**Univerzitet u Nišu, Elektronski fakultet**

**Katedra za računarstvo**



Sigurnost SQL Server baze podataka

Mentor: Doc. dr Aleksandar Stanimirović

Student: Ana Milenković 1524

Univerzitet u Nišu

Elektronski Fakultet Niš

Sadržaj

[Uvod 2](#_Toc2041725469)

[1. Tipovi bezbednosti kod SQL Servera 3](#_Toc711348078)

[1.1 Sigurnost platforme i mreže 3](#_Toc1394569061)

[1.1.1 Fizička bezbednost 3](#_Toc2091541910)

[1.1.2 Sigurnost operativnog sistema 4](#_Toc2017736497)

[1.1.3 Sigurnost fajlova SQL Server operativnog sistema 4](#_Toc6372165)

[1.2 Principi i bezbednost objekata baze podataka 4](#_Toc818784964)

[1.2.1 Šifrovanje i sertifikati 5](#_Toc1617355414)

[1.3 Aplikaciona sigurnost 5](#_Toc910633316)

[1.4 SQL Server bezbednosni alati, pogledi i funkcije 5](#_Toc513144376)

[2. Bezbednost preduzeća 7](#_Toc1669168979)

[2.1 Otkrivanje i klasifikacija SQL podataka 8](#_Toc852735676)

[2.2 Procena ranjivosti SQL-a 11](#_Toc1653088420)

[3. Enkripcija podataka 13](#_Toc1598054304)

[3.1 Hijerarhija enkripcije 14](#_Toc1301147261)

[3.2 Mehanizmi šifrovanja 17](#_Toc1625341167)

[4. Uvek Šifrovano - Always Encrypted 18](#_Toc1755056951)

[4.1 Konfiguracija Always Encrypted funkcije 19](#_Toc97384675)

[4.2 Algoritam šifrovanja podataka 20](#_Toc1552905635)

[4.3 Kako rade upiti nad šifrovanim kolonama 23](#_Toc1454919543)

[4.4 Primeri za Always Encrypted na realnoj bazi 24](#_Toc1496475320)

[Zaključak 28](#_Toc903005934)

[Literatura 29](#_Toc124413407)

# Uvod

SQL Server je popularna relaciona baza podataka razvijena od strane Microsoft-a. Široko je rasprostranjena i koriste je brojne organizacije i kompanije za čuvanje kritičnih podataka, poput finansijske evidencije, informacija o korisnicima, i druge osetljive podatke. Sigurnost je kritičan aspekt bilo kog sistema za rukovođenjem bazama podataka. SQL Server obezbeđuje širok spektar bezbednosnih mehanizama koji omogućavaju administratorima da obezbede bazu i zaštite je od potencijalnih pretnji.  
  
Danas je sve međusobno povezano i podaci kruže i razmenjuju se bez prestanka. Bezbednost ovih podataka postala je od najvećeg značaja. Poslednjih godina,, moglo se zapaziti da su se odigrali proboji baza podataka koji su eksponirali osetljive informacije miliona pojedinaca i izazvali značajne finansijske gubitke i oštetile reputaciju zahvaćenih organizacija. Primer jednog ovakvog napada i proboja desio se aprila 2021. godine kada su objavljene informacije poput imena, brojeva telefona i šifri za preko 530 miliona korisnika društvene mreže Facebook.   
  
Ovakvi incidenti su utvrdili značaj sigurnosti sistema za upravljanje bazama podataka. Cilj ovog rada je da obezbedi pregled i analizu bezbednosnih koncepata koji su prisutni kod SQL Server baze, kao i da prikaže najbolja rešenja i pristupe za osiguravanje baze. Biće obrađeni različiti tipovi bezbednosnih mehanizama, njihova efektivnost, kao i tipovi napada od kojih se baza štiti.

# 1. Tipovi bezbednosti kod SQL Servera

Osiguravanje SQL servera se može posmatrati kao niz koraka, koji uključuju četiri oblasti: platformu, objekte (gde spadaju i podaci), aplikacije koje pristupaju sistemu, i autentifikaciju. Potrebno je obezbediti sve oblasti kako bi se baza podataka maksimalno osigurala.

## 1.1 Sigurnost platforme i mreže

Platforma za SQL Server uključuje fizički hardver i mrežne sisteme koji povezuju klijente na servere baze, kao i binarne fajlove koji se koriste za obradu zahteva.

### 1.1.1 Fizička bezbednost

Najbolja praksa za fizičku bezbednost je strogo ograničiti pristup fizičkim serverima i hardverskim komponentama. Na primer, koristiti zaključane sobe sa ograničenim pristupom za servere i hardver, kao i za mrežne uređaje. Dodatno, potrebno je ograničiti pristup i rezervnim medijima čuvajući ih na odvojenoj, bezbednoj lokaciji.

Implementacija bezbednosti fizičke mrežne bezbednosti počinje sa čuvanjem mreže dalje od korisnika koji nemaju adekvatnu autorizaciju.

### 1.1.2 Sigurnost operativnog sistema

Paketi za ažuriranje operativnog sistema sadrže brojna bitna bezbednosna poboljšanja. Potrebno je primeniti sva ažuriranja nad operativnim sistemom nakon što su testirana sa aplikacijom za bazu.

Firewall takođe pruža efikasne načine za implementaciju bezbednosti.Firewall je separator ili mrežnog saobraćaja, koji se može konfigurisati da sprovodi politiku bezbednosti podataka organizacije. Ako se koristi firewall, povećava se bezbednost na nivou operativnog sistema tako što će obezbediti prigušenu tačku na koju se bezbednosne mere mogu fokusirati.

Smanjenje površine je bezbednosna mera koja uključuje zaustavljanje ili onemogućavanje nekorišćenih komponenti. Smanjenje površine pomaže u poboljšanju bezbednosti obezbeđivanjem manje mogućnosti za potencijalne napade na sistem. Ključ za ograničavanje površine SQL Servera uključuje pokretanje potrebnih servisa koji imaju „najmanje privilegije“ dajući uslugama i korisnicima samo odgovarajuća prava.

### 1.1.3 Sigurnost fajlova SQL Server operativnog sistema

SQL Server koristi datoteke operativnog sistema za rad i skladištenje podataka. Najbolje prakse za bezbednost datoteka zahtevaju da se ograniči pristup ovim datotekama.

SQL Server servisni paketi i nadogradnje pružaju poboljšanu bezbednost. Da bi se utvrdili najnoviji dostupni servisni paket dostupan za SQL Server, potrebno je pogledati veb lokaciju SQL Servera.

Može se koristiti sledeća skripta da bi se odredio servisni paket instaliran na sistemu:

SELECT CONVERT(char(20), SERVERPROPERTY('productlevel'));

## 1.2 Principi i bezbednost objekata baze podataka

Principi su pojedinci, grupe i procesi kojima je odobren pristup SQL Serveru. „Osiguranici“ (eng. *Securables*) su server, baza podataka i objekti koje baza podataka sadrži. Svaki osigurani objekat ima skup dozvola koje se mogu konfigurisati da bi se smanjila površina SQL Servera.

### 1.2.1 Šifrovanje i sertifikati

Šifrovanje ne rešava probleme kontrole pristupa. Međutim, poboljšava bezbednost ograničavanjem gubitka podataka čak i u retkim slučajevima kada se zaobiđu kontrole pristupa. Na primer, ako je računar host baze podataka pogrešno konfigurisan i zlonamerni korisnik dobije osetljive podatke, kao što su brojevi kreditnih kartica, te ukradene informacije mogu biti beskorisne ako su šifrovane.

