Univerzitet u Nišu,

Elektronski fakultet Niš

Logo

Description automatically generated

*Seminarski rad*

Sistemi za upravljanje bazama podataka

**BACKUPRESTORE KOD MICROSOFT SQL SERVER-A**

Mentor: Student:

Aleksandar Stanimirović Andrija Milosavljević 1481

Niš, april 2023. godine

**Sadržaj**

[**1.** **Uvod** 3](#_Toc135491158)

[**2.** **Tipovi rezervnih kopija** 5](#_Toc135491159)

[**2.1.** **Full backup** 6](#_Toc135491160)

[**2.2.** **Differential backup** 7](#_Toc135491161)

[**2.3.** **Transaction log backup** 9](#_Toc135491162)

[**2.4.** **Copy-only backup** 10](#_Toc135491163)

[**2.5.** **File backups** 11](#_Toc135491164)

[**3.** **Restore baze podataka** 13](#_Toc135491165)

[**4.** **Zaključak** 16](#_Toc135491166)

[**5.** **Literatura** 17](#_Toc135491167)

# **Uvod**

Podaci su postali nezaobilazni deo savremenog društva i imaju ključnu ulogu u svim sferama života. Oni su važni u poslovnom svetu jer omogućavaju kompanijama da bolje razumeju svoje klijente i tržište, što im pomaže da se dalje razvijaju i poboljšavaju svoje proizvode i usluge. Bez podataka, kompanije ne bi bile u mogućnosti da optimizuju svoje procese, povećaju produktivnost i smanje troškove. Samim tim zaštita podataka je od velike važnosti, jer njihov gubitak može imati ozbiljne posledice.

Backup je proces izrade rezervne kopije podataka kako bi se podaci zaštitili od gubitka usled oštećenja, kvarova hardvera ili softvera, napada ili drugih nepredviđenih događaja. Backup proces obično podrazumeva kreiranje redovnih kopija podataka u planiranim vremenskim intervalima, često svakodnevno ili nedeljno, i čuvanje tih kopija na sigurnom mestu.

Sistem za upravljanje bazama podataka (DBMS) omogućava korisnicima da kreiraju, uređuju, skladište i upravljaju bazama podataka. DBMS se sastoji od skupa programa koji korisnicima pružaju efikasno i pouzdano upravljanje velikim količinama podataka. On je odgovoran za osiguranje integriteta i sigurnost podataka, tako što pruža sigurnosne mehanizme za zaštitu podataka, enkripciju podataka i kontrolu pristupa. On takođe omogućava korisnicima kreiranje rezervne kopije podataka i u slučaju da dođe do gubitka podataka, da brzo obnove podatke iz rezervnih kopija i vrate bazu podataka u prethodno stabilno stanje.

Pravljenje rezervne kopije podataka podrazumeva kreiranje kopije podataka i skladištenje na bezbednoj lokaciji, kao što su eksterni hard diskovi ili sve češće u današnjem vremenu, cloud rešenja. Posedovanje važećih rezervnih kopija baze podataka je od suštinskog značaja za oporavak ukoliko dođe do problema. Problemi koji uzrokuju oštećenje ili gubitak podataka, i dobar su razlog zašto treba redovno praviti rezervne kopije, možemo podeliti u nekoliko kategorije [1]:

* Korisničke greške – na primer, nenamerno brisanje tabele ili izmena podataka
* Hardverske greške – na primer, oštećenje hard diska ili oštećenje servera
* Destruktivni uticaji – na primer, virusi, hakeri, itd.
* Prirodne katastrofe – na primer, požar, poplava, zemljotres, itd.

Rezervna kopija podataka osim što se koristi za vraćanje podataka u slučaju gubitka, takođe se koristi i za kopiranje baze podataka sa jednog servera na drugi. Ovo je korisno kada treba prebaciti bazu podataka na novi server ili za kreiranje kopije radi testiranja baze podataka.

