



Univerzitet u Nišu
Elektronski fakultet
Katedra za računarstvo



Aleksandar Stamenković

MSSQL Server – Backup/restore baza podataka

Seminarski rad

Master akademske studije

Predmet: Sistemi za upravljanje bazama podataka

Smer: Računarstvo i informatika

Modul: Bezbednost računarskih sistema

Student:

Aleksandar Stamenković

Br. indeksa: 1403

Niš, maj 2022. god.

Sadržaj

1	Uvod	3
2	Razlozi za backup	3
3	Modeli oporavka baza podataka	3
3.1	Prosti (eng. <i>simple</i>)	3
3.2	Celokupni (eng. <i>full</i>)	4
3.3	Bulk-logged.....	4
4	Backup tipovi.....	5
4.1	Copy-only backup.....	5
4.2	Backup baze podataka (full backup)	5
4.3	Diferencijalni backup.....	6
4.4	Log backup.....	7
4.5	Fajl backup.....	7
4.6	Parcijalni backup.....	8
5	Kreiranje backup-a.....	8
6	Restore baze podataka.....	10
6.1	Primer full i diferencijalnog backup-a	12
7	Zaključak.....	13
8	Literatura.....	14

1 Uvod

U ovom radu će biti opisane prednosti backup-a baza podataka, osnovni termini koji se koriste za backup i restore, kao i strategije koje se pri tom koriste.

Komponenta SQL servera koja je zadužena za backup i restore omogućava osnovu za sigurnost tako što čuva kritične podatke koji se skladište u samoj bazi. Da bi se minimizirali rizici katastrofalnog gubitka podataka, potrebno je backup-ovati baze podataka da bi se sačuvala promene nad podacima. Dobro isplanirana backup/restore strategija pomaže da se zaštite baze podataka od eventualnog gubitka podataka ozrokovana različitim greškama.

2 Razlozi za backup

Backup je jedini način da se zaštite podaci. Sa validnim backupovima, moguće je oporaviti podatke od strane različitih grešaka, kao na primer:

- Korisničke greške (npr. nenamerno brisanje tabele),
- Hardverske greške (npr. oštećen hard disk, trajan gubitak servera),
- Prirodne katastrofe (korišćenjem backup-a na nekom od storage servisa, npr. Azure Blob storage servis, moguće je skladištiti backupove na različitim stranama sveta tako da se mogućnosti pojave ove greške svedu na minimum).

Takođe, backup baza podataka je korisna za razne administrativne rutine kao što su kopiranje baze podataka s jednog servera na drugi itd.

3 Modeli oporavka baza podataka

Model oporavka je konfiguracija baze podataka koja određuje tip backup-a koji se može izvesti i omogućava sposobnost obnove podataka ili oporavka od greške. U osnovi, model oporavka odlučuje kako se log transakcija održava i štiti promenu podataka u specifičnoj sekvenci koji se kasnije može koristiti za restore baze podataka.

MSSQL Server podržava 3 modela oporavka i oni su opisani u nastavku.

3.1 Prosti (eng. *simple*)

Prosti model oporavka je najprostiji od mogućih modela. Podržava potpuni (eng. *full*), diferencijalni (eng. *differential*) i backup na nivou fajla. Backup logova transakcija nije podržan. Prostor logova se iznova koristi kad god se desi checkpoint u SQL Server pozadinskom procesu. Neaktivni deo log fajla se briše i čini dostupnim za ponovno korišćenje. Tzv. point-in-time i restore strane nije podržano. Podržan je samo restore sekundarnih read-only fajlova.

Razlozi za izbor prostog modela oporavka:

- Odgovara razvoju i testiranju baza podataka

- Greške se odstranjuju samo celokupnim i diferencijalnim backup-ovima
- Nema administrativnog overhead-a

Podržava:

- Full backup
- Diferencijalni backup
- Copy-only backup
- Fajl backup
- Parcijalni backup

3.2 Celokupni (eng. *full*)

U ovom modelu oporavka, sve transakcije (DDL – Data Definition Language, DML – Data Manipulation Language) se celokupno beleže u logu transakcija. Sekvenca log fajla je neslomljena i čuva se za restore operacije. Za razliku od prostog modela oporavka, fajl loga transakcija nije prebrisan za vreme checkpoint operacija. Ovde su sve restore operacije podržane, uključujući point-in-time restore, restore strane i restore fajla.