Sertifikati su softverski „ključevi“ koji se dele između dva servera koji omogućavaju bezbednu komunikaciju putem jake autentifikacije. Možete kreirati i koristiti sertifikate u SKL Serveru da poboljšate bezbednost objekata i veze.

## 1.3 Aplikaciona sigurnost

Klijentski programi

Najbolje prakse za bezbednost SQL Servera uključuju pisanje bezbednih klijentskih aplikacija. Za više informacija o tome kako pomoći da se zaštite klijentske aplikacije na mrežnom sloju, pogledati Konfiguraciju mreže klijenta.

Kontrola aplikacija Windows Defender-a (WDAC)

Windows Defender Application Control (WDAC) sprečava neovlašćeno izvršavanje koda. WDAC je efikasan način da se ublaži pretnja zlonamernog softvera zasnovanog na izvršnim datotekama. Za više informacija pogledati dokumentaciju o kontroli aplikacija Windows Defender-a.

## 1.4 SQL Server bezbednosni alati, pogledi i funkcije

SQL Server obezbeđuje alate, uslužne programe, poglede i funkcije koje se mogu koristiti za konfigurisanje i administriranje bezbednosti.

SQL Server bezbednosni alati i uslužni programi, između ostalog, uključuju i:

* Povezivanje, konfigurisanje i kontrolu SQL Servera u okviru SQL Server Management Studija,
* Povezivanje na SQL Server i pokretanje upita na komandnoj liniji sklcmd Utility
* Mrežna konfiguracija i kontrola za SQL Server u SQL Server Konfiguracionom Menadžeru
* Manipulisanje simetričnim ključevima za rskeimgmt Utility servera izveštaja (SSRS)

Pogledi i funkcije SQL Server bezbednosnog kataloga  
Database Engine izlaže bezbednosne informacije u nekoliko pogleda i funkcija koje su optimizovane za performanse. Tu spadaju:

* Pogledi kataloga bezbednosti SQL Servera, koji vraćaju informacije o dozvolama na nivou baze podataka i servera, principalima, ulogama i tako dalje. Pored toga, postoje prikazi kataloga koji pružaju informacije o ključevima za šifrovanje i sertifikatima
* SQL Server bezbednosne funkcije, koje vraćaju informacije o trenutnom korisniku, dozvolama i šemama
* Dinamički prikazi upravljanja bezbednošću SQL Servera.

# 2. Bezbednost preduzeća

Obezbeđivanje osetljivih podataka i poštovanje industrijskih propisa kao što su, na primer, PCI-DSS (Standard bezbednosti podataka industrije platnih kartica, eng. *Payment Card Industry Data Security Standard*) i GDPR (Generalna uredba o zaštiti podataka, eng. *General Data Protection Regulation*) je veoma važno. Kompromitovani sistem baze podataka može da vodi do gubitka prihoda, regulatornih kazni i negativnog uticaja na reputaciju i poslovanje zahvaćenih preduzeća.

Praćenje usklađenosti i održavanje bezbednosti baze podataka zahteva značajne administratorske resurse. SQL Server dostavlja alate kao što su alat za Otkrivanje i Klasifikaciju podataka *(Data Discovery and Classification)* i alati za procenu SQL ranjivosti (*SQL Vulnerability Assessment*) koji omogućavaju administratorima da identifikuju probleme kod usklađenosti i da označe i klasifikuju u specifične skupove podataka kako bi se osigurala usklađenost. SQL Server nudi mnoge bezbednosne funkcije koje rešavaju ove izazove, kao što su TDE (Transparentno šifrovanje podataka, *SQL Vulnerability Assessment*), uvek šifrovano (*Always Encrypted*), revizija, dinamičko maskiranje podataka i sigurnost na nivou reda (*Row-Level Security*).

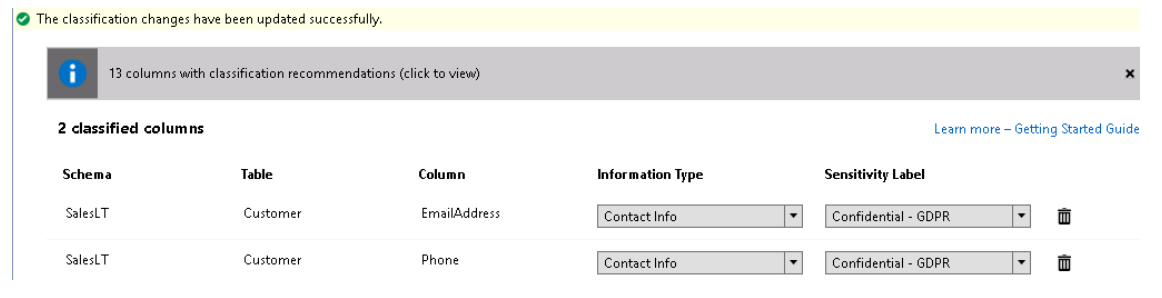
U kombinaciji sa daljim poboljšanjima upravljanja sertifikatima u SQL Serveru, podršku za TLS 1.2 i poverljive računarske inicijative kao što su bezbedne enklave, obezbeđena su rešenja po najvišim bezbednosnim standardima koja su istovremeno usaglašena sa GDPR i PCI-DSS. Sve ove funkcije su takođe dostupne u okviru Azure SQL baze podataka.

## 2.1 Otkrivanje i klasifikacija SQL podataka

Mogućnost otkrivanja i klasifikacije podataka omogućava identifikaciju, klasifikaciju i obeležavanje podataka koji se nalaze u SQL Server okruženju. Velika količina podataka koja se sada čuva u bazama podataka čini ovaj proces izazovnim, uz činjenicu da propisi kao što su GDPR, SOX i PCI zahtevaju zaštitu osetljivih podataka.

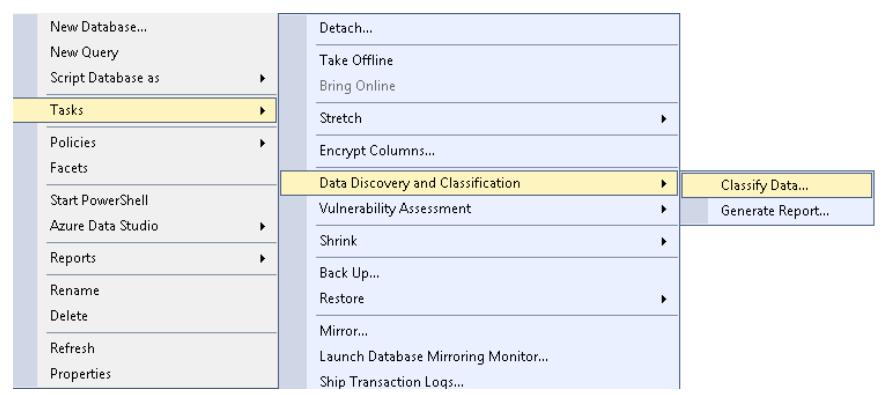
Može se videti kako ova mogućnost pomaže. Pre razvijanja strategije za bezbednost SQL Server baza podataka, logično je znati koje podatke poseduje baza, i na osnovu toga klasifikovati i obeležiti osetljive podatke i primeniti odgovarajuće bezbednosne kontrole, čime se minimizira potencijalno curenje osetljivih podataka.