Proces vraćanja (eng. restore) baze podataka uključuje oporavak baze podataka iz rezervne kopije i vraćanje na njen prethodni validan trenutak. Pravljenje rezervne kopije i vraćanje podataka treba prilagoditi određenom okruženju i raspoloživim resursima. Samim tim, pouzdana upotreba rezervne kopije i vraćanje podataka zahteva strategiju za pravljenje rezervnih kopija i vraćanje podataka. Dobro osmišljena strategija treba da uravnoteži poslovne zahteve za maksimalnu dostupnost podataka i minimalan gubitak podataka, uzimajući u obzir troškove održavanja i čuvanja rezervnih kopija.

Strategija sadrži dva dela: deo za pravljenje rezervne kopije i deo za vraćanje podataka. Deo strategije za backup definiše tip i učestalost pravljenja rezervnih kopija, prirodu i brzinu hardvera koji je zadužen za njih, način na koji se testiraju kreirane kopije, kao i gde i kako treba čuvati rezervne kopije. Deo strategije za restore definiše ko je odgovoran za obavljanje vraćanja, kako se vraćanje treba izvršiti kako bi se postigli ciljevi vezani za dostupnost baze podataka i minimiziranje gubitka podataka, kao i način na koji se restore proces testira. Važno je da obe strane strategije budu pažljivo osmišljene i testirane kako bi se obezbedila pouzdana sigurnost podataka i efikasan oporavak [1].

Da bi se osmislila efikasna strategija pravljenja rezervne kopije i vraćanja, potrebno je pažljivo planiranje, implementacija i testiranje. Strategija nije uspešna sve dok se sve rezervne kopije uspešno ne vrate u svim kombinacijama koje su uključene u strategiju vraćanja podataka i dok se ne testira fizička konzistentnost vraćene baze podataka. Određivanje adekvatne strategije je složen proces i zahteva detaljno razmatranje faktora kao što su: zahtevi za dostupnost i zaštitu podataka, veličina i priroda sadržaja baze podataka, ograničenja resursa kao što su hardver, osoblje, prostor za skladištenje rezervnih sadržaja, itd.

# **Tipovi rezervnih kopija**

Microsoft SQL Server nudi nekoliko vrsta rezervnih kopija, od kojih će svaka detaljno biti obrađena u narednim poglavljima [2]:

* Full backup,
* Differential backup,
* Transaction log backup,
* Tail-log backup,
* Copy-only backup,
* File backup,
* Partial backup.

Sve vrste rezervnih kopija je moguće kreirati i pomoću *SQL Server Management Studio*-a (*Database → Tasks → Back Up…*) (*Slika 2.1.*).

A screenshot of a computer

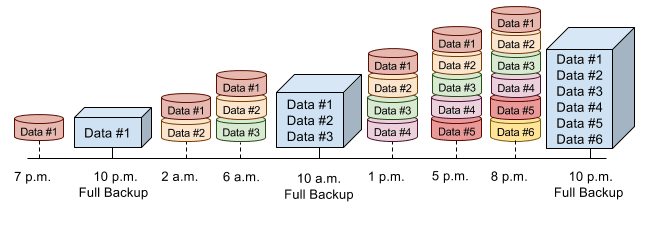
Description automatically generated

Slika 2.1. Kreiranje backup-a pomoću SQL Server Management Studio-a

## **Full backup**

Full backup (ili potpuna rezervna kopija), kao što samo ime govori, kopira celu bazu podataka, uključujući sve podatke i objekte baze podataka, kao što su: tabele, procedure, funkcije, poglede, indekse, itd. Ona uključuje i deo transaction log-a, kako bi se baza podataka mogla u potpunosti vratiti na trenutak kada je rezervna kopija napravljena.