Razlozi za izbor celokupnog modela oporavka:

- Podržava kritične aplikacije u smislu zadataka
- Visoka dostupnost
- Olakšava oporavak svih podataka za nikakvom ili minimalnom mogućnošću gubitka podataka
- Podržava proizvoljne point-in-time restor-ove
- Restore individualnih stranica

Podržava sve navedene backup-ove:

- Full backup
- Diferencijalni backup
- Backup loga transakcija
- Copy-only backup
- Fajl backup, backup grupe fajlova
- Parcijalni backup

3.3 Bulk-logged

Predstavlja konfiguraciju za posebne namene i radi slično kao i full model oporavka osim toga što određene bulk operacije se mogu minimalno logovati. Fajl loga transakcija koristi specijalnu tehniku za minimalno logovanje bulk operacija. Cena koja se plaća je da nije moguće restore-ovati specifične point-in-time podatke.

Razlozi za izbor bulk-logged modela oporavka:

- Koristi tehniku minimalnog logovanja da spreči rast log fajla
- Ako se baza podataka zasniva na periodično korišćenje bulk operacija

Podržava sve tipove backup-a kao i full model oporavka.

4 Backup tipovi

MSSQL Server podržava nekoliko tipova backup-a, a to su:

4.1 Copy-only backup

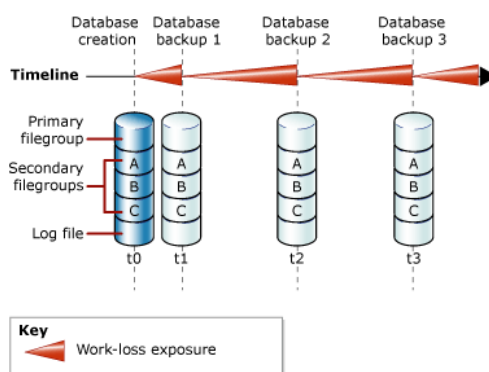
Backup koji je nezavisan od sekvence ostalih backup-a. Ponekad je potrebno napraviti backup bez uticaja na ostale backup-e.

4.2 Backup baze podataka (full backup)

Backup-uje celu bazu podataka. Ovo uključuje deo loga transakcija tako da je moguće obnoviti celu bazu podataka nakon što se cela baza podataka restore-uje. Backup celokupne baze podataka predstavlja bazu podataka u trenutnu kada je backup kreiran.

- **Backup baze podataka pod prostim modelom oporavka**

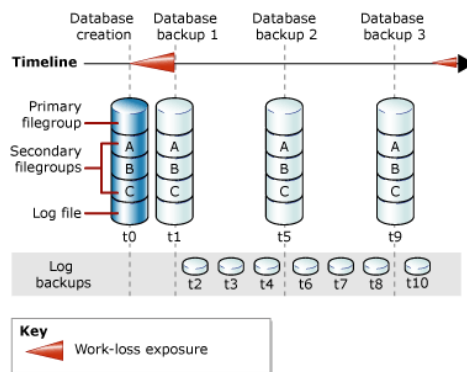
Pod prostim modelom oporavka, nakon svakog backupa, baza podataka je izložena potencijalnom gubitku podataka ako se dogodi greška. Izloženost raste svakom izmenom baze podataka do sledećeg backup-a, kada se ta izloženost opet vrati na nulu i opet kreće da raste. Ta izloženost raste između svaka dva backup-a. Na sledećoj slici je prikazana ta izloženost gubicima podataka:



Slika 1 Rizik od gubitaka podataka kod prostog modela oporavka

- **Backup baze podataka pod full modelom oporavka**

Za baze podatak koje koriste full ili bulk-logge model oporavka, backup baze podataka je potreban ali ne i dovoljan. Takođe su neophodni backup-i logova transakcija. Na sledećoj slici je prikazan full model opravka.



