A logo with text and hands

Description automatically generated with medium confidenceA blue and white logo

Description automatically generated

**Univerzitet u Nišu**

**Elektronski faktultet**

**Backup i Restore kod MS SQL baze podataka**

Studijski program: Računarstvo i informatika

Modul: Softversko inženjerstvo

Student: Profesor:

Katarina Maksimović, 1599 Prof. dr. Aleksandar Stanimirović

Niš, Maj 2024. godina

Sadržaj

[1. Uvod 3](#_Toc167491948)

[2. Recovery models 4](#_Toc167491949)

[3. Backup baze podataka 4](#_Toc167491950)

[3.1. Važnost backup-a 5](#_Toc167491951)

[3.2. Vrste backup-a 5](#_Toc167491952)

[3.2.1. Full backup 6](#_Toc167491953)

[3.2.2 Differential backup 12](#_Toc167491954)

[3.2.3. Transaction log backup 17](#_Toc167491955)

[3.2.4. Tail-log backup 19](#_Toc167491956)

[3.2.5. Full File backup 22](#_Toc167491957)

[3.2.6. Partial backup 27](#_Toc167491958)

[3.2.7. Copy-only backup 28](#_Toc167491959)

[4. Restore baze podataka 32](#_Toc167491960)

[4.1. Restore u SSMS-u 32](#_Toc167491961)

[4.2. Restore pomoću T-SQL komandi 40](#_Toc167491962)

[4.2.1. Restore Full backup-a 40](#_Toc167491963)

[4.2.2. Restore Diferencijalnog backup-a 41](#_Toc167491964)

[4.2.3. Restore Transaction log backup-ova 41](#_Toc167491965)

[4.2.4. Restore Tail-log backup-a 42](#_Toc167491966)

[4.2.5. Restore Full File backup-a 43](#_Toc167491967)

[4.2.6. Restore Partial backup-a 43](#_Toc167491968)

[4.2.7. Restore Copy-Only backup 44](#_Toc167491969)

[5. Zaključak 45](#_Toc167491970)

[6. Literatura 46](#_Toc167491971)

# Uvod

Backup i restore su temeljni procesi za upravljanje podacima u SQL Server-u, ključni za osiguranje kontinuiteta poslovanja i zaštitu podataka. Ovi procesi omogućavaju kreiranje kopija podataka baze podataka koje se mogu koristiti za vraćanje sistema u slučaju gubitka podataka, korupcije baze podataka ili drugih katastrofalnih događaja. Efikasna strategija backup-ovanja i restore-ovanja može značajno smanjiti vreme zastoja i rizik od gubitka podataka, osiguravajući da kritične poslovne operacije ostanu neprekidne.

Backup proces uključuje kreiranje kopija podataka baze podataka koje se skladište na sigurnim mestima. Postoji nekoliko strategija backup-ovanja koje se mogu koristiti, zavisno od specifičnih potreba i karakteristika okruženja. Na primer, organizacije mogu koristiti full backup baze podataka koji uključuje sve podatke, ili mogu kreirati diferencijalni backup koji sadrže samo promene nastale od poslednjeg full backup-a. Ključna svrha backup-a je da osigura da postoji tačna i ažurna kopija podataka koja se može koristiti za oporavak u slučaju gubitka podataka.

Restore proces je suprotan od backup-a i uključuje vraćanje podataka iz kreiranih backup-ova u bazu podataka. Restore proces može uključivati vraćanje cele baze podataka iz jednog backup-a ili kombinovanje više backup-a kako bi se baza podataka vratila u tačno određeno stanje pre kvara. Restore-ovanje može biti složeno, posebno u okruženjima sa velikim količinama podataka i visokom učestalošću transakcija, ali je neophodno za obnavljanje normalnog rada sistema nakon gubitka podataka.

Efikasna strategija backup-ovanja i restore-ovanja zahteva pažljivo planiranje i implementaciju. Ključni elementi ovog plana uključuju definisanje učestalosti backup-ovanja, izbor odgovarajuće vrste backup-ova, određivanje lokacija za skladištenje backup-ova, te testiranje procesa restore-ovanja kako bi se osiguralo da su backup-ovi funkcionalni i upotrebljivi. Organizacije moraju redovno evaluirati i ažurirati svoje strategije backup-ovanja kako bi odgovorile na promene u poslovnim potrebama i IT infrastrukturi.

Jedan od glavnih izazova u upravljanju backup-ovima je balansiranje između potrebe za sveobuhvatnom zaštitom podataka i ograničenja resursa kao što su skladišni prostor i vreme. Kako baze podataka rastu, tako raste i veličina backup-ova, što može povećati troškove skladištenja i produžiti vreme potrebno za kreiranje i vraćanje backup-ova. Stoga je važno optimizovati procese backup-ovanja i implementirati tehnologije i metode koje minimiziraju resurse potrebne za backup, a istovremeno osiguravaju visoku dostupnost podataka.

Osim tehničkih aspekata, backup i restore procesi zahtevaju i adekvatnu organizacionu podršku. To uključuje obuku IT osoblja, uspostavljanje politika i procedura za backup i restore, te osiguranje da svi zaposleni razumeju važnost ovih procesa i znaju kako da reaguju u slučaju incidenta. Regularna revizija i testiranje planova za oporavak pomažu u identifikaciji slabih tačaka i omogućavaju kontinuirano poboljšanje strategija zaštite podataka.

# Recovery models

Operacije backup-a i restore-a u SQL Server-u odvijaju se u kontekstu modela oporavka (recovery model) baze podataka. Modeli oporavka dizajnirani su da kontrolišu održavanje transakcionog loga. Model oporavka je svojstvo baze podataka koje kontroliše način na koji se transakcije beleže u logu, da li je potrebno (i dozvoljeno) praviti backup transakcionog loga, i koje vrste operacija restore-a su dostupne. Postoje tri modela oporavka: jednostavan (simple), pun (full) i bulk-logovani (bulk-logged). Tipično, baza podataka koristi pun ili jednostavan model oporavka. Baza podataka može biti prebačena na drugi model oporavka u bilo koje vreme.

# Backup baze podataka

Backup je proces pravljenja rezervne kopije sistema na nekoj drugoj lokaciji, bilo na hard disku, lokalnom serveru ili nekom drugom, u cloud-u ili kombinaciji svega toga.

Uglavnom, većina kompanija sledi protokol backup-a 3-2-1, koji pravi tri kopije podataka na dva različita sistema, od kojih se jedan čuva van lokacije. Pravljenje backup-a se može vršiti ručno, ali poslednjih godina se pojavio SaaS za pravljenje backup-a u cloud-u kako bi se omogućilo brže, automatizovano i bezbednije pravljenje backup-a.

Postoje mnoge prednosti automatizovanog softvera za pravljenje backup-a u cloud-u; konkretno, smanjuje IT troškove i složenost u organizaciji i omogućava manjim kompanijama koje nemaju IT osoblje da pristupe alatima za preduzeća bez ulaganja u plate i infrastrukturu.

Današnji poslovni pejzaž je veoma konkurentan i svaki zastoj može uticati na kontinuitet. Dostupan backup i spreman za vraćanje podataka kada je to potrebno znači da se zainteresovane strane mogu više fokusirati na razvoj svog poslovanja, a manje na potencijalne pretnje.

Razne backups strategije se koriste za oporavak podataka ili smanjenje rizika od gubitka podataka u slučaju kvara.

# Važnost backup-a

Postoji niz kvarova koji se mogu pojaviti u različitim fazama životnog veka sistema. Iako se mogu preduzeti neke proaktivne mere kako bi se osiguralo da se kvarovi svedu na minimum, nije garantovano da se to neće dogoditi. Međutim, mogu se postaviti odgovarajuće backup i restore strategije kako bi se osiguralo da ako se desi bilo kakav kvar, postoji mehanizam kojim se može oporaviti od kvara.