Potpuna rezervna kopija predstavlja osnovu za ostale tipove rezervnih kopija i omogućava najjednostavniji način vraćanja baze podataka, jer se svi njeni sadržaji nalaze u jednoj datoteci. Ona takođe pruža najpotpuniju zaštitu od gubitka podataka i najmanje zavisi od drugih tipova backup-a. Međutim, kako se baza podataka povećava, za izradu ove vrste kopije je potrebno mnogo više vremena i resursa, što se mora uzeti u obzir prilikom planiranja strategije. Na *slici 2.2.* možemo videti raspored periodičnog pravljenja potpune rezervne kopije [5].



Slika 2.2. Raspored pravljenja potpune rezervne kopije u određenom periodu

U sledećem primeru možemo videti T-SQL naredbu kreiranja potpune rezervne kopije baze podataka:



Primer 2.1. Naredba kreiranje potpune rezervne kopije

U situaciji kada je baza podataka veoma velika, rezervnu kopiju možemo podeliti na manje delove ili zasebne datoteke (*Primer 2.2*). Ovo omogućava da proces pravljenja rezervne kopije bude brži i efikasniji. Podelom rezervne kopije, svaka datoteka sadrži samo deo podataka, što olakšava prenos ili skladištenje. Pored toga, smanjuje se rizik od gubitka podataka, jer ako se jedna od datoteka ošteti ili izgubi, preostale datoteke se mogu koristiti za obnavljanje baze podataka.



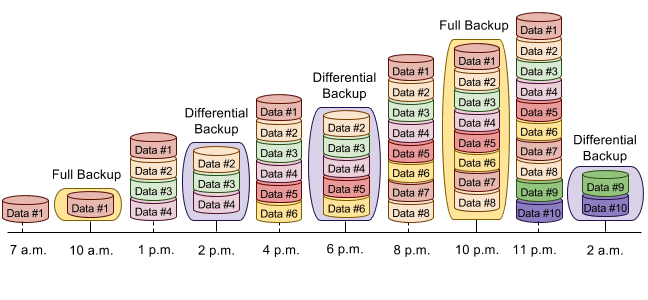
Primer 2.2. Naredba kreiranja potpune rezervne kopije podeljene u više zasebnih datoteka

## **Differential backup**

Diferencijalna rezervna kopija baze podataka je dizajnirana kako bi se smanjilo vreme trajanja izrade rezervne kopije. Princip izrade diferencijalne rezervne kopije podataka je kopiranje samo onih stranica baze podataka koje su se promenile od poslednje potpune rezervne kopije. Ovo olakšava pravljenje čestih rezervnih kopija, što smanjuje rizik od gubitka podataka.

Diferencijalna rezervna kopija se temelji na potpunoj rezervnoj kopiji baze podataka, ona takođe kopira i transaction log kako bi se podaci mogli vratiti u potpunosti. Osnovna prednost diferencijalne kopije u odnosu na potpunu je manje vremena za pravljenje kopije, jer sadrži manje podataka, ukoliko se rezervne kopije izvode redovno. Osim toga, izrada diferencijalne rezervne kopije zahteva manje resursa, jer ne kopira sve podatke, pa se tako može uštedeti prostor [4].

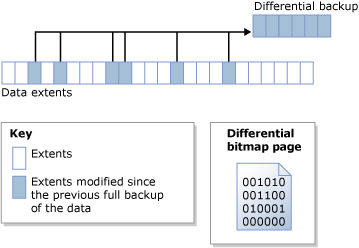
Izrada diferencijalne rezervne kopije može biti mnogo brža u odnosu na izradu potpune rezervne kopije baze podataka. Međutim, prilikom povratka diferencijalne rezervne kopije, potrebno je prvo vratiti najnoviju potpunu rezervnu kopiju, a zatim primeniti diferencijalnu rezervnu kopiju. Kako veličina diferencijalne rezervne kopije raste prilikom svake sledeće izrade, ona gubi prednost u odnosu na potpunu rezervnu kopiju, a to su brži oporavak podataka i manje iskorišćenih resursa. Samim tim je potrebno periodično izrađivati potpune rezervne kopije, kako bi svaka nova diferencijalna kopija koristila što manje resursa (*Slika 2.3*).