Ključni elementi ove mogućnosti uključuju dve vrste meta-podataka: oznake (labels) i vrste informacija (information types). Oznake se koriste za definisanje osetljivosti podataka. Vrste informacija se koriste kako bi se pružila dodatna granularnost o vrstama podataka koji se čuvaju u koloni. Kao što se može videti na slici 2.1.1, e-adrese i brojevi telefona su klasifikovani kao kontakt informacije pod oznakom GDPR.



Slika 2.1.1 Primer klasifikacije

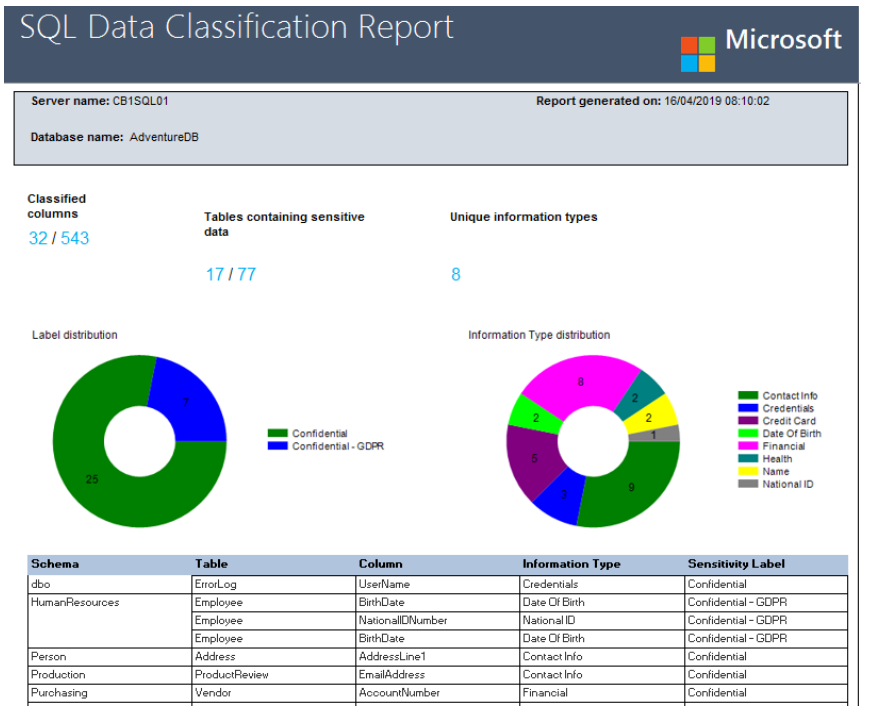
Za početak procesa klasifikacije, potrebno je desnim klikom na bazu podataka pronaći opciju "Otkrivanje podataka i klasifikacija" (Slika 2.2).



Slika 2.1.2: Pristupanje opciji "Klasifikacija podataka..." iz menija

Dok je ostvarena konekcija sa bazom podataka putem SSMS (SQL Server Management Studio), može se izdati sledeći upit kako bi se dobio veoma dobar pregled klasifikacije koja je upravo izvršena:

SELECT schema\_name(O.schema\_id) AS schema\_name, O.NAME AS table\_name, C.NAME AS column\_name, information\_type, sensitivity\_label   
FROM (   
 SELECT IT.major\_id, 48 | Enterprise Security IT.minor\_id, IT.information\_type, L.sensitivity\_label   
 FROM (   
 SELECT major\_id, minor\_id, value AS information\_type   
 FROM sys.extended\_properties   
 WHERE NAME = 'sys\_information\_type\_name' ) IT   
 FULL OUTER JOIN (   
 SELECT major\_id, minor\_id, value AS sensitivity\_label   
 FROM sys.extended\_properties   
 WHERE NAME = 'sys\_sensitivity\_label\_name' ) L   
 ON IT.major\_id = L.major\_id AND IT.minor\_id = L.minor\_id ) EP  
JOIN sys.objects O ON EP.major\_id = O.object\_id   
JOIN sys.columns C ON EP.major\_id = C.object\_id   
AND EP.minor\_id = C.column\_id  
Ovo se može delegirati SQL Serveru i pustiti ga da izvrši pregled podataka i automatsku implementaciju procesa klasifikacije.  
  
Još jedna prednost ove mogućnosti je vidljivost stanja klasifikacije u obliku izveštaja, koji se može eksportovati u različitim formatima prema potrebi. To će biti od koristiti u kod provere usklađenosti i revizije. Sledeći snimak ekrana prikazuje kopiju izveštaja u Excel formatu:



Slika 2.1.3: Izveštaj o klasifikaciji SQL podataka

Kada administratori razumeju postojeće podatke putem procesa klasifikacije, može zatim iskoristiti različite mogućnosti SQL Servera 2019, kao što su Always Encrypted ili Maskiranje podataka, kako bi zaštitili ove osetljive kolone.

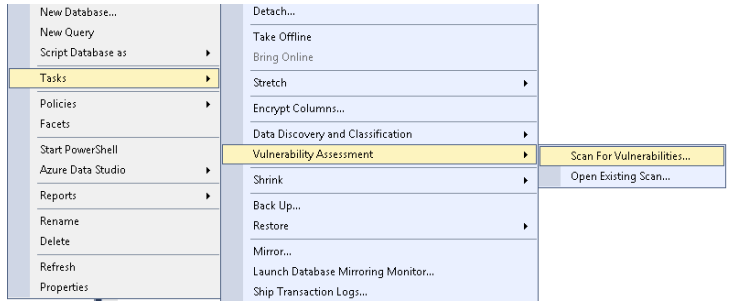
## 2.2 Procena ranjivosti SQL-a

Kada je reč o pouzdanom bezbednosnom planu za SQL Server, važno je adresirati trenutne sigurnosne probleme koji postoje unutar baze podataka. Gde treba početi? Koje tehničke aktivnosti su potrebne da bi se rešili pronađeni problemi?

Alat koji je za to zadužen je “Procena ranjivosti SQL-a". Ovaj alat će omogućiti poboljšanje internih procesa i jačanje sigurnosti u dinamičnom i promenjivom okruženju baze podataka.

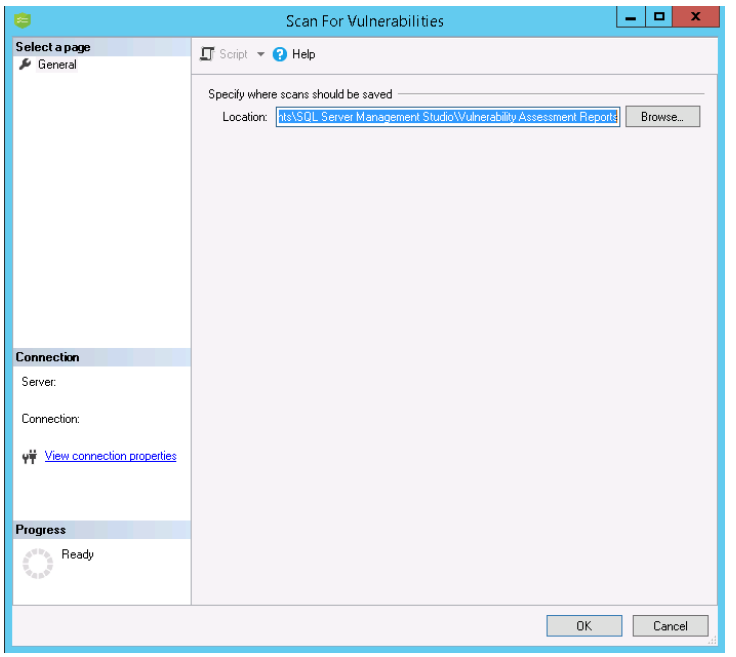
Napomena: Procena ranjivosti podržana je za SQL Server 2012 i kasnije verzije i zahteva SSMS 17.4 ili noviji.

