



UNIVERZITET U NIŠU  
ELEKTRONSKI FAKULTET  
Katedra za računarstvo



## **Pravljenje rezervnih kopija i oporavak baza podataka**

### **Sistemi za upravljanje bazama podataka**

**Profesor:** Doc. dr Aleksandar Stanimirović

**Student:** Minja Ignjatović 1000

Uvod	<b>3</b>
1. Tipovi rezervnih kopija	<b>4</b>
1.1. Puna rezervna kopija	4
1.2. Diferencijalne rezervne kopije	6
1.3. Dnevnik transakcija	7
1.4. <i>Tail-log</i> rezervne kopije	7
1.5. Rezervna kopija samo za kopiranje	8
1.6. Parcijalne rezervne kopije	8
1.7. Rezervne kopije fajlova i fajl grupa	9
2. Modeli oporavka	<b>11</b>
2.1. <i>SIMPLE</i> model oporavka	11
2.2. <i>FULL</i> model oporavka	12
2.3. <i>BULK_LOGGED</i> model oporavka	13
2.4. Promena modela oporavka	13
3. Strategije za pravljenje rezervnih kopija i oporavak nakon otkaza	<b>15</b>
3.1. Najbolja praksa za pravljenje strategija	15
3.2. Kreiranje zakazanih backup-ova korišćenjem planova održavanja	16
3.3. Kreiranje zakazanih backup-ova korišćenjem SQL Server agenta	19
Zaključak	<b>20</b>
Reference	<b>21</b>

# Uvod

Pravljenje rezervnih kopija predstavlja neophodan deo svakog dobrog plana za oporavak od katastrofa. Skladišteni podaci moraju biti zaštićeni i sačuvani i u slučaju najgorih scenarija kako ne bi došlo do trajnog gubitka podataka. Postoje različite situacije u kojima može doći do gubitka podataka. Iako možemo proaktivno reagovati kako bi se osigurali da su otkazi svedeni na minimum, ne postoji garancija da do njih neće doći. Upravo zbog toga, potrebno je obezbediti *backup* i *restore* strategije da bi se obezbedio mehanizam za oporavak u slučaju katastrofe.

Postoji više tipova otkaza koji se mogu javiti. Hardverski otkazi se tiču otkaza instance SQL Servera, mrežnih otkaza, otkaza diska. Iako su ovakvi otkazi uglavnom svedeni na minimum, oni se mogu javiti u situacijama kada npr kontroler na disku otkaze ili se podaci na disku oštete. Pored ovih otkaza, mogu se javiti i sistemski ili softverski otkazi. Oni se tiču grešaka u operativnom sistemu, CPU otkaza ili otkaza glavne memorije kompjuterskog sistema. Otkaz aplikacije tiče se neuspešne aplikacije, primer može biti pogrešno ažuriranje ili brisanje podataka. Poseban tip otkaza su otkazi koji su nastali usled ljudske greške. To se može desiti ukoliko korisnik umesto jednog zapisa u tabeli, obriše čitavu tabelu.

U nekim slučajevima, postoje određena *third-party* rešenja koja pružaju oporavak transakcije ili grupa transakcija na osnovu log fajla, međutim u određenim slučajevima kada je sam disk oštećen potrebno je osloniti se na *backup* fajlove.

U radu će biti predstavljeni različiti tipovi pravljenja rezervnih kopija, uz praktične primere. Pored toga, biće razmatrane opcije za automatsko pravljenje rezervnih kopija i predstavljeni modeli oporavka. Korišćeni DBMS je *Microsoft Sql Server*. Praktični primeri su prikazani korišćenjem *AdventureWorks 2017* baze podataka čija kopija je javno dostupna u okviru *Microsoft* dokumentacije.

# 1. Tipovi rezervnih kopija

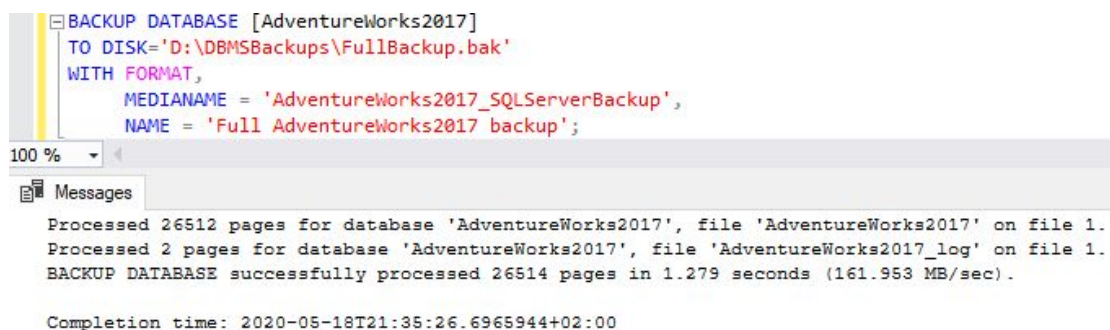
U zavisnosti od korisničkih potreba moguće je izabrati različite tipove *backup*-ova. Ukoliko je gubitak podataka u određenoj meri prihvatljiv, dovoljno je koristiti jednostavan proces pravljenja rezervnih kopija podataka. Međutim, ukoliko gubitak podataka nije prihvatljiv, potrebno je napraviti sistem takav da postoji oporavak podataka koji su bili dostupni u trenutku otkaza. Neki od tipova pravljenja *backup* fajlova su:

1. Puna rezervna kopija
2. Diferencijalna rezervna kopija
3. Dnevnik transakcija
4. *Tail Log* rezervne kopije
5. Rezervna kopija samo za kopiranje
6. Parcijalne rezervne kopije
7. Rezervne kopije fajlova i fajl grupa

## 1.1. Puna rezervna kopija

Puna rezervna kopija (eng. *Full backup*) pravi rezervnu kopiju svega. Predstavlja osnovu svih ostalih tipova i sadrži celokupnu kopiju koja čuva sve objekte baze podataka. Sadrži tabele, procedure, funkcije, poglede, indekse itd. Pomoću ovakvog tipa *backup*-a moguće je lako vratiti bazu podataka u isti oblik u kom se nalazila u trenutku kad je rezervna kopija napravljena. Pored pomenutih objekata, pravi se kopija transakcionih logova kako bi baza podataka mogla biti vraćena. Potpuni *backup* mora biti izvršen najmanje jednom pre nego što se bilo koji od ostalih tipova može izvršiti.