Slika 2.3. Raspored pravljenja diferencijalne rezervne kopije nakon potpune kopije

Diferencijalne rezervne kopije baze podataka mogu biti veoma korisne u situacijama kada se određeni delovi baze podataka često menjaju. Na primer, ako baza podataka sadrži veliku tabelu koja se često ažurira, a većina drugih tabela je relativno statična. Često pravljenje rezervnih kopija cele baze podataka bi predstavljao gubitak i vremena i resursa. U ovom slučaju, diferencijalna rezervna kopija se može koristiti samo za promene napravljenih u tabeli koja se često ažurira.

Diferencijalna rezervna kopija beleži stanje svih *extent*s-a (kolekcija od 8 fizički susednih stranica) koje su se promenile od poslednje potpune rezervne kopije. Na *slici 2.4.* možemo videti princip izrade diferencijalne rezervne kopije baze podataka. Prikazana su 24 *extents*-a, od kojih su 6 označeni kao promenjeni od poslednje potpune rezervne kopije. Izrada diferencijalne kopije se oslanja na stranicu polja bitova (eng. bitmap page) koja sadrži jedan bit za svaki *extents*. Za svaki *extents* koji je promenjen, bit se postavlja na 1 u stranici polja bitova. Diferencijalna kopija baze podataka se sastoji samo od *extents*-a kojima su bitovi u stranici polja bitova postavljeni na vrednost 1.



Slika 2.4. Princip izrade diferencijalne rezervne kopije

U sledećem primeru možemo videti T-SQL naredbu kreiranja diferencijalne rezervne kopije baze podataka:



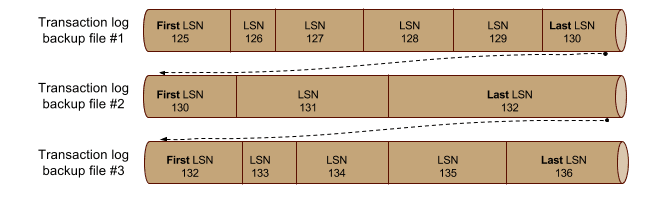
Primer 2.3. Naredba kreiranja diferencijalne rezervne kopije

## **Transaction log backup**

Transaction log (ili evidencija izvršenih transakcija) podrazumeva skladištenje evidencija transakcija koje su se izvršile od poslednje potpune rezervne kopije baze podataka. Samim tim je potrebno izraditi barem jednu potpunu rezervnu kopiju kako bi bilo moguće izraditi evidenciju izvršenih transakcija. Svaka evidencija sadrži detalje koje se odnose na kreiranje i izmenu objekata, korišćenjem DDL naredbi, kao i svaku izmenu podataka, korišćenjem DML naredbi.

Evidencije izvršenih transakcija se zapisuju sekvencijalno, pa je u slučaju oporavka podataka, moguće povratiti operacije nad objektima i podacima u istom redosledu kao što su se i izvršile. Na taj način, moguće je vratiti bazu podataka u tačno određeno stanje. Svaka evidencija izvršenih transakcija je označena sekvencijalnim brojem evidencije (eng. Log Sequence Number – LSN). Najmanji LSN označava najstariju evidenciju i početak transakcije, dok najveći broj označava najnoviju evidenciju i kraj transakcije.

Na *slici 2.5*. vidimo da su evidencije transakcija povezane u lanac. Lanac je neprekidni niz koji sadrži sve evidencije neophodne za oporavak baze podataka do određenog trenutka. Lanac uvek počinje potpunom rezervnom kopijom baze podataka i nastavlja se nizom evidencija izvršenih transakcija [5].



Slika 2.5. Način skladištenja evidencija izvršenih transakcija

Primarni razlog za izradu evidencija izvršenih transakcija je čuvanje transakcija koje su se dogodile od poslednje potpune rezervne kopije baze podataka, kako bi se pomoću transakcija baza podataka dovela u pređašnje stabilno stanje.

