  - Initial size: definira veličinu datoteke na disku nakon primjene naredbe (CREATE DATABASE ili ALTER DATABASE)
  - Automatic Growth: je li automatski rast datoteke moguć ili ne
    - Ako je moguć, definiramo koliki će biti u postotcima ili u fiksnom iznosu MiB

Grupe datoteka – filegroups (1)

- **Imenovane kolekcije podatkovnih datoteka**
- **Svaka grupa datoteka sadrži jednu ili više podatkovnih datoteka**
- **Koriste se za razdvajanje tablica ili indeksa na posebne diskove**
  - Kod kreiranja tablice ili indeksa možemo odrediti u kojoj će se grupi nalaziti
  - Ne možemo odrediti u kojoj datoteci će se nalaziti

Grupe datoteka – filegroups (2)

- **Kroz koncept grupa datoteka dobivaju se sljedeće mogućnosti:**
  - Mogućnost *scale-out* baze na dodatne diskove
  - Odvajanje podataka s različitim zahtjevima za administriranje
    - Primjerice, sigurnosne kopije možemo izrađivati za svaku grupu posebno
  - Pojedine grupe datoteka mogu se definirati kao *read-only*

Grupe datoteka – filegroups (3)

- **Tipovi:**
  - Primarna grupa datoteka
    - Svaka baza mora imati primarnu grupu datoteka
    - Obavezno sadrži primarnu datoteku
  - Korisnički definirane grupe
    - Baza ne mora imati niti jednu korisnički definiranu grupu datoteka ili može imati jednu ili više njih
- ***Defaultna* grupa datoteka**
  - Grupa u koju će se smjestiti nove tablice ako se ništa posebno ne specificira

Grupe datoteka – filegroups (4)

- **Napomene:**
  - Grupe datoteka mogu se kreirati odmah pri kreiranju baze ili naknadno
  - Datoteke se ne mogu premještati u druge grupe
  - Mnoge baze radit će dobro i sa samo jednom podatkovnom i jednom log datotekom, bez razdvajanja u grupe datoteka
  - Treba procijeniti koliko se dobiva uvođenjem korisnički definiranih grupa, a koliko se gubi na lakoći održavanja

Primjer 1

- **Imamo bazu podataka u kojoj smo odvojili podatkovnu i transakcijsku datoteku na posebne diskove. Na disku s podatkovnom datotekom uskoro neće biti mjesta. Što je najlakše napraviti?**
  - Bazi dodati još jednu podatkovnu datoteku (.ndf) u *filegroup* kojeg ćemo smjestiti na drugi disk
  - Posljedično, popravit će se i performanse

Primjer 2

- **Imamo bazu podataka u kojoj smo odvojili podatkovnu i transakcijsku datoteku na posebne diskove. Tablice Racun i Stavka su ogromne i vrlo često se radi njihovo spajanje koje traje predugo. Što možemo napraviti?**
  - Bazi dodati još jednu grupu datoteka
  - U novu grupu datoteka dodati novi .ndf
  - U novu grupu datoteka prebaciti Stavke

Tema: sigurnosne kopije

Sadržaj

- Potreba za izradom sigurnosnih kopija
- Tipovi sigurnosnih kopija
- Određivanje rasporeda izrade sigurnosnih kopija
- Preporuke za izradu sigurnosnih kopija

Potreba za izradom sigurnosnih kopija (1)

- Razne pogreške mogu uzrokovati nedostupnost baza ili oštećenje podataka
  - Hardverske greške
    - Kvar diskova, memorije, matične ploče, mrežni problemi
    - Razne tehnike (RAID, *failover*, UPS) mogu povećati pouzdanost, ali ne mogu sasvim otkloniti takve greške
  - Softverske greške
    - *Bugovi* u DBMS-u, operativnom sustavu ili nekom drugom softveru
  - Ljudske pogreške
    - Pogrešno ažuriranje podataka, nehotično brisanje, …

Potreba za izradom sigurnosnih kopija (2)

- **Ponekad se zastoj može popraviti i može se nastaviti s radom bez posebnih intervencija na bazi**
  - Npr. zamjena radne memorije
- **Ponekad je za nastavak rada potrebno vratiti bazu u neko stanje prije nastanka problema**
- **Oporavak (engl. *recovery*)** - vraćanje baze u neko ranije stanje
- **Sigurnosne kopije (engl. *backups*)** – kopije baze iz kojih se može napraviti oporavak

Tipovi sigurnosnih kopija (1)

