2023 – 2024 Spring Term

CS306 – Database System Course

Term Project Phase 2

-

Berke Ayyıldızlı

31018

Q2. Choosing Two Tables

The tables (relations) that I selected are PaymentMethod and User. Since in our project, we had 4 tables that had a foreign key – primary key relation, my team member Kerem Tatari and I had to use the User table in common. The respected CREATE TABLE statements are as follows:

```
CREATE TABLE PaymentMethod (
pid INT AUTO_INCREMENT,
type VARCHAR(50) NOT NULL,
uid INT NOT NULL,
primary key (pid),
FOREIGN KEY (uid) REFERENCES User(uid)
);
```

```
CREATE TABLE User (
uid INT AUTO_INCREMENT,
name VARCHAR(50) NOT NULL,
email VARCHAR(50) NOT NULL UNIQUE,
password VARCHAR(50) NOT NULL,
primary key (uid)
);
```

Q3. Functional Dependencies and Checking BCNF

To evaluate if the tables are in BCNF, we need to first show the functional dependencies in each table.

For the PaymentMethod Table:

pid → type, uid

We can see here that, the primary key pid, uniquely determines the types of payment methods and user id's used with that payment method.

For the User Table:

uid → name, email, password

email → uid, name, password (because it is unique to each user, just like uid)

We can see here that, the primary key uid determines uniquely the names, emails and passwords of each user. Although email is not a primary key, because it is considered as "UNIQUE", it also can do that.

Now that we have evaluated all the FD's, we can check if they are in the BCNF or not.

We know that a relation is in BCNF iff each functional dependency (x→y) has a determinant x that is a superkey, which means it determines every other aspects of the relation.

For the PaymentMethod Table:

Since on the relation, pid → type, uid; pid determines all the other aspects, it acts as a superkey, thus maintaining the BCNF.

For the User Table:

Since on the relation, uid → name, email, password; uid determines all the other aspects, it acts as a superkey, thus maintaining the BCNF.

Also, although it is not a primary key, on the relation, email → uid, name, password, email acts as a alternate key, because of the fact that it is "UNIQUE" and it can determine other attributes, maintaining the BCNF rules.

From these statements, we can see that **both** tables are in BCNF, and there is no need for decomposition.

Q4. Inserting 10 Rows to Each Table

```
INSERT INTO PaymentMethod (uid, type) VALUES
(1, 'Credit Card'),
(2, 'PayPal'),
(3, 'Debit Card'),
(4, 'Credit Card'),
(5, 'PayPal'),
(6, 'Debit Card'),
(7, 'Credit Card'),
(8, 'PayPal'),
(9, 'Debit Card'),
(10, 'Credit Card');
```

```
INSERT INTO User (name, email, password) VALUES
('John Doe', 'johndoe@example.com', 'password123'),
('Jane Smith', 'janesmith@example.com', 'password456'),
('Alice Johnson', 'alicejohnson@example.com', 'password789'),
('Bob Brown', 'bobbrown@example.com', 'password101'),
('Charlie Davis', 'charliedavis@example.com', 'password202'),
('Diana Evans', 'dianaevans@example.com', 'password303'),
('Frank Green', 'frankgreen@example.com', 'password404'),
('Gina Harris', 'ginaharris@example.com', 'password505'),
('Henry Wilson', 'henrywilson@example.com', 'password606'),
('Ivy Young', 'ivyyoung@example.com', 'password707');
```

Q5. Displaying All the Rows

Query: SELECT * FROM PaymentMethod;

Result:

pid	type	uid
1	Credit Card	1
2	PayPal	2
3	Debit Card	3
4	Credit Card	4
5	PayPal	5
6	Debit Card	6
7	Credit Card	7
8	PayPal	8
9	Debit Card	9
10	Credit Card	10

Query: SELECT * FROM User

Result:

uid	name	email	password
1	John Doe	johndoe@example.com	password123
2	Jane Smith	janesmith@example.com	password456
3	Alice Johnson	alicejohnson@example.com	password789
4	Bob Brown	bobbrown@example.com	password101
5	Charlie Davis	charliedavis@example.com	password202
6	Diana Evans	dianaevans@example.com	password303
7	Frank Green	frankgreen@example.com	password404
8	Gina Harris	ginaharris@example.com	password505
9	Henry Wilson	henrywilson@example.com	password606
10	Ivy Young	ivyyoung@example.com	password707

Q6. Joining Tables in English and Relational Algebra

I selected this sentence so that it would require for a join between two tables:

“Retrieve all user names with their corresponding payment methods.”