Postoji nekoliko vrsta čestih kvarova:

* Otkazivanje hardvera (kvar instance SQL Servera, network greška, kvar medija i diska) – Iako sa najnovijim hardverom na tržištu i sa ugrađenim mogućnostima redundantnosti i prelaska na grešku, šanse za kvar su znatno smanjene, to se ipak može dogoditi. Na primer, kontroler na disku možda neće raditi, podaci na disku se mogu oštetiti itd.
* Otkaz sistema i softvera – Ovo je povezano sa kvarom operativnog sistema, CPU-a ili glavne memorije računarskog sistema. Otkazivanje sistema i softvera može biti uzrokovano nestankom struje, padom aplikacije ili operativnog sistema, greškom u memoriji ili nekim drugim razlogom.
* Neuspeh aplikacije – Ova vrsta kvara može da se desi zbog nekih grešaka u aplikaciji, koje su neželjeno izmenile (ubacile\ažurirale\izbrisale) podatke. Na primer, ažuriranje plate za zaposlenog bez klauzule WHERE. Drugi primer bi mogao da se odnosi na logiku, na primer, plata zaposlenom je trebalo da bude uvećana za 20%, a zapravo je povećana za 200%.
* Greška korisnika – Greška korisnika je slična neuspehu aplikacije sa izuzetkom što korisnik neželjeno modifikuje (ubacuje\ažurira\briše) podatke. Na primer, korisnik je želeo da izbriše jedan određeni red iz tabele, ali je izvršio naredbu DELETE bez klauzule WHERE.

# Vrste backup-a

Generalno, postoje različite backup vrste u SQL-u i upotreba zavisi od zahteva korisnika. Postoji sledeći backup-ovi:

1) Full backup

2) Differential backup

3) Transaction log backup

4) Tail-log backup

5) Full File backup

6) Partial backup

7) Copy-Only backup

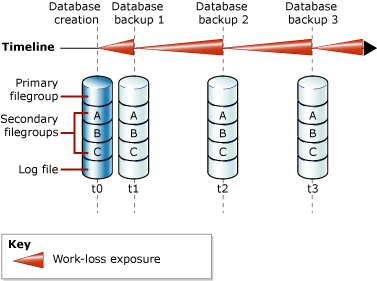
# Full backup

Full backup baze podataka backup-uje celu bazu podataka. Ovo uključuje deo transakcionog loga tako da cela baza podataka može biti obnovljena nakon što se restore-uje full backup baze podataka. Full backup baze podataka predstavlja bazu podataka u trenutku kada je backup završen.

Kako se baza podataka povećava, full backup-ovi baze podataka se duže izvršavaju i zahtevaju više prostora za skladištenje. Stoga, za veliku bazu podataka, može biti efikasnije da se full backup baze podataka kombinuje sa serijom diferencijalnih backup-ova.

Full backup-ovi u kombinaciji sa simple recovery model-om:

U simple recovery modelu, nakon svakog backup-a, baza podataka je izložena potencijalnom gubitku podataka ako bi se desila katastrofa (npr. hardverski kvar, korupcija podataka). Rizik od gubitka podataka se povećava sa svakim ažuriranjem sve dok se ne napravi novi backup, kada se rizik od gubitka podataka vraća na nulu i počinje novi ciklus rizika. Rizik od gubitka podataka se povećava tokom vremena između dva backup-a. Sledeća ilustracija prikazuje rizik od gubitka podataka za strategiju backup-a koja koristi samo full backup-ove baze podataka.



Slika 1. Rizik od gubitka podataka za strategiju backup-a koja koristi samo full backup-ove baze podataka

Pretpostavimo da se pravi full backup svakog ponedeljka ujutro:

* Ponedeljak:
  + 08:00: Pravi se full backup (svi podaci do tada su sačuvani).
  + 09:00: Novi podaci počinju da se unose. Izloženost gubitku rada počinje da raste.
* Sreda:
  + Izloženost gubitku rada se povećava jer sve više podataka nije sačuvano.
* Nedelja:
  + Ako dođe do kvara u nedelju uveče, izgubiće se svi podaci unešeni od ponedeljka ujutru do tog trenutka.
* Sledeći ponedeljak:
  + 08:00: Novi full backup resetuje izloženost na nulu.

Prednosti Simple Recovery Modela:

* Jednostavna administracija: Nema potrebe za pravljenjem transaction log backup-a.
* Manja veličina transakcijskog loga: Automatsko oslobađanje prostora u logu.

Nedostaci Simple Recovery Modela:

* Veća izloženost gubitku podataka između full backup-ova.
* Nema mogućnosti tačkastog vraćanja podataka (point-in-time recovery).

Pravljenja Full Backup-a u Simple Recovery modelu:

Korišćenje SQL Server Management Studio (SSMS)

1. Nakon povezivanja na odgovarajuću instancu Microsoft SQL Server Database Engine-a, u Object Explorer-u, proširiti stablo servera.
2. Proširiti Databases i izabrati korisničku bazu podataka ili proširiti System Databases i izabrati sistemsku bazu podataka.
3. Desni klik na bazu podataka koju je potrebno backup-ovati i izabrati Tasks -> Back Up....

A screenshot of a computer

Description automatically generated

Slika 2. Backup

1. U prozoru Back Up Database, izabrati Full kao Backup type.
2. U sekciji Destination, kliknuti na Add... za dodavanje lokacije gde će se sačuvati backup. Uneti putanju i naziv fajla (npr. C:\Backups\MyDatabase.bak).
3. Kliknuti na OK kako bi se započep proces backup-ovanja.

Korišćenje T-SQL komande:

A screenshot of a computer

Description automatically generated

Slika 3. Full Backup u Simple Recovery modelu

A screenshot of a computer

Description automatically generated

Slika 4. Rezultat

Backup baza podataka pod full recovery modelom:

Za baze podataka koje koriste full I bulk-logged recovery model, backup-ovi baze podataka su neophodni, ali nisu dovoljni. Potrebni su I backup-ovi transakcionih logova. Sledeća ilustracija prikazuje najmanje složenu strategiju backup-a koja je moguća pod full recovery modelom.

A diagram of a backup system

Description automatically generated

Slika 5. Ilustracija najmanje složenu strategiju backup-a koja je moguća pod full recovery modelom.

Ključni Koraci u Strategiji

1. Redovni Full backup-ovi:

* Prave se periodično, obično jednom dnevno ili nedeljno, zavisno od kritičnosti podataka i količine promena.

1. Frequent Transaction log backup-ovi:

* Prave se često (npr. svakih 15 minuta, svakih sat vremena), što omogućava minimalan gubitak podataka i redovno oslobađanje prostora u transakcijskom logu.
* Backup-ovi loga omogućavaju vraćanje baze podataka do bilo koje tačke između dva full backup-a.

Primer (Transact-SQL):

Sledeći primer pokazuje kako napraviti full backup baze podataka korišćenjem WITH FORMAT-a za prepisivanje svih postojećih backup-ova i kreiranje novog skupa medija. Zatim, primer backup-uje log transakciju. U stvarnoj situaciji, moralo bi da se izvrši niz redovnih rezervnih backup-ova. Za ovaj primer, primer baze podataka Sport school je podešen da koristi full recovery model.

A screenshot of a computer program

Description automatically generated

Slika 6. Full backup baze podataka korišćenjem WITH FORMAT-a

A screenshot of a computer

Description automatically generated

Slika 7. Rezultat

Zašto je Full Backup važan?

1. Osnova za druge backup-ove: Svi drugi tipovi backup-a (differential, transaction log, file backup, partial) zavise od full backup-a.
2. Kompletna kopija: Sadrži sve podatke baze podataka u trenutku pravljenja backup-a.
3. Jednostavna obnova: Može se koristiti samostalno za obnovu baze podataka, bez potrebe za kombinovanjem sa drugim backup-ovima.

Detaljan primer koristeći SSMS:

Pretpostavimo da imamo bazu podataka pod nazivom Sport school i da je potrebno napraviti full backup na lokaciji “C:\Backups\”.

1. Pokrenuti SSMS i konektovati se na SQL Server instancu.
2. Proširiti Databases, desni klik na MyDatabase, i izabrati Tasks -> Back Up....
3. U prozoru Back Up Database, izabrati Full kao Backup type.
4. Kliknuti na Add..., zatim u do C:\Backups\ uneti ime fajla Sport school\_FULL.bak.
5. Kliknuti na OK kako bi se započeo proces backup-ovanja.