- Izrada sigurnosnih kopija baza može biti puno složenija nego za “obične” datoteke
  - Baze se sastoje od podatkovnih i log datoteka
    - Treba uskladiti njihov sadržaj prilikom kopiranja
  - Može postojati potreba da se kopiraju samo promjene
    - Ako je baza velika, kopiranje cijele baze može trajati jako dugo
  - Treba znati koji tip kopije odabrati u određenoj situaciji

Tipovi sigurnosnih kopija (2)

- Poslovanje nameće zahtjeve za:
  - Brzinom oporavka
  - Maksimalnom količinom izgubljenih podataka

- Treba uspostaviti odgovarajuću strategiju izrade sigurnosnih kopija → treba poznavati tipove sigurnosnih kopija u konkretnom DBMS-u te njihove karakteristike

Tipovi sigurnosnih kopija (3)

- **On-line sigurnosne kopije**
  - Izrađuju se za vrijeme dok korisnici rade s bazom (*hot backups*)
  - U trenutku završetka kopiranja podaci u sigurnosnoj kopiji mogu biti nekonzistentni
  - Prednost: baza je cijelo vrijeme dostupna
  - Nedostatak: duže trajanje kopiranja, moguć pad performansi
- **Off-line sigurnosne kopije**
  - Izrađuju se na "spuštenoj" bazi (*cold backups*)
  - Prilikom stavljanja off-line, DBMS se pobrine da podaci u bazi budu konzistentni
  - Prednost: kopiranje traje kraće
  - Nedostatak: potrebno je bazu učiniti nedostupnom

Tipovi sigurnosnih kopija (4)

- Potpune sigurnosne kopije (*full backups*)
  - Sadrže sve stranice svih podatkovnih datoteka
- Inkrementalne kopije (*incremental backups*)
  - Sadrže samo stranice promijenjene nakon zadnje potpune ili inkrementalne kopije
- Diferencijalne
  - Promjene od zadnje potpune kopije

Određivanje rasporeda izrade sigurnosnih kopija (1)

- **Oprečni zahtjevi kod izrade sigurnosnih kopija:**
  - Neka potencijalni gubitak podataka bude što manji => sigurnosne kopije se trebaju izrađivati što češće
  - Neka izrade sigurnosnih kopija što manje ometaju normalan rad sustava => sigurnosne kopije se trebaju izrađivati što rjeđe

- **U suradnji s poslovnim korisnicima treba analizirati prirodu podataka i njihovu važnost**

Određivanje rasporeda izrade sigurnosnih kopija (2)

- **Analiza treba dati odgovore na pitanja:**
  - Koliko se često podaci mijenjaju?
  - Koliko su podaci važni za poslovanje?
  - Koliko je loše ako određene podatke izgubimo?
  - Koliko je skupo vrijeme nedostupnosti zbog oporavka?
  - Mogu li se podaci lako rekreirati iz nekih drugih izvora?
  - Postoji li vrijeme kad baza može biti nedostupna?

- **Nakon analize, treba postaviti odgovarajuću strategiju izrade sigurnosnih kopija**

Savjeti pri izradi sigurnosnih kopija

- **Da bismo bili što sigurniji da ćemo bazu moći oporaviti iz sigurnosnih kopija, poželjno je:**
  - Provjeriti je li kopija ispravna izvođenjem oporavka
  - Kopirati sigurnosnu datoteku na drugi disk
  - Čuvati barem dvije generacije sigurnosnih kopija
  - Ako sigurnosnu kopiju kopiramo na traku, zadržati je i na disku
  - Kopirati i datoteke koje su dio baze, ali je u bazi navedena samo putanja do njih
  - Kopirati sistemske baze i datoteke
  - Napraviti potpunu kopiju nakon što se napravi oporavak do odabranog trenutka

Tema: Oporavak baza

Sadržaj

- Način rada transakcijskog loga
- Proces oporavka baza
- Tipovi oporavka

Način rada transakcijskog loga (1)

- Moderni DBMS-i imaju tzv. ***write-ahead* log**
  - Promjene na bazi prvo se zabilježe u log, a tek kasnije u podatkovnu datoteku
  - Kod promjene podataka događa se sljedeće:
    - DBMS učita odgovarajuću stranicu u međuspremnik
    - Podaci se promijene u toj stranici u međuspremniku
    - Promjena se zabilježi u log datoteci na disku
    - Stranica u međuspremniku razlikuje se od stranice u podatkovnoj datoteci (**prljava stranica – *dirty page***)
    - DBMS u nekom trenutku okida ***checkpoint*** – promjene iz međuspremnika propagiraju se u podatkovnu datoteku (***flushing***)

Način rada transakcijskog loga (2)