U sledećem primeru možemo videti T-SQL naredbu kreiranja transaction log rezervne kopije baze podataka:



Primer 2.4. Naredba kreiranja transaction log backup-a

Tail-log backup je vrsta transaction log backup-a koja obuhvata sve potvrđene transakcije koje su izvršene od trenutka poslednjeg transaction log backup-a. Ova vrsta rezervne kopije se koristi u situacijama kada je potrebno oporaviti bazu podataka do poslednjeg trenutka pre nego što je nastao problem koji je doveo do neuspeha baze podataka.

Kada se radi tail-log backup, beleže se sve transakcije koje su završene i potvrđene, ali koje nisu bile uključene u prethodnu evidenciju izvršenih transakcija. Ovo se radi kako bi se obezbedilo da nijedna transakcija ne bude izgubljena prilikom oporavka baze podataka. Važno je napomenuti da se ova vrsta rezervne kopije može izvršiti samo kada je baza podataka u režimu potpunog (eng. full recovery) ili bulk-logged oporavka. Ova vrsta rezervne kopije se koristi u kritičnim situacijama, gde je minimalni gubitak podataka od vitalnog značaja [4].

U sledećem primeru možemo videti T-SQL naredbu kreiranja tail-log rezervne kopije baze podataka:



Primer 2.5. Naredba kreiranja tail-log backup-a

## **Copy-only backup**

Copy-only backup je vrsta backup-a u SQL Serveru koja je nezavisna od ostale grupe backup-ova. Obično, pravljenje rezervnih kopija menja bazu podataka i utiče na način kasnijeg oporavka, međutim povremeno je korisno napraviti rezervnu kopiju u posebne svrhe, a da pri tome ne utiče na opšte postupke rezervnih kopija i oporavka baze podataka. Baš u takvim situacijama se koristi copy-only backup.

Kada se napravi copy-only rezervna kopija, ne prave se nikakve promene na bazi podataka koje bi uticale na redosled i način oporavka kasnijih rezervnih kopija. Copy-only rezervna kopija se koristi kada želimo napraviti rezervnu kopiju za posebne potrebe, kao što je testiranje, kopiranje baze podataka na drugi server, arhiviranje, a da pri tome ne utičemo na standardne postupke pravljenja rezervnih kopija i oporavka.

U sledećem primeru možemo videti T-SQL naredbu kreiranja copy-only rezervne kopije baze podataka:

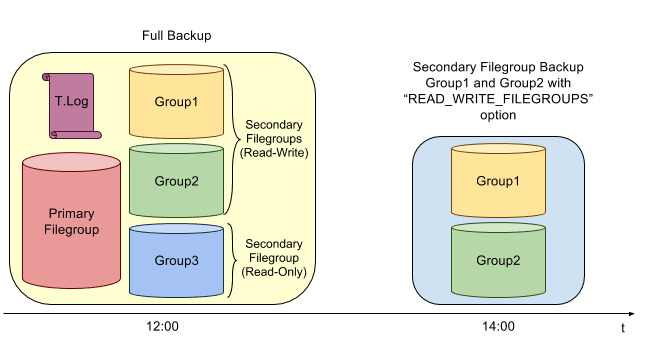


Primer 2.6. Naredba kreiranja copy-only backup-a

## **File backups**

File backup (ili rezervna kopija datoteka) omogućavaju kreiranje kopije samo određenih fajlova ili grupe fajlova koji čine bazu podataka, umesto da se pravi potpuna rezervna kopija baze podataka. Postoji nekoliko vrsta rezervnih kopija datoteka: potpuna rezervna kopija datoteka (eng. full file backup), diferencijalna rezervna kopija datoteka (eng. differential file backup) i delimična rezervna kopija datoteka (eng. partial file backup).

Potpuna rezervna kopija datoteka kopira sve podatke i objekte u datoteci ili grupi datoteka. Korišćenjem ove vrste rezervnih kopija se može povećati brzina oporavka, dopuštajući oporavak samo oštećenih datoteka, bez potrebe za oporavkom cele baze podataka. Zatim, povećava se fleksibilnost u dizajniranju rasporeda strategije pravljenja rezervnih kopija [5].



