13. Taldea 2. Laborategiaren Txostena

EGILEAK: Martín Amézola eta Igor Uriarte

OHARRA: CopyPaste egiten bada Ubuntu Terminalean kontuz komillak desberdin agertzen direlako Terminalean. Aldatu behar dira komanduak ondo exekutatzeko.

DBCOIN

```
CREATE DATABASE DBcoin;

CREATE USER admDBCOIN IDENTIFIED BY ‘igor2020’;

GRANT RELOAD ON *.* TO admDBCOIN; # Hau da flush privileges egin ahal izateko.
```

1) “Transfer” izeneko taula sortu (gako arrotzak definitu barik).

```
# HASIERAN ADM EZ DU TAULA SORTZEKO BAIMENIK BERAZ EMATEN DIOGU:
ROOT >
GRANT CREATE ON *.* TO admDBCOIN;

admDBOCOIN >
CREATE TABLE TRANSFER(TID INTEGER NOT NULL AUTO_INCREMENT, TDATE DATE,
TSENDER INTEGER NOT NULL, TRECEIVER INTEGER NOT NULL, TCOIN INTEGER NOT
NULL, PRIMARY KEY(TID));
```

2) “Owner” eta “Coin” izeneko taulak sortu.

```
admDBCOIN >
CREATE TABLE OWNER(OID INTEGER NOT NULL AUTO_INCREMENT,  ONAME
VARCHAR(20) NOT NULL, OEMAIL VARCHAR(20), BALANCE INTEGER, PRIMARY KEY(OID));


ROOT >
GRANT REFERENCES ON DBcoin.OWNER TO admDBCOIN;


ADM --> CREATE TABLE COIN(CID INTEGER NOT NULL AUTO_INCREMENT, COWNER
INTEGER NOT NULL, CUNITS INTEGER NOT NULL, CVALUE INTEGER, PRIMARY KEY(CID),
FOREIGN KEY(COWNER) REFERENCES OWNER(OID));
```

3) “Transfer” izeneko taularen gako arrotzak sortu.

```
ROOT >
GRANT ALTER ON DBcoin.TRANSFER TO admDBCOIN;
GRANT SELECT(OID) ON DBcoin.OWNER TO admDBCOIN;
GRANT SELECT(CID) ON DBcoin.COIN TO admDBCOIN;
```

```
admDBCOIN >
ALTER TABLE TRANSFER ADD FOREIGN KEY (TSENDER) REFERENCES OWNER(OID);
ALTER TABLE TRANSFER ADD FOREIGN KEY (TRECEIVER) REFERENCES OWNER(OID);
ALTER TABLE TRANSFER ADD FOREIGN KEY (TCOIN) REFERENCES COIN(CID);
```

4) “Rippler” izena duen jabe bat sartu “Owner” izeneko taulan.

```
ROOT >
GRANT INSERT (ONAME) ON DBcoin.OWNER TO admDBCOIN;
```

```
admDBCOIN >
INSERT INTO OWNER(ONAME) VALUES (“RIPPLER”);
```

5) “Ethereum” izena duen beste bezero bat sartu “Owner” izeneko taulan.

```
admDBCOIN >
INSERT INTO OWNER(ONAME) VALUES (“ETHEREUM”);
```

6) "Rippler" jabeko txanpon berri bat sartu “Coin” izeneko taulan.

```
ROOT >
GRANT INSERT ON DBcoin.COIN TO admDBCOIN;
```

```
admDBCOIN >
INSERT INTO COIN(COWNER, CUNITS) VALUES (1, 1);  # 1 DA RIPPLER JABEAREN ID.
```

7) “Ethereum” jabeko beste txanpon berri bat sartu “Coin” izeneko taulan.

```
admDBCOIN >
INSERT INTO COIN(COWNER, CUNITS) VALUES(2, 1);  # 2 DA ETHEREUM JABEAREN ID.
```

8) “Transfer” taulan 2021eko otsailaren 1ean egindako 1. txanponaren (6. urratsean sortutakoa)
transferentzia berri bat sartu non "Ethereum" igorlea eta "Ripple" hartzailea diren.

```
ROOT >
GRANT SELECT FROM DBCOIN.COIN TO admDBCOIN;
GRANT INSERT(TSENDER, TRECEIVER, TCOIN, TDATE) ON DBcoin.TRANSFER TO
admDBCOIN;
```

```
admDBCOIN >
INSERT INTO TRANSFER(TSENDER, TRECEIVER, TCOIN, TDATE) VALUES(2, 1, 1,
‘2021-02-01’);
```

9) “Transfer” taulan 2021eko otsailaren 2an egindako 2. txanponaren (7. urratsean sortutakoa) transferentzia berri bat sartu non "Ripple" igorlea eta "Ethereum" hartzailea diren.

```
admDBCOIN >
INSERT INTO TRANSFER(TSENDER, TRECEIVER, TCOIN, TDATE) VALUES(1, 2, 2,
‘2021-02-02’);
```

10) Aurreko urratsean gordetako transferentziaren data aldatu 2021eko otsailaren 3 baliora.

```
ROOT >
GRANT UPDATE(TDATE) ON DBcoin.TRANSFER TO admDBCOIN;
GRANT SELECT(TDATE) ON DBcoin.TRANSFER TO admDBCOIN;
```

```
admDBCOIN >
UPDATE TRANSFER SET TDATE = ‘2021-02-03’ WHERE TDATE = ‘2021-02-02’;
```

11) “MIser1” eta “MIser2” izeneko MySQL erabiltzaileak sortu.

```
ROOT >
GRANT CREATE USER ON *.* TO admDBCOIN;
```

```
admDBCOIN >
CREATE USER MIser1  IDENTIFIED BY ‘miser12020’;
CREATE USER MIser2  IDENTIFIED BY ‘miser22020’;
```

12) Laborategian emandako orrian deskribatzen diren baimenak esleitu.

```
# GURE TALDEA 13. TALDEA DA, ETA HAUEK DIRA EMANDAKO BAIMENAK:
```

MISER1 ERABILTZAILEARI EMANDAKO BAIMENAK

OWNER TAULAN:

```
ROOT >
GRANT SELECT(OID, OEMAIL) ON DBcoin.OWNER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT SELECT(OID, OEMAIL) ON DBcoin.OWNER TO MIser1;


ROOT>
GRANT INSERT(ONAME, OEMAIL, OBALANCE) ON DBcoin.OWNER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT INSERT(ONAME, OEMAIL, OBALANCE) ON DBcoin.OWNER TO MIser1;



ROOT >
GRANT UPDATE(OID, ONAME, OBALANCE) ON DBcoin.OWNER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT UPDATE(OID, ONAME, OBALANCE) ON DBcoin.OWNER TO MIser1;
```

TRANSFER TAULAN:

```
ROOT >
GRANT SELECT(TID, TRECEIVER, TCOIN, TDATE) ON DBcoin.TRANSFER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT SELECT(TID, TRECEIVER, TCOIN, TDATE) ON DBcoin.TRANSFER TO MIser1;

ROOT >
GRANT INSERT ON DBcoin.TRANSFER TO admDBCOIN WITH GRANT OPTION;


admDBCOIN >
GRANT INSERT ON DBcoin.TRANSFER TO MIser1;

ROOT >
GRANT UPDATE ON DBcoin.TRANSFER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
```

```
GRANT UPDATE ON DBcoin.TRANSFER TO MIser1;
```

COIN TAULAN:

```
ROOT >
GRANT SELECT(COWNER, CUNITS, CVALUE) ON DBcoin.COIN TO admDBCOIN WITH
GRANT OPTION;


admDBCOIN >
GRANT SELECT(COWNER, CUNITS, CVALUE) ON DBcoin.COIN TO MIser1;

ROOT >
GRANT INSERT(CID, COWNER, CVALUE) ON DBcoin.COIN TO admDBCOIN WITH GRANT
OPTION;


admDBCOIN >
GRANT INSERT(CID, COWNER, CVALUE) ON DBcoin.COIN TO MIser1;

ROOT >
GRANT UPDATE ON DBcoin.COIN TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT UPDATE ON DBcoin.COIN TO MIser1;
```

Balantze negatiboa duten jabeen identifikatzailea eta izena lortu .

```
ROOT >
GRANT CREATE VIEW ON *.* TO admDBCOIN WITH GRANT OPTION;
GRANT SELECT(ONAME, OID, OBALANCE) ON DBcoin.OWNER TO admDBCOIN WITH
GRANT OPTION;


admDBCOIN >
GRANT CREATE VIEW ON *.* TO MIser1;
GRANT SELECT(ONAME, OID, OBALANCE) ON DBcoin.OWNER TO MIser1;

MIser1 >

CREATE VIEW DBcoin.NEGATIBOAKLORTU(OWNERNAME, ID) AS SELECT OID, ONAME
FROM OWNER WHERE OBALANCE < 0;
```

Jabeen identifikatzailea eta korreoa kontsultatu.

```
ROOT >
```

```
GRANT SELECT(OID, OEMAIL) ON DBcoin.OWNER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT SELECT(OID, OEMAIL) ON DBcoin.OWNER TO MIser1;

MIser1 >
CREATE VIEW DBcoin.JABEAKKONTSULTATU AS SELECT OID, OEMAIL FROM OWNER;
```

Jabeen balantzea eguneratu bere balioa %10 bat gehienez aldatuz eta identifikatzailea adieraziz.

```
Delimiter//
create procedure  DBcoin.eguneratu( IN pertzentaia double,IN  id int)
begin update DBcoin.Owner set oBalance=oBalance*pertzentaia where oid=id and
pertzentaia <1.10 and pertzentaia >0.9
end//
delimiter;
```

Jabe baten balantzea eguneratu identifikatzailea adieraziz

```
Delimiter//
create procedure  DBcoin.eguneratu2(IN balance int,IN  id int)
begin update DBcoin.Owner set oBalance=balance where oid=id end//
delimiter;
```

MISER2 ERABILTZAILEARI EMANDAKO BAIMENAK

OWNER TAULAN:

```
ROOT  >
GRANT SELECT(OID, OEMAIL) ON DBcoin.OWNER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT SELECT(OID, OEMAIL) ON DBcoin.OWNER TO MIser2;

ROOT >
GRANT INSERT(OID, ONAME, OEMAIL) ON DBcoin.OWNER TO admDBCOIN WITH GRANT
OPTION;

admDBCOIN >
GRANT INSERT(OID, ONAME, OEMAIL) ON DBcoin.OWNER TO MIser2;

ROOT >
GRANT UPDATE(ONAME, OEMAIL, OBALANCE) ON DBcoin.OWNER TO admDBCOIN WITH
GRANT OPTION;
```

```
admDBCOIN >
GRANT UPDATE(ONAME, OEMAIL, OBALANCE) ON DBcoin.OWNER TO MIser2;
```

TRANSFER TAULAN:

```
ROOT >
GRANT SELECT ON DBcoin.TRANSFER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT SELECT ON DBcoin.TRANSFER TO MIser2;

ROOT >
GRANT INSERT(TRECEIVER, TCOIN, TDATE) ON DBcoin.TRANSFER TO admDBCOIN WITH
GRANT OPTION;

admDBCOIN >
GRANT INSERT(TRECEIVER, TCOIN, TDATE) ON DBcoin.TRANSFER TO MIser2;

ROOT >
GRANT UPDATE(TID, TSENDER, TRECEIVER) ON DBcoin.TRANSFER TO admDBCOIN WITH
GRANT OPTION;

admDBCOIN >
GRANT UPDATE(TID, TSENDER, TRECEIVER) ON DBcoin.TRANSFER TO MIser2;
```

COIN TAULAN:

```
ROOT >
GRANT SELECT(CID, CUNITS, CVALUE) ON DBcoin.COIN TO admDBCOIN WITH GRANT
OPTION;

admDBCOIN >
GRANT SELECT(CID, CUNITS, CVALUE) ON DBcoin.COIN TO MIser2;

ROOT >
GRANT INSERT(COWNER, CUNITS) ON DBcoin.COIN TO admDBCOIN WITH GRANT
OPTION;

admDBCOIN >
GRANT INSERT(COWNER, CUNITS) ON DBcoin.COIN TO MIser2;

ROOT >
GRANT UPDATE(CID, COWNER, CVALUE) ON DBcoin.COIN TO admDBCOIN WITH GRANT
OPTION;

admDBCOIN >
GRANT UPDATE(CID, COWNER, CVALUE) ON DBcoin.COIN TO MIser2;
```

Tansferentzien identifikatzailea eta data kontsultatu.

```
ROOT >
GRANT SELECT(TID, TDATE) ON DBcoin.TRANSFER TO admDBCOIN WITH GRANT OPTION;
GRANT CREATE VIEW ON *.* TO admDBCOIN WITH GRANT OPTION;


admDBCOIN >
GRANT SELECT(TID, TDATE) ON DBcoin.TRANSFER TO MIser2;
GRANT CREATE VIEW ON *.* TO MIser2;


MIser2 >
CREATE VIEW DBcoin.TRANSFERENTZIAKKONTSULTATU AS SELECT TID, TDATE FROM
TRANSFER;
```

Igorleek balantze negatiboa duten transferentzien identifikatzailea eta data kontsultatu.

```
ROOT >
GRANT SELECT(OBALANCE) ON DBcoin.OWNER TO admDBCOIN WITH GRANT OPTION;
GRANT SELECT(TSENDER ) ON DBcoin.TRANSFER TO admDBCOIN WITH GRANT OPTION;

admDBCOIN >
GRANT SELECT(OBALANCE) ON DBcoin.OWNER TO MIser2;
GRANT SELECT(TSENDER ) ON DBcoin.TRANSFER TO MIser2;


MIser2 >
CREATE VIEW DBcoin.BALANTZENEGATIBOIGORLEAK AS SELECT TID, TDATE FROM
TRANSFER WHERE
```

DBET

```
# HASI AURRETIK

ROOT >
CREATE USER admDBET IDENTIFIED BY ‘martin2020’;

GRANT CREATE ON *.* TO admDBET;

GRANT RELOAD ON *.* TO admDBCOIN; # Hau da flush privileges egin ahal izateko.

admDBET
CREATE DATABASE DBet;

USE DBet;
```

1.“Bet” izeneko taula sortu (gako arrotzak definitu barik).

```
admDBET >
CREATE TABLE BET(BID INTEGER NOT NULL AUTO_INCREMENT, BGAMBLER INTEGER
NOT NULL, BEVENT INTEGER NOT NULL, BAMOUNT INTEGER, BRESULT VARCHAR(20),
PRIMARY KEY(BID));
```

2.“Gambler” eta “Event” izeneko taulak sortu.

```
admDBET >
CREATE TABLE GAMBLER(GID INTEGER NOT NULL AUTO_INCREMENT, GNAME
VARCHAR(20) NOT NULL, GEMAIL VARCHAR(20), GCREDIT INTEGER, PRIMARY KEY(GID));

CREATE TABLE EVENT(EID INTEGER NOT NULL AUTO_INCREMENT, ENAME VARCHAR(20)
NOT NULL, EDATE DATE, ERESULT VARCHAR(20), PRIMARY KEY(EID));
```

3.“Bet” izeneko taularen gako arrotzak sortu.

```
ROOT >
GRANT ALTER ON DBet.BET TO admDBET;
GRANT REFERENCES ON DBet.GAMBLER TO admDBET;
GRANT REFERENCES ON DBet.EVENT TO admDBET;


admDBET >
ALTER TABLE BET ADD FOREIGN KEY(BGAMBLER) REFERENCES GAMBLER(GID);
ALTER TABLE BET ADD FOREIGN KEY(BEVENT) REFERENCES EVENT(EID);
```

4.“MacBeth” izena duen apustulari bat sartu “Gambler” izeneko taulan.

```
ROOT >
GRANT INSERT(GNAME) ON DBet.GAMBLER TO admDBET;

admDBET >
INSERT INTO GAMBLER(GNAME) VALUES('MacBeth');
```

5.“Betty” izena duen beste apustulari bat sartu “Gambler” izeneko taulan.

```
admDBET >
INSERT INTO GAMBLER(GNAME) VALUES(‘Betty’);
```

6.2021eko otsailaren 1ean ospatuko den eta "SnailRace" izena duen ekitaldi berri bat sartu “Event” izeneko taulan.

```
ROOT >
GRANT INSERT(ENAME) ON DBet.EVENT TO admDBET;

admDBET >
INSERT INTO EVENT(ENAME) VALUES(‘SnailRace’);
```

7.2021eko otsailaren 2an ospatuko den eta "CaterpillarFight" izena duen beste ekitaldi berri bat sartu “Event” izeneko taulan.

```
ROOT >
GRANT INSERT(ENAME, EDATE) ON DBet.EVENT TO admDBET;

admDBET >
INSERT INTO EVENT(ENAME, EDATE) VALUES(‘CaterpillarFight’, ‘2021-02-02’);
```

8."SnailRace" ekitaldian “MacBeth” apustulariak "Turbo wins" alde egindako apustu berri bat sartu "Bet" izeneko taulan.

```
ROOT >
GRANT INSERT(BGAMBLER, BEVENT, BRESULT) ON DBet.BET TO admDBET;

admDBET >
INSERT INTO BET(BGAMBLER, BEVENT, BRESULT) VALUES(1, 1, ‘Turbo Wins’);
```

9.“Caterpillar Fight" ekitaldian “Betty” apustulariak "Butterfly loses" alde egindako apustu berri bat sartu "Bet" izeneko taulan.

```
admDBET >
INSERT INTO BET(BGAMBLER, BEVENT, BRESULT) VALUES(2, 2, ‘Butterfly Loses’);
```

10."SnailRace" ekitaldiaren data 2021eko otsailaren 3 baliora eguneratu.

```
ROOT >
GRANT UPDATE(EDATE) ON DBet.EVENT TO admDBET;
GRANT SELECT(EDATE) ON DBet.EVENT TO admDBET;
```

```
admDBET >
UPDATE SET EDATE = ‘2021-02-03’ WHERE EDATE = ‘2021-02-02’;
```

11.“LUser1” eta “LUser2” izeneko MySQL erabiltzaileak sortu.

```
ROOT >
GRANT CREATE USER ON *.* TO admDBET;
```

```
admDBET >
CREATE USER LUser1 IDENTIFIED BY ‘luser12020’;
CREATE USER LUser2 IDENTIFIED BY ‘luser22020’;
```

12.Laborategian emandako orrian deskribatzen diren baimenak esleitu.

```
# GURE TALDEA 13. TALDEA DA, ETA HAUEK DIRA EMANDAKO BAIMENAK:
```

LUSER1 ERABILTZAILEARI EMANDAKO BAIMENAK

GAMBLER TAULAN:

```
ROOT >
GRANT SELECT ON DBet.GAMBLER TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT SELECT ON DBet.GAMBLER TO LUser1;
```

```
ROOT >
```

```
GRANT INSERT(GID, GNAME, GEMAIL) ON DBet.GAMBLER TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT INSERT(GID, GNAME, GEMAIL) ON DBet.GAMBLER TO LUser1;
```

```
ROOT>
GRANT UPDATE(GNAME, GEMAIL, GCREDIT) ON DBet.GAMBLER TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT UPDATE(GNAME, GEMAIL, GCREDIT) ON DBet.GAMBLER TO LUser1;
```

BET TAULAN:

```
ROOT >
GRANT SELECT(BID, BEVENT, BAMOUNT, BRESULT) ON DBet.BET TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT SELECT(BID, BEVENT, BAMOUNT, BRESULT) ON DBet.BET TO LUser1;
```

```
ROOT >
GRANT INSERT(BID, BEVENT, BAMOUNT, BRESULT) ON DBet.BET TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT INSERT(BID, BEVENT, BAMOUNT, BRESULT) ON DBet.BET TO LUser1;
```

```
ROOT >
GRANT UPDATE ON DBet.BET TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT UPDATE ON DBet.BET TO LUser1;
```

EVENT TAULAN:

```
ROOT >
GRANT SELECT(EID, ENAME, ERESULT) ON DBet.EVENT TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT SELECT(EID, ENAME, ERESULT) ON DBet.EVENT TO LUser1;
```

```
ROOT >
GRANT INSERT ON DBet.EVENT TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT INSERT ON DBet.EVENT TO LUser1;
```

```
ROOT>
GRANT UPDATE(EID, ERESULT) ON DBet.EVENT TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT UPDATE(EID, ERESULT) ON DBet.EVENT TO LUser1;
```

LUSER2 ERABILTZAILEARI EMANDAKO BAIMENAK

GAMBLER TAULAN:

```
ROOT >
GRANT SELECT(GID, GNAME, GCREDIT) ON DBet.GAMBLER TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT SELECT(GID, GNAME, GCREDIT) ON DBet.GAMBLER TO LUser2;
```

```
ROOT >
GRANT INSERT ON DBet.GAMBLER TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT INSERT ON DBet.GAMBLER TO LUser2;
```

```
ROOT>
GRANT UPDATE(GID, GNAME, GCREDIT) ON DBet.GAMBLER TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT UPDATE(GID, GNAME, GCREDIT) ON DBet.GAMBLER TO LUser2;
```

BET TAULAN:

```
ROOT >
GRANT SELECT ON DBet.BET TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT SELECT ON DBet.BET TO LUser2;
```

```
ROOT >
GRANT INSERT(BID, BGAMBLER, BEVENT) ON DBet.BET TO admDBET WITH GRANT
OPTION;
```

```
admDBET >
GRANT INSERT(BID, BGAMBLER, BEVENT) ON DBet.BET TO LUser2;
```

```
ROOT >
GRANT UPDATE(BID, BAMOUNT) ON DBet.BET TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT UPDATE(BID, BAMOUNT) ON DBet.BET TO LUser2;
```

EVENT TAULAN:

```
ROOT >
GRANT SELECT(EID, ENAME, EDATE) ON DBet.EVENT TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT SELECT(EID, ENAME, EDATE) ON DBet.EVENT TO LUser2;
```

```
ROOT >
GRANT INSERT(EID, ENAME, EDATE) ON DBet.EVENT TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT INSERT(EID, ENAME, EDATE)  ON DBet.EVENT TO LUser2;
```

```
ROOT>
GRANT UPDATE ON DBet.EVENT TO admDBET WITH GRANT OPTION;
```

```
admDBET >
GRANT UPDATE ON DBet.EVENT TO LUser2;
```

2021 urtean apusturen bat egin duten apustularien identifikatzailea eta izena kontsultatu.

```
CREATE VIEW 2020Transferentziak AS SELECT tId,tSender,tReceiver FROM
Transfer WHERE Extract(YEAR from tDate)='2020';
```