- **Ako odaberemo neki proizvoljni trenutak, možemo zateći sljedeće stanje:**
  - Neke transakcije mogu biti potvrđene, ali njihove promjene još nisu propagirane do podatkovne datoteke
    - Promjene su zabilježene na stranicama u međuspremniku
    - Još se nije dogodio *checkpoint* koji bi napravio potrebni *flush*
  - Neke transakcije se još uvijek izvode, a promjene su već propagirane do podatkovne datoteke
    - *Checkpoint* se dogodio za vrijeme izvođenja transakcije
    - Podaci su *flushani* u podatkovnu datoteku
    - Budući da transakcije nisu potvrđene, podaci u podatkovnoj datoteci nisu “konačni”

Način rada transakcijskog loga (3)

- Primjer: zbog nestanka struje baza je pala
  - U ponovnom dizanju baze on-line potrebno je dovesti bazu u konzistentno stanje:
    - Ponoviti transakciju Tran2
      - Ona je potvrđena tek nakon zadnjeg *checkpointa*
    - Poništiti transakciju Tran3
      - Ona nije bila potvrđena

Proces oporavka baza (1)

- **Oporavak baze (*database recovery*)** – vraćanje baze i podataka u njoj u stanje kakvo je bilo u nekom trenutku u prošlosti
  - Taj se trenutak naziva **točkom oporavka (*recovery point*)**

- Oporavak mora završiti dovođenjem baze u konzistentno stanje

- Oporavak baze ne mora nužno značiti da se dogodio neki problem s bazom
  - Dizanje baze on-line nakon restartanja servera također se naziva oporavkom – oporavak bez korištenja sigurnosnih kopija

Proces oporavka baza (2)

- Faze u oporavku baza iz sigurnosnih kopija:
  - **Restauriranje (*restore*)** baze
    - Baza se izgradi prema sadržaju sigurnosne kopije
  - Dovođenje baze u konzistentno stanje
    - ***Roll forward*** – transakcije koje su potvrđene, ali se njihove promjene ne nalaze u podatkovnim datotekama, moraju se ponovo izvesti
    - ***Roll back*** – transakcije koje nisu potvrđene, a neke su promjene već prenesene u podatkovne datoteke, moraju se poništiti