A screenshot of a computer

Description automatically generated

Slika 8. Full backup

A screenshot of a computer

Description automatically generated

Slika 9. Rezultat

Kreiranje full backup koristeći T-SQL komande:

A screenshot of a computer

Description automatically generated

Slika 10. Full backup koristeći T-SQL komande

Opis naredbe:

* Sport school je naziv baze podataka.
* ‘C:\Backups\Sport school\_FULL.bak’ je putanja I naziv fajla gde će backup biti sačuvan.
* FORMAT: Ova opcija prepisuje bilo koji postojeći backup set na odredištu.
* INIT: Inicijalizuje (prepisuje) medijum za backup, brišući postojeće backup setove.
* NAME: Opisni naziv za backup set.

A screenshot of a computer

Description automatically generated

Slika 11. Rezultat

# Differential backup

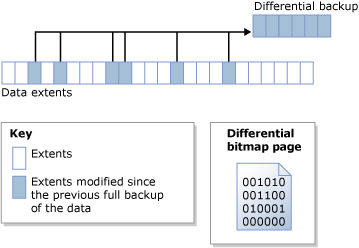
Diferencijalni backup se zasniva na najskorijem full backup-u. Diferencijalni backup beleži samo podatke koji su se promenili od poslednjeg full backup-a. Full backup na osnovu kojeg se pravi diferencijalni backup poznat je kao osnova diferencijalnog backup-a. Full backup-ovi, osim kopiranih (copy-only) backup-a, mogu poslužiti kao osnova za seriju diferencijalnih backup-a, uključujući full backup-ove, delimične backup-ove i file backup-ove.

Kreiranje diferencijalnog backup-a može biti znatno brže od kreiranja full backup-a. Diferencijalni backup beleži samo podatke koji su se promenili od full backup-a na osnovu kojeg je napravljen diferencijalni backup. Ovo olakšava pravljenje čestih backup-ova podataka, što smanjuje rizik od gubitka podataka. Međutim, pre nego što se restore-uje diferencijalni backup, mora se restore-ovati njegova osnova. Stoga, restore-ovanje iz diferencijalnog backup-a će nužno zahtevati više koraka i vremena nego restore-ovanje iz full backup-a jer su potrebna dva backup fajla. Diferencijalni backup-ovi baza podataka su posebno korisni ako se podskup baze podataka češće menja od ostatka baze podataka. U ovim slučajevima, diferencijalni backup-ovi baza podataka omogućavaju da se često prave backup-ovi bez dodatnih troškova full backup-a.

Pod full recovery modelom, korišćenje diferencijalnih backup-a može smanjiti broj log backup-a koji se moraju restore-ovati.

Diferencijalni backup beleži stanje bilo kojih extents-a (kolekcija od osam fizički susednih stranica) koje su se promenile između trenutka kada je kreirana diferencijalna osnova i trenutka kada je kreiran diferencijalni backup. To znači da veličina određenog diferencijalnog backup-a zavisi od količine podataka koji se promenili od osnove. Generalno, što je osnova starija, to će novi diferencijalni backup biti veći. U seriji diferencijalnih backup-a, extent koji se često ažurira verovatno će sadržati različite podatke u svakom diferencijalnom backup-u.

Sledeći primer prikazuje kako diferencijalni backup funkcioniše. Na slici su prikazana 24 extents-a podataka, od kojih se 6 promenilo. Diferencijalni backup sadrži samo ova šest extents-a podataka. Operacija diferencijalnog backup-a se oslanja na stranicu bitova koja sadrži bit za svaki extent. Za svaki extent koji je ažuriran, bit se postavlja na 1 u bitmape.



Slika 12. Prikaz funkcionisanja diferencijalnog backup-a

Diferencijalni backup koji je napravljen relativno brzo nakon svoje osnove može biti značajno manji od diferencijalne osnove. Ovo štedi prostor za skladištenje i vreme backup-a. Međutim, kako se baza podataka menja tokom vremena, razlika između baze podataka i određene diferencijalne osnove se povećava. Što je duže vreme između diferencijalnog backup-a i njegove osnove, to je veća verovatnoća da će diferencijalni backup biti veći. To znači da se diferencijalni backup-ovi na kraju mogu približiti veličini diferencijalne osnove. Veliki diferencijalni backup gubi prednosti bržeg i manjeg backup-a.

Kako veličina diferencijalnih backup-ova raste, obnova diferencijalnog backup-a može značajno povećati vreme potrebno za obnavljanje baze podataka. Stoga se preporučuje da se redovno pravi novi full backup u određenim intervalima kako bi se uspostavila nova diferencijalna osnova za podatke. Na primer, može se praviti nedeljni full backup cele baze podataka, a zatim tokom nedelje praviti redovnu seriju diferencijalnih backup-ova baze podataka.

Prilikom restore-ovanja, pre nego što se restore-uje diferencijalni backup, mora se restore-ovati njegova osnova. Zatim, restore-uje se samo najnoviji diferencijalni backup kako bi se baza podataka obnovila na vreme kada je taj diferencijalni backup napravljen. Dakle, tipično prvo se obnovi najnoviji full backup, a zatim najnoviji diferencijalni backup koji se bazira na tom full backup-u.

Diferencijalni backup-i read-only baza podataka:

Za read-only baze podataka, full backup-ovi su lakši za upravljanje kada se koriste samostalno nego kada se koriste sa diferencijalnim backup-ovima. Kada je baza podataka read-only, operacije backup-a i ostale operacije ne mogu promeniti metapodatke koji se nalaze u fajlu. Stoga, metapodaci potrebni za diferencijalni backup, kao što je log sequence number sa kojim diferencijalni backup počinje (diferencijalna osnovna LSN) se čuvaju u master bazi podataka. Ako je diferencijalna osnova preuzeta kada je baza podataka read-only, diferencijalni bitkod pokazuje više promena nego što su se zaista desile od osnovnog backup-a. Dodatni podaci se čitaju tokom backup-a, ali se ne pišu u backup, jer se diferencijal\_base\_lsn čuva u backupset sistemu tabela kako bi se odredilo da li su podaci promenjeni od osnove.

Kada se read-only baza podataka rebuilduje, restoreuje ili diskonektuje i ponovo povezuje, informacije o osnovnoj diferencijalnoj osnovi se gube. To se dešava jer master baza podataka nije sinhronizovana sa korisničkom bazom podataka. SQL Server Database Engine ne može otkriti ili sprečiti ovaj problem. Svaki kasniji diferencijalni backup nije baziran na najnovijem full backup-u i može pružiti neočekivane rezultate. Da bi se uspostavila nova diferencijalna osnova, potrebno je da se napravi novi full backup baze podataka.

Najbolje prakse za korišćenje diferencijalnih backup-a sa read-only bazama podataka su sledeće:

* Nakon što se napravi full backup read-only baze podataka, ukoliko se planira kasniji diferencijalni backup, potrebno je napraviti backup master baze podataka.
* Ako je master baza podataka izgubljena, potrebno je da se ona restore-uje pre nego što se restore-uje bilo koji diferencijalni backup korisničke baze podataka.
* Ako se diskonektuje i ponovo konektuje read-only baza podataka za koju se planira da se kasnije koriste diferencijalni backup-ovi, čim bude moguće, potrebno je napraviti full backup obe baze podataka, kako korisničke tako i master baze podataka.

Primer kreiranja Differential backup-a:

Korišćenje SQL Server Management Studio (SSMS)

1. Nakon povezivanja na odgovarajuću instancu Microsoft SQL Server Database Engine-a, u Object Explorer-u kliknuti na ime servera da bi se proširilo stablo servera.
2. Proširiti Databases I u zavisnosti od baze podataka ili izabrati korisničku bazu podataka ili proširiti System Databases I izabrati sistemsku bazu podataka.
3. Desni klik na bazu podataka koju je potrebno backup-ovati I izabrati Tasks -> Back Up....
4. U prozoru Back Up Database, izabrati Differential kao Backup type.
5. U sekciji Destination, kliknuti na Add... za dodavanje lokacije gde će backup biti sačuvan. Uneti putanju I naziv fajla (npr. C:\Backups\MyDatabase\_Diff.bak).