The respected relational algebra equivelant is as follows:

$$\pi_{name,type}(User \bowtie_{User.uid=PaymentMethod.uid} PaymentMethod)$$

Q7. The SQL Version of The Query

Query:

```
SELECT User.name, PaymentMethod.type

FROM User

INNER JOIN PaymentMethod ON User.uid = PaymentMethod.uid;
```

Result:

name	type
John Doe	Credit Card
Jane Smith	PayPal
Alice Johnson	Debit Card
Bob Brown	Credit Card
Charlie Davis	PayPal
Diana Evans	Debit Card
Frank Green	Credit Card
Gina Harris	PayPal
Henry Wilson	Debit Card
Ivy Young	Credit Card

Q8. Group By Query in English and SQL

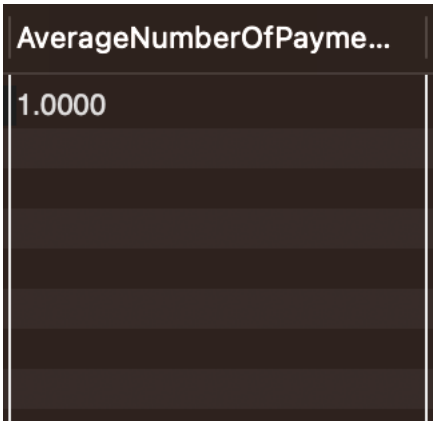
I selected this sentence so that it requires using group by operation:

"Calculate the average number of payment methods per user."

Query:

```
SELECT AVG(PaymentCount) AS AverageNumberOfPayments
FROM (
  SELECT User.uid, COUNT(PaymentMethod.pid) AS PaymentCount
  FROM User
  JOIN PaymentMethod ON User.uid = PaymentMethod.uid
  GROUP BY User.uid
) AS UserPayments;
```

Result:



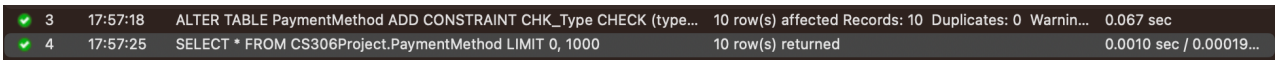
Q9. Adding a Constraint

Since I am working with the PaymentMethod table, the “type” column holds an important value for us. Because as the project discription dictates, we are entitled to expect payments only via credit cards (credit,debit,PayPal etc.). Thus, I am adding a constraint on the payment method types, as from now on, it will not accept any values other than the specified ones.

Query:

```
ALTER TABLE PaymentMethod
ADD CONSTRAINT CHK_Type CHECK (type IN ('Credit Card', 'Debit Card', 'PayPal'));
```

Result:

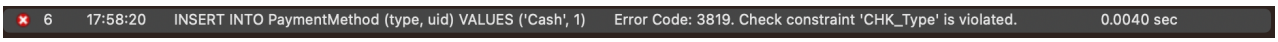


3	17:57:18	ALTER TABLE PaymentMethod ADD CONSTRAINT CHK_Type CHECK (type...	10 row(s) affected Records: 10 Duplicates: 0 Warnin...	0.067 sec
4	17:57:25	SELECT * FROM CS306Project.PaymentMethod LIMIT 0, 1000	10 row(s) returned	0.0010 sec / 0.00019...

Query:

```
INSERT INTO PaymentMethod (type, uid) VALUES ('Cash', 1);
```

Result:



6	17:58:20	INSERT INTO PaymentMethod (type, uid) VALUES ('Cash', 1)	Error Code: 3819. Check constraint 'CHK_Type' is violated.	0.0040 sec

As we have seen from this query result, trying to add Cash and 1 to they payment method table throws an exception, as it mentions check contrained is violated while trying to insert this expression.

This is the end of the project phase 2.