Slika 2.6. Raspored pravljenja poptune rezervne kopije datoteka nakon potpune kopije

U sledećem primeru možemo videti T-SQL naredbu kreiranja potpune rezervne kopije datoteka:



Primer 2.7. Naredba kreiranja potpune rezervne kopije datoteka

Diferencijalna rezervna kopija datoteka (eng. differential file backup) kopira samo one stranice baze podataka koje su promenjene od poslednje potpune rezervne kopije datoteka. Kao i u diferencijalnoj rezervnoj kopiji baze podataka, i rezervna kopija datoteka može smanjiti broj zapisa izvršenih promena nad datotekama, koje su potrebne prilikom oporavka tih datoteka. U poređenju sa diferencijalnom kopijom baze podataka, prednost diferencijalne kopije datoteka je kreće vreme izvođenja i mogućnost kopiranja promena u tačno određenim datotekama.

Delimična rezervna kopija datoteka (eng. partial file backup), kao i potpuna i diferencijalna, omogućava kopiranje delova baze podataka, bez potrebe za kopiranjem cele baze podataka. Međutim, delimična rezervna kopija datoteka ne uključuje grupe fajlova koji služe samo za čitanje (eng. read-only filegroups). Ova vrsta rezervne kopije je korisna kada baza podataka sadrži jednu ili više grupa fajlova koje su samo za čitanje i nema potrebe da se oni kopiraju svaki put.

Delimična rezervna kopija omogućava da se kopiraju samo aktivni i promenljivi delovi baze podataka, dok se read-only grupe fajlova preskaču. Ovo može biti korisno za velike baze podataka koje sadrže velike read-only grupe fajlova, kao što su arhivski podaci koji se retko menjaju. Time se smanjuje vreme i resursi potrebni za pravljenje rezervne kopije.

Kao i kod kreiranja rezervnih kopija baze podataka, takođe možemo kreirati rezervne kopije datoteka pomoću *SQL Server Management Studio*-a, izborom „*Files and filegroups*“ komponente umesto „*Database*“ (*Slika 2.7.*).

A screenshot of a computer

Description automatically generated

Slika 2.7. Kreiranje file backup-a pomoću SQL Server Management Studio-a

# **Restore baze podataka**

Model oporavka baze podataka definiše način na koji se čuvaju i održavaju transakcije, kao i način na koji se vrši oporavak i vraćanje podataka u slučaju neuspeha. SQL Server podržava 3 modela oporavka [3]:

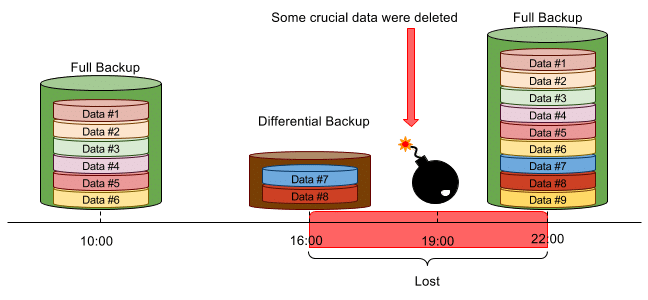
1. Jednostavan model oporavka (eng. Simple recovery model) – moguće je izvršiti backup potpune i diferencijalne rezervne kopije baze podataka, dok nije moguće izvršiti transaction log backup. Ovaj model oporavka je pogodan za jednostavne baze podataka sa manjim zahtevima za oporavak.
2. Potpuni model oporavka (eng. Full recovery model) – za razliku od jednostavnog modela, u ovom modelu se sve transakcije čuvaju u transaction log backup-u. Prilikom opravka podataka se koristi kombinacija potpunog, diferencijalnog i transaction log backup-a.
3. Bulk-logged model oporavka – sličan je potpunom modelu, ali ima određena ograničenja. On je optimizovan za masovne operacije upisivanja/ažuriranja kako bi se smanjio uticaj na performanse. Prilikom oporavka podataka koristi se kombinacija potpunog, diferencijalnog i transaction log backup-a.