Na Slici 1 prikazana je komanda za izvršenje celokupnog *backup*-a baze podataka.



```
BACKUP DATABASE [AdventureWorks2017]
TO DISK='D:\DBMSBackups\FullBackup.bak'
WITH FORMAT,
MEDIANAME = 'AdventureWorks2017_SQLServerBackup',
NAME = 'Full AdventureWorks2017 backup';
```

100 %

Messages

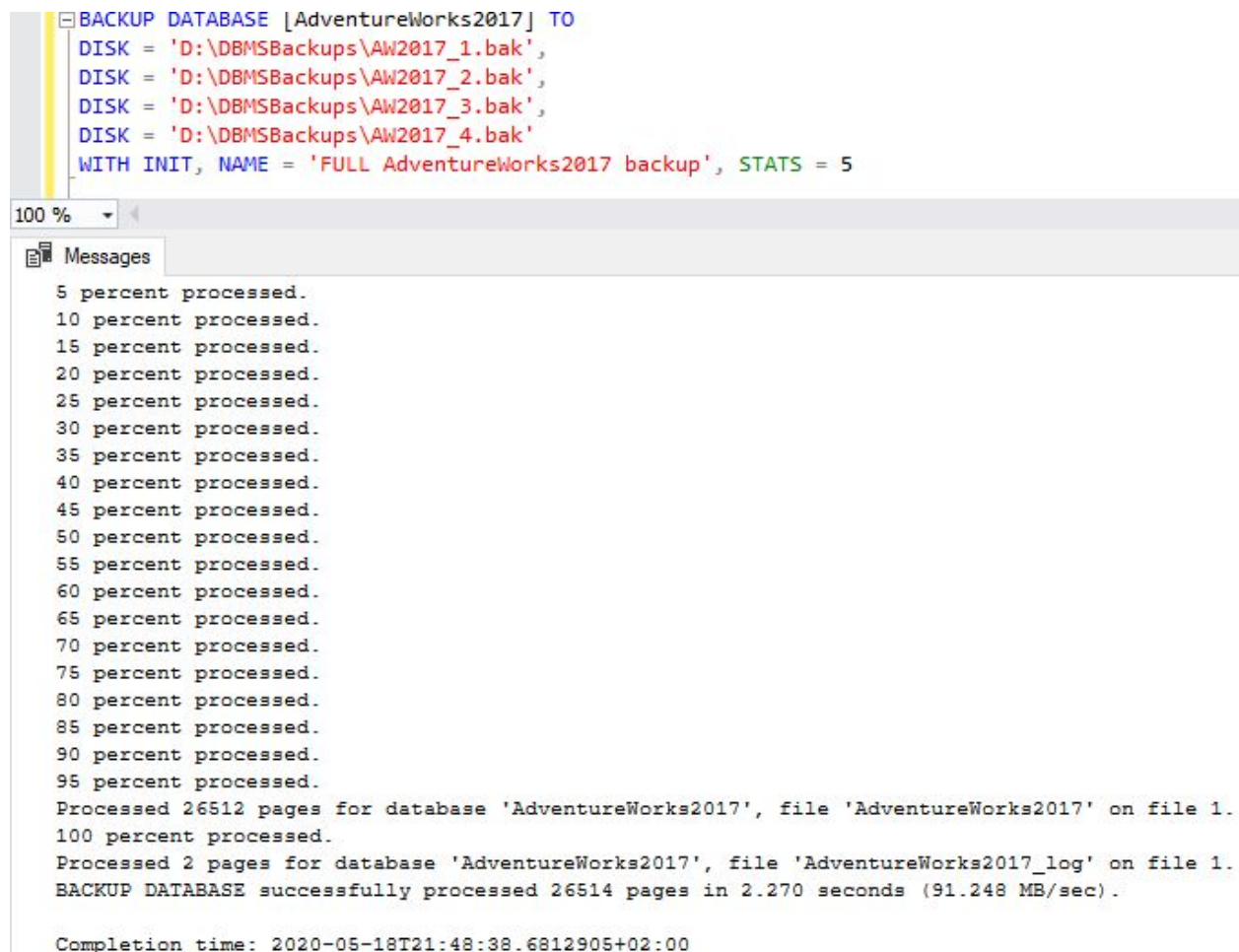
Processed 26512 pages for database 'AdventureWorks2017', file 'AdventureWorks2017' on file 1.  
Processed 2 pages for database 'AdventureWorks2017', file 'AdventureWorks2017\_log' on file 1.  
BACKUP DATABASE successfully processed 26514 pages in 1.279 seconds (161.953 MB/sec).

Completion time: 2020-05-18T21:35:26.6965944+02:00

Slika 1 - Komanda za izvršenje *full backup*-a

Parametri koje je potrebno navesti kod komande su najpre lokacija gde će se nalaziti kreirani *backup* fajl a zatim opcija *FORMAT* koja određuje da li zaglavlje medija treba da se piše za ovaj postupak pravljenja kopija, prepisujući bilo koje postojeće zaglavlje. *NAME* referencira specifični *backup* skup a *MEDIANAME* je labela za *backup* uređaj.

Postoje slučajevi kada je potrebno napraviti rezervnu kopiju baze podataka koja zauzima dosta memorijskog prostora. Da bi to bilo moguće, postoji opcija deljenja *backup* fajla u više manjih celina od koje svaka predstavlja zasebni fajl. Komanda je prikazana na Slici 2.



```
BACKUP DATABASE [AdventureWorks2017] TO
DISK = 'D:\DBMSBackups\AW2017_1.bak',
DISK = 'D:\DBMSBackups\AW2017_2.bak',
DISK = 'D:\DBMSBackups\AW2017_3.bak',
DISK = 'D:\DBMSBackups\AW2017_4.bak'
WITH INIT, NAME = 'FULL AdventureWorks2017 backup', STATS = 5
```

100 %

Messages

5 percent processed.  
10 percent processed.  
15 percent processed.  
20 percent processed.  
25 percent processed.  
30 percent processed.  
35 percent processed.  
40 percent processed.  
45 percent processed.  
50 percent processed.  
55 percent processed.  
60 percent processed.  
65 percent processed.  
70 percent processed.  
75 percent processed.  
80 percent processed.  
85 percent processed.  
90 percent processed.  
95 percent processed.  
Processed 26512 pages for database 'AdventureWorks2017', file 'AdventureWorks2017' on file 1.  
100 percent processed.  
Processed 2 pages for database 'AdventureWorks2017', file 'AdventureWorks2017\_log' on file 1.  
BACKUP DATABASE successfully processed 26514 pages in 2.270 seconds (91.248 MB/sec).  
  