Slika 2 Rizik od gubitaka podataka kod full modela oporavka

4.3 Diferencijalni backup

Zasniva se na najskorijem posljednjem full backup-u. Diferencijalni backup beleži samo podatke koji se menjani od poslenjeg full backup-a. Full backup na koga se sledeći diferencijalni backup-ovi odnose se zove osnova diferencijala (eng. base of the differential). Full backup-ovi, osim copy-only backup-ova, mogu da posluže kao osnove za seriju diferencijala backup-ova, uključujući backup baze podataka, parcijalnih backup-ova i backup fajlova.

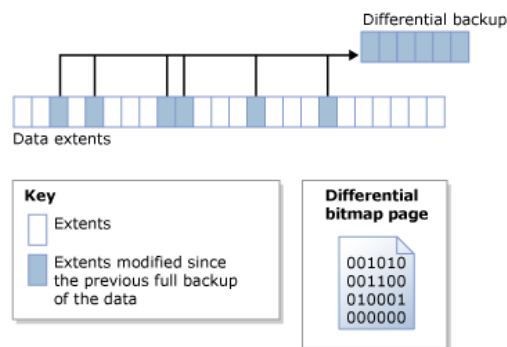
- **Prednosti**

Kreiranje diferencijalnih backup-ova može biti značajno brže od kreiranja full backup-a. Diferencijalni backup beleži samo promene nad podacima koje su se desile nakon poslednjeg full backup-a. Ovo značajno olakšava kreiranje učestalih backup-ova koji umanjuju mogućnost gubitka podataka. Međutim, pre nego što se izvrši restore diferencijalnog backup-a najpre treba izvršiti restore njegove osnove. Prema tome, restore iz diferencijalnog backup-a zahteva više koraka i vremena od nego restore full backup-a.

Diferencijalni backup-ovi su naročito korisni ukoliko se podskup baze podataka menja znatno češće u odnosu na ostatak baze podataka. U tom slučaju, diferencijalni backup omogućava čest backup bez overhead-a koji bi uneo backup cele baze podataka.

Ukoliko se koristi full model oporavka, korišćenje diferencijalnog backup-a može smanjiti broj backup-ova logova koje je potrebno restore-ovati.

Na seldećoj slici je data ilustracija rada diferencijalnih backup-ova. Slika prikazuje 24 extent-a podataka (kolekacija od 8 fizički usednih stranica), od čega je 6 modifikovano. Diferencijalni backup sadrži samo ova 6 extenta podataka. Operacija diferencijalnog backup-a se oslanja na bitmapi koja sadrži bit za svaki extent. Kad god se neki extent modifikuje od trenutka kreiranja osnove diferencijalnog backup-a, bit za odgovarajući extent se postavlja na 1.



Slika 3 Princip rada diferencijalnog bckup-a

4.4 Log backup

Predstavlja backup logova transakcija koji sadrže sve log record-e koji nisu backup-ovani u prethodnih log backup-u. Može se primeniti samo ukoliko je postavljen full model oporavka baze podataka. Potrebno je da postoji bar jedan full backup pre nego što se kreira log backup. Preporučljivo je raditi log backup često, da bi se smanjili rizici za gubicima podataka kao i da bi se smanjio log transakcija novim backup-om.

4.5 Fajl backup

Predstavlja backup jednog ili više fajlova ili grupe fajlova baze podataka. Fajlovi baze podataka se mogu backup-ovati i restore-ovati individualno. Takođe, moguće je specificirati celu grupu fajlova umesto specificiranja svakog fajla individualno. Međutim, ukoliko je jedan fajl iz grupe fajlova offline (na primer dok se radi njihov restore), cela grupa fajlova je offline i ne može se backup-ovati.