Može da se napravi diferencijalni backup za bilo koji model oporavka (full, bulk-logged ili simple).

1. Kliknuti na OK da bi se započeo process backup-ovanja.

A screenshot of a computer

Description automatically generated

Slika 13. Kreiranje diferencijalnog backup-a

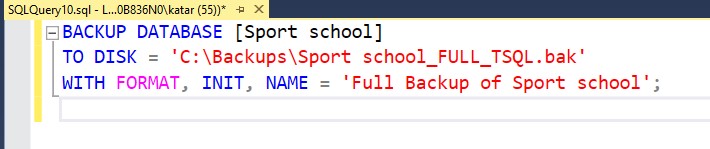
A screenshot of a computer

Description automatically generated

Slika 14. Rezultat

Korišćenje T-SQL Skripti:

1. Full backup (potrebno pre kreiranja differential backup-a):



Slika 15. Kreiranje Full backup-a pre diferencijalnog backup-a

1. Differential Backup Script:

A screenshot of a computer

Description automatically generated

Slika 16. Kreiranje diferencijalnog backup-a

A screenshot of a computer

Description automatically generated

Slika 17. Rezultat

# 3.2.3. Transaction log backup

Transakcioni log je datoteka koja beleži sve transakcije koje su izvršene u bazi podataka. Ove transakcije uključuju sve promene podataka koje su izvršene nad bazom podataka, uključujući INSERT, UPDATE, DELETE i DDL (Data Definition Language) operacije.

Neophodno je da se kreira barem jedan full backup pre nego što se može kreirati bilo kakav log backup. Nakon toga, transakcioni log može biti backup-ovan u bilo koje vreme osim ako već nije backup-ovan. Pogodno je da se često prave log backup-ovi kako bi se minizovao rizik od gubitka podataka.

Administrator baze podataka obično povremeno kreira full backup baze podataka, npr nedeljno, i opciono, kreira seriju diferencijalnih backup-ova baze podataka sa kraćim intervalom, npr dnevno. Nezavisno od backup-a baze podataka, administrator baze podataka backup-uje transakcioni log u čestim intervalima. Za određeni tip backup-a, optimalni interval zavisi od faktora kao što su važnost podataka, veličina baze podataka i opterećenje servera.

Kako funkcioniše sekvenca log backup-ova:

Sekvenca transakcionih log backup-ova, odnosno lanac logova, nezavisan je od backup-a podataka. Na primer, pretpostavimo sledeći niz događaja.

| **Time** | **Event** |
| --- | --- |
| 8:00 AM | Back up database. |
| Noon | Back up transaction log. |
| 4:00 PM | Back up transaction log. |
| 6:00 PM | Back up database. |
| 8:00 PM | Back up transaction log. |

Log backup kreiran u 20:00 časova sadrži zapise transakcionog loga od 16:00 do 20:00 časova, obuhvatajući vreme kada je full backup baze podataka kreiran u 18:00 časova. Sekvenca transakcionih log backup-ova je kontinuirana, od početnog punog backup-a baze podataka kreiranog u 08:00 časova, do poslednjeg log backup-a kreiranog u 20:00 časova.

Kako izvršiti backup transakcionog loga?

1. SQL Server Management Studio (SSMS):

* Otvoriti SQL Server Management Studio i konektovati se sa serverom koji sadrži bazu podataka.
* U Object Explorer prozoru, desni klik na bazu podataka za koju je potrebno napraviti backup transakcionog loga.
* Izabrati Tasks, a zatim Backup.
* U dijalogu za backup, izabrati Transaction Log kao tip backup-a.
* Odabrati lokaciju gde će se sačuvati backup i dati odgovarajuće ime datoteci.
* Kliknuti na "OK" kako bi se pokrenuo backup proces.

A screenshot of a computer

Description automatically generated

Slika 18. Backup transakcionog loga

A screenshot of a computer

Description automatically generated

Slika 19. Rezultat

1. Transact-SQL (T-SQL) komande:

Može se izvršiti backup transakcionog loga korišćenjem T-SQL skripti:

BACKUP LOG [ImeBazePodataka]

TO DISK = ‘PutanjaDoLokacije\ImeBackupDatoteke.bak’

WITH INIT

Zameniti “[ImeBazePodataka]” sa imenom baze podataka, a “PutanjaDoLokacije” I “ImeBackupDatoteke.bak” sa odgovarajućom putanjom I imenom datoteke za backup.

A screenshot of a computer

Description automatically generated

Slika 20. Kreiranje transaction log backup-a

A screenshot of a computer

Description automatically generated

Slika 21. Rezultat

# Tail-log backup

Tail-log backup je vrsta transakcionog log backup-a koji se pravi u specifičnim situacijama kako bi se obezbedila poslednja verzija transakcionog loga pre nego što se izvrši operacija vraćanja baze podataka (restore).

Tail-log backup snima sve transakcione logove koji još nisu backup-ovani (rep loga), kako bi se sprečio gubitak podataka I održala celovitost log lanca. Pre nego što se restore-uje SQL Server baza podataka, mora se napraviti tail-log backup. Tail- log backup je poslednji backup od interesa u recovery planu za bazu podataka. Ako je baza podataka oštećena ili nedostupna, tail-log backup može pomoći da se uhvate sve transakcije koje su se dogodile neposredno pre kvara.

Ne zahtevaju svi scenariji restore-a tail-log backup. Nije potreban tail-log backup ako je tačka oporavka sadržana u ranijem log backup-u. Tail-log backup je nepotreban ako se premesta ili zamenjuje (prepisuje) baza podataka I ne treba da se restore-uje posle njenog poslednjeg backup-a.

Situacije koje zahtevaju tail-log backup:

* Ako je baza podataka online I planira se izvršenje restore operacije li bazom podataka, prvo je potrebno kreirati tail-log backup pre nego što se započne restore operaciju kako bi se osiguralo da su sve nedavne transakcije uključene u backup. Da bi se izbegla greška za online bazu podataka, mora se koristiti opcija WITH NORECOVERY iz BACKUP Transact-SQL naredbe.
* Ako je baza podataka offline I ne može se pokrenuti, a potrebno je da se restore-uje, prvo je potrebno kreirati tail-log backup. Budući da u tom trenutku ne može doći do transakcija, koristi se opcija WITH NO\_TRUNCATE. NO\_TRUNCATE je efektivno isto kao I copy-only transaction log backup. Korišćenje opcije WITH NORECOVERY je opciono, jer u tom trenutku ne može doći do transakcija.
* Ako je baza podataka oštećena, može se pokušati kreiranje tail-log backup koristeći opciju WITH CONTINUE\_AFTER\_ERROR BACKUP naredbe.

Na oštećenoj bazi podataka, tail-log backup može uspeti samo ako su log fajlovi neoštećeni, baza podataka je u stanju koje podržava tail-log backup, I baza podataka ne sadrži nikakve bulk-logged promene. Ako tail-log backup ne može da se napravi, sve transakcije koje su izvršene nakon poslednjeg log backup-a su izgubljene.

Tail-log backup-ovi koji imaju nepotpune backup metapodatke:

Tail-log backup-ovi beleže kraj tail-loga čak I ako je baza podataka offline, oštećena lia ko nedostaju fajlovi podataka. To može dovesti do nepotpunih metapodataka prilikom komandi za obnovu informacija I u msdb. Međutim, samo su metapodaci nepotpuni; transakcioni log je potpun I upotrebljiv.

Ako tail-log backup ima nepotpune metapodatke, u tabeli backupset, polje has\_incomplete\_metadata će biti postavljeno na 1. Takođe, u izlazu RESTORE HEADERONLY, polje HasIncompleteMetadata će biti postavljeno na 1.

Ako su metapodaci u tail-log backup-u nepotpuni, tabeli backupfilegroup će nedostajati većina informacija o grupama fajlova u trenutku tail-log backup-a. Većina kolona u tabeli backupfilegroup će biti NULL; jedine relevantne kolone su:

* backup\_set\_id
* filegroup\_id
* type
* type\_desc
* is\_readonly

Pretpostavimo da imamo bazu podataka pod nazivom Sport school koja je pretrpela kvar, I želimo da napravimo tail-log backup pre nego što izvršimo restore operaciju.