Completion time: 2020-05-18T21:48:38.6812905+02:00

Slika 2 - *Full backup* sa više fajlova

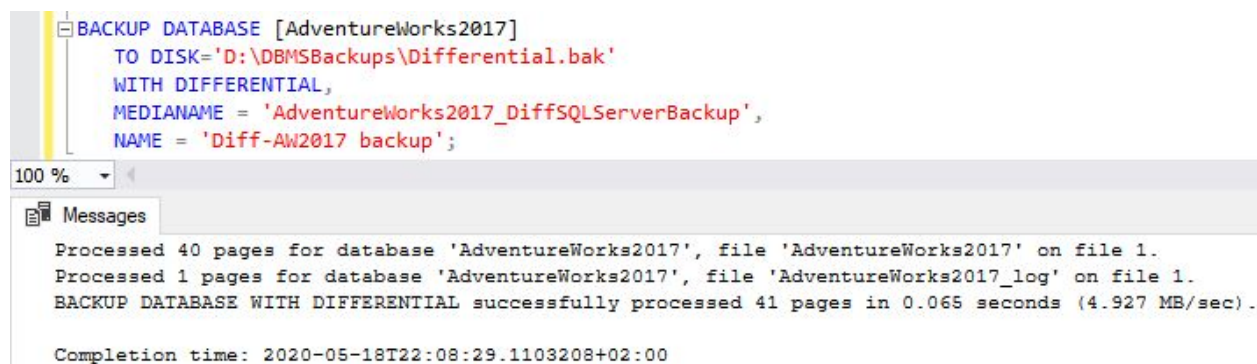
## 1.2. Diferencijalne rezervne kopije

Diferencijalni *backup* obuhvata samo opseg tj zbirke od osam fizički povezanih stranica, koje su izmenjene od poslednje potpune rezervne kopije baze podataka. SQL Server pregledava stranicu *Differential Changed Map* (DCM) kako bi identifikovao sve ekstenzije koje su promenjene od poslednje sigurnosne kopije baze podataka i snima samo njih. Dakle, diferencijalno sigurnosno kopiranje je često veoma brzo u poređenju sa pravljenjem potpune rezervne kopije podataka.

DCM je bitmapa koja sadrži po jedan bit za svaki opseg i prati samo one koji su promenjeni od poslednje potpune rezervne kopije. Prilikom samog pravljenja kopije, SQL Server identifikuje sve opsege za koje bit ima vrednost 1.

Poslednja rezervna kopija je obavezna pre samog počinjanja diferencijalne rezervne kopije. Na potpunoj kopiji se temelji diferencijalna sigurnosna kopija. Uopšteno važi da što je starija diferencijalna kopija to će biti nova veća kopija jer se razlika između trenutne baze podataka i specifične diferencijalne rezervne baze povećava. Ovu osobinu je potrebno uzeti u obzir jer velika razlika povećava vreme za obnavljanje baze podataka. Kako bi se to prevazišlo, preporučuje se da se potpuna rezervna kopija pravi redovno, u određenim intervalima kako bi se uspostavila nova diferencijalna kopija. Primer dobre prakse je da se potpuno sigurnosno kopiranje obavlja jednom nedeljno a da se zatim redovno u toku nedelje obavlja diferencijalno pravljenje rezervnih kopija. Naravno, za donošenje odluke je potrebno uzeti u obzir specifičnost situacije.

Na Slici 3 prikazana je komanda za kreiranje diferencijalne rezervne kopije. Karakteristični parametri koje je potrebno navesti su: ime baze podataka, *backup device* i klauzulu *DIFFERENTIAL*.



```
BACKUP DATABASE [AdventureWorks2017]
  TO DISK='D:\DBMSBackups\Differential.bak'
  WITH DIFFERENTIAL,
  MEDIANAME = 'AdventureWorks2017_DiffSQLServerBackup',
  NAME = 'Diff-AW2017 backup';
```

100 %

Messages

Processed 40 pages for database 'AdventureWorks2017', file 'AdventureWorks2017' on file 1.  
Processed 1 pages for database 'AdventureWorks2017', file 'AdventureWorks2017\_log' on file 1.  
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 41 pages in 0.065 seconds (4.927 MB/sec).

Completion time: 2020-05-18T22:08:29.1103208+02:00

Slika 3 - Kreiranje diferencijalne rezervne kopije baze podataka

### 1.3. Dnevnik transakcija

Dnevnik transakcija (eng. *Transaction Log Backup*) je podržan samo s modelom potpunog oporavka ili s modelom oporavka sa velikim brojem zapisa. Kao i diferencijalne sigurnosne kopije baze podataka, za pravljenje rezervnih kopija dnevnika transakcija neophodno je imati potpunu kopiju baze podataka ili poslednje sigurnosne kopije dnevnika transakcija. Preporučuje se da se rezervne kopije prave dovoljno često kako bi se smanjila mogućnost gubitka podataka i skratio dnevnik transakcija kako ne bi narastao. Postoje slučajevi kada nije moguće skratiti dnevnik transakcija. Jedan od slučajeva je kada je navedena opcija *AlwaysOn, Database Mirroring, Transactional replication*. Za razliku od diferencijalnih rezervnih kopija koje pamte samo poslednje promene kada je red menjan više puta, transakcioni logovi čuvaju sve promene. Takođe, dnevnici transakcija su inkrementalni i kumulativni. To znači da, ukoliko želimo da povratimo bazu podataka na neko određeno stanje u vremenu, potrebno je da izvršimo potpuni oporavak baze podataka, zatim diferencijalni i na kraju sve odgovarajuće transakcije koje garantuju da će baza podataka biti u željenom stanju. Na Slici 4 je prikazan način kreiranja rezervne kopije dnevnika transakcija.

```
BACKUP LOG [AdventureWorks2017]
TO DISK='D:\DBMSBackups\TransactionalLog.bak'
WITH
MEDIANAME = 'AW2017_SQLServerLogBackup',
NAME = 'Log-AW2017 backup';
GO
```