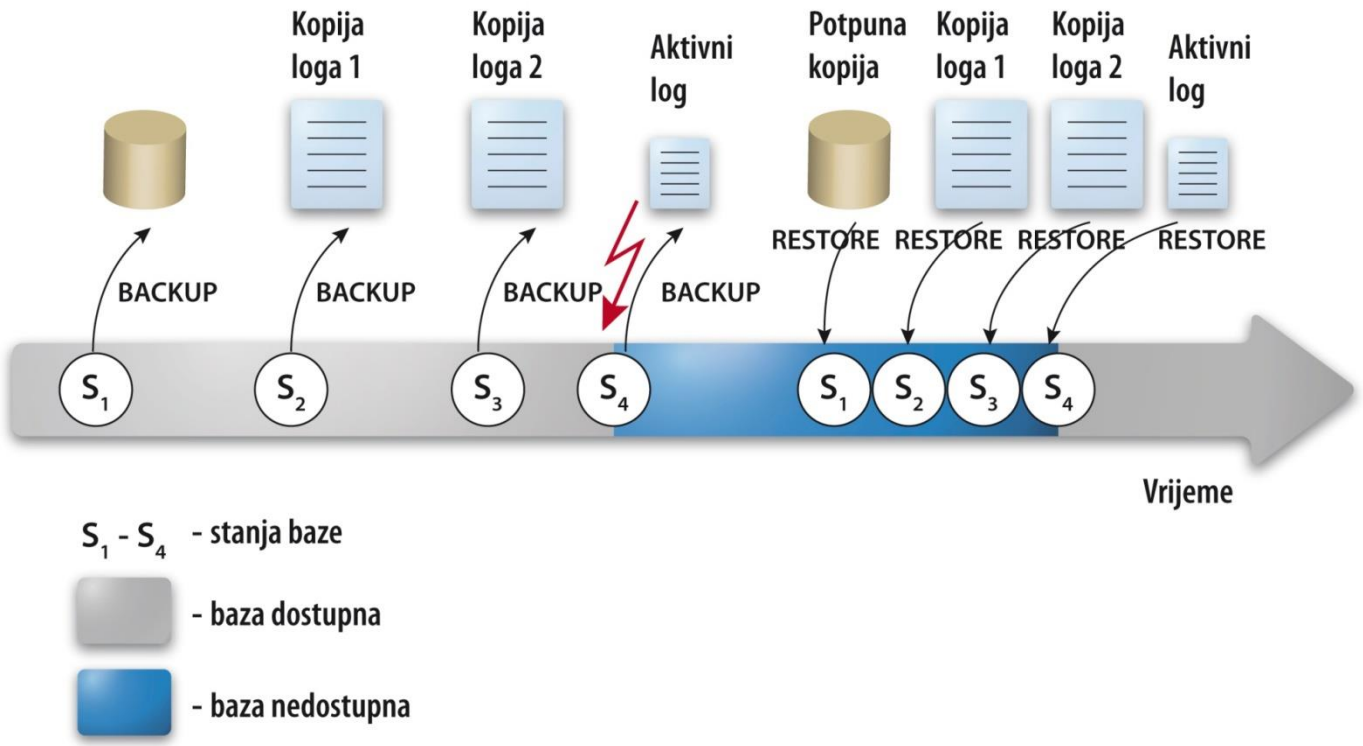
Tipovi oporavka

- Najčešći tipovi oporavka:
  - Oporavak do točke pada
  - Oporavak do odabranog trenutka

Oporavak do točke pada (1)

- **Izvodi se kad je bazu potrebno vratiti u stanje neposredno prije pada**
  - *Recover to current, point of failure recovery*
- **Tijek oporavka:**
  - Restauriranje baze iz potpune kopije
  - Restauriranje iz zadnje diferencijalne kopije
  - Roll forward iz arhiviranih transakcijskih logova
  - Roll forward iz aktivnog loga
  - Roll back transakcija koje nisu potvrđene
- **Ne može se napraviti ako nemamo potpunu kopiju**
- **Ne može se napraviti ako je uništen aktivni log**

Oporavak do točke pada (2)



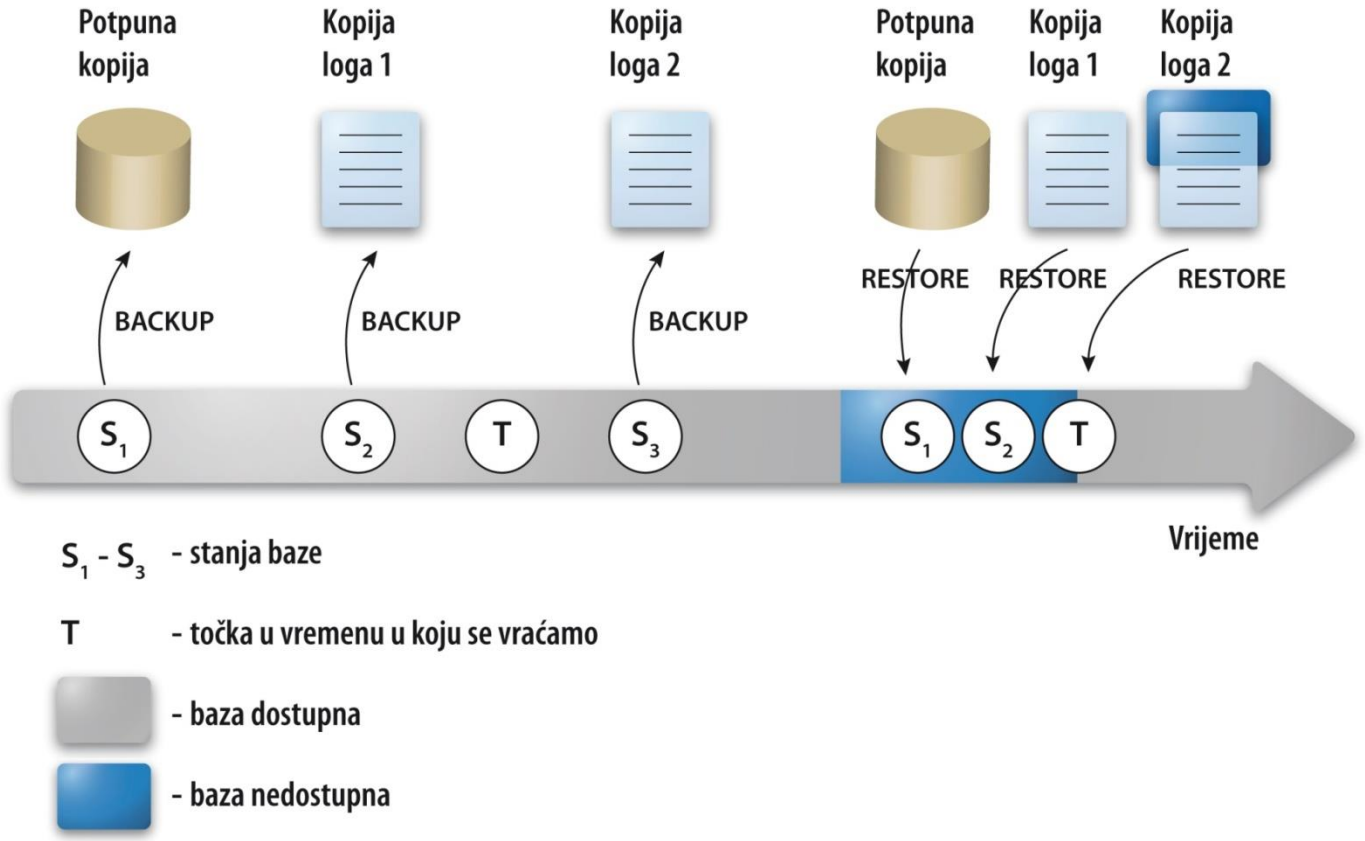
Oporavak do odabranog trenutka (1)