T-SQL komanda za Tail-Log backup:

A close-up of a computer screen

Description automatically generated

Slika 22. Kreiranje Tail-log backup-a

Objašnjenje komandi:

* BACKUP LOG [Sport school]: Inicijalizuje backup procesa za transakcioni log baze podataka Sport school.
* TO DISK = ‘C:\Backups\ Sport school\_TAILLOG\_TQSL.bak’: Putanja I ime tail-log backup datoteke.
* WITH NORECOVERY: Ostavlja bazu u stanju čekanja, omogućavajući dalji restore rocess bez promene trenutnog stanja baze.
* NO\_TRUNCATE: Omogućava backup log zapisa čak I ako baza nije dostupna (npr. ako je baza u stanju kvara).

A screenshot of a computer program

Description automatically generated

Slika 23. Rezultat

# Full File backup

Fajlovi u SQL Server bazi podataka mogu se pojedinačno backup-ovati i restore-ovati. Takođe, može se specificirati cela grupa fajlova umesto pojedinačno svaki fajl. Ako je bilo koji fajl u grupi fajlova offline (na primer, jer se fajl restore-uje), cela grupa fajlova je offline i ne može se backup-ovati.

File backup read-only grupa fajlova može se kombinovati sa parcijalnim backup-om. Parcijalni backup-ovi uključuju sve read/write grupne fajlove i, opciono, jednu ili više read-only grupa fajlova.

File backup može služiti kao diferencijalna baza za diferencijalne fajl backup-ove.

File backup nudi sledeće prednosti u odnosu na full backup baze podataka:

* Korišćenje File backup-a može ubrzati recovery proces omogućavajući restore-ovanje samo oštećenih fajlova, bez restore-ovanja ostatka baze podataka.

Na primer, ako baza podataka sadrži nekoliko fajlova koji se nalaze na različitim diskovima i jedan disk otkaže, samo fajl na oštećenom disku može biti restore-ovan. Oštećeni fajl može biti brzo restore-ovan, a oporavak je brži nego što bi bio za celu bazu podataka.

* File backup povećava fleksibilnost u rasporedu i rukovanju backup-a u odnosu na full backup, što za veoma velike baze podataka može postati neupravljivo. Povećana fleksibilnost backup-a fajlova ili grupa fajlova takođe je korisna za velike baze podataka koje sadrže podatke sa različitim karakteristikama ažuriranja.

Nedostaci Full backup-a:

* Glavni nedostatak File backup-a u poređenju sa full backup-om baza podataka je dodatna administrativna kompleksnost. Održavanje i praćenje kompletne kolekcije ovih backup-a može biti vremenski zahtevan zadatak koji može prevazići zahteve prostora potrebnog za Full backup baze podataka.
* Kvar backup-a može učiniti celu bazu podataka neoporavljivom ako oštećeni fajl nema backup. Stoga se mora održavati kompletan set file backup-a i, za full/bulk-logged recovery model, jedan ili više log backup-ova koji pokrivaju minimalni interval između prvog full file backup-a i poslednjeg full file backup-a.

Full file backup bekapuje sve podatke u jednom ili više fajlova ili grupa fajlova. Podrazumevano, file backup sadrži dovoljno transakcionih logova da unapred prebaci fajl na kraj operacije backup-ovanja.

Backup-ovanje read-only fajla ili grupe fajlova isto je za svaki recovery model. Pod full recovery modelom, kompletni setovi full file backup-a, zajedno sa dovoljnim brojem log backup-a koji pokrivaju sve full file backup-ove, ekvivalentni su full backup-u baze podataka.

Samo jedna operacija file backup-a može se desiti istovremeno. Može se backup-ovati više fajlova u jednoj operaciji, ali to može produžiti vreme oporavka ako se mora restore-vati samo jedna datoteka. To je zato što se, kako bi se pronašao taj fajl, čita ceo backup.

File backup-ovi i Simple recovery model

Pod Simple recovery modelom, svi read/write fajlovi moraju biti backup-ovati zajedno. To osigurava da se baza podataka može vratiti u konzistentno stanje u određenom trenutku. Umesto pojedinačnog navođenja svakog read/write fajla ili grupe fajlova, koristi se opcija READ\_WRITE\_FILEGROUPS. Ova opcija backup-uje sve read/write grupe fajlova u bazi podataka. Backup koji je kreiran navođenjem READ\_WRITE\_FILEGROUPS opcije poznat je kao parcijalni backup.

File backup i Full recovery model:

Pod full recovery modelom, mora se backup-ovati transakcioni log, bez obzira na ostatak strategije backup-vanja. Kompletan set full file backup-ovanja, zajedno sa dovoljnim brojem backup-ova transakcionog loga koji pokriva sve file backup-ove od početka prvog file backup-a, ekvivalentan je full backup-u baze podataka.

Restore-ovanje baze podataka samo koristeći file i log backup-ove može biti kompleksno. Stoga, ako je moguće, najbolja praksa je izvršiti full backup baze podataka i započeti log backup pre prvog file backup-a. Sledeća ilustracija prikazuje strategiju u kojoj se pravi full backup baze podataka (u trenutku t1) odmah nakon što je baza podataka kreirana (u trenutku t0). Ovaj prvi backup baze podataka omogućava početak backup transakcionog log-a. Backup-ovi transakcionog loga su zakazani da se vrše u određenim intervalima. File backup-ovi se vrše u intervalima koji najbolje odgovaraju poslovnim zahtevima baze podataka. Ova ilustracija pokazuje da se svaka od četiri grupe fajlova backup-uje jedna po jedna. Redosled u kojem su backup-ovane (A, C, B, A) odražava poslovne zahteve baze podataka.

A diagram of a data backup

Description automatically generated

Slika 24. File backup u kombinaciji sa Full recovery modelom

Kako izvršiti Full File Backup?

1. SQL Server Management Studio (SSMS):

* Otvoriti SQL Server Management Studio I povezati se sa serverom koji sadrži bazu podataka.
* U Object Explorer prozoru, proširiti bazu podataka za koju je potrebno napraviti backup.
* Proširiti stavku Databases I zatim proširiti odgovarajuću bazu podataka.
* Kliknuti desnim tasterom miša na Files u okviru baze podataka, I izabrati Tasks, a zatim Back Up. Pojavljuje se dijalog Back Up Database.
* Na listi Database proveriti ime baze podataka. Opciono može da se izaberete druga baza podataka sa liste.
* U listi Backup type, izabrati Full ili Differential.
* Za opciju Backup component izabrati File and Filegroups.
* U dijalogu Select Files and Filegroups izabrati datoteke I grupe datoteka za koje je potrebno napraviti backup. Može se odabrati jedna ili više pojedinačnih datoteka ili označiti polje za grupu datoteka da bi se automatski odabrali sve datoteke u toj grupi datoteka.

A screenshot of a computer

Description automatically generated

Slika 25. Dijalog Select Files and Filegroups

* Može se ili prihvatiti podrazumevani naziv backup seta predložen u polju Name ili uneti drugo ime za backup set.
* (opciono) U polju Description može se uneti opis backup seta.
* Može se navesti kada će backup isteći:
  + Da bi backup set istekao nakon određenog broja dana, kliknuti na After (podrazumevana opcija) i unesite broj dana nakon kreiranja skupa kada će skup isteći. Ova vrednost može biti od 0 do 99999 dana; vrednost od 0 dana znači da backup skup nikada neće isteći.
  + Da bi backup skup istekao na određeni datum, kliknuti na On i uneti datum kada će skup isteći.
* Može se izabrati tip odredišta za backup klikom na Disk ili Tape.
* Za napredne opcije kliknuti na Media Options u Select a page panelu.
* Odabrati lokaciju gde će se sačuvati backup i dati odgovarajuće ime datoteci.
* Kliknuti na OK kako bi se pokrenuo backup proces.