Slika 4 - Kreiranje rezervne kopije transakcionih logova

### 1.4. *Tail-log* rezervne kopije

U situacijama kada se rezervne kopije kreiraju na svakih sat vremena, može doći do problema ukoliko u periodu između pravljenja dve kopije dođe do otkaza. U tim situacijama *tail-log* rezervne kopije mogu biti od koristi. Ukoliko se baza podataka ošteti ali je dnevnik transakcija dostupan, treba napraviti rezervnu kopiju evidencije koja beleži sve zapise dnevnika koji još nisu sigurnosno kopirani (rep zapisnik) kako bi se sprečio gubitak podataka i lanac transakcija održao netaknut. Ovaj način je moguć samo ukoliko fajl dnevnika transakcija nije oštećen i ukoliko je baza podataka u stanju koji podržava sigurnosne kopije replika. *Tail-log backup* je potrebno izvršiti pre početka oporavka baze podataka i on je moguć ukoliko baza

podataka radi u *FULL* ili *BULK\_LOGGED* modelu oporavka. Na Slici 5 prikazan je način formiranja pomenute rezervne kopije.

```
BACKUP LOG [AdventureWorks2017]
TO DISK = 'D:\DBMSBackups\TailLog.log'
WITH CONTINUE_AFTER_ERROR;
GO
```

Slika 5 - Kreiranje *Tail-Log backup* fajla

## 1.5. Rezervna kopija samo za kopiranje

Rezervna kopija samo za kopiranje (eng. *Copy-only backup*) je nezavisna od konvencionalne sekvence rezervnih kopija. Razlika između ove rezervne kopije i potpune rezervne kopije je ta što se *Copy-only* rezervna kopija ne koristi kao osnova za sledeću diferencijalnu rezervnu kopiju. Potpuna rezervna kopija funkcioniše na svim modelima oporavka baze podataka dok *Copy-only* rezervna kopija funkcioniše sa *FULL* ili *BULK-LOGGED* modelima oporavka. Na Slici 6 prikazan je način formiranja *Copy-only backup* fajla.

```
BACKUP DATABASE [AdventureWorks2017]
TO DISK='D:\DBMSBackups\CopyOnly.bak'
WITH COPY_ONLY,
    MEDIANAME = 'Native_SQLServerFullBackup',
    NAME = 'Full-AW2017 backup';
BACKUP LOG [AdventureWorks2017]
TO DISK = 'D:\DBMSBackups\AW2017CopyOnly.log'
WITH COPY_ONLY;
GO
```

Slika 6 - Formiranje *Copy-only* rezervne kopije

## 1.6. Parcijalne rezervne kopije

Parcijalne rezervne kopije su najmanje korišćeni metod koji je dostupan u okviru SQL Servera. Svi modeli oporavka podržavaju parcijalne rezervne kopije, s tim što se najčešće koriste sa *simple* modelom oporavka kako bi se unapredila fleksibilnost prilikom pravljanja rezervnih kopija baza podataka koje zauzimaju dosta memorijskog prostora i sadrže *read-only* fajl grupe.



## 1.7. Rezervne kopije fajlova i fajl grupa

Rezervne kopije fajlova i fajl grupa imaju smisla u SQL Server bazama podataka koje sadrže više fajlova ili grupa fajlova. Pravljenje rezervnih kopija *read-only* fajl grupa može se kombinovati sa parcijalnim rezervnim kopijama. Parcijalne rezervne kopije uključuju sve *read* i *write* fajl grupe i, opciono, jednu ili više *read-only* fajl grupa.

Komanda sa Slike 7 kreira bazu podataka sa više fajlova i fajl grupa koja će služiti kao osnova za kreiranje rezervnih kopija.

```
CREATE DATABASE SQLFileBackup ON PRIMARY
( NAME = N'SQLFileBackup_1',
  FILENAME = N'D:\DBMSBackups\SQLFileBackup_1.mdf' ,
  SIZE = 5000KB , FILEGROWTH = 1024KB ),
FILEGROUP [Secondary]
( NAME = N'SQLFileBackup_2',
  FILENAME = N'D:\DBMSBackups\SQLShackBackup_2.ndf' ,
  SIZE = 5000KB , FILEGROWTH = 1024KB )
LOG ON
( NAME = N'SQLFileBackup_Log',
  FILENAME = N'D:\DBMSBackups\SQLFileBackup_Log.ldf' ,
  SIZE = 1024KB , FILEGROWTH = 10%)
GO
```

Slika 7 - Kreiranje baze podata sa više fajlova

Naredni primeri prikazuju način kreiranja rezervnih kopija na nivou fajlova (Slika 8).

```
BACKUP DATABASE SQLFileBackup
FILE = 'SQLFileBackup_1',
FILE = 'SQLFileBackup_2'
TO DISK = 'D:\DBMSBackups\SQLGroupfiles.bak';
GO
```

Messages

Processed 328 pages for database 'SQLFileBackup', file 'SQLFileBackup\_1' on file 1.  
Processed 8 pages for database 'SQLFileBackup', file 'SQLFileBackup\_2' on file 1.  
Processed 4 pages for database 'SQLFileBackup', file 'SQLFileBackup\_Log' on file 1.  
BACKUP DATABASE...FILE=<name> successfully processed 340 pages in 0.297 seconds (8.930 MB/sec).