- **Vraćanje baze u stanje kakvo je bilo u odabranom trenutku u prošlosti**
  - *Point-in-time recovery*
  - Obično se primjenjuje kad se naprave kriva ažuriranja ili brisanja podataka koja nije lako na drugačiji način ispraviti
  - Točka oporavka može biti zadana:
    - Datumom i vremenom
    - Preko adrese u transakcijskom logu

Oporavak do odabranog trenutka (2)

- **Tok oporavka:**
  - Restauriranje baze iz potpune kopije
  - Restauriranje iz zadnje diferencijalne kopije
  - Roll forward i roll back iz arhiviranih logova
- **Oporavak do točke pada je specijalan slučaj oporavka do odabranog trenutka**
- **Ovisno o tome koji je način rada transakcijskog loga odabran, neće uvijek biti moguće napraviti oporavak do odabranog trenutka**

Oporavak do odabranog trenutka (3)



Plan oporavka baze

- **Za oporavak baza treba napraviti plan**
- **U plan je potrebno uključiti:**
  - Detaljnu razradu svakog koraka oporavka za sve baze
  - Skripte za oporavak
- **Plan treba testirati**
  - Testiranje je poželjno provoditi redovito
  - Testiranje će administratoru baze biti svojevrstan trening da bude što spremniji kad se dogodi potreba za oporavkom

Tema: Sigurnosne kopije i oporavak baza na SQL Serveru

*Detalji naredbi će biti obrađeni na vježbama

Sadržaj

- Tipovi sigurnosnih kopija na SQL Serveru
- Modeli oporavka
- Strategije izrade sigurnosnih kopija
- Izrada sigurnosnih kopija
- Restauriranje baza

Tipovi sigurnosnih kopija na SQL Serveru

- Potpune kopije (*full backups*)
- Kopije transakcijskog loga (*transaction log backups*)
- Kopije kraja loga (*tail-log backups*)
- Diferencijalne kopije (*differential backups*)
- Obične kopije
- Kopije datoteka ili grupa (*file/filegroup backups*)

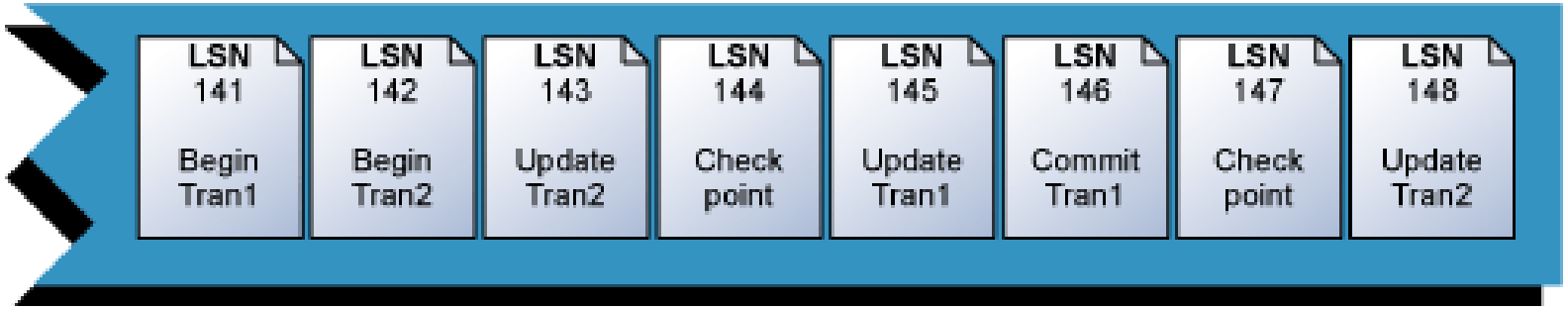
Potpune kopije

- **Kopiraju se svi podaci iz podatkovnih datoteka**
  - Kopira se i dovoljan dio loga kako bi se baza mogla oporaviti u trenutak završetka izrade kopije

