in2 GROUP

Tomislav Kušanić

Blockchain je više od

hroug '23
godišnja konferencija

Tko smo mi

- Vodeći pružatelj IT usluga u javnom i privatnom sektoru u Hrvatskoj
- Dio Constellation Software Incorporated grupe
- 30 godina iskustva
- 600 zaposlenih
- 350 klijenata
- 650 projekata

O meni

- Tomislav Kušanić
  - tomislav.kusanic@in2.hr
- 15 godina iskustva sa Oracle tehnologijama (E-Business Suite, Apex, BI)

Sadržaj predavanja

- Općenito o Blockchain tehnologiji
- Blockchain tablice u Oracle bazi

Uvod u Blockchain

- Digitalni dnevnik dijeljen u mreži kompjutera
- Bilo kakva vrijednost može se pratiti u blochckain mreži
- Svaki blok u lancu ima jedinstveni digitalni potpis
- Blokovi vezani kronološkim redom formiraju lanac
- Otporni na petljanje i hakiranje

Digitalni dnevnik dijeljen u mreži kompjutera
Nema potrebe za centralnim autoritetom
Bilo kakva vrijednost može se pratiti u blochckain mreži
Svaki blok u lancu ima jedinstveni digitalni potpis
Blokovi vezani kronološkim redom formiraju lanac
Otporni na petljanje i hakiranje

Uvod u Blockchain

- Nepromjenjivi zapisi
  - niti jedan sudionik ne može mijenjati ili petljati sa transakcijama u dnevniku
  - originalne transakcije se ne mogu mijenjati ili brisati
  - ispravke se rade ispravljajućim transakcijama

Uvod u Blockchain

- Pametni ugovori
  - jednostavni programi pohranjeni na samom blockchainu
  - izvode se kad su ispunjeni uvjeti
  - automatiziraju izvođenje dogovora
  - sudionici su odmah sigurni u rezultat
  - nema posrednika i gubljenja vremena
  - mogu automatizirati proces automatskim pokretanjem sljedeće akcije

Primjeri upotrebe

1. Decentralizirana pohrana podataka/audita
2. Sljedivost u opskrbnom lancu i autentikacija u trgovačkoj zajednici
3. Transakcije razmjena više strana:
    1. plaćanja
    2. prijenos sredstava
    3. praćenje prava vlasništva
4. Digitalni identitet ili certifikacija kroz više različitih izdavača
5. Poslovne transakcije bazirane na usklađivanje predmeta i dokumenata više strana
6. Multi-brand sustavi lojalnosti

Blockchain u Oracle bazi

- Blockchain tablice predstavljene u 21c
- Dodane u 19c
- Nove funkcionalnosti u 23c

Blockchain tablice

- Insert only tablice
- Organiziraju retke u lance
- Svaki redak vezan na prethodni redak u lancu
- Petljanje sa retkom mijenja njegov hash čime se ukazuje na manipulaciju podacima
- Opcionalni korisnički potpis

Insert only tablice koje organiziraju retke u lance
Svaki redak je vezan na prethodni redak u lancu korištenjem kriptografskog hasha
Kriptografski hash retka je baziran na podacima retka i hasha prethodnog retka
Petljanje sa retkom mijenja njegov hash, a time utječe i na sve sljedeće retke u lancu čime se ukazuje na manipulaciju podacima
Opcionalni korisnički potpis se može dodati rad povećane zaštite od prevara, zahtijeva digitalni certifikat

Blockchain tablice

- Indeksirane i particionirane
- Kontrolirano brisanje tablica
- Reci mogu biti selektivno brisani i sačuvani
- Koriste se u transakcijama i upitima zajedno sa regularnim tablicama

Blockchain tablice

- Sprečavaju neautorizirane promjene podataka
- Samo kreiranje zapisa
- Definirani period zadržavanja

Blockchain tablice sprečavaju neautorizirane promjene podataka od strane zaposlenih ili hakera sa ukradenim lozinkama zaposlenih
Moguće je samo kreiranje zapisa
Korisnici ne mogu brisati retke unutar definiranog perioda zadržavanja

Blockchain tablice

- Nepromjenjiva definicija blockchain tablice
- Nije dozvoljena konverzija između blockchain i normalnih tablica
- Nepromjenjivi podaci o tablici u database dicitionaryu

Baza ne dopušta korisnicima da mijenjaju definiciju blockchain tablice
Nije dozvoljena konverzija između blockchain i normalnih tablica
Podaci o tablici u database dicitionaryu se ne mogu mijenjati

Blockchain tablice

- Sažetak tablice se generira na zahtjev i potpisuje
- Sažetak baziran na kolonama metapodataka za zadnji redak
- Promjena rezultira promjenom vrijednosti sažetka

Kriptografski sažetak blockchain tablice se generira na zahtjev i potpisuje se privatnim ključem vlasnika bazne sheme
Sažetak je baziran na kolonama metapodataka za zadnji redak svakog lanca u tablici
Bilo kakva promjena rezultira promjenom vrijednosti sažetka

Blockchain tablice

- Sažetak periodički generiran i pohranjen
- Provjera sažetka za raspon redaka između dva vremenska trenutka

Kriptografski sažetak se može periodički generirati i pohraniti u distribuirani osigurani repozitorij
Provjera sažetka za raspon redaka između dva vremenska trenutka može otkriti prikrivene neautorizirane promjene

Blockchain tablice

- Sprečava neprimijećene, neautorizirane promjene podataka
- Krajnji korisnici mogu potpisati novi redak
- Sprečava lažno predstavljanje
- Omogućuje provjeru integriteta podataka

Sprečava neprimijećene, neautorizirane promjene podataka korištenjem ukradenih lozinki krajnjih korisnika
Krajnji korisnici mogu kriptografski potpisati novi redak i time se potvrđuje uloga korisnika u kreiranju retka
Digitalni certifikat i privatni ključ sprečavaju lažno predstavljanje i provjeru integriteta podataka

Blockchain tablice

- Blockchain tehnologija je direktno integrirana u Oracle bazu
- Koristi napredne funkcionalnosti baze
- Minimalne promjene na postojeće aplikacije
- Nema novih infrastrukturnih zahtjeva
- Kombiniranje blockchain i običnih tablica u upitima i transakcijama

Blockchain tehnologija je direktno integrirana u Oracle bazu za naprednu zaštitu podataka
Koristi napredne funkcionalnosti baze, uključujući analitiku nad kriptografski osiguranim podacima
Minimalne promjene na postojeće aplikacije bez novih infrastrukturnih zahtjeva
Omogućava korisnicima kombiniranje blockchain i običnih tablica u upitima i transakcijama

Ulančavanje redaka

- Redak vezan na prethodni redak u lancu
- Lanac redaka može provjeriti svaki sudionik blockchaina
- 32 lanca
- Lanci identificirani jedinstvenom kombinacijom ID instance i ID lanca

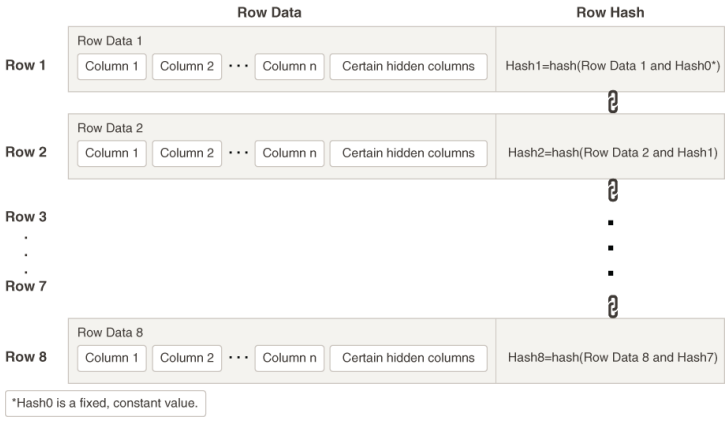
Redak u blockchain tablici se vezan na prethodni redak u lancu
Lanac redaka može provjeriti svaki sudionik blockchaina
Svaka blockchain tablica sadrži 32 lanca numerirana od 0 do 31
Lanci se identificiraju jedinstvenom kombinacijom ID instance i ID lanca

Ulančavanje redaka

- Korisničke i skrivene kolone
- Jedinstvena sekvenca unutar lanca
- Svaki redak vezan za prethodni redak

Reci se sastoje od korisničkih i skrivenih kolona
Nakon unosa u bazu, retku se dodjeljuje jedinstvena sekvenca unutar lanca za 1 veća od prethodnog retka u lancu
Svaki redak je vezan za prethodni redak

Ulančavanje redaka



Ulančavanje redaka

- Reci jedinstveno identificirani kombinacijom ID instance, ID lanca i sekvence
- Generiran SHA-512 hash
- Hash0

Reci se jedinstveno identificiraju kombinacijom ID instance, ID lanca i sekvence zapisa unutar lanca
Kad se redak unese u bazu, SHA-512 hash se generira na temelju podataka retka i hash vrijednosti prethodnog retka
Prvi redak koristi fiksnu, konstantnu vrijednost (Hash0) kao hash prethodnog retka

