

Elektronski fakultet, Niš

Smer: Računarstvo i informatika, Softversko inženjerstvo

Predmet: **Sistemi za upravljanje bazama podataka**

Tema:

**Sigurnost Oracle baze podataka**

Student: Lena Petrović 1295

Mentor: Doc. dr Aleksandar Stanimirović

Niš, Maj 2022

<b>O Oracle bezbednosti baze podataka</b>	<b>3</b>
<b>Obezbeđivanje korisničkih naloga i privilegija</b>	<b>4</b>
<b>Obezbeđivanje uloga</b>	<b>9</b>
<b>Obezbeđivanje lozinki</b>	<b>10</b>
<b>Revizija aktivnosti</b>	<b>12</b>
<b>Obezbeđivanje podataka</b>	<b>17</b>
<b>Transparent data encryption - TDE</b>	<b>18</b>
<b>Alati za procenu sigurnosti Oracle baze podataka</b>	<b>24</b>
<b>Reference</b>	<b>27</b>

## O Oracle bezbednosti baze podataka

Oracle Database baza podataka pruža veliki skup podrazumevanih bezbednosnih funkcija za upravljanje korisničkim nalogima, autentifikacijom, privilegijama, bezbednošću aplikacija, šifrovanjem, mrežnim saobraćajem i revizijom.

Mogu se koristiti podrazumevane funkcije Oracle baze podataka da bi se konfigurisala bezbednost u nekoliko oblasti za instalaciju Oracle baze podataka. Oblasti u kojima se može konfigurisati bezbednost su sledeće:

- Korisnički nalozi - Kada se kreiraju korisnički nalozi, moguće ih je zaštititi na različite načine.
- Autentifikacija - Oracle baza podataka pruža nekoliko načina za konfigurisanje autentifikacije za korisnike i administratore baze podataka. Na primer, mogu se autentifikovati korisnici na nivou baze podataka, iz operativnog sistema i na mreži.
- Privilegije i uloge - Mogu se koristiti privilegije i uloge da se ograniči korisnički pristup podacima.
- Sigurnost aplikacije - Prvi korak u kreiranju aplikacije baze podataka je da se osigura da je ona ispravno osigurana.
- Informacije o sesiji korisnika koristeći kontekst aplikacije - Kontekst aplikacije je par ime-vrednost (key-value) koji sadrži informacije o sesiji. Informacije o sesiji korisnika, kao što su korisničko ime ili termina, mogu se preuzeti, a zatim se može ograničiti pristup bazi podataka i aplikaciji za tog korisnika.
- Pristup bazi podataka na nivou reda i kolone koristeći Virtual Private Database - Politika Virtual Private Database dinamički ugrađuje predikat "where" u SQL upit koji korisnik izdaje.
- Klasifikacija i zaštita podataka u različitim kategorijama - Mogu se pronaći sve kolone tabele u bazi podataka koje sadrže osetljive podatke (kao što su brojevi kreditne kartice ili socijalnog osiguranja), zatim se ovi podaci mogu klasifikovati. Na kraju se može kreirati politika koja štiti ove podatke kao celinu.
- Jaka autentifikacija - Baza podataka se može konfigurisati tako da koristi jaku autentifikaciju pomoću Oracle adaptera (Oracle authentication adapters) za potvrdu identiteta koji podržavaju različite usluge autentifikacije nezavisnih proizvođača, uključujući SSL sa digitalnim sertifikatima.

- Revizija aktivnosti baze podataka - Omogućena je revizija aktivnosti baze podataka uopšteno, kao i revizija SQL upita, SQL privilegija, objekata šeme...

Bezbednost informacija, privatnost i zaštita korporativne imovine i podataka su od veoma velikog značaja za svaki posao.

Oracle baza podataka na sveobuhvatan način rešava potrebu za bezbednošću informacija obezbeđujući najsavremenije bezbednosne funkcije kao što su duboka zaštita podataka, revizija, skalabilna bezbednost, bezbedno hostovanje i razmena podataka.

Oracle baza podataka predvodi u oblasti bezbednosti. Da bi se maksimizirale bezbednosne karakteristike koje nudi Oracle u bilo kom poslovnom okruženju, neophodno je da sama baza podataka bude dobro zaštićena.

Bezbednosne smernice pružaju savete o tome kako da se konfiguriše Oracle baza podataka, tako da bude bezbedna time što će se pridržavati preporučenih standarda i preporučljive bezbednosne prakse za bazu podataka.

## Obezbeđivanje korisničkih naloga i privilegija

### 1. Zaključavanje korisničkih naloga

Oracle baza podataka se instalira sa nekoliko podrazumevanih korisničkih naloga baze podataka. Nakon uspešne instalacije baze podataka, "Database Configuration Assistant" automatski zaključava i "ističe" (podrazumeva da je istekla) većinu podrazumevanih korisničkih naloga baze podataka.

Kod klasične instalacije dobija se 37 korisnika. Informacije o svima njima nalaze se u ALL\_USERS tabeli, dok je na slici prikazano samo prvih par:

```
SQL> select * from all_users;
```

USERNAME	USER_ID	CREATED	COM	O	INH	DEFAULT_COLLATION
SYS	0	30-MAY-19	YES	Y	NO	USING_NLS_COMP
AUDSYS	8	30-MAY-19	YES	Y	NO	USING_NLS_COMP
SYSTEM	9	30-MAY-19	YES	Y	NO	USING_NLS_COMP
SYSBACKUP	2147483617	30-MAY-19	YES	Y	NO	USING_NLS_COMP
SYSDBG	2147483618	30-MAY-19	YES	Y	NO	USING_NLS_COMP
SYSKM	2147483619	30-MAY-19	YES	Y	NO	USING_NLS_COMP
SYSRAC	2147483620	30-MAY-19	YES	Y	NO	USING_NLS_COMP
OUTLN	13	30-MAY-19	YES	Y	NO	USING_NLS_COMP
XS\$NULL	2147483638	30-MAY-19	YES	Y	NO	USING_NLS_COMP
GSMADMIN_INT	22	30-MAY-19	YES	Y	NO	USING_NLS_COMP
ERNAL						

Ako se koristi ručna instalacija Oracle baze podataka (bez korišćenja "Database Configuration Assistant"), tada nijedan podrazumevani korisnik baze podataka neće biti

zaključan nakon uspešne instalacije. Ili, ako se vrši nadogradnja sa prethodnog izdanja Oracle baze podataka, mogu se uključiti podrazumevani nalozi iz ranijih izdanja. Ostavljeni "otvoreni" u svojim podrazumevanim stanjima, ovi korisnički nalozi mogu da se eksploatišu za dobijanje neovlašćenog pristupa podacima ili ometanje operacija baze podataka.

Preporučuje se zaključavanje i isticanje svih podrazumevanih korisničkih naloga baze. Zaključani nalozi ne mogu napraviti konekciju ka bazi. U Oracle bazi podataka obezbeđen je SQL upit kojim je ovo moguće postići:

```
SQL> alter user sara account lock;  
User altered.
```

Da bi se otključao korisnik, jednostavno zameniti „lock“ sa „unlock“. Oracle se instalira sa nekoliko podrazumevanih naloga koji nikada ne bi trebalo da budu zaključani ili otpušteni. To uključuje: SYS, SYSTEM, SYSMAN... Međutim, uvek bi trebalo promeniti lozinku za ove korisnike.