Da bi se izvršio oporavak baze podataka nakon neuspeha, potrebno je vratiti skup rezervnih kopija u logički ispravnom i smislenom redosledu. SQL Server podržava nekoliko tipova vraćanja podataka [3]:

1. Vraćanje celokupne baze podataka (eng. complete database restore) – obnavlja celu bazu podataka iz rezervnih kopija, uključujući sve fajlove i strukture podataka koji čine bazu podataka. Ovo je najopširniji oblik vraćanja podataka i koristi se kada je potrebno vratiti celokupnu bazu podataka nakon neuspeha.
2. Vraćanje datoteke sa podacima (eng. file restore) – obnavlja pojedinačne fajlove ili grupe fajlova sa podacima iz rezervnih kopija. Ovaj postupak je koristan kada je samo određeni fajl sa podacima oštećen ili izgubljen, a ostatak baze podataka je sačuvan. Na ovaj način možemo obnoviti samo oštećeni deo podataka.
3. Vraćanje stranice sa podacima (eng. page restore) – obnavlja pojedinačne stranice sa podacima iz rezervnih kopija. Ova vrsta vraćanja podataka se koristi u retkim slučajevima kada je samo mali deo podataka oštećen ili izgubljen. Obnavljanje stranica sa podacima se obično vrši automatski tokom procesa oporavka, ali se može i ručno izvršiti ukoliko je potrebno.

U narednim primerima možemo videti način na koji se koristi vraćanje podataka i upotreba *RESTORE* naredbe u SQL Serveru [5]:

1. *Primer jednostavnog oporavka* (*Slika 3.1.*): Potpuna rezervna kopija baze podataka je napravljena u 10h i 22h, a diferencijalna rezervna kopija u 16h. Pretpostavimo da se u 19h dogodio problem i da su neki ključni podaci obrisani. U ovom slučaju, najbolje je vratiti potpunu rezervnu kopiju (iz 10h), a zatim diferencijalnu rezervnu kopiju (iz 16h) (*Primer 3.1.*). Problem u ovom slučaju oporavka je to da će sve promene koje su nastale između 16h i 22h biti izgubljene.

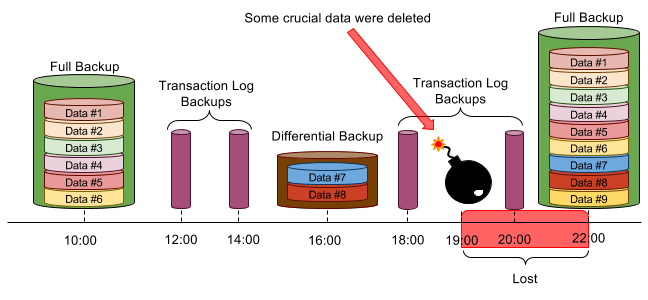


Slika 3.1. Raspored pravljenja backup-a i nastanak problema



Primer 3.1. Naredbe vraćanja podataka prilikom jednostavnog oporavka baze podataka

1. *Primer potpunog oporavka* (*Slika 3.2.*):Potpuna rezervna kopija baze podataka je napravljena u 10h i 22h, diferencijalna rezervna kopija u 16h, a transaction log backup u 12, 14, 18 i 20h. Problem je nastao u 19h, što znači da će sve promene između 19h i 22h biti izgubljene. Najbolji način za oporavak podataka u ovom slučaju jeste vraćanje potpune rezervne kopije (iz 10h), diferencijalne kopije (iz 16h) i rezervne kopije evidencije izvršenih transakcija (18h i 20h) (*Primer 3.2.*).