A screenshot of a computer

Description automatically generated

Slika 26. Full File backup

A screenshot of a computer

Description automatically generated

Slika 27. Rezultat

1. Transact-SQL (T-SQL) komande:

Može se izvršiti Full File backup korišćenjem T-SQL komandi. Evo detaljnog primera:

A screenshot of a computer

Description automatically generated

Slika 28. Kreiranje Full File backup-a pomoću T-SQL komandi

A screenshot of a computer

Description automatically generated

Slika 29. Rezultat

Primer sa više datoteka:

Ako je potrebno backup-ovati više datoteka, može se koristiti sledeća T-SQL skripta:

BACKUP DATABASE [ImeBazePodataka]

FILE = 'ImeDatoteke1',

FILE = 'ImeDatoteke2'

TO DISK = 'PutanjaDoLokacije\ImeBackupDatoteke.bak'

WITH INIT;

U ovom primeru, zameniti ImeDatoteke1 i ImeDatoteke2 sa imenima datoteka koje je potrebno backup-ovati.

# Partial backup

Parcijalni backup-ovi su korisni kad je potrebno isključiti read-only grupe fajlova. Parcijalni backup podseća na full backup baze podataka, ali parcijalni backup ne sadrži sve grupe fajlova. Umesto toga, za read/write bazu podataka, parcijalni backup sadrži podatke u primarnoj grupi fajlova, svakoj read/write grupi fajlova i, opciono, jednu ili više read-only grupe fajlova. Parcijalni backup read-only baze podataka sadrži samo primarnu grupu fajlova. Koristi se kada baza podataka ima velike read-only delove koji se retko menjaju.

Omogućava češće backup-ovanje delova baze koji se često menjaju, dok se read-only delovi mogu backup-ovati ređe.

Detaljan primer kako se radi parcijalni backup

Priprema:

* Identifikuju se read/write filegroup-ove i read-only filegroup-ove.
* Planira se kada će se izvoditi parcijalni backup-i i kada će se izvoditi full backup-ovi read-only filegroup-ova.

Kreiranje Parcijalnog backup-a pomoću T-SQL komadni:

A screenshot of a computer

Description automatically generated

Slika 30. Kreiranje Parcijalnog backup-a pomoću T-SQL komadni

Objašnjenje komandi:

* READ\_WRITE\_FILEGROUPS: Ključna reč koja uključuje sve read/write filegroup-ove u parcijalni backup.
* TO DISK = ‘PutanjaDoLokacije\ImeBackupDatoteke.bak’: Specifikuje putanju I ime backup datoteke.
* WITH INIT: Inicijalizuje novi backup set, prepisujući prethodne backup-ove na istoj destinaciji.

A screenshot of a computer

Description automatically generated

Slika 31. Rezultat

# Copy-only backup

Copy-only backup je SQL Server backup koji je nezavisan od redosleda konvencionalnih SQL Server backup-ova. Obično, pravljenje backup-a menja bazu podataka i utiče na to kako se kasniji backup-ovi restore-ju. Međutim, povremeno je korisno napraviti backup za posebnu svrhu, a da ne utiče na celokupne procedure backup-a i restore-a baze podataka. Copy-only backup-ovi služe upravo za tu svrhu.

Tipovi copy-only backup-a su:

* Copy-only full backup-ovi (svi recovery modeli)
* Copy-only backup ne može da posluži kao diferencijalna osnova ili diferencijalni backup i ne utiče na diferencijalnu osnovu.
* Restore-ovanje copy-only full backup-a je isto kao restore-ovanje bilo kog drugog full backup-a.
* Copy-only log backup (samo u full i bulk-logged recovery modelu)
* Copy-only log backup čuva postojeću tačku arhiviranja logova i, stoga, ne utiče na sekvencu redovnih log backup-a. Copy-only log backup-ovi su obično nepotrebni. Umesto toga, može se napraviti novi rutinski log backup (koristeći WITH NORECOVERY) i koristiti taj backup zajedno sa svim prethodnim log backup-ovima koji su potrebni za sekvencu restore-ovanja. Međutim, copy-only log backup ponekad može biti koristan za obavljanje online restore-a.
* Transakcioni log nikada nije skraćen nakon copy-only backup-a.

Copy-only backup-ovi su zabeleženi u koloni is\_copy\_only u tabeli backupset.

Kada koristiti Copy-Only Backup?

1. Testiranje i razvoj: Kada je potrebno testirati ili razvijati na bazi podataka bez menjanja postojećeg backup-a.
2. Migracija: Kada je potrebno migrirati podatke na drugi server bez ometanja postojećeg sistema sigurnosnog kopiranja.
3. Ad-hoc backup: Kada je potrebno napraviti trenutnu kopiju baze podataka za posebne potrebe bez ometanja redovnog rasporeda backup-a.

Postoje dva glavna načina za pravljenje copy-only backup-a u SQL Serveru: putem SQL Server Management Studio (SSMS) i putem T-SQL komandi.

* Korišćenje SQL Server Management Studio (SSMS):

1. Potrebno je pokrenuti SSMS i povezati se na odgovarajuću SQL Server instancu.
2. Proširiti čvor Databases i desni klik na bazu podataka koju je potrebno backup-ovati.
3. Izabrati Tasks -> Back Up...
4. U prozoru Back Up Database, u okviru sekcije Backup type, izabrati Full.

A screenshot of a computer

Description automatically generated

Slika 32. Tipa backup-a

1. Označiti polje Copy-only backup.

A screenshot of a computer

Description automatically generated

Slika 33. Izbor Copy-only backup-a

1. Konfigurisati destinaciju backup-a (lokalni fajl, mrežni fajl, itd.).

A screenshot of a computer

Description automatically generated

Slika 34. Lokacija gde će backup biti smešten

1. Kliknuti na OK kako bi se započeo process backup-ovanja.

A screenshot of a computer

Description automatically generated

Slika 35. Rezultat

* Korišćenje T-SQL komandi:

Mogu se koristiti T-SQL komande za pravljenje copy-only backup-a. Sledeći primer prikazuje kako to uraditi:

BACKUP DATABASE [ImeBazePodataka]

TO DISK = 'C:\Putanja\Do\Fajla\ImeBazePodataka\_CopyOnly.bak'

WITH COPY\_ONLY;

Gde je:

* + [ImeBazePodataka] je naziv baze podataka.
  + 'C:\Putanja\Do\Fajla\ImeBazePodataka\_CopyOnly.bak' je lokacija na kojoj će se sačuvati backup.

Ova T-SQL komanda će kreirati copy-only backup fajl na zadatoj lokaciji.

Primer:

A screenshot of a computer

Description automatically generated

Slika 36. Pravljenje copy-only backup-a pomoću T-SQL komandi

A screenshot of a computer

Description automatically generated

Slika 37. Rezultat

# Restore baze podataka

Baza podataka se može vratiti na nekoliko načina u zavisnosti od nanete štete i dostupnih rezervnih kopija baze podataka. U najboljem slučaju, baza podataka se može oporaviti sa minimalnim zastojima i bez gubitka podataka.

Obnavljanje baze podataka zahteva ponovno učitavanje backup-a baze podataka (u bilo kom scenariju). Zbog toga je oporavak baze podataka nemoguć bez pouzdanog backup-a baze podataka. Zbog toga, kompanije moraju uspostaviti pouzdanu strategiju pravljenja backup-a baze podataka kako bi osigurale da se podaci i logovi redovno (i pouzdano) kopiraju na drugi uređaj za skladištenje ili platformu (lokalno, fizički van preduzeća ili u cloudu).

Dobro osmišljen i izvršen plan backup-a smanjuje rizik od gubitka podataka, minimizira vreme zastoja i obezbeđuje uspešnu rekonstrukciju baze podataka. Ipak, čak i ako kompanija ima sveobuhvatnu strategiju oporavka, preporučuje se pravljenje backup-a zasnovano na SQL Server-u i Windows-u.

Pre nego što se restore-ujete baza podataka u kombinaciji sa full ili bulk-logged recovery modelom, potrebno je napravite backup aktivnog transaction log-a (poznatog kao tail of the log).

Da bi se restore-ovala šifrovana baza podataka, potreban je pristup sertifikatu ili asimetričnom ključu koji se koristi za šifrovanje te baze podataka. Bez sertifikata ili asimetričnog ključa, ne može da se restore-uje ta baza podataka. Neophodno je zapamtiti sertifikat koji se koristi za šifrovanje baze podataka dok god je potrebno čuvanje backup-a.