Instaliranje dodatnih proizvoda i komponenti nakon početne instalacije takođe rezultira stvaranjem više podrazumevanih naloga baze podataka. "Database Configuration Assistant" automatski zaključava i ističe sve dodatno kreirane korisničke naloge baze podataka. Treba otključati samo one naloge kojima treba redovno pristupati i dobra praksa je da im se dodeli jaka i smislena lozinka.

Ako je iz bilo kog razloga potreban bilo koji podrazumevani korisnički nalog baze podataka osim onih koji su ostali otvoreni, onda administrator baze podataka (DBA) mora da otključa i aktivira taj nalog novom, bezbednom lozinkom.

## 2. Odgovoriti korisnike od korišćenja NOLOGGING klauzule u SQL upitima

U nekim SQL izrazima, korisnik ima opciju da navede NOLOGGING klauzulu, što ukazuje da operacija baze podataka nije evidentirana u onlajn datoteci dnevnika ponavljanja. Iako korisnik specificira klauzulu, zapis ponavljanja se i dalje upisuje u onlajn log fajlu. Dakle, ne postoje podaci povezani sa ovim zapisom. Zbog toga, korišćenje NOLOGGING klauzule ima potencijal da se zlonamerni kod unese bez provere.

Opcija NOLOGGING je odličan način da se ubrza umetanje i kreiranje indeksa. Zaobilazi pisanje logogva, značajno poboljšavajući performanse. Međutim, ovaj pristup može biti prilično opasan.

NOLOGGING se može koristiti u sledećim situacijama:

- Create Table As Select (CTAS)
- ALTER TABLE
- INSERT /\*+APPEND\*/
- CREATE INDEX
- ALTER INDEX

```
SQL> CREATE TABLE persons_nologging
2  (
3    PersonID int,
4    LastName varchar(255),
5    FirstName varchar(255),
6    Address varchar(255),
7    City varchar(255)
8  )
9  nologging;
Table created.
```

### 3. Princip najmanje privilegija

#### 1. Dodeliti samo neophodne privilegije

Nije pametno dodeljivati korisnicima baze podataka ili ulogama (roles) više privilegija nego što im je zaista potrebno. Ako je moguće, treba dodeliti privilegije ulogama, a ne korisnicima. Drugim rečima, princip najmanje privilegija je da se korisnicima daju samo one privilegije koje su zaista potrebne za efikasno obavljanje njihovih poslova.

Za primenu ovog principa, sledeće stvari treba ograničiti i smanjiti što je više moguće:

- Broj SYSTEM i OBJECT privilegija dodeljenih korisnicima baze podataka
- Broj ljudi kojima je dozvoljeno da naprave SYS privilegovane veze sa bazom podataka.
- Broj korisnika kojima su dodeljene ANY privilegije, kao što je privilegija DROP ANY TABLE. Na primer, generalno nema potrebe da se dodeljuju privilegije CREATE ANY TABLE korisniku koji nema DBA privilegije (administratorske).
- Broj korisnika kojima je dozvoljeno da izvode operacije koje kreiraju, menjaju ili brišu objekte baze podataka (na primer, TRUNCATE TABLE, DELETE TABLE, DROP TABLE...)

#### 2. Ponovo proceniti privilegiju SELECT objekta i SELECT ANY TABLE sistemske privilegije koje su dodeljene korisnicima

Ukoliko je potrebno da se korisnici ograniče samo na postavljanje upita prema tabelama, pogledima onda je moguće dodeliti korisnicima READ privilegiju za objekat. Pouzdanim korisnicima je moguće dodeliti i READ ANY TABLE privilegiju.

Sa druge strane, ukoliko je potrebno da korisnicima obezbedi i zaključavanje tabele ili izvršenje SELECT...FOR UPDATE upita, tada se može dodeliti SELECT privilegija (za pouzdane korisnike, SELECT ANY TABLE privilegija).

Iz ovoga se može zaključiti da READ i SELECT privilegija rade identičnu stvar, s tim što korisnik sa select privilegijom može da zaključa tabelu.

```
SQL> grant read on system.Persons to professor;
Grant succeeded.
```

### 3. Zabraniti korisnicima pristup podacima iz SYS šeme

Zabraniti običnim korisnicima (korisnicima koji nisu administratori) da menjaju redove tabele ili objekte šeme u SYS šemi, jer to može da ugrozi integritet podataka. Ograničiti upotrebu izraza kao što su DROP TABLE, TRUNCATE TABLE, DELETE, INSERT i sličnih izraza za modifikaciju objekata u SYS šemi.

Na prvoj slici je prikazano kako je moguće oduzeti/zabraniti neku privilegiju određenoj ulozi, dok je na drugoj slici pokazano kako to izgleda u praksi.

```
SQL> revoke select on system.Persons from professor;
Revoke succeeded.
```

```
SQL> select * from system.Persons;
select * from system.Persons
                        *
ERROR at line 1:
ORA-01031: insufficient privileges
```

### 4. Dodela privilegija samo ulogama

Dodeljivanje privilegija ulogama, a ne pojedinačnim korisnicima, znatno olakšava upravljanje i praćenje privilegija. U nastavku se može videti dodeljivanje privilegija ulogama, a zatim dodeljivanje uloga korisnicima.

Pre svega je potrebno kreirati korisnika. Kreirani korisnik ima username 'lena', vrednost lozinke kojom se identifikuje je ista kao i username.

```
SQL> create user lena identified by lena;
User created.
SQL> grant sysdba to lena;
```

Slika ispod predstavlja dodeljivanje privilegija koje ima korisnik 'sysdba' korisniku 'lena'. Ovo u praksi ne treba primenjivati jer je korisnik 'sysdba' korisnik sa visokim privilegijama i obični korisnici ne bi trebalo da imaju toliko privilegija.

```
SQL> grant sysdba to lena;  
Grant succeeded.
```

Sledi kreiranje uloge 'professor' koje je identifikovano lozinkom. A odmah zatim i odobravanje operacija koje će biti dodeljene ulozi 'professor'. Radi se o operacijama selekcije, ažuriranja i brisanja koje će korisnici sa ulogom 'professor' moći da izvrše nad tabelom system.Persons.

```
SQL> create role professor identified by professor1;  
Role created.
```

```
SQL> grant select, update, delete on System.Persons to professor;  
Grant succeeded.
```

Osim kreiranja korisnika, potrebno mu je obezbediti i mogućnost prijavljivanja, odnosno kreiranja sesije kako bi on mogao da se poveže na samu bazu podataka. Primer neuspele konekcije i obezbeđivanje kreiranja sesije prikazani su na slikama:

```
Enter user-name: lena  
Enter password:  
ERROR:  
ORA-01045: user LENA lacks CREATE SESSION privilege; logon denied
```

```
SQL> grant create session to lena;  
Grant succeeded.
```

Za sada postoji korisnik 'lena' kome su dodelje privilegije koje ima korisnik 'sysdba' (da mu nisu dodeljene ove privilegije - kao što se ne preporučuje, korisnik 'lena' ne bi imao nikakve privilegije). Takođe postoji i privilegije 'professor' koja još nije nikome dodeljena. Da bi se ove dve stvari povezale, tj. da bi korisnik dobio kreirane privilegije potrebno je izvršiti sledeću grant naredbu i sledeću set naredbu.

```
SQL> grant professor to lena;  
Grant succeeded.
```

```
SQL> set role professor identified by professor1;  
Role set.
```