Ova mogućnost izvršava skeniranje baze podataka koristeći unapred izgrađenu bazu znanja pravila koja će označiti sigurnosne zabrinutosti kao što su privilegovani nalozi i sigurnosne konfiguracije. Da bi se započela ova procena, treba desnim klikom na bazu podataka kliknuti na "Procena ranjivosti" (kao što je prikazano na sledećem snimku ekrana) i pokrenuti skeniranje:

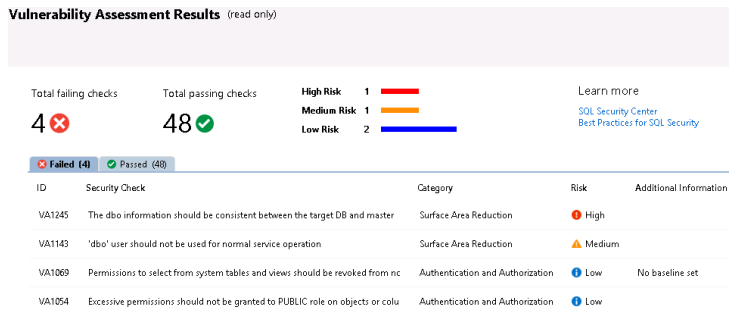


Slika 2.2.1: Pristupanje skeniranju ranjivosti iz menija Zadaci

Postoji zahtev za određivanjem lokacije na kojoj će se sačuvati procena. To će biti mesto na kojem se mogu otvoriti i pregledati istorijski izveštaji:



Slika 2.2.2: Dijalog okvira za skeniranje



Slika 2.2.3: Rezultati procene ranjivosti

Kao što se može videti, vrši se širok spektar različitih provera. One koje ne uspeju zahtevaju posebnu pažnju, pogotovu ako su označene kao visoki rizik.

Dok se vrši pregled rezultata procene, mogu se obeležiti određeni rezultati kao prihvatljivi u konkretnom okruženju.

# 3. Enkripcija podataka

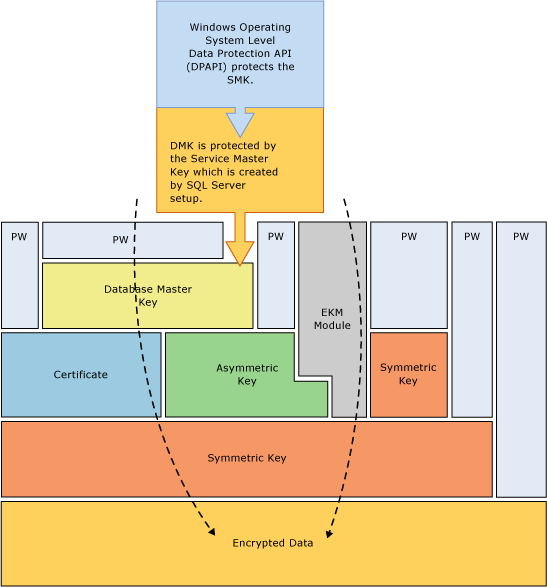
Šifrovanje ne rešava probleme kontrole pristupa. Međutim, ono poboljšava bezbednost ograničavanjem gubitka podataka čak i u retkim slučajevima kada se zaobiđu kontrole pristupa. Na primer, ako je računar host baze podataka pogrešno konfigurisan i zlonamerni korisnik dobije osetljive podatke, kao što su brojevi kreditnih kartica, te ukradene informacije su beskorisne ako su šifrovane.

SQL Server šifruje podatke hijerarhijskom enkripcijom i infrastrukturom za upravljanje ključevima. Svaki sloj šifruje sloj ispod sebe koristeći kombinaciju sertifikata, asimetričnih ključeva i simetričnih ključeva. Asimetrični ključevi i simetrični ključevi mogu da se čuvaju van SQL Servera u modulu za upravljanje proširenim ključevima. Sertifikati su softverski „ključevi“ koji se dele između dva servera koji omogućavaju bezbednu komunikaciju putem autentifikacije.

## 3.1 Hijerarhija enkripcije

Sledeća ilustracija, na slici 3.1.1, pokazuje da svaki sloj hijerarhije šifrovanja šifruje sloj ispod sebe i prikazuje najčešće konfiguracije šifrovanja. Pristup početku hijerarhije je obično zaštićen lozinkom.  
Treba uvek imati na umu sledeće koncepte kod SQL Server šifrovanja:

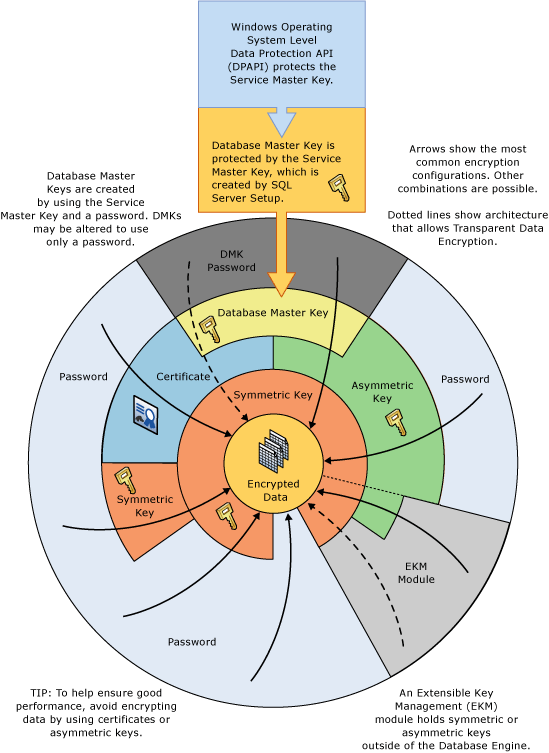
* Za najbolje performanse, podatke treba šifrovati korišćenjem simetričnih ključeva umesto sertifikata ili asimetričnih ključeva.
* Master ključevi baze podataka su zaštíćeni Service Master ključem. Service Master ključ kreira se podešavanjem SQL Servera i šifrovan je pomoću Windows API-ja za zaštitu podataka (*Windows Data Protection API,* DPAPI).
* Moguće su i druge hijerarhije šifrovanja koje implementiraju dodatne slojeve.
* Modul Extensible Key iManagement (EKM) čuva simetrične ili asimetrične ključeve izvan SQL Servera.
* Transparentno šifrovanje podataka (*Transparent Data Encryption*, TDE) mora da koristi simetrični ključ koji se zove ključ za šifrovanje baze podataka koji je zaštićen ili sertifikatom zaštićenim master ključem baze podataka glavne baze podataka, ili asimetričnim ključem uskladištenim u EKM-u.
* Service Master ključ i svi master ključevi baze podataka su simetrični ključevi.