Completion time: 2020-05-18T23:56:42.6799184+02:00

Slika 8 - Kreiranje rezervnih kopija na nivou fajlova

Moguće je kreirati rezervne kopije svih fajlova u primarnim i sekundarnim fajl grupama što je prikazano na Slici 9.

```
BACKUP DATABASE SQLFileBackup
FILEGROUP = 'PRIMARY',
FILEGROUP = 'Secondary'
TO DISK = 'D:\DBMSBackups\SQLGroupfilegroup.bak';
GO
```

Messages

Processed 328 pages for database 'SQLFileBackup', file 'SQLFileBackup\_1' on file 1.  
Processed 8 pages for database 'SQLFileBackup', file 'SQLFileBackup\_2' on file 1.  
Processed 2 pages for database 'SQLFileBackup', file 'SQLFileBackup\_Log' on file 1.  
BACKUP DATABASE...FILE=<name> successfully processed 338 pages in 0.251 seconds (10.520 MB/sec).

Completion time: 2020-05-18T23:58:42.3747006+02:00

Slika 9 - Kreiranje rezervnih kopija u primarnim i sekundarnim fajl grupama

## 2. Modeli oporavka

Model oporavka predstavlja konfiguracionu opciju baze podataka koja određuje tip *backup*-a koji će se koristiti. Modeli oporavka pružaju mogućnost povrata podataka i oporavka od otkaza. Postoji nekoliko tipova modela oporavka. Svi SQL Server *backup* tipovi, načini oporavka i operacije oporavka su bazirani na jednom od tri dostupna modela. To su:

1. *SIMPLE*
2. *FULL*
3. *BULK\_LOGGED*

### 2.1. *SIMPLE* model oporavka

*SIMPLE* model oporavka je najjednostavniji model od navedenih. Podržava potpuni, diferencijalni, parcijalni, *copy-only* i *backup* na nivou fajlova dok rezervne kopije na nivou dnevnika transakcija nisu podržane. Oporavak sekundarnih *read-only* fajlova je dozvoljen. Svaka transakcija se upisuje u dnevnik transakcija, međutim čim se transakcija obavi i podaci upišu u fajl za podatke, prostor koji je korišćen za taj zapis biva oslobođen i može se koristiti za nove zapise. Upravo zbog toga moguće je izvršiti poslednji celokupni ili poslednji diferencijalni *backup*. Razlozi za odabir ovakvog modela oporavka mogu biti:

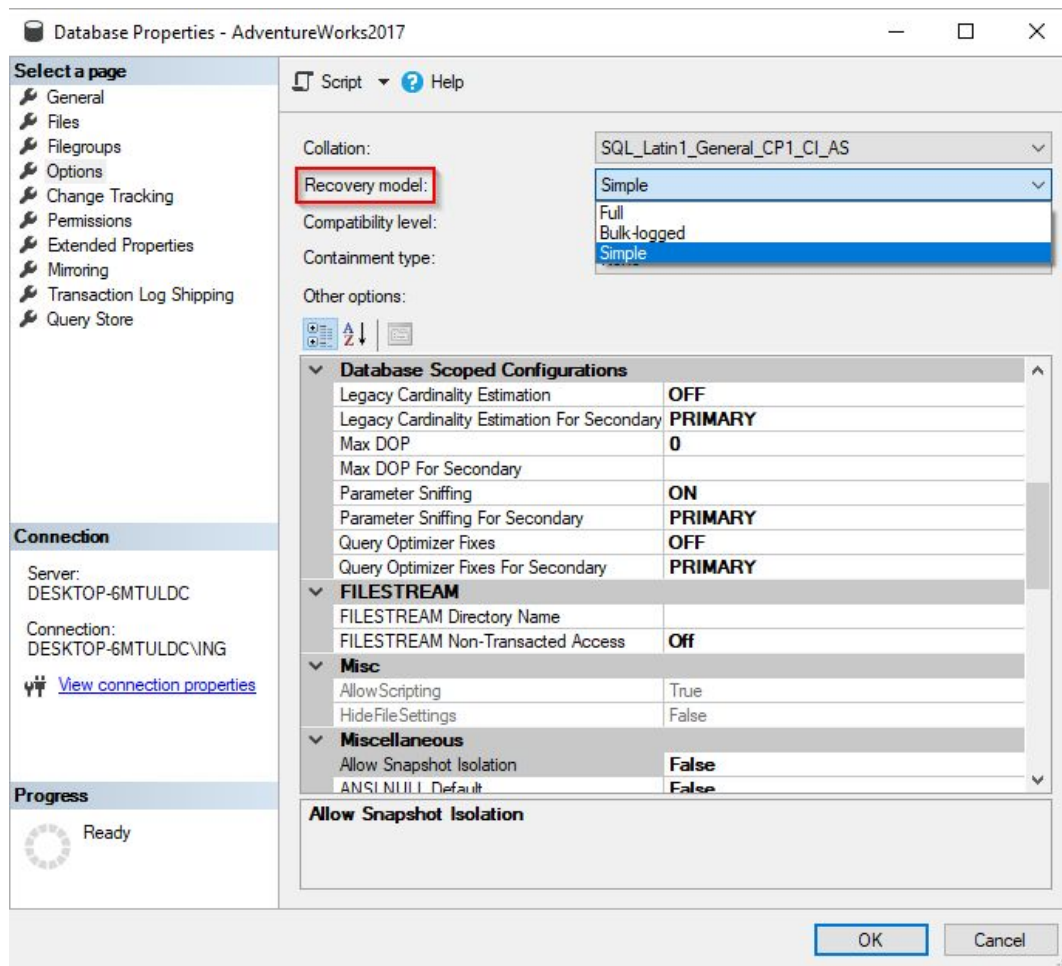
- Najbolje je prilagođen dev i test bazama podataka
- Jednostavna baza podataka za izveštavanje ili aplikacije gde je gubitak podataka prihvatljiv
- Podaci nisu kritični i lako se mogu rekreirati
- Podaci su statički i nema promena nad njima
- Nema administrativnih troškova

Za postavljanje modela oporavka dovoljno je izvršiti komandu sa Slike 10.

```
ALTER DATABASE AdventureWorks2017
SET RECOVERY SIMPLE
GO
```

Slika 10 - Podešavanje *SIMPLE* modela oporavka

Drugi način za promenu modela oporavka je kroz *Microsoft SQL Server Managament Studio*, odabirom jedne od ponuđenih opcija (Slika 11).