- **Prednosti**

Korišćenje backup fajlova može ubrzati oporavak baze podataka tako što omogućava restore samo onih fajlova koji su oštećeni, bez restore-a ostatka baze podataka. Npr. ukoliko se baza podataka sastoji od nekoliko fajlova koji su smešteni na više različitih diskova i jedan disk otkaže, samo fajl koji se nalazi na tom disku treba biti restore-ovan. Oštećen fajl se može brzo restore-ovati i oporavak je značajno brži nego da se radi oporavk cele baze podataka. Još jedna prednost je takođe da backup fajlova omogućava veću fleksibilnost u raspoređivanju backupova što je naročito važno kod velikih baza podataka.

- **Nedostaci**

Glavni nedostatak backupa fajlova u odnosu na backup cele baze podataka je dodatna administrativna kompleksnost. Održavanje i čuvanje skupova backup-ova fajlova može biti vremenski zahtevan zadatak koji brzo može prevazići prostor namenjen za potrebe full backup-a.

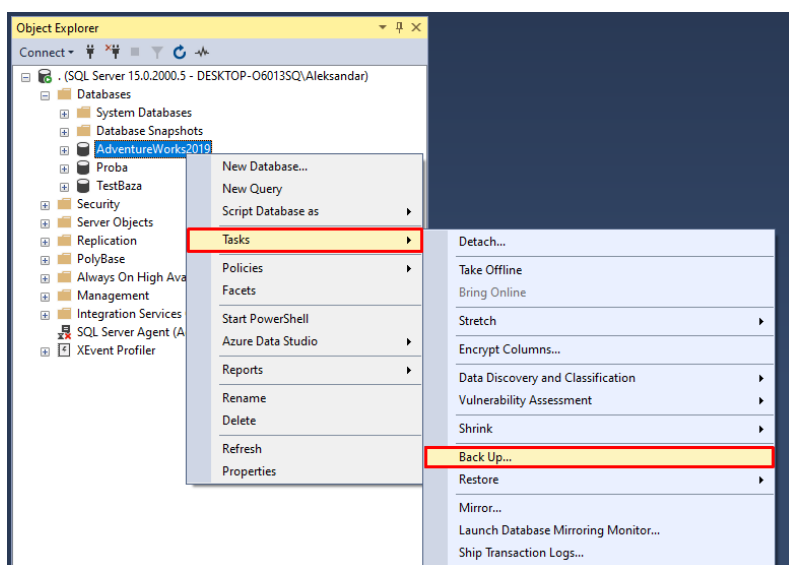
4.6 Parcijalni backup

Svi modeli oporavka podržavaju parcijalni backup. Međutim parcijalni backup je dizajniran najviše za prosti model oporavka da bi se poboljšala fleksibilnost za backup veoma velikih baza podataka koje sadrže jednu ili više read-only grupa fajlova. Parcijalni backup je korisan kad god je neophodno da se isključi backup read-only grupa fajlova. Parcijalni backup liči na full backup, s tim što parcijalni backup ne uključuje sve grupe fajlova.