Slika 3.1.1 Hijerarhijska struktura šifrovanja

Sledeća ilustracija 3.1.2 prikazuje iste informacije na drugačiji način. Ilustruje sledeće dodatne koncepte:

* Na ovoj ilustraciji, strelice označavaju uobičajene hijerarhije šifrovanja.
* Simetrični i asimetrični ključevi u EKM-u mogu zaštititi pristup simetričnim i asimetričnim ključevima uskladištenim u SQL Serveru. Isprekidana linija povezana sa EKM-om označava da ključevi u EKM-u mogu zameniti simetrične i asimetrične ključeve uskladištene u SQL Server-u.



Slika 3.1.2 hijerarhijska struktura šifrovanja, druga verzija

## 3.2 Mehanizmi šifrovanja

SQL Server obezbeđuje sledeće mehanizme za šifrovanje:

1. Transact-SQL funkcije
2. Asimetrični ključevi
3. Simetrični ključevi
4. Sertifikati
5. Transparentno šifrovanje podataka

Transact-SQL funkcije  
Pojedinačne stavke mogu biti šifrovane dok se ubacuju ili ažuriraju pomoću Transact-SQL funkcija, poput ENCRYPTBYPASSPHRASE i DECRYPTBYPASSPHRASE. ENCRYPTBZPASSPHRASE funkcija šifruje podatke pristupnom frazom koristeći TRIPLE DES algoritam sa dužinom bita ključa od 128. SQL Server 2017 i novije verzije šifruju podatke pomoću šifre i AES256 ključa. Sintaksa naredbe:  
EncryptByPassPhrase ( { 'passphrase' | @passphrase } , { 'cleartext' | @cleartext }   
 [ , { add\_authenticator | @add\_authenticator } , { authenticator | @authenticator } ] )

Asimetrični ključevi  
Asimetrični ključ se sastoji od privatnog ključa i odgovarajućeg javnog ključa. Svaki ključ može dešifrovati podatke koje je šifrovao drugi. Asimetrično šifrovanje i dešifrovanje su relativno zahtevni po pitanju resursa, ali pružaju viši nivo bezbednosti od simetrične enkripcije. Asimetrični ključ se može koristiti za šifrovanje simetričnog ključa za skladištenje u bazi podataka.

Simetrični ključevi  
Simetrični ključ je jedan ključ koji se koristi i za šifrovanje i za dešifrovanje. Šifrovanje i dešifrovanje korišćenjem simetričnog ključa je brzo i pogodno za rutinsku upotrebu sa osetljivim podacima u bazi podataka.

Sertifikati  
Sertifikat javnog ključa, koji se obično naziva samo sertifikat, je digitalno potpisana izjava koja vezuje vrednost javnog ključa za identitet osobe, uređaja ili usluge koja drži odgovarajući privatni ključ. Sertifikate izdaje i potpisuje autoritet za sertifikaciju (CA). Entitet koji primi sertifikat od CA je predmet tog sertifikata. Obično sertifikati sadrže sledeće informacije:

* Javni ključ subjekta.
* Informacije o identifikatoru subjekta, kao što su ime i adresa e-pošte.
* Period važenja. Ovo je vremenski period za koji se sertifikat smatra važeim.
* Sertifikat važi samo za period koji je u njemu naveden; svaki sertifikat sadrži datume Važi od i Važi do. Ovi datumi postavljaju granice perioda važenja. Kada istekne period važenja sertifikata, subjekt sertifikata koji je sada istekao mora da zatraži novi sertifikat.
* Informacije o identifikatoru izdavaoca.
* Digitalni potpis izdavaoca.

Ovaj potpis potvrđuje validnost veze između javnog ključa i informacija o identifikatoru subjekta. (Proces digitalnog potpisivanja informacija podrazumeva transformaciju informacija, kao i nekih tajnih informacija koje drži pošiljalac, u oznaku koja se zove potpis.)

Primarna prednost sertifikata je da oslobađaju domaćine potrebe da održavaju skup lozinki za pojedinačne subjekte. Umesto toga, domaćin samo uspostavlja poverenje u izdavaoca sertifikata, koji onda može potpisati neograničen broj sertifikata.

Kada host, kao što je bezbedni veb server, odredi izdavaoca kao pouzdani osnovni autoritet, host implicitno veruje smernicama koje je izdavalac koristio za uspostavljanje vezivanja sertifikata koje izdaje. U stvari, domaćin veruje da je izdavalac verifikovao identitet subjekta sertifikata. Host označava izdavaoca kao pouzdano osnovno ovlašćenje tako što stavlja samopotpisani sertifikat izdavaoca, koji sadrži javni ključ izdavaoca, u skladište sertifikata pouzdanog korenskog autoriteta za sertifikaciju na računaru domaćina. Srednji ili podređeni autoriteti za sertifikaciju imaju poverenja samo ako imaju važeću putanju za sertifikaciju od pouzdanog osnovnog sertifikacionog tela.

# 4. Uvek Šifrovano - Always Encrypted

Always Encrypted je funkcija dizajnirana da zaštiti osetljive podatke, kao što su brojevi kreditnih kartica ili nacionalni/regionalni identifikacioni brojevi (na primer, brojevi socijalnog osiguranja u SAD), uskladišteni u Azure SQL bazi podataka, Azure SQL upravljanoj instanci i SQL Server bazama podataka. Always Encrypted omogućava klijentima da šifruju osetljive podatke unutar klijentskih aplikacija i nikada ne otkrivaju ključeve za šifrovanje Database Engine-u. Ovo obezbeđuje razdvajanje između onih koji poseduju podatke i mogu da ih pregledaju i onih koji upravljaju podacima, ali ne bi trebalo da imaju pristup – administratora baze podataka na lokaciji, operatera baze podataka u oblaku ili drugih neovlašćenih korisnika sa visokim privilegijama. Kao rezultat toga, Always Encrypted omogućava korisnicima da pouzdano čuvaju svoje osetljive podatke u oblaku i da smanje verovatnoću krađe podataka od strane zlonamernih insajdera.

Always Encrypted se može konfigurisati da podržava ograničene poverljive upite o šifrovanim podacima, upite koji uključuju poređenja jednakosti. Na primer, traženje tačaka ili spajanje jednakosti. Takvi upiti koriste determinističku enkripciju.

Always Encrypted čini šifrovanje transparentnim za aplikacije. Uvek šifrovani drajver instaliran na klijentskom računaru to postiže automatskim šifrovanjem i dešifrovanjem osetljivih podataka u klijentskoj aplikaciji. Drajver šifruje podatke u osetljivim kolonama pre nego što ih prosleđuje Database Engine-u i automatski prepisuje upite tako da se očuva semantika aplikacije. Slično tome, drajver transparentno dešifruje podatke, uskladištene u šifrovanim kolonama baze podataka, sadržane u rezultatima upita.

## 4.1 Konfiguracija Always Encrypted funkcije

Da bi se podesio Always Encrypted u bazi podataka, potrebno je:

1. Obezbediti kriptografske ključeve da bi se zaštitili podaci. Always Encrypted koristi dve vrste ključeva:
   1. Ključevi za šifrovanje kolona.
   2. Master ključevi kolone.