Primer uspešnog korišćenja select naredbe nad system.Persons tabelom od strane 'lena' korisnika koji ima privilegiju za ovu operaciju:



```
SQL> select firstname, lastname from system.Persons;
```

firstname	lastname
Lena	Pet
Lena	Petrovic
Sava	Savic
Jova	Jovic
Ana	Anic

Primer neuspešnog korišćenja insert naredbe nad system.Persons tabelom od strane 'lena' korisnika koji nema privilegiju za ovu operaciju:

```
SQL> insert into system.Persons(personid, firstname, lastname, address, city) values (1, 'Ivan', 'Ivanovic', 'Ivanova 12', 'BG');
insert into system.Persons(personid, firstname, lastname, address, city) values (1, 'Ivan', 'Ivanovic', 'Ivanova 12', 'BG')
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

Zanimljivo je da grant naredba može da se odnosi na kolonu, ali samo za dodavanje i ažuriranje (insert i update) operacije. S toga, ukoliko je potrebno nešto više od ovoga, potrebno je koristiti Virtual Private Database.

5. Praćenje dodele sledećih privilegija samo korisnicima i ulogama kojima su te privilegije potrebne

Oracle baza podataka podrazumevano proverava sledeće privilegije: ALTER SYSTEM, AUDIT SYSTEM i CREATE EXTERNAL JOB. Međutim, Oracle preporučuje da se takođe vrši revizija sledećih privilegija: DBMS\_BACKUP\_RESTORE paket, SELECT ANY TABLE, privilegije koje imaju klauzulu WITH ADMIN... Revizija je prikazana pri kraju ovog dokumenta.

## Obezbeđivanje uloga

1. Dodeliti ulogu korisnicima samo ako su im potrebne sve privilegije uloge

Uloge (grupe privilegija) su korisne za brzo i lako davanje dozvola korisnicima. Iako mogu da se koriste uloge koje definiše Oracle, više kontrole i kontinuiteta se postiže ukoliko se kreiraju sopstvene uloge koje sadrže samo privilegije koje se odnose na potrebne zahteve. Oracle može da promeni ili ukloni privilegije koje su definisane s njegove strane. Ova situacija se dogodila sa CONNECT ulogom koja je ranije podrazumevala 8 privilegija, a sada podržava samo CREATE SESSION privilegiju.

Treba biti siguran da uloge koje se definišu sadže samo privilegije koje su potrebne za uspešno izvršenje posla. Ako korisnicima nisu potrebne sve privilegije obuhvaćene postojećom

ulogom, onda treba razmotriti drugačiji pristup. Može se koristiti drugi skup privilegija ili se može kreirati i dodeliti ograničena uloga.

Recimo da je 'lena' dobro poznat nalog koji može biti osetljiv i iskorišćen za zloupotrebu. Neophodno je striktno ograničiti privilegije korisnika 'lena'. Na primer, privilegija CREATE DBLINK dozvoljava pristup iz jedne baze podataka u drugu. Korisnik 'lena' ne bi trebalo da ima ovu vrstu mogućnosti, te je s toga treba odbaciti. Dalje se može izbaciti cela uloga za korisnika, jer se privilegije stečene pomoću uloge ne mogu odbaciti pojedinačno. Ponovo se može kreirati svoja uloga sa samo potrebnim privilegijama i dodeliti tu novu ulogu korisniku.

## 2. Ne odobriti korisničke uloge programerima aplikacija

Uloge ne treba da koriste programeri aplikacija, jer privilegije za pristup objektima šeme unutar uskladištenih programskih konstrukcija moraju biti direktno dodeljene.

## 3. Kreiranje i dodela uloge specifične za svaku instalaciju Oracle baze podataka

Ovaj princip omogućava organizaciji da zadrži detaljnu kontrolu nad svojim ulogama i privilegijama. Ovo takođe izbegava potrebu prilagođavanja ako Oracle Database promeni ili ukloni uloge definisane Oracle Database, kao što je to slučaj sa CONNECT.

# Obezbeđivanje lozinki

Kada se kreira korisnički nalog, Oracle baza podataka dodeljuje podrazumevanu politiku lozinke za tog korisnika. Politika lozinke definiše pravila za način na koji lozinka treba da se kreira, kao što je minimalni broj znakova, kada ističe i tako dalje. Lozinke se mogu pojačati.

## 1. Pažljivo biranje lozinke

Postoje minimalni zahtevi za lozinke koji moraju biti ispoštovani prilikom kreiranja iste. Sledeće stavke je moguće iskoristiti prilikom kreiranja ili menjanja lozinke:

- Neka lozinka ima dužinu između 12 i 30 bajtova (preporuka je uključiti i abecedne znakove i cifre)
- Neka lozinka sadrži najmanje jednu cifru, jedan znak velikih slova i jedan znak malih slova
- Neka lozinka sadrži posebne znakove

## 2. Da bi se napravila duža, složenija lozinka od kraće lozinke, mogu se iskoristiti prva slova reči neke poznate rečenice.

Na primer, "Ovo je projekat za predmet sistemi za upravljanje bazama podataka za koji je rok izrade 3 nedelje" bi bilo: `ijpzpszubpkjri3n`.

### 3. Dovoljno složena lozinka

Oracle Database obezbeđuje rutinu verifikacije složenosti lozinke, PL/SQL script, koja se može pokrenuti radi provere složenosti lozinke (da li je dovoljno složena ili ne). U pitanju je `utlpwdmg.sql` skripta koje se dobija nakon instalacije. Ako je potrebno, skripta se može izmeniti tako da obezbedi jaču lozinku.

```
SQL> @?\\rdbms\\admin\\utlpwdmg.sql;
Profile altered.
```

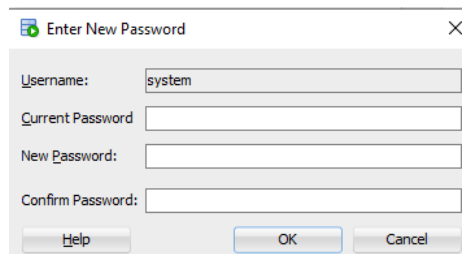
### 4. Promena podrazumevane korisničke lozinke

Oracle baza podataka se instalira sa skupom unapred definisanih, podrazumevanih korisničkih naloga. Bezbednost se najlakše narušava kada podrazumevani korisnički nalog baze podataka i dalje ima podrazumevanu lozinku. Ovo posebno važi za korisnički nalog 'lena', koji je dobro poznat nalog koji može biti ranjiv na uljeze. Da se izvršio pregled korisničkih naloga koji imaju podrazumevane lozinke, potrebno je prikazati podatke iz tabele `DBA_USERS_WITH_DEFPWD` rečnika.

U nastavku su prikazana tri načina za promenu lozinke koja se odnose na `ALTER` komandu, `password` komandu i promenu lozinke kroz SQL developer alat.