- **Baza oporavljena iz potpune kopije:**
  - Odražava stanje baze u trenutku završetka izrade potpune kopije
  - Ne sadrži transakcije koje u tom trenutku nisu bile potvrđene

Kopije transakcijskog loga (1)

Primjer: Izgled transakcijskog loga u nekom trenutku



- **Svaki zapis ima svoj redni broj – LSN (*log sequence number*)**
- **Nakon *checkpointa* (LSN 147) zapisi o Tran1 više nisu potrebni jer su propagirani na disk i transakcija je potvrđena**
- **Zapisi o Tran2 još su uvijek potrebni jer su se neki podaci već upisali u podatkovnu datoteku, a transakcija još nije potvrđena**
- **LSN 142 – minimum recovery LSN (MinLSN)**
  - Zapisi lijevo od njega čine **neaktivni dio loga** – nisu više potrebni za oporavak baze
  - Ostatak loga je **aktivni dio loga** – zapisi potrebni za oporavak baze

Kopije transakcijskog loga (2)

- **Sigurnosna kopija transakcijskog loga**
  - Sadrži zapise iz loga koji dotad nisu bili kopirani kroz log backup (od prvog nekopiranog LSN-a do zadnjeg LSN-a)
- **Nakon izrade kopije loga, neaktivni dio loga označava se slobodnim za ponovno korištenje**
  - Ta se operacija naziva **truncate loga**
  - Truncate loga ne smanjuje fizičku veličinu log datoteke!
- **Kopija loga neke baze ne može se napraviti ako nikad nije napravljena potpuna kopija te baze**

Kopije kraja loga

- Sadrži sve LSN-ove koji još nisu bili kopirani
- Koristi se pri oporavku baze do točke pada ako je podatkovna datoteka oštećena, a transakcijski log ispravan
- Za razliku od "običnih" kopija loga, one ne okidaju truncate loga

Diferencijalne kopije

- Obuhvaćene su samo promjene od izrade zadnje potpune kopije
- Ne mogu se napraviti ako nikad nije napravljena potpuna kopija baze

Obične kopije

- **Kod oporavka baze, moguće je da će biti potrebne različite sigurnosne kopije, koje su “ulančane”**
- **U nekim slučajevima, neplanirana izrada sigurnosne kopije može prekinuti uspostavljeni lanac kopija**
- **Obične kopije ne prekidaju taj lanac**
- **Tipovi običnih kopija:**
  - Potpuna obična kopija
  - Diferencijalna obična kopija
  - Obična kopija loga

Modeli oporavka

- **Modeli oporavka (*recovery models*) – načini rada transakcijskog loga**
  - Simple
  - Full
  - Bulk-logged

- **Razlike:**
  - Detaljnost bilježenja promjena na bazi
  - Stupanj mogućeg oporavka baze

Simple recovery model (1)

- **Promjene u transakcijskom logu ne bilježe se detaljno**
  - Samo koliko je nužno za oporavak baze u slučaju pada
- **Nije moguće napraviti sigurnosnu kopiju loga**
- **Oporavak baze ograničen je na točku u kojoj je završila izrada potpune ili diferencijalne kopije**

Simple recovery model (2)

- **Truncate loga događa se kod svakog okidanja checkpointa**
  - Zbog toga u simple recovery modelu obično nema problema s veličinom log datoteke

- **Pogodan u slučajevima kad nije potrebna mogućnost oporavka do odabranog trenutka**
  - Npr. razvojne baze

Full recovery model

- **Sve se promjene maksimalno detaljno zapisuju u log**
- **Postoji mogućnost oporavka do odabranog trenutka**
- **Truncate loga ne događa se prilikom checkpointa, već samo kod izrade kopije loga**
  - Potrebno uspostaviti odgovarajuću strategiju izrade sigurnosnih kopija, koja mora uključivati i izradu kopija loga
  - Ako se log ne kopira, može previše narasti
- **Treba ga koristiti ako je oporavak u bilo koju točku u vremenu najviši prioritet**

Bulk-logged recovery model

- **Sličan full recovery modelu, ali neke promjene ne bilježi tako detaljno:**
  - Bulk load operacije
  - SELECT INTO
  - CREATE INDEX

- **Oporavak do odabranog trenutka nije uvijek moguć**

Brzi rast transakcijskog loga (1)

- **Čest problem kod "administratora silom prilika"**
  - Bazu drže u full recovery modelu, a ne backupiraju log