Ključ za šifrovanje kolone se koristi za šifrovanje podataka u šifrovanoj koloni. Master ključ kolone je ključ za zaštitu ključa koji šifruje jedan ili više ključeva za šifrovanje kolone.  
Morate da skladištite glavne ključeve kolona u pouzdano skladište ključeva izvan sistema baze podataka, kao što je Azure Key Vault, Windows skladište sertifikata ili hardverski bezbednosni modul.  
Zatim su vam potrebni ključevi za šifrovanje kolona i šifrujte svaki od njih Master ključem kolone.  
Konačno, morate da uskladištite metapodatke o ključevima u vašoj bazi podataka.  
Metapodaci glavnog ključa kolone beleže lokaciju glavnog ključa kolone.  
Metapodaci ključa za šifrovanje kolone sadrže šifrovanu vrednost ključa za šifrovanje kolone. Database Engine nikada ne skladišti niti koristi ključeve bilo koje vrste u otvorenom tekstu.

1. Konfigurišite šifrovanje za izabrane kolone baze podataka koje sadrže osetljive podatke koje treba zaštititi. Ovo može uključivati kreiranje novih tabela sa šifrovanim kolonama ili šifrovanje postojećih kolona baze podataka i postojećih podataka. Kada podešavate šifrovanje za kolonu, navodite informacije o algoritmu za šifrovanje, ključu za šifrovanje kolone da biste zaštitili podatke u koloni i tip šifrovanja. Always Encrypted podržava dva tipa šifrovanja:
   1. Determinističko šifrovanje uvek generiše istu šifrovanu vrednost za datu vrednost otvorenog teksta. Korišćenje determinističkog šifrovanja omogućava traženje tačaka, spajanje jednakosti, grupisanje i indeksiranje na šifrovanim kolonama. Međutim, takođe može dozvoliti neovlašćenim korisnicima da pogode informacije o šifrovanim vrednostima ispitivanjem obrazaca u šifrovanoj koloni, posebno ako postoji mali skup mogućih šifrovanih vrednosti, kao što su Tačno/Netačno ili Sever/Jug/Istok/Zapad region.
   2. Nasumično šifrovanje koristi metod koji šifruje podatke na manje predvidljiv način. Nasumično šifrovanje je bezbednije, ali sprečava pretragu, grupisanje, indeksiranje i pridruživanje šifrovanim kolonama.

Koristite determinističko šifrovanje za kolone koje će se koristiti kao parametri za pretragu ili grupisanje. Na primer, državni identifikacioni broj. Koristite nasumično šifrovanje za podatke kao što su poverljivi komentari istrage, koji nisu grupisani sa drugim zapisima i ne koriste se za spajanje tabela.

## 4.2 Algoritam šifrovanja podataka

Always Encrypted koristi AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256 algoritam za šifrovanje podataka u bazi podataka.

AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256 koristi šemu šifrovanja sa autentifikacijom sa pridruženim podacima, pritom prateći pristup Šifrovanje-pa-MAC. To jest, otvoreni tekst se prvo šifruje, a MAC se proizvodi na osnovu rezultujućeg šifrovanog teksta.

Da bi prikrio obrasce, AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256 koristi način rada Cipher Block Chaining (CBC), gde se početna vrednost unosi u sistem pod nazivom vektor inicijalizacije (*Initialization Vector - IV*). AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256 izračunava vrednost šifrovanog teksta za datu vrednost otvorenog teksta koristeći sledeće korake.

**Korak 1: Generisanje vektora inicijalizacije (IV)**

Always Encrypted podržava dve varijacije AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256:

* Nasumično
* Deterministički

Za nasumično šifrovanje, vektor inicijalizacije se generiše nasumično. Kao rezultat, svaki put kada se šifruje isti otvoreni tekst, generiše se drugačiji šifrovani tekst, koji sprečava bilo kakvo otkrivanje informacija:

IV = Kriptografski generiše nasumičnih 128 bita

Ako postoji determinističko šifrovanje, IV se ne generiše nasumično, već se izvodi iz vrednosti otvorenog teksta koristeći sledeći algoritam:

IV = HMAC-SHA-256( iv\_key, cell\_data ) skraćeno na 128 bita.

Gde je iv\_key izveden iz CEK-a na sledeći način:

iv\_key = HMAC-SHA-256(CEK, "ključ ćelije IV Microsoft SQL Server" + algoritam + CEK\_length)

Skraćivanje HMAC vrednosti se vrši da bi se uklopio jedan blok podataka po potrebi za IV.

Kao rezultat, determinističko šifrovanje uvek proizvodi isti šifrovani tekst za datu vrednost otvorenog teksta, što omogućava zaključivanje da li su dve vrednosti otvorenog teksta jednake upoređivanjem njihovih odgovarajućih vrednosti šifrovanog teksta. Ovo ograničeno otkrivanje informacija omogućava sistemu baze podataka da podrži poređenje jednakosti na šifrovanim vrednostima kolona.

Determinističko šifrovanje je efikasnije u prikrivanju obrazaca, u poređenju sa alternativama, kao što je korišćenje unapred definisane IV vrednosti.

**Korak 2: Računanje AES\_256\_CBC šifrovanog teksta**

Nakon izračunavanja vektora inicijalizacije, generiše se AES\_256\_CBC šifrovani tekst:  
aes\_256\_cbc\_ciphertext = AES-CBC-256(enc\_key, IV, cell\_data) sa dopunom PKCS7.

Gde je ključ za šifrovanje (enc\_key) izveden iz CEK-a na sledeći način.  
enc\_key = HMAC-SHA-256(CEK, "SQL Server ključ za šifrovanje ćelije" + algoritam + CEK\_length)

**Korak 3: Računanje MAC-a**

Nakon toga, MAC se izračunava korišćenjem sledećeg algoritma:

MAC = HMAC-SHA-256(mac\_ključ, versionbyte + IV + Ciphertext + versionbyte\_length)

Gde su:

versionbyte = 0x01 i  
versionbyte\_length = 1 mac\_key = HMAC-SHA-256(CEK, " QL Server ključ za šifrovanje ćelije " + algoritam + CEK\_length)

**Korak 4: Povezivanje**

Konačno, šifrovana vrednost se proizvodi spajanjem bajta verzije algoritma, MAC-a, vektora inicijalizacije i AES\_256\_CBC šifrovanog teksta:

aead\_aes\_256\_cbc\_hmac\_sha\_256 = versionbyte + MAC + IV + aes\_256\_cbc\_ciphertext

## 4.3 Kako rade upiti nad šifrovanim kolonama

Da bi se izvršio upit za šifrovane kolone baze podataka, umetnuli podaci u šifrovane kolone, preuzele vrednosti otvorenog teksta iz šifrovanih kolona, korisnik ili aplikacija koja izdaje upit moraju ispuniti sledeće preduslove:

* Posedovati pristup master ključu kolone koji štiti podatke. Pristup ključu je neophodan pored dozvola na nivou baze podataka, kao što je SELECT nad tabelom koja sadrži podatke.
* Povezivanje sa bazom podataka sa omogućenim Always Encrypted u konekciji sa bazom podataka. Većina SQL alata i drajvera SQL klijenta podržavaju omogućavanje Always Encrypted za konekcije sa bazom podataka.

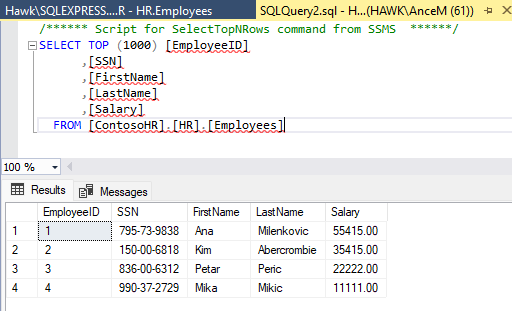
Ako korisnik ima potrebne dozvole baze podataka za čitanje podataka, ali nema pristup ključevima koji ih štite, korisnik i dalje može da preuzme šifrovane (šifrovane) podatke tako što će se povezati sa bazom podataka bez omogućavanja Always Encrypted u konekciji sa bazom podataka.