5 Kreiranje backup-a

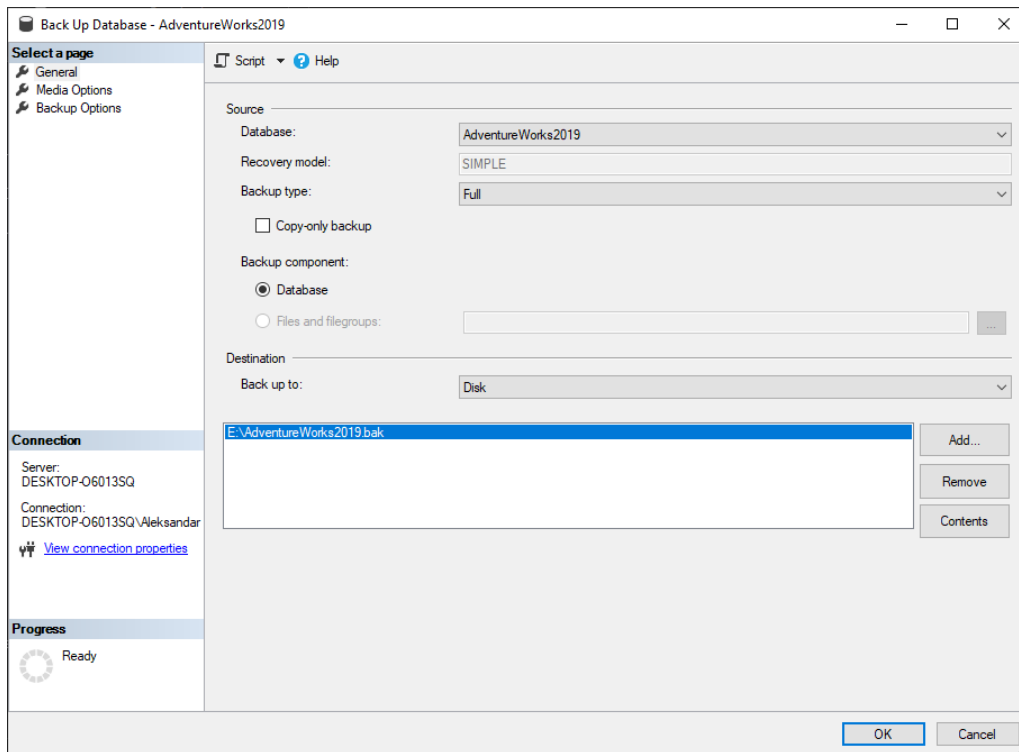
Za potrebe kreiranja backup-a biće korišćena baza podataka AdventureWorks2019 (koja je takođe korišćena u prvom seminarskom radu iz ovog predmeta) i klijentska aplikacija Microsoft SQL Server Management Studio preko koje će se vršiti konekcija na DBMS.

Da bi se napravio backup neke baze podataka potrebno je pronaći željenu bazu podataka pod folderom Databases u Object Exploreru, zatim kliknuti desnim klikom na bazu podataka i doći do komande Back up kao što je to prikazano na sledećoj slici:



Slika 4 SSMS - Backup komanda iz Object Explorer-a

Zatim će se otvoriti prozor kao što je prikazano na sledećoj slici. Tu je moguće dodatno podešavanje backup-a, od kojih su najbitnija model oporavka (eng. recovery model), tip backup-a (eng. backup type) i putanja backup fajla gde će se čuvati backup. Ovde je neophodno da se fajlu postavi ekstenzija *.bak*.



Slika 5 SSMS - Backup prozor

Osim preko interfejsa, backup je moguće uraditi i korišćenjem SQL skripte. Postoji posebna naredba za to, kao što je prikazano ispod:

```
BACKUP DATABASE { database_name | @database_name_var }
TO <backup_device> [ ,...n ]
[ <MIRROR TO clause> ] [ next-mirror-to ]
[ WITH { DIFFERENTIAL
      | <general_WITH_options> [ ,...n ] } ]
[;]

<backup_device>::=
{
  { logical_device_name | @logical_device_name_var }
  | { DISK
    | TAPE
    | URL } =
    { 'physical_device_name' | @physical_device_name_var | 'NUL' }
}

<MIRROR TO clause>::=
MIRROR TO <backup_device> [ ,...n ]

<general_WITH_options> [ ,...n ]::=
--Backup Set Options
COPY_ONLY
| { COMPRESSION | NO_COMPRESSION }
| DESCRIPTION = { 'text' | @text_variable }
| NAME = { backup_set_name | @backup_set_name_var }
| CREDENTIAL
| ENCRYPTION
| FILE_SNAPSHOT
| { EXPIREDATE = { 'date' | @date_var }
  | RETAINDAYS = { days | @days_var } }
```

Slika 6 Backup SQL sintaksa

Konkretni primer skripte je dat na slici ispod:

```
BACKUP DATABASE [AdventureWorks2019]
TO DISK = N'E:\primer123.bak'
GO
```