```
SQL> alter user sara identified by sara123;
User altered.
```

```
SQL> password
Changing password for SARA
Old password:
New password:
Retype new password:
Password changed
```



### 5. Promena podrazumevane lozinke administrativnih korisnika

Za admin naloge kao što su `SYS`, `SYSTEM`, `SYSMAN` i `DBSNMP` mogu se koristiti iste ili različite lozinke. Oracle preporučuje da se koriste različite lozinke za svaki od njih. U bilo kom Oracle okruženju (proizvodnom ili testnom), treba dodeliti jake, bezbedne i različite lozinke. Ako koristite Database Configuration Assistant da biste kreirali novu bazu podataka, tada je potrebno da unesete lozinke za `SYS` i `SISTEM` naloge, čime se onemogućavaju podrazumevane lozinke.

## 6. Upravljanje lozinkama

Dobro je primeniti osnovna pravila za upravljanje lozinkama (kao što su dužina lozinke, istorija, složenost...) na sve korisničke lozinke. Informacije o korisničkim nalogima se mogu pronaći postavljanjem upita za prikaz sadržaja DBA\_USERS tabele. Kolona PASSWORD tabele DBA\_USERS pokazuje da li je lozinka globalna, spoljna ili nulta. Prikaz DBA\_USERS pruža korisne informacije kao što su status korisničkog naloga, da li je nalog zaključan i verzije lozinke.

Oracle takođe preporučuje, ako je moguće, korišćenje Oracle jake autentifikacije sa uslugama mrežne provere autentičnosti (kao što je Kerberos), karticama tokena ili pametnim karticama. Ove usluge obezbeđuju snažnu autentifikaciju korisnika i pružaju zaštitu od neovlašćenog pristupa Oracle bazi podataka.

## 7. Bez čuvanja korisničke lozinke kao tekst u Oracle tabelama

Radi bolje bezbednosti, nije pametno čuvati lozinku u formatu čitljivom ljudima u Oracle tabelama. Ovaj problem se može rešiti korišćenjem "bezbednog spoljnog skladišta lozinke za šifrovanje lozinke u okviru Oracle novčanika". Oracle novčanik je bezbedni softverski kontejner koji čuva podatke za autentifikaciju i potpisivanje.

Kada se kreira ili izmeni lozinka za korisnički nalog, Oracle baza podataka automatski kreira kriptografski heš ili sažetak lozinke. Pregled sadržaja DBA\_USERS tabele takođe ima kolonu pod nazivom PASSWORD\_VERSIONS, koja navodi tipove kriptografskog heša koji postoje za lozinku korisnika (11G ili 12C).

## Revizija aktivnosti

Revizija je praćenje i evidentiranje odabranih radnji baze podataka korisnika. U standardnoj reviziji, koristi se parametar AUDIT za reviziju SQL izraza, privilegija i objekata šeme. Postoje i aktivnosti koje Oracle uvek prati, bez obzira da li je revizija omogućena. Ove aktivnosti su uglavnom vezane za administrativne privilegije, pokretanje baze podataka i gašenje baze podataka.

Postoji još jedna vrsta revizije, takozvana fina-zrnasta revizija. Fino-zrnasta revizija omogućava da se revizija izvrši na najpreciznijem nivou. Moguće ju je koristiti za reviziju aktivnosti na osnovu pristupa ili promena u kolona. Mogu se kreirati pravila koja definišu specifične uslove koji se moraju desiti da bi se revizija desila. Na primer, može se izvršiti revizija određene kolone tabele da bi se saznalo kada i ko je pokušao da joj pristupi tokom određenog vremenskog perioda.

Revizija se tipično koristi za obavljanje sledećih aktivnosti:

- Istraživanje sumnjivih aktivnosti. Na primer, ako korisnik briše podatke iz tabela, administrator može odlučiti da izvrši reviziju svih veza sa bazom podataka i svih uspešnih i neuspešnih brisanja redova iz svih tabela u bazi podataka.
- Obaveštavanje revizora o radnjama neovlašćenog korisnika. Na primer, neovlašćeni korisnik bi mogao da promeni ili izbriše podatke, ili korisnik ima više privilegija nego što se očekivalo, što može dovesti do ponovne procene autorizacije korisnika.
- Otkrivanje problema sa implementacijom autorizacije ili kontrole pristupa. Na primer, možegu se kreirati pravila revizije za koje se očekuje da nikada neće generisati zapis revizije (jer su podaci zaštićeni na druge načine). Međutim, ako ova pravila generišu zapise revizije, onda dolazi do problema.
- Praćenje i prikupljanje podataka o specifičnim aktivnostima baze podataka. Na primer, administrator baze podataka može da prikupi statistiku o tome koje tabele se ažuriraju, koliko logičkih I/O operacija se izvodi ili koliko se istovremenih korisnika povezuje u vršnim trenucima...

Oracle beleži aktivnosti revizije u evidenciji revizije. Revizorski zapisi pružaju informacije o operaciji koja je revidirana, korisniku koji obavlja operaciju i datumu i vremenu operacije. Revizorski zapisi mogu biti uskladišteni ili u tabeli rečnika podataka ili u fajlovima operativnog sistema.

Sam proces revizije prikazan je na slikama ispod. Sastoji se od više koraka i primera. Pre svega je potrebno izvršiti komandu `audit table` kojom se omogućuje revizija tabela pod kojom se podrazumeva kreiranje i brisanje tabela (`create` i `drop`).

```
SQL> audit table;
Audit succeeded.
```

Nakon ovoga potrebno je, konkretno navesti operacije čija se revizija želi, tako da se pažnja usmeri samo na te operacije. Na slici je prikazano revidiranje `select`, `update` i `delete` operacija, ali ne i `insert`. Što znači da će se pratiti i pamtiti samo SQL upiti koji se odnose na selektovanje, ažuriranje i brisanje podataka iz tabele `system.Persons`.

Što se tiče 'by access' dela na kraju, on se odnosi na to da ako korisnik istu naredbu pokrene više puta, svaki put će biti registrovan. Umesto 'by access' dela na kraju, može da stoji i 'by session', tada će pokretanje iste naredbe više puta tokom sesije biti registrovano samo jednom.

```
SQL> audit select, update, delete on system.Persons by access;
Audit succeeded.
```

Pregledom vrednosti `audit_file_dest` parametra, može se videti lokacija u fajl sistemu na kojoj će biti čuvani revizorski zapisi.