- **Preporuke:**
  - Full recovery model → treba backupirati log
  - Nije važna mogućnost oporavka do bilo kojeg odabranog trenutka → simple recovery model

Brzi rast transakcijskog loga (2)

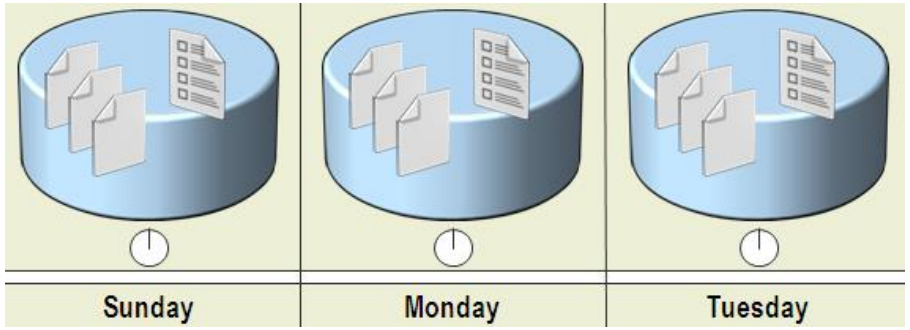
- Kad log toliko naraste da je disk gotovo pun, "vatrogasno" rješenje je:

1. `USE master`
2. `ALTER DATABASE <naziv_baze> SET RECOVERY SIMPLE`
3. `USE <naziv_baze>`
4. `DBCC SHRINKFILE(<naziv_log_datoteke>)`
5. `ALTER DATABASE <naziv_baze> SET RECOVERY FULL`
6. Napraviti full backup baze

Strategije izrade sigurnosnih kopija

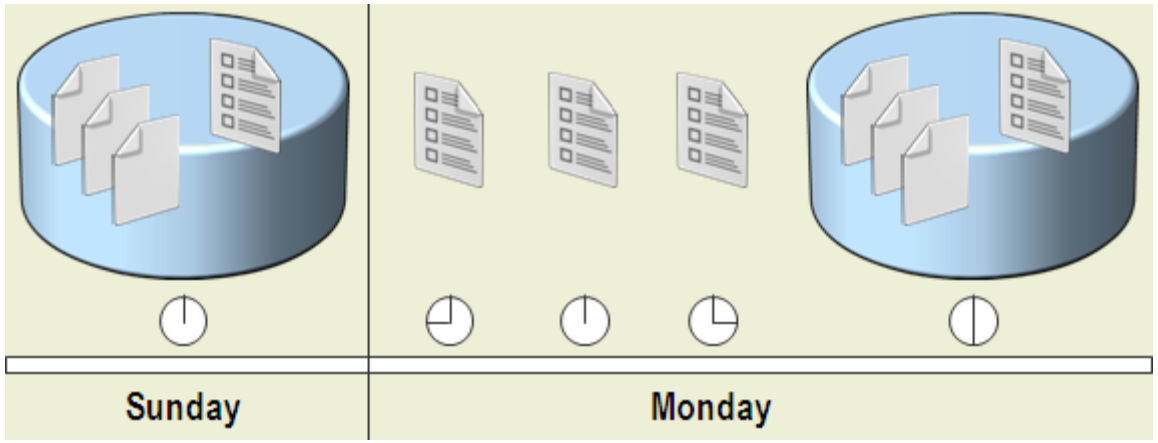
- **Najčešće strategije:**
  - Izrada samo potpunih sigurnosnih kopija
  - Izrada potpunih kopija i kopija loga
  - Strategija koja uključuje diferencijalne kopije

Izrada samo potpunih sigurnosnih kopija



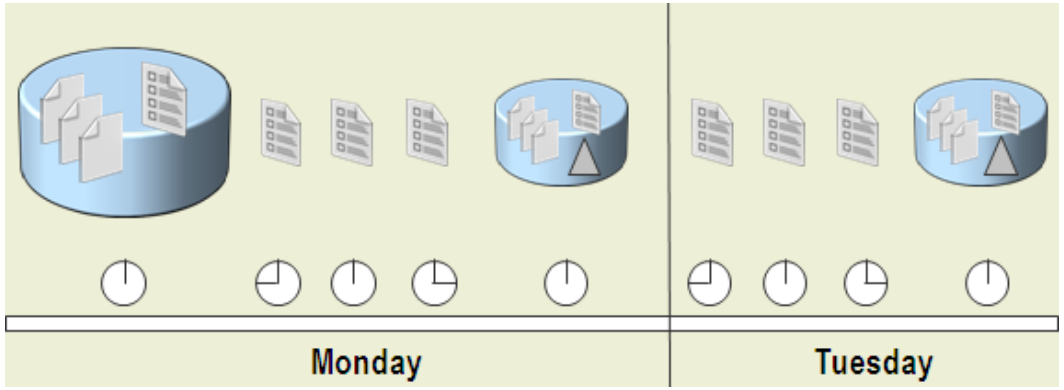
- **Prikladna ako su baze male i prihvatljiv je manji gubitak podataka**
  - Uz takvu strategiju prikladan je simple recovery model
    - Moguć oporavak samo u točke kad su izrađeni backupi
  - Samo full backupi + full recovery model → opasnost od brzog rasta loga