# Restore u SSMS-u

1. **Restore full backup-a baze podataka**
2. U Object Explorer-u, povezati se sa instancom SQL Server Database Engine-a, a zatim proširiti tu instancu.
3. Kliknuti desnim tasterom miša na Databases i izaberite Restore Database...

A screenshot of a computer

Description automatically generated

Slika 38. Restore

1. Na General stranici, koristiti sekciju Source gde se navodi source i lokacija backup skupova za restore-ovanje. Izabrati jednu od sledećih opcija:

* Database: Izabrati bazu podataka koju je potrebno restore-ovati sa padajuće liste. Lista sadrži samo baze podataka za koje je napravljen backup prema msdb backup istoriji. Ako je backup preuzet sa drugog servera, odredišni server neće imati informacije o istoriji backup-a za navedenu bazu podataka. U tom slučaju, izabrati Device da bi se ručno odredila datoteka ili uređaj za vraćanje.
* Device: Izabrati dugme za pretragu (...) da bi se otvorio dijalog Select backup devices.
* Dijalog Select backup devices

Backup media type: Izabrati tip medija sa padajuće liste Backup media type. Napomena: Opcija Tape se pojavljuje samo ako je tape drive montiran na računar, a opcija Backup Device se pojavljuje samo ako postoji barem jedan backup uređaj.

Add: U zavisnosti od tipa medija koji se izabere sa padajuće liste Tip backup medija, klikom na Add otvara se jedan od sledećih dijaloga. (Ako je lista u polju Backup media za listu puna, dugme Add nije dostupno.)

Remove: Uklanja jednu ili više izabranih fajlova, traka ili logičkih uređaja.

Contents: Prikazuje medijski sadržaj izabranog fajla, trake ili logičkog backup uređaja. Ovo dugme možda neće funkcionisati ako je tip medija URL.

Backup media: Prikazuje listu izabranih medija.

Nakon što se dodaju željeni uređaji na Backup media listu, izabrati OK za povratak na General stranicu.

A screenshot of a computer

Description automatically generated

Slika 39. Dijalog Select backup devices

1. U delu Destination, polje Database se automatski popunjava imenom baze podataka koju treba restore-ovati. Da bi se promenilo ime baze podataka, potrebno je uneti novo ime u polje Database.
2. U okviru Restore to, ostaviti podrazumevanu vrednost kao To the last backup taken ili izabrati Timeline da bi se pristupilo Backup Timeline dijalogu kako bi se ručno izabrala tačka u vremenu za zaustavljanje radnje oporavka.
3. U Backup sets to restore grid-u izabrati backup-ove za restore-ovanje. Ovaj grid prikazuje backup-ove dostupne za navedenu lokaciju. Podrazumevano se predlaže recovery plan. Da bi se zamenili predloženi recovery plan, može se promeniti izbor u grid-u. Backup-ovi koji zavise od obnavljanja ranijeg backup-a se automatski poništavaju kada se poništi izbor ranijih backup-ova.
4. Opciono, izabrati Files u Select a page da bi se pristupilo dijalogu Files. Odavde se može restore-ovati baza podataka na novu lokaciju tako što se navede novo odredište restore-ovanja za svaki fajl u Restore the database files as grid-u.
5. Da bi se videle napredne opcije, na stranici Options, u panelu Restore options, može se izabrati bilo koja od sledećih opcija, ako je prikladno za konkretnu situaciju:
6. WITH opcijama (nije obavezna):

* Zameniti postojeću bazu podataka (WITH REPLACE)
* Sačuvati podešavanja replikacije (WITH KEEP\_REPLICATION)
* Ograničiti pristup restore-ovanoj bazi podataka (WITH RESTRICTED\_USER)

1. Izabrati opciju Recovery state. Ovo polje određuje stanje baze podataka nakon operacije restore-ovanja.

* RESTORE WITH RECOVERY je podrazumevano ponašanje koje ostavlja bazu podataka spremnom za upotrebu vraćanjem uncommited transakcija. Ne mogu se restore-ovati nikakve dodatne log transakcije. Izabrati ovu opciju ako se restore-uju svi potrebni backup-ovi.
* RESTORE WITH NORECOVERY što ostavlja bazu podataka neoperativnom i ne vraća uncommited transakcije. Dodatne log transakcije se mogu restore-ovati. Baza podataka se ne može koristiti dok se ne restore-uje.
* RESTORE WITH STANDBY što stavlja bazu podataka u read-only režimu. Poništava uncommited transakcije, ali čuva akcije poništavanja u standby fajlu tako da se efekti oporavka mogu vratiti.

1. Napraviti tail-log backup pre restore-ovanja. Ne zahtevaju sve restore situacije tail-log backup-ove.
2. Restore operacije mogu biti neuspešne ako postoje aktivne konekcije sa bazom podataka. Čekirati opciju Close existing connections da bi se osiguralo da su sve aktivne konekcije između Management Studio-a i baze podataka zatvorene. Ova opcija setuje bazu podataka u single user režimu pre restore operacije, a kada završi setuje je u multi-user režimu.
3. Izabrati Prompt before restoring each backup je potebno da iskače obaveštenje između svake restore operacije. Ovo nije neophodno osim ako je baza podataka velika i ako je potrebno pratiti status restore operacije.

A screenshot of a computer

Description automatically generated

Slika 40. Napredne opcije kod restore-a

1. Selektovati OK.

A screenshot of a computer

Description automatically generated

Slika 41. Restore full backup-a

1. **Restore ranijeg backup sa diska preko postojeće baze podataka**

Sledeći primer restore-uje raniji backup sa diska za Sport school i zamenjuje postojeću bazu podataka Sport school.

1. U Object Explorer-u, povezati se sa instancom SQL Server Database Engine-a, a zatim proširiti tu instancu.
2. Kliknuti desnim tasterom miša na Databases I izabrati Restore Database...
3. Na stranici General, izabrati Device u sekciji Source.
4. Izabrati dugme za pretragu (...) da bi se otvorio dijalog Select backup devices. Izabrati Add I navigirati se do backup-a. Izabrati OK nakon što se izebere backup fajl/fajlove diska.

A screenshot of a computer

Description automatically generated

Slika 42. Dijalog Select backup devices

1. Izabrati OK za povratak na stranicu General.
2. Izabrati Options u panelu Select a page.
3. U sekciji Restore options, izabrati Overwrite the existing database (WITH REPLACE).
4. U sekciji Tail-log backup, odčekirati Take tail-log backup before restore.
5. U sekciji Server connections, čekirati Close existing connections to destination database.

A screenshot of a computer

Description automatically generated

Slika 43. Dijalog Options

1. Izabrati OK.

A screenshot of a computer

Description automatically generated

Slika 44. Backup sa diska

**C. Restore ranijeg backup sa diska sa novim imenom baze podataka gde originalna baza podataka još uvek postoji**

Sledeći primer restore-uje raniji backup sa diska za Sport school € kreira novu bazu podataka pod nazivom SportSchoolTest. Originalna baza podataka, Sport school, još uvek postoji na server.

1. U Object Explorer-u, povezati se sa instancom SQL Server Database Engine-a, a zatim proširiti tu instancu.
2. Kliknuti desnim tasterom miša na Databases € izabrati Restore Database...
3. Na stranici General izabrati Device u sekciji Source.
4. Izabrati dugme za pretragu (...) da kako bi se otvorio dijalog Select backup devices. Izabrati Add € navigirati se do backup-a. Izabrati OK nakon što se izabere backup fajl€ diska.
5. Izabrati OK za povratak na stranicu General.
6. U sekciji Destination, polje Database se automatski popunjava imenom baze podataka koju treba restore-ovati. Da bi se promenilo ime baze podataka, uneti novo ime u polje Database.
7. Izabrati Options u panelu Select a page.
8. U sekciji Tail-log backup, odčekirati „Take tail-log backup before restore“.

A screenshot of a computer

Description automatically generated

Slika 45. Opcije pri restore-ovanja

1. Izabrati OK.

A screenshot of a computer

Description automatically generated

Slika 46. Restore-uje raniji backup sa diska sa novim imenom baze podataka

A screenshot of a computer

Description automatically generated

Slika 47. Restore-ovana baza pod novim nazivom

### Restore pomoću T-SQL komandi

### Restore Full backup-a

Full backup je osnovna vrsta backup-a koja sadrži celu bazu podataka. Restore procesa full backup-a je prvi korak pri oporavku baze podataka. Recimo da je full backup uskladišten u C:\Backups\Sport school\_FULL\_TSQL.bak i da ga je potrebno vratiti u bazu podataka Sport school. Potrebno je izvršiti sledeću komandu:

A screenshot of a computer

Description automatically generated

Slika 48. Restore full backup-a

Ako je nakon toga potrebno nastaviti sa vraćanjem diferencijalnih ili transaction log backup-ova, potrebno je dodati opciju NORECOVERY. Ovo će ostaviti backup proces u restoring stanje i omogućiti da se vrate diferencijalni ili transaction log backupovi.

A screenshot of a computer

Description automatically generated

Slika 49. Restore full backup-a sa opcijom NORECOVERY

# Restore Diferencijalnog backup-a

Diferencijalni backup sadrži samo promene napravljene od poslednjeg full backup-a. Poslednji diferencijalni backup akumulira sve promene tako da nisu potrebni svi prethodni diferencijalni backup-ovi da bi se restore-ovala baza podataka, već samo poslednji. Pre nego što se restore-uje diferencijalni backup, mora se prvo restore-ovati poslednji full backup sa opcijom NORECOVERY, a zatim poslednji diferencijalni backup sa opcijom RECOVERY:

A screenshot of a computer

Description automatically generated

Slika 50. Restore diferencijalnog backup-a

# Restore Transaction log backup-ova

Transaction Log Backup-ovi sadrže sve transakcije koje su se desile između Transaction Log Backup-a (ili prvog full backup-a) I trenutka kada je rocess pravljenja backup-a završen. Potrebno je da se restore-uju svi transaction log backup-ovi napravljeni nakon poslednjeg diferencijalnog backup-a istim redosledom u kom su napravljeni. Transaction log backup-ovi se restore-uju nakon što se završi sa full I diferencijalnim backup-ovima:

A screenshot of a computer

Description automatically generated

Slika 51. Restore transaction log backup-a

# Restore Tail-log backup-a

Tail-log backup se koristi da uhvati sve transakcije koje su se dogodile neposredno pre problema. On se vraća poslednji u procesu oporavka.

Pretpostavimo da imamo bazu podataka Sport school, koju treba oporaviti. Pre nego što se započne restore proces, napravljen je tail-log backup, a sada ga treba vratiti.

Pre nego što započnete restore operacije, napravite tail-log backup kako biste sačuvali sve trenutne transakcije.

Prvo je potrebno restore-ovati full backup baze podataka. Zatim ako postoji diferencijalni backup, vratiti ga nakon full backup-a. Nakon toga vratiti sve log backup-ove nastale nakon poslednjeg diferencijalnog backup-a. Na kraju, vratiti tail-log backup kako bi se sačuvale sve poslednje transakcije:

A screenshot of a computer

Description automatically generated

Slika 52. Restore tail-log backup-a

# Restore Full File backup-a

Ponekad je potrebno vratiti pojedinačne fajlove ili grupe fajlova unutar baze podataka. Ovo je posebno korisno za velike baze podataka ili baze koje sadrže podatke na različitim diskovima.

Prvo, mora se vratiti full backup fajla ili filegroup-a. Ovaj korak vraća sve podatke u fajlu do tačke kada je backup napravljen. Ako postoji diferencijalni backup za fajl, mora se vratiti nakon vraćanja full backup-a. Ovaj korak vraća sve promene koje su se desile od poslednjeg full backup-a do trenutka pravljenja diferencijalnog backup-a. Nakon što se vrati full I diferencijalni backup fajla, moraju se vratiti svi log backup-ovi koji pokrivaju vreme od poslednjeg full ili diferencijalnog backup-a do trenutka pravljenja tail-log backup-a. Ovo obnavlja sve transakcije koje su se dogodile u tom eriod. Konačno, vratiti tail-log backup kako bi se sačuvale sve transakcije koje su se dogodile neposredno pre kvara ili pre početka restore procesa, I postavili bazu u online stanje.

A screenshot of a computer program

Description automatically generated

Slika 53. Restore full file backup-a

# Restore Partial backup-a

Partial backup uključuje sve read/write fajlove i opcionalno jednu ili više read-only grupa fajlova.

A screen shot of a computer

Description automatically generated

Slika 54. Restore partial backup-a

# Restore Copy-Only backup

Restore process za copy-only backup je isti kao I za redovni backup. Važno je napomenuti da copy-only backup ne menja sekvencu backup-ova, tako da se može nastaviti sa regularnim restore operacijama.

T-SQL komanda za restore Copy-Only backup-a:

A close-up of a computer screen

Description automatically generated

Slika 55. Restore copy-only backup-a

# Zaključak

Backup i restore kod SQL Server baza podataka imaju vitalnu ulogu u osiguravanju kontinuiteta poslovanja i zaštiti podataka. Ovi procesi su neophodni za prevenciju gubitka podataka i omogućavanje brzog oporavka u slučaju gubitka podataka. Efikasna strategija backup-ovanja omogućava organizacijama da redovno prave kopije podataka, smanjujući rizik od trajnog gubitka važnih informacija.

Implementacija različitih tipova backup-ova, kao što su full backup, diferencijalni backup i transakcioni log backup, omogućava fleksibilnost i efikasnost u upravljanju podacima. Full backup pruža potpunu kopiju baze podataka, dok diferencijalni i transakcioni log backup-ovi omogućavaju brže i manje backup-ove, fokusirajući se na promene nastale od poslednjeg full backup-a. Ova kombinacija različitih backup strategija osigurava optimalnu zaštitu podataka uz efikasno korišćenje resursa.

Restore procesi su ključni za vraćanje podataka u slučaju gubitka ili korupcije. Mogućnost brzog i pouzdanog vraćanja podataka osigurava minimalno vreme zastoja i održava kontinuitet poslovnih operacija. Redovno testiranje restore procedura je neophodno kako bi se osiguralo da su backup-ovi ispravni i da se mogu uspešno koristiti u slučaju potrebe.

Uspešno upravljanje backup-om i restore-om zahteva kontinuiranu pažnju i resurse. Organizacije moraju ulagati u adekvatnu infrastrukturu, obuku osoblja i redovnu reviziju svojih strategija backup-ovanja kako bi ostale efikasne i prilagodljive promenama u poslovnom okruženju. Uz pravilno planiranje i implementaciju, backup i restore procesi postaju pouzdan temelj za zaštitu podataka, omogućavajući organizacijama da se suoče sa izazovima i neprekidno održavaju svoje operacije.

Backup i restore procesi nisu samo tehnički zadaci, već ključni elementi strategije upravljanja rizicima i osiguranja kontinuiteta poslovanja. Pažljivo planiranje, redovna implementacija i kontinuirano poboljšavanje ovih procesa osiguravaju da podaci ostanu sigurni i dostupni, čime se podržava stabilnost i dugoročni uspeh organizacije.

# Literatura

[Backup overview (SQL Server) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/backup-overview-sql-server?view=sql-server-ver16)

[Recovery Models (SQL Server) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/recovery-models-sql-server?view=sql-server-ver16)

[Copy-only backups - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/copy-only-backups-sql-server?view=sql-server-ver16)

[Full database backups (SQL Server) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/full-database-backups-sql-server?view=sql-server-ver16)

[Differential Backups (SQL Server) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/differential-backups-sql-server?view=sql-server-ver16)

[Transaction log backups - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/transaction-log-backups-sql-server?view=sql-server-ver16)

[Full File Backups (SQL Server) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/full-file-backups-sql-server?view=sql-server-ver16)

[Partial Backups (SQL Server) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/partial-backups-sql-server?view=sql-server-ver16)

<https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/tail-log-backups-sql-server?view=sql-server-ver16>

<https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-database-backup-using-ssms?view=sql-server-ver16>

<https://www.acronis.com/en-us/blog/posts/how-to-restore-database-from-backup/>

<https://sqlbackupandftp.com/blog/restore-database-backup/>