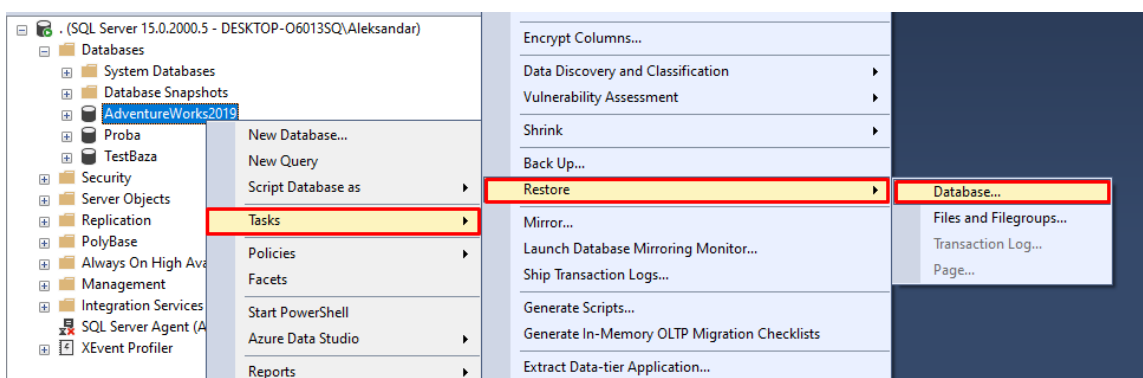
Slika 7 SQL primer backup-a

6 Restore baze podataka

Da bi se oporavila baza podataka od greške, administrator baze podataka treba da restore-uje (prev. obnovi) skup backup-a u logički korektnoj i smislenoj sekvenci. SQL Server restore i oporavak podržava restore podataka iz backup-a cele baze podataka, jednog fajla podataka ili jedne strane podataka:

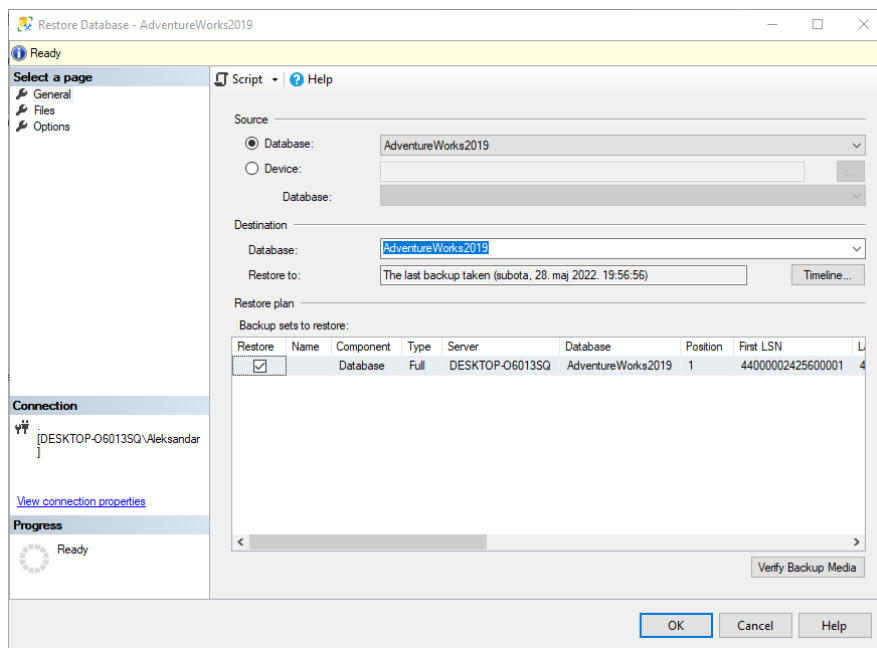
- **Baza podataka** (*kompletna obnova baze podataka*)
Cela baza podataka je obnovljena i oporavljena, i baza podataka je offline za vreme obnavljanja i oporavljanja.
- **Fajl podataka** (*obnova fajla*)
Fajl podataka ili skup fajlova je oporavljen i oporavljen. Za vreme ove operacije, grupe fajlova (eng. filegroups) koje sadrže ove fajlove su automatski offline za vreme restore-a. Svaki pokušaj pristupa offline grupi fajlova proukovaće grešku.
- **Strana podataka** (*obnova strane*)
Ako se koristi model potpunog oporavka (eng. full recovery model) ili bulk-logged model oporavka, moguće je restore-ovati individualne stranice. Restore stranice se može izvesti na bilo kojoj bazi podataka, bez obzira na broj grupe fajlova.

Da bi se restore-ovala baza podataka preko SSMS-a neophodno je najpre pronaći željenu bazu podataka u folderu Databases u Object Explorer-u, zatim pronaći komandu za Restore kao što je to prikazano na slici ispod:



Slika 8 SSMS - Restore komanda iz Object Explorer-a

Zatim se otvara prozor kao na sledećoj slici:



Slika 9 SSMS - Restore prozor

Obično se restore radi iz nekog fajla (mada može i iz postojeće baze podataka), tako da je neophodno selektovati u odeljku Source opciju Device i pronaći bak fajl u fajl sistemu. Dodatno se mogu podesiti i putanje gde se čuvaju podaci kao i logovi, kao i razne restore opcije.

Pored korišćenja korisničkog interfejsa, restore baze podataka moguće je uraditi i korišćenjem SQL skripti. Sintaksa je prikazana na sledećoj slici:

```
RESTORE DATABASE { database_name | @database_name_var }
[ FROM <backup_device> [ ,...n ] ]
[ WITH
    {
        [ RECOVERY | NORECOVERY | STANDBY =
            {standby_file_name | @standby_file_name_var }
        ]
        | , <general_WITH_options> [ ,...n ]
        | , <replication_WITH_option>
        | , <change_data_capture_WITH_option>
        | , <FILESTREAM_WITH_option>
        | , <service_broker_WITH_options>
        | , \<point_in_time_WITH_options-RESTORE_DATABASE>
    } [ ,...n ]
]
[:]

<backup_device>::=
{
    { logical_backup_device_name |
        @logical_backup_device_name_var }
    | { DISK
        | TAPE
        | URL
    } = { 'physical_backup_device_name' |
        @physical_backup_device_name_var }
}

<general_WITH_options> [ ,...n ]::=
--Restore Operation Options
    MOVE 'logical_file_name_in_backup' TO 'operating_system_file_name'
    [ ,...n ]
| REPLACE
| RESTART
| RESTRICTED_USER | CREDENTIAL
```

Slika 10 Restore SQL sintaksa

Konkretan minimalan primer skripte je dat na slici ispod:

```
USE [master]
RESTORE DATABASE [AdventureWorks2019]
FROM DISK = N'E:\primer123.bak'
GO
```

Slika 11 SQL primer restore-a

6.1 Primer full i diferencijalnog backup-a

Često se samo jedan deo baze podataka ažurira (a ne cela baza), tako da je u tom slučaju dobar pristup ne raditi svaki put full backup baze podataka već nakon jednog full backup-a dalje se može raditi parcijalni backup-ovi. Na taj način će se značajno uštedeti vreme i prostor na disku s obzirom na to da su full backup-ovi značajno veći od diferencijalnih backup-ova.

Na sledećoj slici je prikaza skripta koja ilustruje primer jednog korišćenja full i diferencijalnog backup-a, kao i način restore-a korišćenjem tako kreiranih backup-ova:

```
declare @full_backup nvarchar(64) = N'E:\full_backup.bak'
declare @diff_backup nvarchar(64) = N'E:\diff_backup.bak'

-- full backup
BACKUP DATABASE [AdventureWorks2019] TO DISK = @full_backup

-- modifikacija podataka
-- . . .

-- diferencijalni backup
BACKUP DATABASE [AdventureWorks2019] TO DISK = @diff_backup WITH DIFFERENTIAL

-- restore
USE [master]
ALTER DATABASE [AdventureWorks2019] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
RESTORE DATABASE [AdventureWorks2019] FROM DISK = @full_backup WITH NORECOVERY
RESTORE DATABASE [AdventureWorks2019] FROM DISK = @diff_backup
ALTER DATABASE [AdventureWorks2019] SET MULTI_USER
```