Slika 3.2. Raspored pravljenja backup-a i nastanak problema

****

Primer 3.2. Naredbe vraćanja podataka prilikom potpunog oporavka baze podataka

Restore proces je moguće izvršiti i pomoću *SQL Server Management Studio*-a   
(*Database Tasks Restore*), izborom vrste restore procesa, zatim vrste backup-a koji koristimo za vraćanje podataka (*Slika 3.3.*).

A screenshot of a computer

Description automatically generated

Slika 3.3. Proces vraćanja podataka pomoću SQL Server Management Studio-a

# **Zaključak**

Backup i restore procesi kod sistema za upravljanje bazama podataka (DBMS) su ključni prilikom zaštite podataka i omogućavaju oporavak u slučaju gubitka ili oštećenja podataka. Backup proces omogućava sigurnost podataka tako što pravi kopije originalnih podataka i čuva ih na sigurnom mestu. U slučaju gubitka ili oštećenja originalnih podataka usled hardverskih kvarova, grešaka korisnika, zlonamernih napada ili prirodnih katastrofa, backup podaci se mogu koristiti za obnovu izgubljenih ili oštećenih podataka. Ako dođe do nenamernog brisanja, izmene ili oštećenja podataka, backup omogućava njihovo vraćanje na prethodno stanje, čime se čuva njihova tačnost i pouzdanost.

Postoji nekoliko osnovnih tipova rezervnih kopija koje podržava Microsoft SQL Server, uključujući potpuni (eng. full backup) koji čuva kompletnu kopiju cele baze podataka, diferencijalni (eng. differential backup) koji čuva samo promene od poslednje potpune rezervne kopije i transaction log backup koji čuva evidencije izvršenih transakcija. Osnovni tipovi rezervnih kopija se mogu kombinovati u cilju pronalaženja što bolje strategije za backup i restore. Rezervne kopije se ne moraju uvek vršiti nad bazom podataka, već je moguće i izraditi rezervnu kopiju datoteka.

Restore proces podrazumeva vraćanje podataka iz rezervne kopije u bazu podataka i pri tom je važno da se poštuje pravilan redosled operacija i da se koristi odgovarajuća strategija u skladu sa zahtevima oporavka podataka. Model oporavka baze podataka takođe utiče na način i vrstu restore procesa koji se primenjuje.

Da bi se obezbedio uspešan backup i restore proces, neophodno je imati dobro osmišljenu strategiju. Strategija uključuje izbor vrste backup-a, frekvenciju izvršavanja backup-a, njegovo testiranje i pronalaženje sigurne lokacije za čuvanje rezervnih kopija. Testiranje strategije vraćanja podataka je ključni korak kako biste bili sigurni da je vaša strategija efikasna i da se može primeniti u slučaju nastanka problema u bazi podataka. Testiranje uključuje izbor određenih backup fajlova, njihovo vraćanje i proveru da li je baza podataka uspešno obnovljena i funkcionalna. Testiranje takođe omogućava identifikovanje eventualnih problema ili nedostataka u strategiji i omogućava vam da ih rešite pre nego što dođe do problema u realnoj situaciji. Takođe je važno odrediti raspored izvršavanja backup-a, kako bi se osiguralo da se podaci redovno čuvaju. Pravilno upravljanje ovim procesima omogućava se zaštita podataka od gubitka ili oštećenja, i brz oporavak u slučaju neuspeha.

# **Literatura**

1. Back Up and Restore of SQL Server Databases, <https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases?view=sql-server-ver16>, pristupljeno: 6.5.2023.
2. Backup Overview (SQL Server), <https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/backup-overview-sql-server?view=sql-server-ver16>, pristupljeno: 6.5.2023.
3. Restore and recovery overview (SQL Server), <https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-and-recovery-overview-sql-server?view=sql-server-ver16>, pristupljeno: 13.5.2023.
4. Understanding SQL Server Backup Types, <https://www.sqlshack.com/understanding-sql-server-backup-types/>, pristupljeno: 19.5.2023.
5. SQL Server Backup Academy, <https://sqlbak.com/academy/>, pristupljeno: 14.5.2023.