Evo kako funkcionišu upiti nad šifrovanim kolonama:

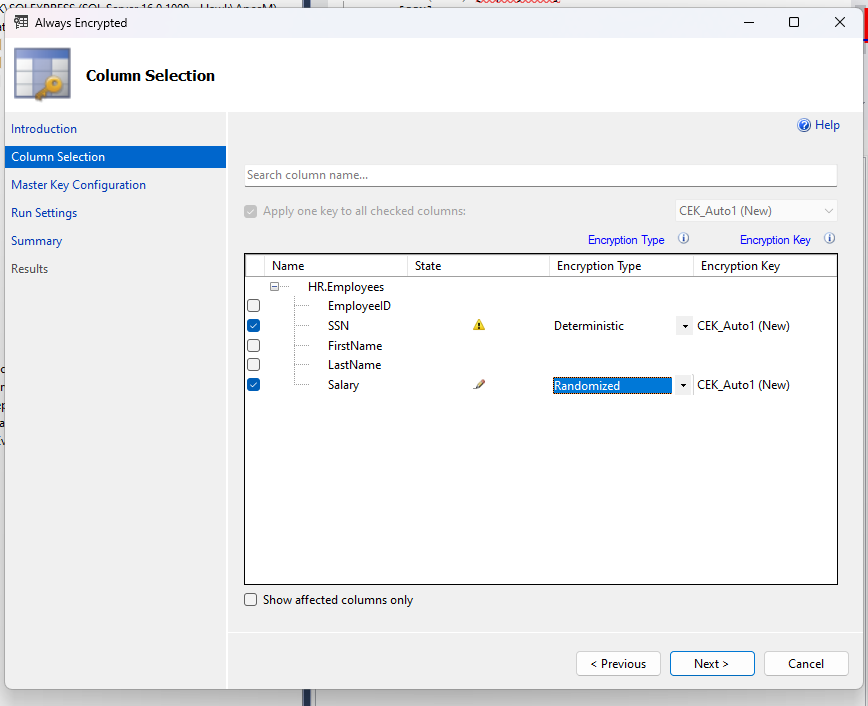
1. Kada aplikacija izda parametrizovani upit, SQL klijentski drajver u okviru aplikacije transparentno kontaktira Database Engine (pozivanjem sp\_describe\_parameter-encryption) da bi odredio koji parametri ciljaju na šifrovane kolone i treba da budu šifrovani. Za svaki parametar koji treba da bude šifrovan, drajver prima algoritam šifrovanja, tip šifrovanja i metapodatke ključa, uključujući šifrovani ključ za šifrovanje kolone i lokaciju odgovarajućeg master ključa kolone.
2. Drajver poziva skladište ključeva, koje sadrži master ključeve kolone da bi dešifrovao vrednosti šifrovanog ključa za šifrovanje kolone. Rezultirajući ključevi za šifrovanje kolone otvorenog teksta se keširaju da bi se smanjio broj povratnih puteva do skladišta ključeva pri narednim upotrebama istih ključeva za šifrovanje kolona.
3. Drajver koristi dobijene ključeve za šifrovanje kolona otvorenog teksta da šifruje parametre upita koji odgovaraju šifrovanim kolonama.
4. Drajver zamenjuje vrednosti otvorenog teksta parametara koji ciljaju na šifrovane kolone sa njihovim šifrovanim vrednostima i šalje upit Database Engine-u na obradu.
5. Database Engine izvršava upit, koji može uključiti poređenja jednakosti nad kolonama koristeći determinističko šifrovanje.
6. Ako rezultati upita uključuju podatke iz šifrovanih kolona, Database Engine prilaže metapodatke šifrovanja za svaku kolonu, uključujući informacije o algoritmu šifrovanja, tipu šifrovanja i metapodacima ključeva skupu rezultata.
7. Database Engine šalje skup rezultata klijentskoj aplikaciji.
8. Za svaku šifrovanu kolonu u primljenom skupu rezultata, drajver prvo pokušava da pronađe ključ za šifrovanje kolone otvorenog teksta u lokalnoj keš memoriji i samo se vraća do skladišta ključeva koji drži glavni ključ kolone ako ne može da pronađe ključ u keš memorija.
9. Drajver dešifruje rezultate i vraća vrednosti otvorenog teksta u aplikaciju.

Klijentski drajver stupa u vezu sa skladištem ključeva, koje sadrži glavni ključ kolone, koriste dobavljač skladištenja glavnog ključa kolone, koji je softverska komponenta na strani klijenta koja inkapsulira skladište ključeva koje sadrži glavni ključ kolone. Dobavljači za uobičajene tipove ključeva dostupni su u Microsoft-ovim bibliotekama drajvera na strani klijenta ili kao samostalni alati. Takođe je moguće implementirati provajdera. Mogućnosti Always Encrypted, uključujući ugrađene dobavljače skladišta glavnog ključa kolone, razlikuju se u zavisnosti od biblioteke drajvera i njene verzije.

## 4.4 Primeri za Always Encrypted na realnoj bazi

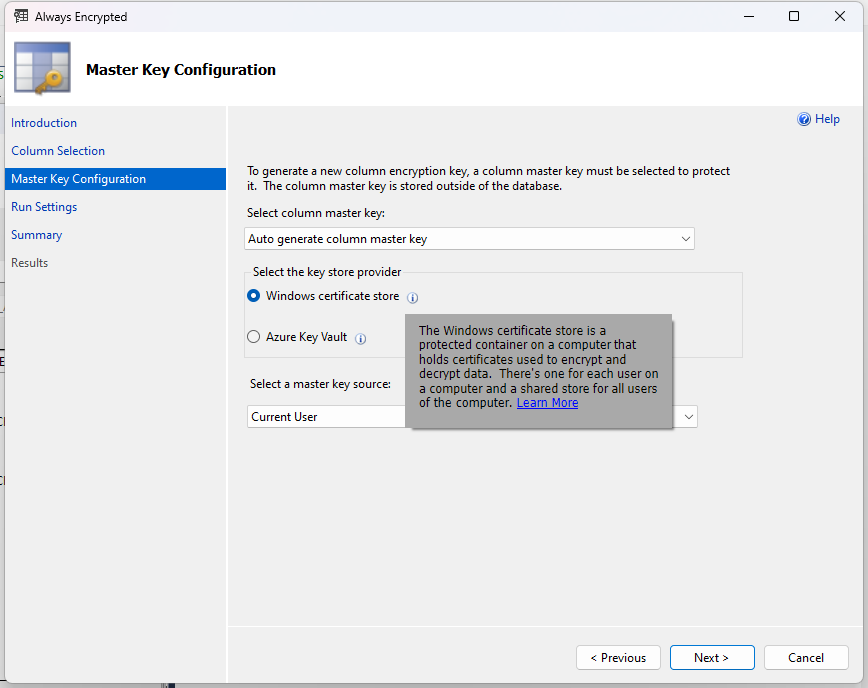
Baza ContosoHR sa tabelom HR.Employees:  


Ova tabela sadrži osetljive informacije: SSN (Social Security Number) i Salary (plata). Desnim klikom na tabelu, odabirom opcije “Encrypt Columns” otvara se sledeći prozor:



Izabrati kolone SSN i Salary za šifrovanje, a kao Tip Šifrovanja za SSN se bira Deterministic, dok za Salary je izabrano Randomized.