Slika 12 SQL primer full i diferencijalno backup i restore

Sa skripte se vidi da se najpre vrši full backup baze podataka, a zatim nakon izmena baze se može izvršiti diferencijalni backup koji će zabeležiti samo izmene nakon kreirane poslenje osnove tj. full backup-a. Nakon toga se baza podataka stavlja u single user modu što znači da se raskidaju sve konekcije s bazom i nakon toga je moguće uraditi restore full backup-a ali nije neophodno odmah zatim uraditi i oporavak baze već se to odlaže tako da se uradi nakon svih neophodnih restore-ova. Sledeće se radi restore diferencijalnog backup-a i nakon toga je potrebno oporaviti bazu kada se ona stavlja u operativni mod i spremna je za korišćenje.

Poređenja radi, data je i slika loga nakon izvršenja gore navedene skripte gde se mogu uporediti vremena izvršenja full i diferencijalnog backup-a odnosno restore-a.

```

Processed 34480 pages for database 'AdventureWorks2019', file 'AdventureWorks2017' on file 5.
Processed 2 pages for database 'AdventureWorks2019', file 'AdventureWorks2017_log' on file 5.
BACKUP DATABASE successfully processed 34482 pages in 7.621 seconds (35.347 MB/sec).
Processed 56 pages for database 'AdventureWorks2019', file 'AdventureWorks2017' on file 5.
Processed 2 pages for database 'AdventureWorks2019', file 'AdventureWorks2017_log' on file 5.
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 58 pages in 0.231 seconds (1.944 MB/sec).
Nonqualified transactions are being rolled back. Estimated rollback completion: 0%.
Nonqualified transactions are being rolled back. Estimated rollback completion: 100%.
Processed 34480 pages for database 'AdventureWorks2019', file 'AdventureWorks2017' on file 1.
Processed 2 pages for database 'AdventureWorks2019', file 'AdventureWorks2017_log' on file 1.
RESTORE DATABASE successfully processed 34482 pages in 7.370 seconds (36.551 MB/sec).
Processed 120 pages for database 'AdventureWorks2019', file 'AdventureWorks2017' on file 1.
Processed 2 pages for database 'AdventureWorks2019', file 'AdventureWorks2017_log' on file 1.
RESTORE DATABASE successfully processed 122 pages in 1.000 seconds (0.949 MB/sec).

```

Slika 13 Log izvršene skripte za backup i restore

7 Zaključak

Backup je jedini način da se zaštitite podaci. Backup-ovanjem baza podataka se osigurava da se podaci ne izgube ili slučajno neželjeno modifikuju. Takođe, postoji opasnost i od hardverskih grešaka, npr. otkaza diska, otkaza servera gde su uskladišteni podaci itd.

S obzirom na to da postoji veliki broj različitih strategija za backup baza podataka, moguće je fleksibilno odabrati način koji je najpogodniji za konkretnu bazu podataka i primeniti je da bi se izbegli katastrofalni gubici podataka. Jedna od mogućih primena backup/restore baza podatak može biti i u razvoju aplikacija kada je potrebno reprodukovati neki događaj sa nekog okruženja lokalno da bi se otkrili uzroci nastanka takvih događaja (bilo željenih ili neželjenih – bagovi). Naročito je to korisno kod aplikacija koji se nalaze pri kraju razvojnog ciklusa, gde su nastali bagovi najverovatnije uzrokovani stanjem podataka a ne samom implementacijom aplikacije.

8 Literatura

- [1] Back Up and Restore of SQL Server Databases <https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases?view=sql-server-ver16>
- [2] How to Restore SQL Server Database from backup <https://blog.sqlbackupandftp.com/restore-database-backup>
- [3] Understanding SQL Server Backup Types <https://www.sqlshack.com/understanding-sql-server-backup-types/>