Slika 11 - Podešavanje modela oporavka kroz *Managament Studio*

## 2.2. *FULL* model oporavka

Kada je u pitanju *FULL* model opravka, sve transakcije pamte se u log fajlu. Sekvenca log fajla je neprekinuta i sačuvana je za operacije obnavljanja baza podataka. Za razliku od jednostavnog modela oporavka, datoteka dnevnika transakcija nije automatski skraćena tokom *CHECKPOINT* operacija. Podržane su sve opcije vraćanja, uključujući *point-in-time* vraćanje i vraćanje fajlova. Neke od karakteristika i povoljnih situacija za korišćenje pomenutog modela su:

- Pruža podršku za kritične aplikacije
- Dizajn rešenja visoke dostupnosti
- Za olakšavanje oporavka svih podataka sa nultim ili minimalnim stepenom gubitka podataka
- Ako je baza podataka dizajnirana da ima više grupa datoteka a potrebno je izvršiti delimično vraćanje sekundarnih grupa datoteka za čitanje ili pisanje
- Dozvoljen je proizvoljan *point-in-time* oporavak
- Uključuje administrativne troškove

### 2.3. *BULK\_LOGGED* model oporavka

*BULK\_LOGGED* model oporavka predstavlja konfiguracionu opciju baze podataka specijalne namene. Karakteristike ovog modela su veoma slične *FULL* modelu oporavka, s tim što pojedine *bulk* operacije mogu biti minimalno logovane. Log fajl za transakcije koristi tehniku koja je poznata kao minimalno logovanje za *bulk* operacije. Ograničenje predstavlja to što nije moguće izvršiti povrat u određeni trenutak (*point-in-time*).

Prednosti korišćenja ovog modela su:

- Korišćenje minimalne tehnike za čuvanje logova kako bi se sprečilo da log fajlovi narastu s vremenom
- Povoljan je u situacijama kada je baza podataka podvrgnuta *bulk* operacijama

### 2.4. Promena modela oporavka

Moguće je izvršiti promenu modela oporavka uz praćenje određenih koraka.

1. *FULL* i *BULK\_LOGGED* modeli oporavka zahtevaju:

- Iniciranje log *backup*-a
- Izvršavanje *bulk* operacija i vraćanje baze podataka na *full* model oporavka
- Iniciranje transakcionog log *backup*-a

2. Prevođenje iz *SIMPLE* modela oporavka u *FULL* model oporavka:
  - Iniciranje potpunog *backup*-a
  - Zakazivanje transakcionog log *backup*-a
3. Prevođenje iz *FULL* ili *BULK\_LOGGED* modela oporavka u *SIMPLE* model oporavka:
  - Zaustaviti *backup job* za transakcione logove
  - Uveriti se da postoji *job* koji kreira potpuni *backup*

### 3. Strategije za pravljenje rezervnih kopija i oporavak nakon otkaza

Jedna od najbitnijih uloga administratora baza podataka je zaštita integriteta baza podataka i održavanje i oporavak usled otkaza. Upravo zbog toga, strategije za oporavak i redovno pravljenje rezervnih kopija je neophodno. Glavna odgovornost administratora je da omogući da baza podataka bude uvek dostupna i da postoje unapred pripremljena rešenja kako se performanse ne bi dovele u pitanje. Takođe, pri tome potrebno je osigurati da nema ili ima veoma malo gubitka podataka. Pomenute strategije se moraju periodično testirati. Postoji više naprednih tehnika koje pružaju visoku dostupnost kao što su *Clustering*, *AlwaysOn*, *LogShipping* i *Mirroring*, međutim neophodno je imati dobro definisan i testiran plan za pravljenje rezervnih kopija i oporavak.

Prilikom odabira neke od strategija potrebno je uzeti u obzir:

- Vremenski period u okviru kog se podaci ažuriraju
- Proces online transakcija
- Stepem promena šeme baze podataka
- Stepem promena u konfiguraciji baze podataka
- Obrasci za učitavanje podataka
- Priroda podataka

#### 3.1. Najbolja praksa za pravljenje strategija

Neke od smernica kojih se treba pridržavati su:

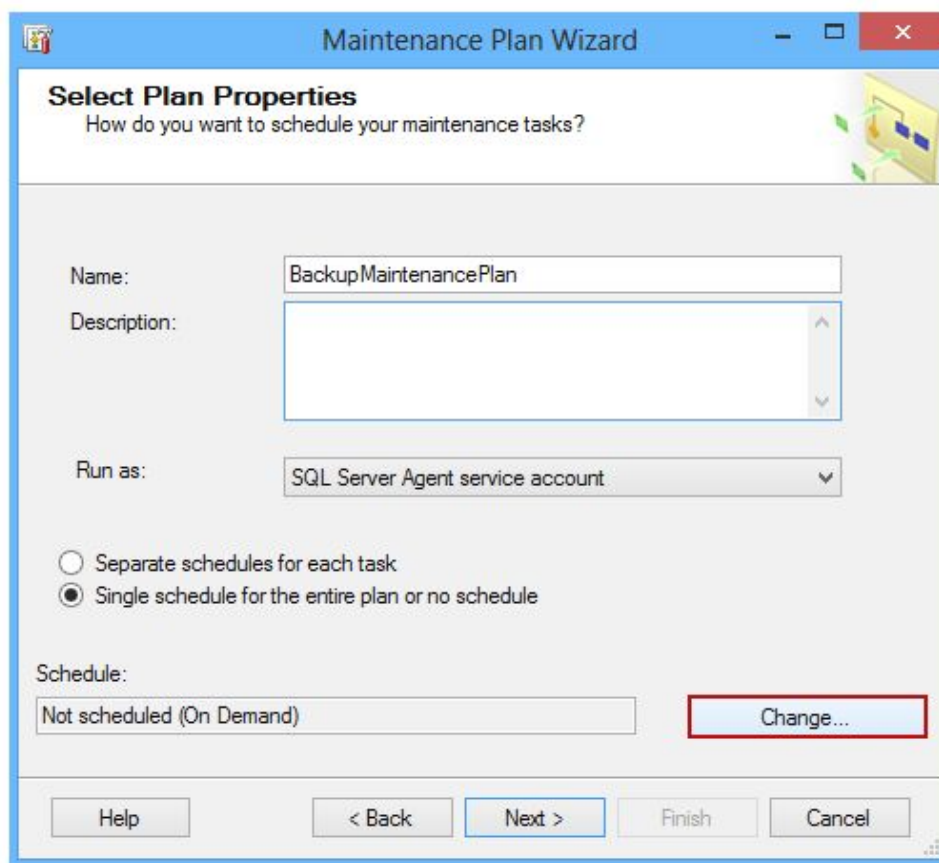
1. Lokacija - Najbolja praksa je da se lokacija gde je smešten *backup* fajl fizički odvoji od lokacije gde se nalazi server u okviru koga je baza podataka. To je potrebno zbog slučajeva kada neki otkazi, kao što je otkaz diska, utiču na mogućnost korišćenja rezervne kopije koja se nalazi na istom fizičkom disku.
2. Automatski i zakazani backup-ovi - Predstavljaju dobru praksu kako bi se osigurali od slučajnog gubitka podataka ili drugih scenarija otkaza. Postojanje redovnih rezervnih kopija smanjuje vreme za koje će aplikacija biti nedostupna usled otkaza. Vremenski