Zatim je potrebno kreirati Master ključ. Za ovaj primer za čuvanje Master ključa izabran je Windows certificate store. Druga opcija je Azure Key Vault.

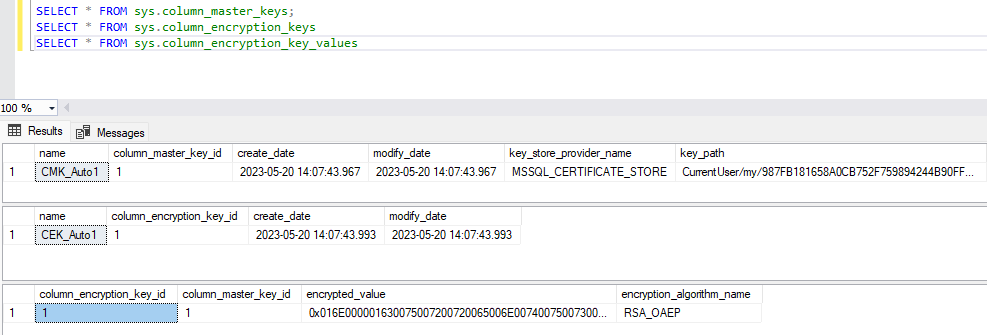


Nakon što je sve podešeno, može se krenuti sa šifrovanjem.  
Sledeći upiti daju pregled onoga što je učinjeno šifrovanjem:

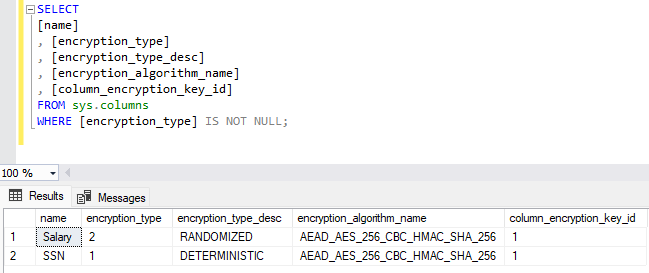
SELECT \* FROM sys.column\_master\_keys;

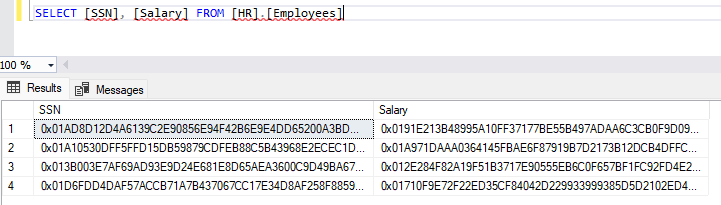
SELECT \* FROM sys.column\_encryption\_keys

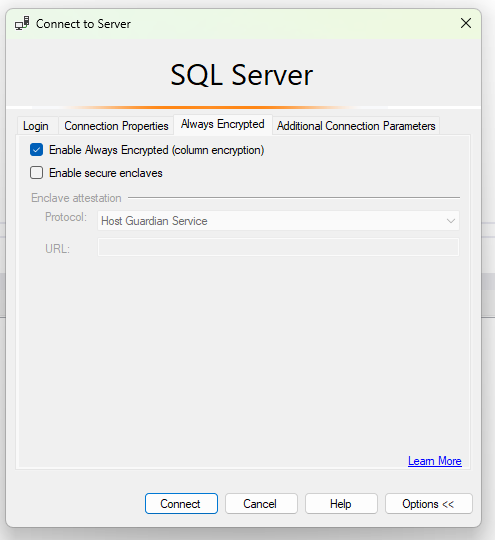
SELECT \* FROM sys.column\_encryption\_key\_values



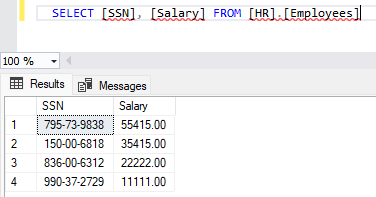
Ovaj upit prikazuje tip algoritma koji se koristi za šifriranje, kao i tip šifriranja.



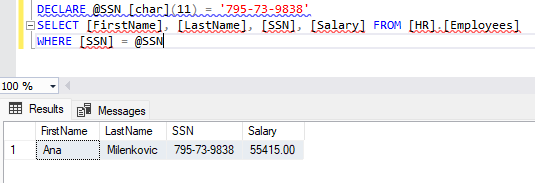
Za izvršavanje upita nad šifrovanim kolonama, postoje dve mogućnosti: ako je Always Encrypted ugašeno prilikom povezivanja na bazu, i ukoliko je Always Encrypted uključeno prilikom povezivanja na bazu. Kada je ugašen, izvršavanje upita nad šifrovanim kolonama vraća binarne šifrirane podatke:  




Ukoliko se sada uključi Always Encrypted, rezultat istog upita je sledeći:



SQL Server nudi i mogućnost parametrizacije za Always Encrypted. Ovo je potrebno ručno uključiti. Nakon što se uključi, moguće je pretražiti tabelu po vrednostima šifrovanih kolona:



# Zaključak

SQL Server je popularna relaciona baza podataka razvijena od strane Microsoft-a. Široko je rasprostranjena i koriste je brojne organizacije i kompanije za čuvanje kritičnih podataka, poput finansijske evidencije, informacija o korisnicima, i druge osetljive podatke. Obzirom na široku primenu i popularnost ove relacione baze, od velikog je značaja da sigurnost podataka bude na najvišem mogućem nivou.

Ovaj rad je obradio sve tipove bezbednosti koji postoje kod SQL Server baze, detaljnije pokrio bezbednost preduzeća, kao i enkripciju: njenu strukturu, hijerarhiju, način implementacije kod SQL Server-a. Dodatno je objašnjen i na praktičnim primerima pokazan koncept nazvan “Always Encrypted”, koji služi da automatski šifrira i dešifruje osetljive podatke.

Glavni zaključak koji nosi ovaj rad je upravo o tome koliko su bezbednost i sigurnost podataka važni. Da se ne bi dešavali proboji baza, kao i da bi se izbegle sve naknadne štete koje bi ti proboji izazvali, potrebno je obratiti veliku pažnju na sigurnost podataka, bilo da je u pitanju razvoj sistema baze podataka, ili klijentskih aplikacija koje tim bazama pristupaju.

# Literatura

Gorman, K., Hirt, A., Noderer, D., Pearson, M., Rowland-Jones, J., Ryan, D., Sirpal, A., & Woody, B. (2020, April 27). *Introducing Microsoft SQL Server 2019: Reliability, Scalability, and Security Both on Premises and in the Cloud*.

J. (2023, February 28). *SQL Server Encryption - SQL Server*. SQL Server Encryption - SQL Server | Microsoft Learn. <https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/sql-server-encryption>

J. (2023, February 28). *Encryption Hierarchy - SQL Server*. Encryption Hierarchy - SQL Server | Microsoft Learn. <https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/encryption-hierarchy>

J. (2023, April 20). *Always Encrypted - SQL Server*. Always Encrypted - SQL Server | Microsoft Learn. <https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine>