نویسنده : محمد رضیئی

شماره دانشجویی : ۹۴۳۱۷۰۴

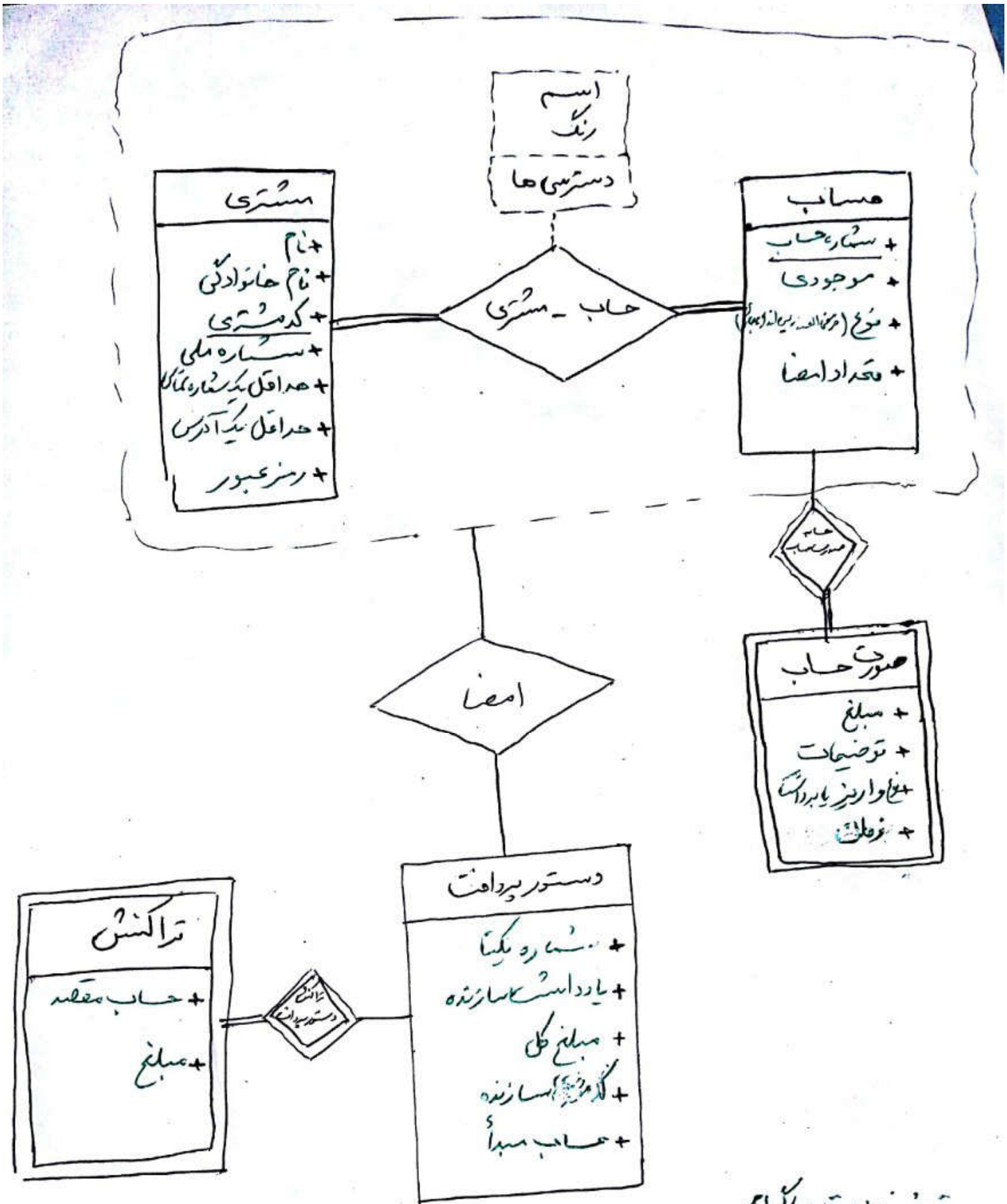
پروژه پایانی دیتابیس

بهار ۹۸

Github

https://github.com/MohammadRaziei/DB2019/tree/master/Workspace/Project

دیاگرام :



توضیحات دیاگرام

۱) کد مشتری که همچنان می‌دانیم که خود کد ملی کافی است اما به این مودل اضافه شده است تا اطلاعات مهمی مانند کد ملی به دلیل بی‌امنیتی پایگاه داده‌ها ذخیره نشود.

جداول نهایی :

```
-- Customer :
CREATE TABLE Customer (
CID                     int,
NationalCode      int Unique,
LastName          varchar(127),
FirstName         varchar(127),
Hash_Password     varchar(63),
CONSTRAINT PK_C PRIMARY KEY (CID)
);
CREATE TABLE  Customer_Address(
CID                     int,
Address                 varchar(255) NOT NULL,
CONSTRAINT PK_CAd PRIMARY KEY (CID),
CONSTRAINT FK_CAd FOREIGN KEY (CID) REFERENCES Customer(CID)
);
CREATE TABLE  Customer_Phone(
CID                     int,
Phone                   varchar(12) NOT NULL,
CONSTRAINT PK_CP PRIMARY KEY (CID),
CONSTRAINT FK_CP FOREIGN KEY (CID) REFERENCES Customer(CID)
);

-- Account :
CREATE TABLE  Account(
AID                     int,
Cash                    int DEFAULT(0),
AType                   varchar(15),
Nsign                   int,
CONSTRAINT PK_A_ID PRIMARY KEY (AID),
CONSTRAINT CHK_A_type CHECK(AType in ('currnet','charity','saving'))
);
CREATE TABLE  Account_soratHesab(
AID                     int,
STime                   timestamp,
Descriptions      varchar(257),
amount                  int,
Stype                   varchar(15) NOT NULL,

CONSTRAINT PK_AS PRIMARY KEY (AID,STime),
CONSTRAINT FK_AS_ID FOREIGN KEY (AID) REFERENCES Account(AID),
CONSTRAINT CHK_AS_type CHECK(Stype in('receive','send'))
);

CREATE TABLE DastorPardakht(
PNumber                 int,
CNote                   varchar(255),
CID                     int,
Origin_AID        int Not Null,
State                   numeric(1,0) DEFAULT(0),
CONSTRAINT PK_D PRIMARY KEY (PNumber),
CONSTRAINT FK_D_CID FOREIGN KEY (CID) REFERENCES Customer(CID),
CONSTRAINT FK_D_Origin_AID FOREIGN KEY (Origin_AID) REFERENCES Account(AID),
CONSTRAINT CHK_AS_type CHECK(State in (0,1))
);
-- Transactions
```

```sql
CREATE TABLE Transactions(
TID                     int NOT NULL AUTO_INCREMENT,
PNumber                 int,
Dest_AID                int,
amount                  int,
CONSTRAINT PK_T PRIMARY KEY (TID),
CONSTRAINT FK_T_PNumber FOREIGN KEY (PNumber) REFERENCES
DastorPardakht(PNumber),
CONSTRAINT FK_T_Dest_AID FOREIGN KEY (Dest_AID) REFERENCES Account(AID)
);
-- Customer_Account
CREATE TABLE Customer_Account(
AID                     int,
CID                     int,
Name                    varchar(127),
Color                   varchar(31),
CONSTRAINT PK_CA PRIMARY KEY (CID,AID),
CONSTRAINT FK_CA_CID FOREIGN KEY (CID) REFERENCES Customer(CID),
CONSTRAINT FK_CA_AID FOREIGN KEY (AID) REFERENCES Account(AID)
);
CREATE TABLE Customer_Account_Access(
AID                     int,
CID                     int,
Access                  varchar(7),
CONSTRAINT PK_CAA PRIMARY KEY (CID,AID,Access),
CONSTRAINT FK_CAA_CID FOREIGN KEY (CID) REFERENCES Customer(CID),
CONSTRAINT FK_CAA_AID FOREIGN KEY (AID) REFERENCES Account(AID),
CONSTRAINT CHK_CAA_Access CHECK(Access in ('pay','visit','sign'))
);

-- Customer_sign
CREATE TABLE Customer_sign(
CID                     int,
PNumber                 int,
CONSTRAINT PK_CS PRIMARY KEY (CID,PNumber),
CONSTRAINT FK_CS_CID FOREIGN KEY (CID) REFERENCES Customer(CID),
CONSTRAINT FK_CS_PNumber FOREIGN KEY (PNumber) REFERENCES
DastorPardakht(PNumber)

);
-- Customer_sign_deleted
CREATE TABLE Customer_sign_deleted(
CID                     int,
PNumber                 int,
CONSTRAINT PK_CSD PRIMARY KEY (CID,PNumber),
CONSTRAINT FK_CSD_CID FOREIGN KEY (CID) REFERENCES Customer(CID),
CONSTRAINT FK_CSD_PNumber FOREIGN KEY (PNumber) REFERENCES
DastorPardakht(PNumber)

);
```

عملیات ها :

```sql
-- Q1
insert into Customer values(  101, 0019773722, 'Mohammad', 'Raziei',
SHA1('password'));
```

```sql
insert into Customer values(  102, 0022621121, 'Ali', 'Raziei',
SHA1('PASS541054'));
insert into Customer values(  103, 00195785104, 'Simon', 'Jakson',
SHA1('PASS54'));


-- Q2
insert into Account(AID, cash, AType, NSign) values(11, 1200, 'currnet', 2);
insert into Account(AID, cash, AType, NSign) values(12, 3300, 'charity', 3);
insert into Account(AID, cash, AType, NSign) values(13, 1049, 'saving', 2);


-- Q3
-- (AID, CID, ACCESS)
insert into Customer_Account values(11, 101, 'MR','blue');
insert into Customer_Account_Access values(11, 101, 'sign');
insert into Customer_Account_Access values(11, 102, 'pay');
insert into Customer_Account values(12, 102, 'AR', 'blue');
insert into Customer_Account_Access values(12, 102, 'sign');
insert into Customer_Account_Access values(12, 103, 'visit');
insert into Customer_Account values(13, 101, 'SJ', 'red');
insert into Customer_Account_Access values(13, 101, 'visit');
insert into Customer_Account_Access values(13, 103, 'pay');

delete from Customer_Account_Access
where (AID, CID, Access) = (13, 103, 'pay');

update Customer_Account_Access
set Access = 'pay'
where (AID, CID) = (13, 101);


-- Q4
insert into DastorPardakht(PNumber, CNote, CID, Origin_AID)
values(6062, 'this is some notes : 6062', 101, 11);


-- Q5
update DastorPardakht
set Origin_AID = 12
where PNumber = 6062;

delete from DastorPardakht
where PNumber = 6062;


-- Q10
insert into DastorPardakht(PNumber, CNote, CID, Origin_AID)
values(6062, 'this is some notes 6062', 101, 11);

insert into Customer_sign values(103, 6062); -- impossible (CID=103 , AID=11)
insert into Customer_sign values(101, 6062); -- possible

delete from Customer_sign
where (CID, PNumber) = (101, 6062);
insert into Customer_sign_deleted values(101, 6062);
```

```
-- Q7
insert into Transactions(PNumber, Dest_AID, amount) values(6062, 12, 300);
insert into Transactions(PNumber, Dest_AID, amount) values(6062, 13, 450);

delete from Transactions
where (PNumber, Dest_AID) = (6062, 13);

update Transactions
set amount = 333
where (PNumber, Dest_AID) = (6062, 12);


-- Q8
insert into Customer_Account_Access values(11, 102, 'sign');
insert into Customer_Account_Access values(11, 103, 'sign');
update DastorPardakht set state = 1 where PNumber = 6062; -- impossible 1 < 2
insert into Customer_sign values(102, 6062);
update DastorPardakht set state = 1 where PNumber = 6062; -- possible 2 => 2



-- Q9
/*
This operation is fully automatic using triggers.
*/

insert into Account_soratHesab(STime, amount, AID, Descriptions, SType)
values (current_timestamp(), 11500, 11, 'exp1', 'receive');

insert into Account_soratHesab(STime, amount, AID, Descriptions, SType)
      values (current_timestamp(), 24600, 12, 'exp2', 'send');
```

فانکشن ها و ویو ها:

```
-- VIEWs :
CREATE VIEW customers_have_sign as
select distinct CID
from Customer_sign;


create view Amaade_Padakht_list  as
select AID, Nsign, count(*) as signs
from Customer_Account_Access natural join (
select Nsign, AID
from (DastorPardakht as P) inner join (Account as A)
on A.AID = P.Origin_AID
)
where Access = 'sign'
group by AID, Nsign
having Nsign <= signs;

CREATE VIEW Payment as
select PNumber, CID, Origin_AID, State, sum(amount) as total,CNote
```

```sql
From Transactions natural join DastorPardakht



-- PROCEDUREs
delimiter $$
CREATE PROCEDURE transactions_list(accountNumber int)
begin
SELECT *
FROM Transactions natural JOIN DastorPardakht
WHERE Dest_AID = accountNumber and state = 0;
end $$
DELIMITER ;

delimiter $$
CREATE PROCEDURE deleted_signs_list(customerNo int)
begin
select *
from Customer_sign_deleted
where CID = customerNo;
end $$
DELIMITER ;

delimiter $$
CREATE PROCEDURE sharing_accounts(customerNo1 int, customerNo2 int)
begin
select C1 as CNumber from (
select distinct CID as C1
from Customer_Account_Access
where CID = customerNo1

) inner join (
select distinct CID as C2
from Customer_Account_Access
where CID = customerNo2
)
where C1 = C2;
end $$
DELIMITER ;

delimiter $$
CREATE PROCEDURE payment_ability_by_User(UID int)
begin
select PNumber, CNote, CID, Origin_AID, total, state
from Customer_Account_Access inner join Payment
on AID = Origin_AID
where Access = 'sign' and CID = UID;
end $$
DELIMITER ;

delimiter $$
CREATE PROCEDURE payments_list_by_user(UID int)
begin
select distinct PNumber
from Payment natural join Transactions
where CID = UID
not in(
select paymentNumber
from Payment inner join Customer_Account_Access
on Origin_AID = AID
where Access = 'sign' and CID = UID);
end $$
DELIMITER ;

delimiter $$
CREATE PROCEDURE transactions_list_by_user(User_AID int, User_amount float)
begin
select *
from Transactions natural join Payment
where state = 1 and Dest_AID = User_AID and amount > User_amount;
end $$
DELIMITER ;


-- HashBytes ( 'MD5', Hash_Password )
```

کوئری:

```
-- Q1
select * from customers_can_sign;

-- Q2
select AID
from Account natural join(
select count(*) as signs, AID
from Custumer_Account_Access
where Access ='sign'
group by AID
)
where signs > Nsign;

-- Q3
select * from Amaade_Padakht_list;

-- Q4
call transactions_list(3);

-- Q5
call deleted_signs_list(2);

-- Q6
call sharing_accounts(1, 2);

-- Q7
call payment_ability_by_User(2);

-- Q8
call payments_list_by_user(1);

-- Q9
call transactions_list_by_user(3, 150);
```

پایان