```
SQL> show parameter audit_file_dest
```

NAME	TYPE	VALUE
audit_file_dest	string	C:\USERS\KORISNIK\DOWNLOADED\ADMIN\ORCL\ADUMP

Kada se koristi revizija, Oracle upisuje zapise revizije u DBA\_AUDIT\_TRAIL tabelu. Dok revizija još nije pokrenuta ova tabela je prazna.

Sada je izvršena select naredba nad system.Persons tabelom. Pošto je prethodno izvršena naredba koja označava praćenje select naredbe, DBA\_AUDIT\_TRAIL tabela više nije prazna. U tabeli se može videti da je ista naredba izvršena tri puta i da su sva tri puta registrovana zahvaljujući 'by access' oznaci. Takođe se može videti:

- username korisnika koji je izvršio SQL naredbu
- vlasnik tabele nad kojom se to desilo
- datum izvršenja
- kod akcije (bolje je koristiti action\_name kako bi bio naziv operacije - select, insert, logon...)
- SQL naredba

```
SQL> select os_username, username, owner,timestamp, action, sql_text
2  from dba_audit_trail;
```

OS_USERNAME	USERNAME	OWNER	TIMESTAMP	ACTION	SQL_TEXT
DESKTOP-PPMGGES\Korisnik	LENA	SYSTEM	28-MAY-22	3	select * from system.Persons
DESKTOP-PPMGGES\Korisnik	LENA	SYSTEM	28-MAY-22	3	select * from system.Persons
DESKTOP-PPMGGES\Korisnik	LENA	SYSTEM	28-MAY-22	3	select * from system.Persons

Na sledećoj slici je izvršena update naredba nad istom tabelom. Zapravo na slici se nalaze dve update naredbe. Prva je neuspešna, nije izvršena, jer id ne postoji kao kolona u tabeli system.Persons. Nakon toga je greška ispravljena, te je update naredba uspešno prošla i podaci u tabeli su promenjeni.

```
SQL> update system.Persons
  2  set firstname='lenalenalena'
  3  where id = 1;
where id = 1
      *
ERROR at line 3:
ORA-00904: "ID": invalid identifier

SQL> update system.Persons set firstname = 'lenalenalena' where personid = 1;

1 row updated.

SQL> select personid, firstname from system.Persons where personid = 1;

   PERSONID
-----
   FIRSTNAME
-----
          1
lenalenalena
```

Ukoliko se nakon ovoga opet pogleda sadržaj tabele DBA\_AUDIT\_TRAIL, možemo videti da za update operaciju ima ukupno 2 zapisa. Ovo znači da je i update naredba koja nije izvršena jer je u sebi imala grešku, ipak revidirana.

Ova situacija je veoma zgodna jer je moguće pratiti neuspešne SQL naredbe. Ukoliko postoji više neuspešnih naredbi smatra se da ovo potencijalno može predstavljati rizik.

```
SQL> select os_username, username, owner, timestamp, sql_text
2 from dba_audit_trail;

OS_USERNAME
-----
-----
USERNAME      OWNER      TIMESTAMP  SQL_TEXT
-----
DESKTOP-PPMGES\Korisnik
LENA          SYSTEM    28-MAY-22  select * from system.Persons

DESKTOP-PPMGES\Korisnik
LENA          SYSTEM    28-MAY-22  select * from system.Persons

DESKTOP-PPMGES\Korisnik
LENA          SYSTEM    28-MAY-22  select * from system.Persons

DESKTOP-PPMGES\Korisnik
LENA          SYSTEM    28-MAY-22  update system.Persons
                                     set firstname='lenalenalena'
                                     where id = 1

DESKTOP-PPMGES\Korisnik
LENA          SYSTEM    28-MAY-22  update system.Persons set firstname = 'lenalenalen
                                     a' where personid = 1

DESKTOP-PPMGES\Korisnik
LENA          SYSTEM    28-MAY-22  select personid, firstname from system.Persons whe
                                     re personid = 1

6 rows selected.
```

Znajući da veliki broj neuspešnih naredbi nije dobar, mogu se pratiti samo neuspešne naredbe. Revizija neuspešnih naredbi se može pratiti bilo za šta uključujući konkretno za nekog korisnika ili tabelu. Naredbom ispod prate se neuspešne naredbe korisnika 'lena' nad tabelama.

```
SQL> audit table by lena whenever not successful;
```

Neuspešna prijava takođe može biti nezgodna. Može označavati pokušaj prijave na nalog koji ne pripada osobi koja pokušava prijavu. Na sledeći način je moguće pratiti neuspešne prijave:

```
SQL> audit create session whenever not successful;

Audit succeeded.
```

Pokušaj prijave korisnika 'lena' sa pogrešnom lozinkom i pokušaj prijave korisnika 'ana' koji uopšte ne postoji:



```
Enter user-name: lena
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied

Enter user-name: ana
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied

Enter user-name: lena
Enter password:
Last Successful login time: Sat May 28 2022 15:37:31 +02:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

Zapisi u tabeli nakon neuspješnih prijava su na slici dole. Treći pokušaj je uspешan, s toga nije zapamćen.

DESKTOP-PPMGGES\Korisnik		
LENA	28-MAY-22	LOGON
DESKTOP-PPMGGES\Korisnik		
ANA	28-MAY-22	LOGON

## Obezbeđivanje podataka

### 1. Ograničiti pristup operativnom sistemu

Da bi se ovo postiglo, moguće je iskoristiti neke od sledećih stavki nabrojanih ispod:

- Ograničenje broj korisnika operativnog sistema
- Ograničenje privilegija naloga operativnog sistema (admin nalog sa root privilegijama) na Oracle Database host računaru na najmanje privilegije koje su potrebne korisniku za obavljanje neophodnih zadataka
- Ograničenje mogućnosti izmene podrazumevanih dozvola za datoteku i instalacioni direktorijum Oracle baze podataka ili njegov sadržaj. Čak i privilegovani korisnici operativnog sistema i Oracle vlasnik ne bi trebalo da menjaju ove dozvole
- Ograničenje simboličkih veza. Treba biti siguran da nevedenu putanju do baze podataka, ni datoteku ni bilo koji deo putanje ne može da menja nepouzdan korisnik. Datoteka i sve komponente putanje treba da budu u vlasništvu administratora baze podataka ili naloga "od poverenja", kao što je root

## 2. Omogućiti zaštitu rečnika podataka

Oracle preporučuje da klijenti implementiraju zaštitu rečnika podataka kako bi sprečili korisnike da imaju ANY sistemsku privilegiju kako bi je koristili nad rečnikom podataka. Da bi se omogućila zaštita rečnika, potrebno je podesiti `O7_DICTIONARY_ACCESSIBILITY` konfiguracioni parametar na `FALSE`, u `init<sid>.ora` kontrolnoj datoteci.

```
O7_DICTIONARY_ACCESSIBILITY = false|
```

## 3. Šifrovanje osetljivih podataka i datoteka koje sadrže podatke baze podataka

U skladu sa uobičajenim zahtevima za usklađenost sa propisima, ukoliko postoje podaci kao što su brojevi kreditnih kartica i lozinke, oni se smatraju za osetljive podatke te ih je potrebno šifrovati. Kada se osetljivi podaci izbrišu iz baze podataka, šifrovani podaci se ne zadržavaju u blokovima podataka, datotekama operativnog sistema ili sektorima na disku.

U većini slučajeva, dobro je koristiti Transparentno šifrovanje podataka (Transparent Data Encryption) radi šifrovanja osetljivih podataka.

## Transparent data encryption - TDE

TDE transparentno šifrue podatke u Oracle bazama podataka. Zaustavlja neovlašćene pokušaje operativnog sistema da pristupi podacima baze podataka uskladištenim u datotekama, bez uticaja na to kako aplikacije pristupaju podacima koristeći SQL. TDE može da šifrue čitave prostore tabela aplikacije ili određene osetljive kolone. TDE je u potpunosti integrisan sa Oracle bazom podataka. Šifrovani podaci ostaju šifrovani u bazi podataka, bilo da se nalaze u datotekama za skladištenje prostora tabele, privremenim prostorima tabela, poništavanjem prostorima tabela ili drugim datotekama na koje se Oracle Database oslanja.