Ulančavanje redaka

- Jedna transakcija insertira u više različitih blockchain tablica
- Jedna transakcija insertira u isti lanac
- Pozicija retka odgovara redoslijedu insertiranja
- Automatski odabir lanca prilikom commita

Jedna transakcija može insertirati retke u više različitih blockchain tablica
Reci koje insertira jedna transakcija se dodaju u isti lanac
Pozicija retka unutar lanca odgovara redoslijedu insertiranja u tablicu
Baza automatski odabire u koji lanac će se dodati reci prilikom commita

Ulančavanje redaka

- Kod paralelnih tranksacija redoslijed dodavanja u lanac ovisi o redoslijedu commita
- Reci vezani u lanac nakon commita
- Viša latencija commita
- Izbjegavati insertiranje velikog broja redaka u jednoj transakciji

Kad više korisnika istovremeno insertira retke u isti lanac, redoslijed dodavanja u lanac ovisi o redoslijedu commita
Reci su vezani u lanac nakon commita
Insertiranje velikog broja redaka u jednoj transakciji rezultira višom latencijom commita
Preporuča se izbjegavanje insertiranja velikog broja redaka u jednoj transakciji

Ograničenja blockchain tablica

- Nepodržani tipovi podataka:
  - ROWID
  - LONG
  - Object type
  - TIMESTAMP WITH TIME ZONE
  - TIMESTAMP WITH LOCAL TIME ZONE
  - BFILE
  - XMLType

Ograničenja blockchain tablica

- Nepodržane funkcionalnosti:
  - kreiranje blockchain tablica u CDB-u
  - update i merge nad recima
  - dodavanje, micanje i preimenovanje kolona
  - truncate tablice
  - brisanje particija
  - insertiranje podataka korištenjem paralelnih DML naredbi
  - BEFORE ROW triggeri za update naredbe
  - definiranje ADO, VPD i OLS postavki
  - online redefinition
  - promjena regularne tablice u blockchain tablicu i obrnuto

Kreiranje blockchain tablica

```
CREATE BLOCKCHAIN TABLE transaction_ledger
(transaction_id NUMBER
,transaction_date   DATE
,transaction_user   VARCHAR2(50))
NO DROP UNTIL 8 DAYS IDLE
NO DELETE UNTIL 356 DAYS AFTER INSERT
HASHING USING "SHA2_512" VERSION "V1";
```

Kreiranje blockchain tablica

```
CREATE BLOCKCHAIN TABLE transaction_ledger_partitioned
(transaction_id NUMBER
,transaction_date    DATE
,transaction_user    VARCHAR2(50))
NO DROP UNTIL 16 DAYS IDLE
NO DELETE UNTIL 356 DAYS AFTER INSERT
HASHING USING "SHA2_512" VERSION "V1"
PARTITION BY RANGE(transaction_date)
(PARTITION p0 VALUES LESS THAN (TO_DATE('31.03.2023','DD.MM.YYYY'))
,PARTITION p1 VALUES LESS THAN (TO_DATE('30.06.2023','DD.MM.YYYY'))
,PARTITION p2 VALUES LESS THAN (TO_DATE('30.09.2023','DD.MM.YYYY'))
,PARTITION p3 VALUES LESS THAN (TO_DATE('31.12.2023','DD.MM.YYYY'))
);
```

Kreiranje blockchain tablica

```
select   *
from     all_tab_columns
where    table_name = 'TRANSACTION_LEDGER';
```

OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE
HROUG2023	TRANSACTION_LEC	TRANSACTION_ID	NUMBER
HROUG2023	TRANSACTION_LEC	TRANSACTION_DAT	DATE
HROUG2023	TRANSACTION_LEC	TRANSACTION_USE	VARCHAR2

Skrivene kolone

Naziv kolone	Tip	Opis
ORABCTAB_INST_ID$	NUMBER (22)	ID instance na kojoj se redak insertira.
ORABCTAB_CHAIN_ID$	NUMBER (22)	ID lanca u koji se redak insertira. Valjane vrijednosti su 0 do 31.
ORABCTAB_SEQ_NUM$	NUMBER(22)	Redni broj retka u lancu. Svaki redak koji se dodaje u lanac dobiva jedinstveni redni broj koji je za 1 veći od prethodnog retka u lancu. Nedostajući reci mogu biti detektirati na temelju te kolone. Kombinacija ID-a instance, lanca i rednog broja jedinstveno identificira redak u blockchain tablici
ORABCTAB_CREATION_TIME$	TIMESTAMP WITH TIME ZONE	Vrijeme kreiranja retka u UTC formatu.
ORABCTAB_USER_NUMBER$	NUMBER (22)	ID baznog korisnika koji je kreirao redak.
ORABCTAB_HASH$	RAW(2000)	Hash vrijednost retka koja se računa na temelju sadržaja retka i hash vrijednosti prethodnog retka u lancu.
ORABCTAB_SIGNATURE$	RAW(2000)	Korisnički potpis retka koji se računa koristeći hash vrijednost retka.
ORABCTAB_SIGNATURE_ALG$	NUMBER(22)	Algoritam koji se koristi za potpisivanje retka.
ORABCTAB_SIGNATURE_CERT$	RAW(16)	GUID certifikata korištenog za potpisivanje retka.
ORABCTAB_SPARE$	RAW(2000)	Rezervirano za buduću upotrebu