period na koji je potrebno kreirati rezervne kopije zavisi od dosta parametara, a zavisi i od zahtevanog biznis plana.

3. Restore i testiranje backup-ova - Backup-ovi moraju biti *restore*-ovani u test okruženju kako bi bilo sigurno da oni odgovaraju svim biznis zahtevima.
4. Verifikovanje backup-ova - Obuhvata provere da je rezervna kopija uspešno kreirana, sve transakcije su konzistentne i *backup* se može *restore*-ovati.

### 3.2. Kreiranje zakazanih *backup*-ova korišćenjem planova održavanja

Postoji više načina za kreiranje *backup*-ova u trenutku koji je unapred definisan. Jedan od načina je kreirati *job* korišćenjem *SQL Server Management Studio*. Da bi se automatski zakazao SQL Server *backup* plan, potrebno je pratiti sledeće korake: odabrati *Maintenance Plan Wizard* opciju u okviru *Management* regije, upisati ime plana i klikom na dugme *Change* zakazati *job* (Slika 12).



Slika 12 - Izgled prozora za zakazivanje *backup*-a



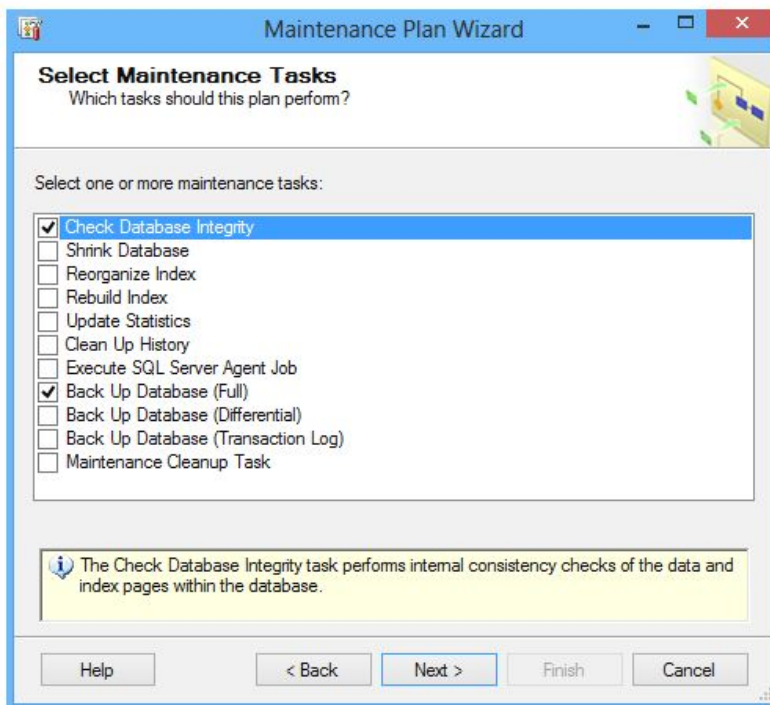
Pored imena rezervne kopije, potrebno je navesti frekvencu pojavljivanja zadatka za pravljenje rezervnih kopija. Od informacija potrebno je uneti koliko često se zadatak obavlja, da li je na dnevnom nivou, u koliko sati ukoliko se obavlja jednom u toku dana ili na koliko sati ukoliko se obavlja na određeni vremenski period u okviru dana. Ove opcije prikazane su na Slici 13.

The image shows a 'New Job Schedule' dialog box with the following configuration:

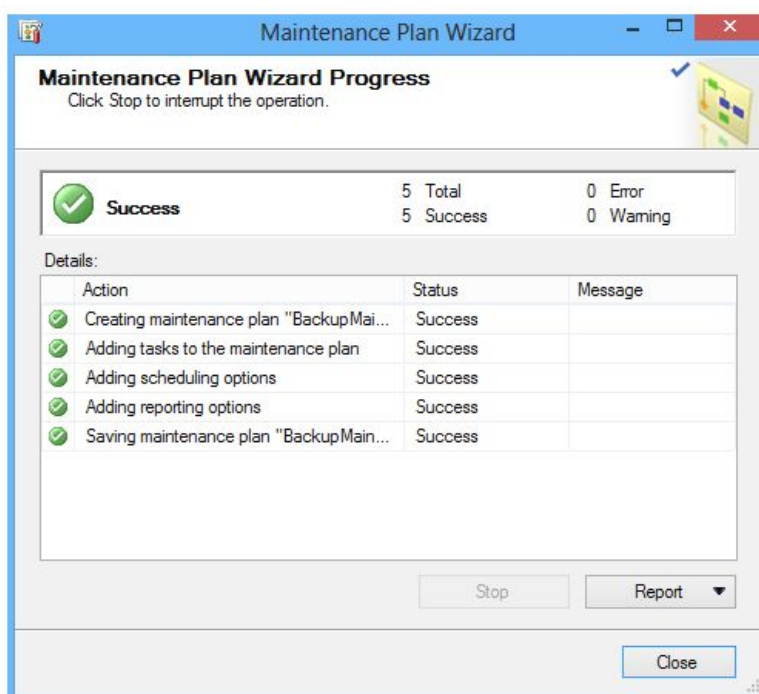
- Name:** BackupMaintenancePlan
- Schedule type:** Recurring (Enabled)
- One-time occurrence:** Date: 3/12/2015, Time: 9:10:58 PM
- Frequency:** Occurs: Daily, Recurs every: 1 day(s)
- Daily frequency:** Occurs once at: 12:00:00 AM, Occurs every: 1 hour(s), Starting at: 12:00:00 AM, Ending at: 11:59:59 PM
- Duration:** Start date: 3/12/2015, End date: 3/12/2015, No end date (selected)
- Summary:** Description: Occurs every day at 12:00:00 AM. Schedule will be used starting on 3/12/2015.

Slika 13 - Prikaz podešavanja za pravljenje rezervnih kopija

Pored navedenih opcija, potrebno je navesti koje zadatke održavanja je potrebno periodično izvršavati, treba izabrati jednu od ponuđenih mogućnosti iz liste dostupnih (Slika 14). Nakon uspešno konfigurisanog plana održavanja prikaz izgleda kao na Slici 15.



Slika 14 - Odabir zadataka održavanja



Slika 15 - Prikaz uspešno konfigurisanog plana održavanja

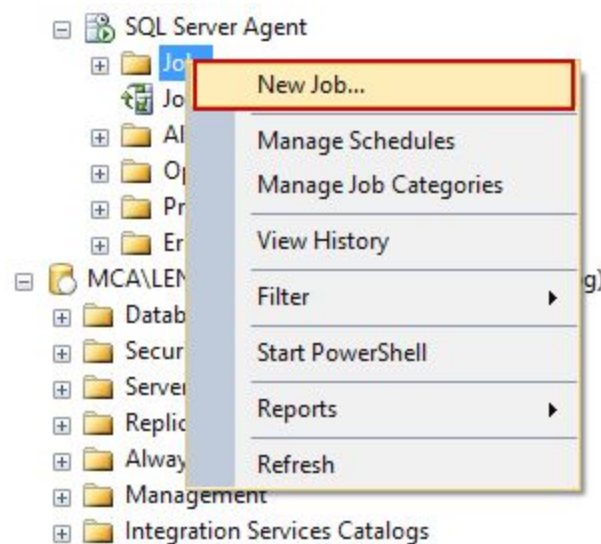
Planovi održavanja predstavljaju dobru praksu za održavanje sigurnosti baze podataka za manje iskusne administratore. Međutim, interfejs, iako pruža dosta mogućnosti, nije dovoljan za kreiranje zahtevnijih planova održavanja.

Mana zakazivanja kreiranja rezervnih kopija korišćenjem planova održavanja je ta što jedan plan može da izvršava samo jedan zadatak jednovremeno. To znači da bi vremenom bio potreban sve veći broj planova koji vremenom postaju jako teški za organizaciju i održavanje.

### 3.3. Kreiranje zakazanih *backup*-ova korišćenjem SQL Server agenta

Za kreiranje zakazanih *backup*-ova potrebno je ispuniti nekoliko koraka. Prvi je kreiranje *job*-a u okviru SQL Server agenta (Slika 16) i dodavanje SQL izraza za kreiranje *backup*-a.

```
BACKUP DATABASE AdventureWorks2017  
TO DISK = N'D:\DBMSBackups\AW2017Backup.bak'  
WITH CHECKSUM;
```



Slika 16 - Kreiranje novog *job*-a u okviru SQL Server agenta

Nakon što se *job* kreira, potrebno je odlučiti u kom vremenskom periodu će biti pravljen rezervne kopije. To se postiže navođenjem frekvence pravljenja rezervnih kopija u okviru *Job Schedule* opcije, na sličan način kao što je slučaj sa zakazivanjem kod planova održavanja.

## Zaključak

Svaki administrator baze podataka mora imati jasan i pouzdan plan za oporavak od potencijalnih katastrofa. Plan treba da bude u stanju da povрати sistem na stanje koje je bilo pre samog otkaza. Postoji više situacija koje je potrebno uzeti u obzir prilikom kreiranja planova izvršenja. Pored toga, postoji više načina za njihovo automatsko izvršavanje, odnosno zakazivanje, kako bi se omogućilo redovno formiranje rezervnih kopija.

U radu su prikazani različiti tipovi rezervnih kopija: celokupne, diferencijalne, parcijalne, rezervne kopije zasnovane na fajlovima. Pored toga, prikazano je koje je modele oporavka moguće konfigurisati u okviru SQL Server baze podataka. Na osnovu odabranog modela i tipa *backup*-a prikazani su načini za automatsko formiranje rezervnih kopija. Da bi ono funkcionisalo validno potrebno je obratiti posebnu pažnju na vremenski period na koji je potrebno formirati rezervne kopije i šta treba prethoditi određenim tipovima *backup*-ova.

Oporavak i pravljenje rezervnih kopija predstavljaju veoma bitnu oblast projektovanja sistema i za njihovo planiranje i testiranje je potrebno izdvojiti dovoljno vremena, testiranja i analiza kako bi se svi potencijalni rizici sveli na minimum a samim tim i sistem bio visoko dostupan.

# Reference

- [1] Arshad Ali - Database Journal - Backup and Restore Strategies in SQL Server  
<https://www.databasejournal.com/features/mssql/backup-and-restore-strategies-in-sql-server-part-1.html>
  
- [2] Prashanth Jayaram - Understanding Sql Server Backup Types  
<https://www.sqlshack.com/understanding-sql-server-backup-types/>
  
- [3] Robert Sales - Multiple methods for scheduling a SQL Server backup automatically  
<https://www.sqlshack.com/multiple-methods-for-scheduling-a-sql-server-backup-automatically/>
  
- [4] MSDN - Backup Overview (SQL Server)  
<https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/backup-overview-sql-server?redirectedfrom=MSDN&view=sql-server-ver15>
  
- [5] Prashanth Jayaram - Understanding SQL Server database recovery models  
<https://www.sqlshack.com/understanding-database-recovery-models/>
  
- [6] MSDN - SQL Server Agent  
<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent?view=sql-server-ver15>