Ukoliko bilo ko, bilo kako dođe u posed datoteke u kojoj se čuvaju šifrovani fajlovi, nikada neće moći da dešifrue podatke.

TDE je potpuno transparentan za aplikacije. Šifrovanje i dešifrovanje se dešavaju na nivou skladištenja baze podataka, bez uticaja na interfejs koji aplikacije koriste. Korisnik ne može primetiti da dolazi do šifrovanja i dešifrovanja podataka.

Treba znati koji osetljivi podaci se čuvaju u bazi. Da bi se ovo saznalo Oracle obezbeđuje Oracle Database Security Assessment Tool (alat za procenu bezbednosti baze podataka). Većina kompanija se odlučuje da šifrue sve podatke aplikacije koristeći šifrovanje prostora tabele (tablespace).

Termini koji će se koristiti:

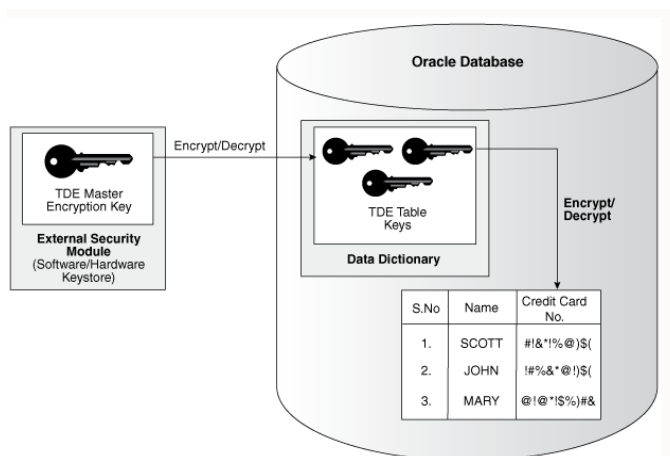
- Glavni ključ za šifrovanje (Master encryption key) – ključ za šifrovanje koji se koristi za šifrovanje sekundarnih ključeva za šifrovanje podataka koji se koriste za šifrovanje kolona i šifrovanje prostora tabele.
- Ključ tabele (Table key) – Ponekad se naziva ključ kolone, ovaj ključ se koristi za šifrovanje jedne ili više specifičnih kolona u datoj tabeli. Ovi ključevi se čuvaju u Oracle rečniku podataka, šifrovani su glavnim ključem za šifrovanje.
- Ključ tabelarnog prostora (Tablespace key) – ključ koji se koristi za šifrovanje prostora tabele. Ovi ključevi su šifrovani pomoću glavnog ključa i čuvaju se u zaglavlju prostora tabele šifrovanog prostora tabele, kao i u zaglavlju svakog operativnog sistema.
- Wallet – formatirana datoteka izvan baze podataka, šifrovana na osnovu šifrovanja zasnovanog na lozinki. Koristi se za čuvanje glavnog ključa TDE.

Osetljivi podaci se mogu šifrovati na nivou kolone ili na nivou prostora tabele. Na nivou kolone mogu se šifrovati osetljivi podaci koji se čuvaju u kolonama tabele. TDE šifrovanje prostora tabele omogućava šifrovanje svih podataka koji se čuvaju u prostoru tabele.

### Šifrovanje na nivou kolone

Šifrovanje kolone štiti poverljive podatke, kao što su brojevi kreditnih kartica i socijalnog osiguranja, koji se čuvaju u određenim kolonama tabele.

TDE šifrovanje kolona koristi dvoslojnu arhitekturu zasnovanu na ključu za transparentno šifrovanje i dešifrovanje osetljivih kolona tabele. TDE glavni ključ za šifrovanje se čuva u eksternom skladištu ključeva, koje može biti Oracle softversko ili hardversko skladište ključeva. Ovaj TDE glavni ključ za šifrovanje šifruije i dešifruije TDE ključ tabele, koji dalje šifruije i dešifruije podatke u koloni tabele.



Eksterno u odnosu na bazu podataka je softversko ili hardversko skladište ključeva. U okviru ovog softverskog ili hardverskog skladišta ključeva, nalazi se i TDE glavni ključ za šifrovanje.

U okviru baze podataka, u rečniku podataka, nalaze se ključevi TDE tabele. Postoje i tabele koje čuvaju šifrovane podatke. Ova ilustracija prikazuje tabelu koja ima šifrovane brojeve kreditnih kartica.

Softversko ili hardversko skladište ključeva se povezuje sa bazom podataka i vrši šifrovanje i dešifrovanje preko ključeva TDE tabele u rečniku podataka, koji onda primenjuju šifrovanje i dešifrovanje na zaštićenu kolonu tabele, u ovom slučaju na brojeve kreditnih kartica.

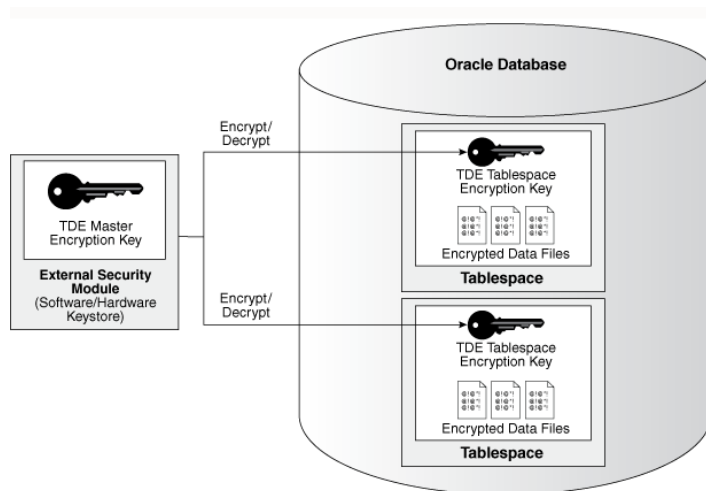
### Šifrovanje na nivou prostora tabele:

Šifrovanje prostora tabele omogućava šifrovanje celog prostora tabele. Svi objekti koji su kreirani u šifrovanom prostoru tabele biće automatski šifrovani. TDE šifrovanje prostora tabele je korisno ako tabele sadrže osetljive podatke u više kolona ili ako je potrebna zaštita cele tabele, a ne samo pojedinačne kolone. Nike potrebna detaljna analiza svake kolone tabele da bi se odredile kolone kojima je potrebno šifrovanje.

Pored ovoga, TDE šifrovanje prostora tabele koristi prednosti masovnog šifrovanja i keširanja kako bi obezbedilo poboljšane performanse.

Svi podaci u šifrovanom prostoru tabele se čuvaju u šifrovanom formatu na disku. Podaci se transparentno dešifruju za ovlašćenog korisnika koji ima potrebne privilegije da pregleda ili modifikuje podatke. Korisnici baze podataka ili aplikacija ne moraju da znaju da li su podaci u određenoj tabeli šifrovani na disku.

TDE šifrovanje prostora tabele koristi dvoslojnu arhitekturu zasnovanu na ključu za transparentno šifrovanje (i dešifrovanje) prostora tabele. TDE glavni ključ za šifrovanje se čuva u spoljnom bezbednosnom modulu (softver ili eksterno skladište ključeva). Ovaj TDE glavni ključ za šifrovanje se koristi za šifrovanje ključa za šifrovanje TDE prostora tabele, koji se zauzvrat koristi za šifrovanje i dešifrovanje podataka u prostoru tabele.



Kao i gore, eksterno u odnosu na bazu podataka je softversko ili hardversko skladište ključeva. U okviru ovog softverskog ili hardverskog skladišta ključeva, nalazi se i TDE glavni ključ za šifrovanje.

U okviru baze podataka nalaze se dva tabele prostora. Softversko ili hardversko skladište ključeva kontroliše šifrovanje za svaki od njih. Svaki prostor tabele ima svoj ključ za šifrovanje prostora tabele, koji šifrue datoteke podataka unutar prostora tabele.

Postoje 4 glavna koraka za omogućavanje Transparent Data Encryption-a:

1. Podešavanje parametara za wallet

Pre svega treba pronaći sqlnet.ora fajl u fajl sistemu. Treba ga otvoriti i u njega upisati sledeće (čime se specificira lokacija wallet-a):

```
ENCRYPTION_WALLET_LOCATION=
(SOURCE=
(METHOD=FILE)
(METHOD_DATA=
(DIRECTORY=$ORACLE_BASE/admin/$ORACLE_SID/wallet)))
```

Nakon ovoga nije loše proveriti ORACLE\_HOME i ORACLE\_UNQNAME, jer će kasnije biti jednostavnije uz njihovo korišćenje.

```
C:\Users\Korisnik>set ORACLE_HOME=C:\Users\Korisnik\Downloads\WINDOWS.X64_193000_db_home
C:\Users\Korisnik>echo %ORACLE_HOME%
C:\Users\Korisnik\Downloads\WINDOWS.X64_193000_db_home
```

```
C:\Users\Korisnik>set ORACLE_UNQNAME=orcl
C:\Users\Korisnik>echo %ORACLE_UNQNAME%
orcl
```

Sada slede konkretni koraci, moguće je koristi SQL plus ili SQL developer, takođe će biti potreban command prompt. Kombinacija ovih alata je korišćenja radi demonstracije.

Podešavanje WALLET\_ROOT parametra koji se koristi za određivanje osnovne lokacije wallet-a. Potrebno ga je podesiti sledećom komandom.

```
SQL> alter system set wallet_root='$ORACLE_BASE/admin/orcl/wallet' scope=spfile sid='*';
```

```
SQL> show parameter wallet_root
```

NAME	TYPE	VALUE
wallet_root	string	C:\USERS\KORISNIK\DOWNLOADS\ADMIN\ORCL\WALLET

Zatim sledi TDE\_CONFIGURATION parametar:

```
SQL> alter system set tde_configuration="KEYSTORE_CONFIGURATION=FILE" scope=both sid='*';
System altered.
```

## 2. Kreiranje skladišta ključeva

Kreiranje skladišta ključeva zaštićeno lozinkom ('welcome1', preporuka je jaka lozinka):

```
SQL> administer key management create keystore identified by welcome1;
```

Kreiranje skladišta ključeva za automatsko prijavljivanje koje omogućava automatsko otvaranje i zatvaranje skladišta ključeva kada god je to potrebno.

```
SQL> administer key management create auto_login keystore from keystore identified by welcome1;
```

Provera kreiranih fajlova gde je moguće primetiti dva fajla. Fajl "ewallet.p12" koji predstavlja skladište ključeva zaštićeno lozinkom, a "cwallet.sso" skladište ključeva za automatsko prijavljivanje.

```
C:\Users\Korisnik\Downloads\admin\orcl\WALLET\TDE>dir
Volume in drive C is SSD NEW WIN 10 X64
Volume Serial Number is 1C8C-5967

Directory of C:\Users\Korisnik\Downloads\admin\orcl\WALLET\TDE

05/16/2022  08:52 PM    <DIR>          .
05/16/2022  08:52 PM    <DIR>          ..
05/16/2022  08:52 PM                4,040 cwallet.sso
05/16/2022  08:52 PM                3,995 ewallet.p12
05/16/2022  08:51 PM                2,555 ewallet_2022051618515319.p12
               3 File(s)                10,590 bytes
               2 Dir(s)  16,136,192,000 bytes free
```

## 3. Podešavanje TDE glavnog ključa

Pre postavljanja TDE glavnog ključa u skladište ključeva, trebalo bi ga otvoriti. Da bi se skladište otvorilo uz pomoć lozinke, treba uraditi sledeće:

```
SQL> administer key management set keystore open force keystore identified by welcome1 container=all;
keystore altered.
```

Provera trenutnog stanja wallet-a:

```
SQL> select con_id, wallet_type, status from v$encryption_wallet;
```

CON_ID	WALLET_TYPE	STATUS
1	AUTOLOGIN	OPEN
2	AUTOLOGIN	OPEN
3	AUTOLOGIN	OPEN_NO_MASTER_KEY

Kao što se vidi po OPEN\_NO\_MASTER\_KEY statusu, u skladištu ključeva ne postoji ništa. Dakle, sada sledi postavljanje TDE glavnog ključa u skladište ključeva i zaključavanje skladišta.

```
SQL> administer key management set key force keystore identified by welcome1 with backup container=all;
keystore altered.
```

```
SQL> administer key management set keystore close identified by welcome1 container=all;
```

#### 4. Šifrovanje podataka

Za potrebe testiranja i provere biće kreirana dva prostora tabela. U jednom će biti šifrovani podaci, dok će u drugom biti normalni podaci.

```
create tablespace NOR_DATA datafile 'norm_data.dbf' size 10m autoextend on maxsize unlimited;

create tablespace TDE_DATA datafile 'tde_data.dbf' size 10m autoextend on maxsize unlimited encryption encrypt;
```

TDE podržava AES256, AES192 (podrazumevano za TDE šifrovanje kolona), AES128 (podrazumevano za TDE šifrovanje prostora tabele), ARIA128, ARIA192, ARIA256, GOST256, SEED128 i 3DES168 algoritme šifriranja. U ovom slučaju, podrazumevano je korišćen AES128. Ukoliko je potrebno koristiti neki drugi algoritam (konkretno AES256), to se može uraditi na sledeći način:

```
create tablespace TDE_DATA datafile 'tde_data.dbf' size 10m autoextend on maxsize unlimited encryption using 'AES256' encrypt;
```

Dodavanje tabela i podataka u prostore tabele:

```
create table nor_persons (
  id varchar2(10),
  firstname varchar2(20),
  lastname varchar2(20)
) tablespace NOR_DATA;

insert into nor_persons (id, firstname, lastname) values ('123', 'Lena', 'Pet');

create table tde_persons (
  id varchar2(10),
  firstname varchar2(20),
  lastname varchar2(20)
) tablespace TDE_DATA;

insert into tde_persons (id, firstname, lastname) values ('321', 'Lena', 'Pet');
```

Sadržaji tabela se nalaze se na slikama ispod. Na levoj slici su normalni podaci, za koje je očekivano da se normalno prikazuju, dok se na desnoj slici nalaze šifrovani podaci. Kao što je pomenuto, proces šifriranja i dešifrovanja podataka je potpuno transparentan, te je s toga slika desno takva kakva jeste.



ID	FIRSTNAME	LASTNAME
1 123	Lena	Pet

ID	FIRSTNAME	LASTNAME
1 321	Lena	Pet

U tabeli DBA\_TABLESPACES se nalaze informacije vezane za prostore tabela, gde je između ostalog moguće videti informacije vezane za šifrovane prostore tabele.

```
select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;
```

	TABLESPACE_NAME	STATUS	ENCRYPTED
1	SYSTEM	ONLINE	NO
2	SYSAUX	ONLINE	NO
3	UNDOTBS1	ONLINE	NO
4	TEMP	ONLINE	NO
5	USERS	ONLINE	NO
6	NOR_DATA	ONLINE	NO
7	TDE_DATA	ONLINE	YES

Može se primetiti da je prostor tabela "NOR\_DATA" normalan, a "TDE\_DATA" zapravo šifrovan. Postavlja se pitanje da li je moguće podesiti svi budući kreirani prostori imena budu automatski šifrovani. Jeste! Na sledeći način:

```
alter system set ENCRYPT_NEW_TABLESPACES = ALWAYS scope = both;
```

```
create tablespace FUTURE_DATA datafile 'future_data.dbf' size 10m autoextend on maxsize unlimited;
```

```
select TABLESPACE_NAME, STATUS, ENCRYPTED from DBA_TABLESPACES;
```

	TABLESPACE_NAME	STATUS	ENCRYPTED
1	SYSTEM	ONLINE	NO
2	SYSAUX	ONLINE	NO
3	UNDOTBS1	ONLINE	NO
4	TEMP	ONLINE	NO
5	USERS	ONLINE	NO
6	NOR_DATA	ONLINE	NO
7	TDE_DATA	ONLINE	YES
8	FUTURE_DATA	ONLINE	YES

## Alati za procenu sigurnosti Oracle baze podataka

Pored bezbednosnih resursa koji su dostupni u podrazumevanoj instalaciji baze podataka, Oracle obezbeđuje i nekoliko dodatnih proizvoda za bezbednost baze podataka. Ovi proizvodi obuhvataju:



- Oracle Advanced Security - koji omogućava zaštitu osjetljivih podataka korišćenjem Transparent Data Encryption i Oracle Data Redaction.
- Oracle Label Security - omogućuju vam da filtrirate korisnički pristup podacima na nivou reda.....
- Oracle Database Vault
- Oracle Data Safe
- Oracle Enterprise User Security
- Oracle Enterprise Manager Data Masking and Subsetting Pack
- Oracle Audit Vault and Database Firewall
- Oracle Key Vault

## Database Security Assessment Tool

Oracle Database Security Assessment Tool (DBSAT) je popularni alat komandne linije koji pomaže kod indentifikacije oblasti kod kojih implementacija, konfiguracija ili rad baze podataka mogu dovesti do rizika. Takođe može preporučiti promene i kontrole za ublažavanje tih rizika. DBSAT pomaže u proceni koliko je bezbedno baza podataka konfigurisana, određuje ko su korisnici i njihova prava i identifikuje gde se u bazi podataka nalaze osjetljivi podaci.

Ukratko, DBSAT je moguće koristiti bez dodatnih novčanih troškova, a omogućava korisnicima da brzo pronađu:

- Probleme sa bezbednosnom konfiguracijom i predlog za otklanjanje
- Korisnike i njihova prava
- Lokaciju, vrstu i količinu osjetljivih podataka

Oracle DBSAT se sastoji od sledećih komponenti:

Collector (kolektor) izvršava SQL upite i pokreće komande operativnog sistema kako bi prikupio podatke iz sistema koji će se koristiti za procenu. To radi prvenstveno tako što ispituje poglede (views) rečnika baze podataka. Prikupljeni podaci se upisuju u JSON datoteku koju koristi DBSAT reporter za analizu.

Reporter analizira prikupljene podatke i generiše izveštaj o proceni bezbednosti Oracle baze podataka u HTML, Excel, JSON i tekstualnim formatima. Reporter može da radi na bilo kojoj mašini: računaru, laptopu ili serveru. Pokretanje reportera nije ograničeno za server baze podataka ili na istoj mašini kao kolektor.

Discoverer (Otkrivač) izvršava SQL upite i prikuplja podatke iz sistema za procenu, na osnovu podešavanja navedenih u konfiguracionim datotekama. To radi prvenstveno tako što ispituje poglede rečnika baze podataka. Prikupljeni podaci se zatim koriste za generisanje izveštaja o proceni osjetljivih podataka Oracle baze podataka u HTML i CSV formatima. Discoverer može da radi na bilo kojoj mašini: računaru, laptopu ili serveru. Pokretanje Discoverer-a nije ograničeno za istu mašinu kao kolektor ili reporter.

## Oracle DataSafe

Oracle Data Safe, koji je do Oracle Cloud Infrastructure-a, omogućava organizacijama da razumeju osetljivost podataka, procenjuju rizike podataka, maskiraju osetljive podatke, primenjuju i nadgledaju bezbednosne kontrole, procenjuju bezbednost korisnika i nadgledaju aktivnosti korisnika. Ove mogućnosti pomažu u upravljanju svakodnevnim zahtevima bezbednosti i usklađenosti Oracle baza podataka, kako lokalno tako i u oblaku.

- Procena korisnika - smanjuje rizik korisnika upravljanjem privilegijama i autentifikacijom. Moguće je brzo identifikovati rizično ponašanje i privilegovane korisnike. DataSafe identifikuje koji korisnici predstavljaju najveći rizik, pregleda privilegije date tim korisnicima i omogućava analizu aktivnosti korisnika. Procenjuje informacije o profilu kao što su tip korisnika, jačina lozinke, poslednje prijavljivanje i starost lozinke.
- Revizija aktivnosti - prikuplja revizijske podatke iz baza podataka i identifikuje anomalne operacije. Omogućava jednostavnije upravljajte politikama revizije i upozorenja.
- Otkrivanje osetljivih podataka - otkrivanje i klasifikacija osetljivih podataka na osnovu biblioteke.

## Reference

<https://www.oracle.com/br/a/tech/docs/technical-resources/twp-transparent-data-encryption-best-practices.pdf>  
<https://logic.edchen.org/how-oracle-enable-tde-on-rac-19c-db/#set-wallet-parameters>  
<https://www.oracle.com/a/tech/docs/dbsec/dbsat-ds-june2021.pdf>  
<https://www.oracle.com/uk/database/technologies/security/dbsat.html>  
[https://www.youtube.com/watch?v=kpDhtutyWNU&ab\\_channel=OracleWorld](https://www.youtube.com/watch?v=kpDhtutyWNU&ab_channel=OracleWorld)  
<https://www.oracle.com/security/database-security/data-safe/>  
[https://www.youtube.com/watch?v=8ZJQTjRI8OE&ab\\_channel=ICyb3r](https://www.youtube.com/watch?v=8ZJQTjRI8OE&ab_channel=ICyb3r)  
<https://logic.edchen.org/how-oracle-enable-tde-on-rac-19c-db/#set-wallet-parameters>  
<https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/introduction-to-oracle-database-security.html#GUID-91DB199F-50F3-4A86-8876-1579149A1930>  
<https://www.oracle.com/uk/database/technologies/security/dbsat.html>