Svaki redak u blockchain tablici sadrži skrivene kolone koje baza popunjava prilikom commita

Izmjena blockchain tablica

- Moguće je samo povećati period zadržavanja tablice i redaka:

```
alter table transaction_ledger no drop until 16 days idle;

alter table transaction_ledger_partitioned
no delete until 712 days after insert locked;
```

Pregled blockchain tablica

```
select   *
from     user_blockchain_tables;
```

TABLE_NAME	ROW_RETENTION	ROW_RETENTION_LOCKED	TABLE_INACTIVITY_RETENTION
TRANSACTION_LEDGER	356	NO	16
TRANSACTION_LEDGER_PARTITIONED	712	YES	16

Insertiranje u blockchain tablice

```
insert into transaction_ledger
(transaction_id
,transaction_date
,transaction_user)
values
(1
,sysdate
,'test_user'
);
```

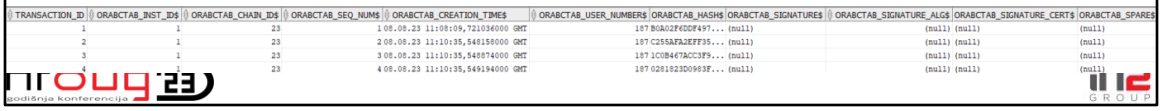
Dohvat podataka iz blockchain tablica

```
select   *
from     transaction_ledger;
```

TRANSACTION_ID	TRANSACTION_DATE	TRANSACTION_USER
1	8/10/2023, 8:09:22 AM	test_user

Dohvat podataka iz blockchain tablica

```
select  transaction_id
,       ORABCTAB_INST_ID$
,       ORABCTAB_CHAIN_ID$
,       ORABCTAB_SEQ_NUM$
,       ORABCTAB_CREATION_TIME$
,       ORABCTAB_USER_NUMBER$
,       ORABCTAB_HASH$
,       ORABCTAB_SIGNATURE$                    commit;
,       ORABCTAB_SIGNATURE_ALG$
,       ORABCTAB_SIGNATURE_CERT$
,       ORABCTAB_SPARE$
from    transaction_ledger;
```



TRANSACTION_ID	ORABCTAB_INST_ID$	ORABCTAB_CHAIN_ID$	ORABCTAB_SEQ_NUM$	ORABCTAB_CREATION_TIME$	ORABCTAB_USER_NUMBER$	ORABCTAB_HASH$	ORABCTAB_SIGNATURE$	ORABCTAB_SIGNATURE_ALG$	ORABCTAB_SIGNATURE_CERT$	ORABCTAB_SPARE$
1	1	23	1	08.08.23 11:08:09,721036000 GMT	187	B0A02F4DDF497...	(null)	(null)	(null)	(null)
2	1	23	2	08.08.23 11:10:35,548158000 GMT	187	C255AFA2EFF35...	(null)	(null)	(null)	(null)
3	1	23	3	08.08.23 11:10:35,548874000 GMT	187	1C0B467ACC3F9...	(null)	(null)	(null)	(null)
4	1	23	4	08.08.23 11:10:35,549194000 GMT	187	0281823D0983F...	(null)	(null)	(null)	(null)

Brisanje redaka iz blockchain tablica

- Samo reci kojima je istekao period zadržavanja mogu biti obrisani:

```
delete   transaction_ledger
where    transaction_id = 4;
```

SQL Error: ORA-05715: operation not allowed on the blockchain or immutable table
05715. 0000 - "operation not allowed on the blockchain or immutable table"
*Cause: The table was insert-only table and, therefore, could not be
updated or deleted.