Izrada potpunih kopija i kopija loga



- Prikladno kad je nužna mogućnost oporavka do odabranog trenutka
- Potpune kopije se izrađuju kad nema velikih aktivnosti na bazi, a inače se izrađuju kopije loga
- Postupak oporavka:
  - Restaurira se potpuna kopija, a zatim redom kopije loga

Strategija koja uključuje diferencijalne kopije



- **Izrada potpunih kopija zamijeni se diferencijalnima**
  - Pogodno ako nema dovoljno prostora ili vremena za potpune kopije
- **Oporavak se usložnjava**
  - Restaurira se potpuna kopija, zadnja diferencijalna kopija i redom kopije loga

Izrada sigurnosnih kopija (1)

- **Kreiranje stalnog backup devicea:**

```
EXEC sp_addumpdevice
   'disk',
   'AdventureWorksBack', 'D:\MyBackupDir\AdventureWorksBack.bak'
```

- **Izrada potpune sigurnosne kopije:**

```
BACKUP DATABASE AdventureWorks
   TO AdventureWorksBack
   WITH STATS
```

Izrada sigurnosnih kopija (3)

- **Izrada sigurnosne kopije loga**

```
BACKUP LOG AdventureWorks
  TO AdventureWorksBack
```

- **Izrada sigurnosne kopije kraja loga**

```
BACKUP LOG AdventureWorks
  TO 'C:\Backup\AWTail.bak'
  WITH NORECOVERY, NO_TRUNCATE
```

  - NORECOVERY – stavlja bazu u stanje "restoring" pa se nad njom više ne mogu raditi promjene
  - NO_TRUNCATE – ne smije se dogoditi truncate loga

Izrada sigurnosnih kopija (4)

- Izrada diferencijalne kopije

```
BACKUP DATABASE AdventureWorks TO
  DISK = 'D:\MyData\MyDiffBackup.bak'
  WITH DIFFERENTIAL
```

Restauriranje baza (1)

- Prikaz sigurnosnih kopija u backup deviceu

```
RESTORE HEADERONLY
  FROM DISK = 'C:\AdventureWorks.BAK'
```

- Prikaz datoteka unutar sigurnosne kopije

```
RESTORE FILELISTONLY
  FROM DISK = 'C:\AdventureWorks.BAK' WITH FILE = 2
```

- Provjera ispravnosti sigurnosne kopije

```
RESTORE VERIFYONLY
  FROM DISK = 'C:\AdventureWorks.BAK' WITH FILE = 2
```

Restauriranje baza (2)

- **Restauriranje potpune kopije**

```
USE master
RESTORE DATABASE AdventureWorks
FROM disk = 'C:\AW.bak'
WITH
  REPLACE,
  MOVE 'AdventureWorks_Data' to 'D:\AdventureWorks.mdf',
  MOVE 'AdventureWorks_Log' to 'E:\AdventureWorks.ldf'
```

  - REPLACE – ako baza već postoji, pregazi se
  - MOVE – određivanje putanje za datoteke baze

Restauriranje baza (3)

- **Restauriranje diferencijalne kopije**

```
USE master
RESTORE DATABASE AdventureWorks
FROM AWBack
WITH NORECOVERY

RESTORE DATABASE AdventureWorks
FROM AWBackDiff
WITH RECOVERY
```

  - NORECOVERY– restauriramo sigurnosnu kopiju i namjeravamo na nju kasnije nadovezati još jednu
  - RECOVERY – nakon restauracije, dižemo bazu on-line

Restauriranje baza (4)

- Restauriranje kopije loga

```
USE master
RESTORE DATABASE AdventureWorks FROM disk = 'C:\AW.bak'
WITH NORECOVERY

RESTORE LOG AdventureWorks FROM disk = 'C:\AW1.trn'
WITH NORECOVERY

RESTORE LOG AdventureWorks FROM disk = 'C:\AW2.trn'
WITH RECOVERY
```