Brisanje redaka iz blockchain tablica

```
DECLARE
    l_num_rows NUMBER;
BEGIN
   DBMS_BLOCKCHAIN_TABLE.DELETE_EXPIRED_ROWS
      (schema_name            => 'HROUG2023'
      ,table_name             => 'TRANSACTION_LEDGER'
      ,before_timestamp       => TO_DATE('31.03.2023','DD.MM.YYYY')
      ,number_of_rows_deleted => l_num_rows);

   DBMS_OUTPUT.PUT_LINE('Number of rows deleted = ' || l_num_rows);
END;
```

PL/SQL procedure successfully completed.
Number of rows deleted = 0

Brisanje blockchain tablica

- Može biti obrisana ako nema redaka ili svim recima istekao rok zadržavanja
- Mora biti u korisničkoj shemi ili korisnik mora imati DROP ANY TABLE privilegiju
- Koristiti Purge opciju

```
drop table transaction_ledger_partitioned purge;
```

Blockchain tablica može biti obrisana ako nema redaka ili ako je svim recima istekao rok zadržavanja
Mora biti u korisničkoj shemi ili korisnik mora imati DROP ANY TABLE privilegiju
Preporučeno je koristiti Purge opciju prilikom brisanja tablica

Dodavanje certifikata

- X.509 digitalni certifikat
- Dodati ceritifikat u bazu kao BLOB
- ID pohranjenog certifikata koristi za potpisivanje i provjeru redaka
- Više različitih certifikata za potpisivanje
- Redak može imati samo jedan potpis

Nabaviti X.509 digitalni certifikat od certifikacijske organizacije
Dodati ceritifikat u bazu kao BLOB
ID pohranjenog certifikata se koristi za potpisivanje i provjeru potpisanih redaka
Više različitih certifikata se može koristiti za potpisivanje retka, ali svaki redak može imati samo jedan potpis

Dodavanje certifikata

- Kreiranje ključa za potpisivanje:

```
openssl genrsa -out bc_signing_key.pem 2048
```

- Kreiranje certifikata:

```
openssl req -new -x509 -outform pem -sha512 -days 3650 \
-nodes \
-out bc_signing_certificate.pem \
-key bc_signing_key.pem \
-subj
"/C=HR/ST=Zagreb/L=Somwhere/O=HROUG2023/OU=IN2/CN=Tomislav
/emailAddress=tomislavku@in2.hr"
```

Dodavanje certifikata

```
DECLARE
    file        BFILE;
    buffer      BLOB;
    amount      NUMBER := 32767;
    cert_id  RAW(16);
BEGIN
    file := BFILENAME('BC_CERT_DIR', 'bc_signing_certificate.pem');
    DBMS_LOB.FILEOPEN(file);
    DBMS_LOB.READ(file, amount, 1, buffer);
    DBMS_LOB.FILECLOSE(file);
    DBMS_USER_CERTS.ADD_CERTIFICATE(buffer, cert_id);
    DBMS_OUTPUT.PUT_LINE('Certificate ID = ' || cert_id);
END;
```

```
Certificate GUID = 02A55DA59D470E5EE0630100007FF6E2
```

Dodavanje certifikata

- Podaci o certifikatima u data dictionary viewovima:
  - DBA_CERTIFICATES
  - CDB_CERTIFICATES
  - USER_CERTIFICATES

CERTIFICATE_ID	USER_NAME	DISTINGUISHED_NAME	CERTIFICATE
02A55DA59D470E5EE0630100007FF6E2	HROUG2023	EMAIL=tomislavku@in2.hr,	(BLOB)

Podaci o postojećim certifikatima se mogu dohvatiti iz data dictionary viewova

Brisanje certifikata

```
declare
    certificate_guid RAW(16):=02A73D7872B912B0E0630100007F3E27';
begin
    DBMS_USER_CERTS.DROP_CERTIFICATE(certificate_guid);
end;
```

Dodavanje potpisa retku blockchain tablice

- Potpisivanje retka opcionalno
- Dodatno osiguranje protiv petljanja po podacima
- Baza provjerava:
  - da trenutni korisnik posjeduje redak koji se ažurira
  - da se korišteni hash poklapa sa hash vrijednošću retka
- Digitalni certifikat potpisuje redak blockchain tablice
- Podržani algoritmi za potpisivanje:
  - SIGN_ALGO_RSA_SHA2_256
  - SIGN_ALGO_RSA_SHA2_384
  - SIGN_ALGO_RSA_SHA2_512

Potpisivanje retka korisničkim potpisom je opcionalno
Pruža dodatno osiguranje protiv petljanja po podacima
Baza provjerava:
    da trenutni korisnik posjeduje redak koji se ažurira
    da se korišteni hash poklapa sa hash vrijednošću retka
Digitalni certifikat se koristi prilikom dodavanja potpisa retku blockchain tablice
Podržani algoritmi za potpisivanje:
    SIGN_ALGO_RSA_SHA2_256
    SIGN_ALGO_RSA_SHA2_384
    SIGN_ALGO_RSA_SHA2_512

Dodavanje potpisa retku blockchain tablice

- Uvjet za dodavanje potpisa retku blockchain tablice:
  - redak ne smije imati potpis
  - INSERT pravo na blockchain tablicu

Dodavanje potpisa retku blockchain tablice

```
select transaction_id
, ORABCTAB_INST_ID$
, ORABCTAB_CHAIN_ID$
, ORABCTAB_SEQ_NUM$
, ORABCTAB_SIGNATURE$
, ORABCTAB_SIGNATURE_ALG$
, ORABCTAB_SIGNATURE_CERT$
, ORABCTAB_SPARE$
from transaction_ledger;
```

TRANSACTION_ID	ORABCTAB_INST_	ORABCTAB_CHAI	ORABCTAB_SEQ_	ORABCTAB_SIGNATURE$	ORABCTAB_SIGNATURE_ALG$	ORABCTAB_SIGNATURE_CERT$	ORABCTAB_SPARE$
1	1	31	1	(null)	(null)	(null)	(null)
3	1	31	2	(null)	(null)	(null)	(null)
2	1	25	1	(null)	(null)	(null)	(null)
4	1	25	2	(null)	(null)	(null)	(null)

Dodavanje potpisa retku blockchain tablice

```
declare
    l_row_data blob;
    l_buffer raw(4000);
    l_inst_id binary_integer;
    l_chain_id binary_integer;
    l_seq_num binary_integer;
    l_row_len binary_integer;
    l_file utl_file.file_type;
begin
select orabctab_inst_id$,orabctab_chain_id$,orabctab_seq_num$
into l_inst_id,l_chain_id,l_seq_num
from transaction_ledger where transaction_id = 2;
dbms_blockchain_table.get_bytes_for_row_signature(schema_name => 'HROUG2023'
                                                 ,table_name  => 'TRANSACTION_LEDGER'
                                                 ,instance_id => l_inst_id
                                                 ,chain_id    => l_chain_id
                                                 ,sequence_id => l_seq_num
                                                 ,data_format => 1
                                                 ,row_data    => l_row_data);
l_row_len := dbms_lob.getlength(l_row_data);
dbms_lob.read(l_row_data, l_row_len, 1, l_buffer);
l_file := utl_file.fopen('BC_CERT_DIR','transaction2.dat','wb', 32767);
utl_file.put_raw(l_file, l_buffer, true);
utl_file.fclose(l_file);
end;
```

Dodavanje potpisa retku blockchain tablice

Potpisivanje sažetka retka:

```
openssl dgst -sha512 \
        -sign bc_signing_key.pem \
        -out transaction2.sha512 \
        transaction2.dat
```

Dodavanje potpisa retku blockchain tablice

```
DECLARE
l_inst_id binary_integer;
l_chain_id binary_integer;
l_sequence_no binary_integer;
l_file BFILE;
l_source_off integer := 1;
l_destination_off integer := 1;
l_signature blob;
l_cert_guid RAW (16) := HEXTORAW('02A55DA59D470E5EE0630100007FF6E2');
BEGIN
select orabctab_inst_id$,orabctab_chain_id$,orabctab_seq_num$
into l_inst_id,l_chain_id,l_sequence_no
from transaction_ledger where transaction_id = 2;
l_file := bfilename('BC_CERT_DIR', 'transaction2.sha512');
dbms_lob.createtemporary(l_signature, false);
dbms_lob.fileopen(l_file);
dbms_lob.loadblobfromfile(l_signature,l_file,dbms_lob.getlength(l_file),l_
estination_off,l_source_off);
dbms_lob.fileclose(l_file);
```

Dodavanje potpisa retku blockchain tablice

```
dbms_blockchain_table.sign_row(schema_name => 'HROUG2023'
,table_name       => 'TRANSACTION_LEDGER'
,instance_id      => l_inst_id
,chain_id         => l_chain_id
,sequence_id      => l_sequence_no
,hash             => NULL
,signature        => l_signature
,certificate_guid => l_cert_guid
,signature_algo   => DBMS_BLOCKCHAIN_TABLE.SIGN_ALGO_RSA_SHA2_512);
END;
```

Dodavanje potpisa retku blockchain tablice

TRANSACTION_ID	ORABCTAB_INST_	ORABCTAB_CHAI	ORABCTAB_SEQ_	ORABCTAB_SIGNATURE$	ORABCTAB_SIGNATURE_ALG$	ORABCTAB_SIGNATURE_CERT$	ORABCTAB_SPARE$
1	1	31	1	(null)	(null)	(null)	(null)
3	1	31	2	(null)	(null)	(null)	(null)
2	1	25	1	0EA24D566588AE2D206956EC	3	028EEC4E44D00A8CE0630100007FE0AF	(null)
4	1	25	2	(null)	(null)	(null)	(null)

Generiranje potpisanog sažetka tablice

- Potpisani sažetak - metapodaci i podaci o zadnjem retku svakog lanca u tablici u određenom trenutku
- Potpis baziran na sadržaju potpisanog sažetka
- Potpis koristi privatni ključ i certifikat
- Potpis i potpisani sažetak pohranjeni u repozitoriju

Potpisani sažetak se sastoji od metapodataka i podataka o zadnjem retku svakog lanca u tablici u određenom trenutku
Potpis je baziran na sadržaju potpisanog sažetka
Potpis koristi privatni ključ i certifikat vlasnika blockchain tablice
Potpis i potpisani sažetak se generiraju u različitim trenutcima i pohranjuju u repozitoriju

Generiranje potpisanog sažetka tablice

- Preduvjeti:
  - certifikat dodan u bazu
  - PKI privatni ključi i certifikat vlasnika blockchain tablice moraju biti pohranjeni u walletu:
    - Za PDB: WALLET_ROOT/pdb_guid/bctable/
    - Za non-CDB: WALLET_ROOT/bctable/

Preduvjeti:

certifikat vlasnika blockchain tablice moraju biti dodani u bazu

PKI privatni ključi i certifikat vlasnika blockchain tablice moraju biti pohranjeni u walletu:

Za PDB: WALLET_ROOT/pdb_guid/bctable/

Za non-CDB: WALLET_ROOT/bctable/

Generiranje potpisanog sažetka tablice

Postaviti WALLET_ROOT (as CDB SYSDBA):

```
ALTER SYSTEM SET WALLET_ROOT =
'/opt/oracle/product/23c/dbhomeFree/admin/FREE' SCOPE=SPFILE;
```

Kreiranje strukture mapa unutar WALLET_ROOTa:

```
SELECT pdb_name, guid FROM dba_pdbs;
```

PDB GUID: F87259FB7D3C3519E0530100007F5D4C

```
cd /opt/oracle/product/23c/dbhomeFree/admin/FREE/
mkdir F87259FB7D3C3519E0530100007F5D4C/bctable
```

Generiranje potpisanog sažetka tablice

Kreiranje wallet-a:

```
orapki wallet create -wallet
/opt/oracle/product/23c/dbhomeFree/admin/FREE/F87259FB7D3C3519E0530100007
F5D4C/bctable/ -auto_login_only
```

Pakiranje potpisanog certifikata i ključa:

```
openssl pkcs12 -export -in bc_signing_certificate.pem \
-inkey bc_signing_key.pem \
-out private_key_bct_owner.p12 -passout pass:"oracle"
```

Generiranje potpisanog sažetka tablice

Dodavanje ključeva wallet-u:

```
orapki wallet import_pkcs12 \
-wallet
/opt/oracle/product/23c/dbhomeFree/admin/FREE/F87259FB7D3C3519E053010000
7F5D4C/bctable/ \
-auto_login_only -pkcs12file
/home/oracle/my_wallet/private_key_bct_owner.p12 \
-pkcs12pwd "oracle"
```

Generiranje potpisanog sažetka tablice

```
DECLARE
l_signed_bytes BLOB;
l_signed_row_array SYS.ORABCTAB_ROW_ARRAY_T;
l_certificate_guid RAW(2000) := '02A55DA59D470E5EE0630100007FF6E2';
l_signature RAW(2000);
BEGIN
dbms_lob.createtemporary(l_signed_bytes, false);
l_signature := DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST
(schema_name => 'HROUG2023'
,table_name => 'TRANSACTION_LEDGER'
,signed_bytes => l_signed_bytes
,signed_rows_indexes => l_signed_row_array
,schema_certificate_guid => l_certificate_guid
,signature_algo => dbms_blockchain_table.SIGN_ALGO_RSA_SHA2_512);
insert into bc_signed_digests (creation_date,signed_digest,signed_bytes)
values(sysdate,to_blob(l_signature),l_signed_bytes);

DBMS_OUTPUT.PUT_LINE('Certificate GUID = ' || l_certificate_guid);
DBMS_OUTPUT.PUT_LINE('Signature length = ' || UTL_RAW.LENGTH(l_signature));
DBMS_OUTPUT.PUT_LINE('Number of chains = ' || l_signed_row_array.count);
DBMS_OUTPUT.PUT_LINE('Signature content buffer length = ' ||
DBMS_LOB.GETLENGTH(l_signed_bytes));
END;
```

Provjera integriteta blockchain tablica

1. DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS provjerava veze u lancima
2. DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST generira potpis i potpisani sažetak blockchaina u trenutku T1
3. DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST generira potpis i potpisani sažetak blockchaina u trenutku T2
4. DBMS_BLOCKCHAIN_TABLE.VERIFY_TABLE_BLOCKCHAIN - provjera integriteta redaka kreiranih između trenutaka T1 i T2

Korake 2 do 4 ponavljati u različitim vremenskim periodima

1. Provjeriti veze u svim lancima u blockchain tablici korištenjem DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS procedure.
   Ako redak sadrži potpis korisnika i on se provjerava
2. Generirati potpis i potpisani sažetak blockchain tablice korištenjem DBMS_BLOCKCHAIN_TABLE.GET_SIGN

ED_BLOCKCHAIN_DIGEST funkcije u trenutku T1

Generirani detalji i datum generiranja bi se trebali pohraniti u repozitoriju koji bi trebao biti izvan baze koja sadrži blockchain tablicu (npr. drugu bazu)

2. Generirati potpis i potpisani sažetak blockchain tablice korištenjem DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST funkcije u trenutku T2. Generirani detalji i datum generiranja bi trebali biti pohranjeni u repozitoriju.
3. Provjera integriteta redaka koji su kreirani između trenutaka T1 i T2 korištenjem DBMS_BLOCKCHAIN_TABLE.VERIFY_TABLE_BLOCKCHAIN procedure. Ulazni parametri za tu proceduru su potpisani sažeci u trenutcima T1 i T2.

Koraci 2 do 4 bi se trebali ponavljati u različitim vremenskim periodima,da se provjeri integrirate redaka insertiranih između ta dva trenutka.

Provjera redaka

```
DECLARE
      l_rows_verified NUMBER;
BEGIN
   DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS
   (schema_name => 'HROUG2023'
   ,table_name => 'TRANSACTION_LEDGER'
   ,low_timestamp => NULL
   ,high_timestamp => NULL
   ,instance_id => 1
   ,chain_id => NULL
   ,number_of_rows_verified => l_rows_verified
   ,verify_signature => TRUE);
   dbms_output.put_line('Number of rows verified in instance id 1 = ' ||
   l_rows_verified);
END;
```

```
Number of rows verified in instance id 1 = 5
```

Provjera integriteta redaka

```
DECLARE
l_signature RAW(2000);
l_signed_row_array SYS.ORABCTAB_ROW_ARRAY_T;
l_signed_bytes1 BLOB;
l_certificate_guid RAW(2000) := '02A55DA59D470E5EE0630100007FF6E2';
l_signed_bytes2 BLOB;
l_rows_verified NUMBER;
BEGIN
SELECT signed_bytes INTO l_signed_bytes1 FROM bc_signed_digests WHERE trunc(creation_date)
= trunc(SYSDATE-1);

l_signature := DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST
(schema_name => 'HROUG2023'
,table_name => 'TRANSACTION_LEDGER'
,signed_bytes => l_signed_bytes2
,signed_rows_indexes => l_signed_row_array
,schema_certificate_guid => l_certificate_guid
,signature_algo => dbms_blockchain_table.SIGN_ALGO_RSA_SHA2_512);
```

Provjera integriteta redaka

```
DBMS_BLOCKCHAIN_TABLE.VERIFY_TABLE_BLOCKCHAIN
(signed_bytes_latest       => l_signed_bytes2
,signed_bytes_previous     => l_signed_bytes1
,number_of_rows_verified   => l_rows_verified);

dbms_output.put_line('Rows verified = ' || l_rows_verified);
END;

Rows verified = 5
```

If you want to learn more

- https://docs.oracle.com/en/database/oracle/oracle-database/23/admin/managing-tables.html#GUID-E7151628-AF04-48D4-9CB4-F72417AFC391
- https://docs.oracle.com/en/database/oracle/oracle-database/23/arpls/dbms_blockchain_table.html#GUID-8B000001-AE8B-42EA-8BF3-E590BCBA6657
- https://apexapps.oracle.com/pls/apex/r/dbpm/livelabs/view-workshop?wid=875&p180_gb_clicked=Y&session=105569203428298

If you want to learn more

- https://www.oracle.com/blockchain/#blockchain-platform-tab
- https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home
- https://www.hyperledger.org/
- https://101blockchains.com/enterprise-blockchain-framework/

hroug'23
godišnja konferencija