

Name: Halil İbrahim Umut  
Surname: Çolak  
ID: 28879

Step 2:

```
CREATE TABLE Customer (  
    Cuid INTEGER,  
    customer_name VARCHAR(40),  
    phone_number VARCHAR(20),  
    PRIMARY KEY (Cuid)  
);
```

```
CREATE TABLE Reviews (  
    Rid INTEGER,  
    Rating INTEGER,  
    Comment CHAR(50),  
    Cuid INTEGER,  
    FOREIGN KEY (Cuid) REFERENCES Customer(Cuid),  
    PRIMARY KEY(Rid)  
);
```

STEP 3:

```
INSERT INTO Customer (Cuid, customer_name, phone_number) VALUES  
(1, 'John Doe', '123-456-7890'),  
(2, 'Jane Smith', '234-567-8901'),  
(3, 'Alice Johnson', '345-678-9012'),  
(4, 'Bob Brown', '456-789-0123'),  
(5, 'Carol White', '567-890-1234'),  
(6, 'David Green', '678-901-2345'),  
(7, 'Eve Black', '789-012-3456'),  
(8, 'Frank Gray', '890-123-4567'),  
(9, 'Grace Lee', '901-234-5678'),  
(10, 'Hank Martin', '012-345-6789');
```

```
INSERT INTO Reviews (Rid, Rating, Comment, Cuid) VALUES  
(1, 4, 'Great service!', 1),  
(2, 5, 'Loved the product.', 2),  
(3, 3, 'Average experience.', 3),  
(4, 2, 'Could be better.', 4),
```

```
(5, 5, 'Excellent!', 5),
(6, 4, 'Very satisfied.', 6),
(7, 1, 'Not happy with the service.', 7),
(8, 3, 'It was okay.', 8),
(9, 4, 'Good, but can improve.', 9),
(10, 5, 'Outstanding experience!', 10);
```

### STEP 4:

1

SELECT \* FROM Reviews

|      |      |                             |      |
|------|------|-----------------------------|------|
| 1    | 4    | Great service!              | 1    |
| 2    | 5    | Loved the product.          | 2    |
| 3    | 3    | Average experience.         | 3    |
| 4    | 2    | Could be better.            | 4    |
| 5    | 5    | Excellent!                  | 5    |
| 6    | 4    | Very satisfied.             | 6    |
| 7    | 1    | Not happy with the service. | 7    |
| 8    | 3    | It was okay.                | 8    |
| 9    | 4    | Good, but can improve.      | 9    |
| 10   | 5    | Outstanding experience!     | 10   |
| NULL | NULL | NULL                        | NULL |

1

SELECT \* FROM Customer

|      |               |              |
|------|---------------|--------------|
| Cuid | customer_name | phone_number |
| 1    | John Doe      | 123-456-7890 |
| 2    | Jane Smith    | 234-567-8901 |
| 3    | Alice Johnson | 345-678-9012 |
| 4    | Bob Brown     | 456-789-0123 |
| 5    | Carol White   | 567-890-1234 |
| 6    | David Green   | 678-901-2345 |
| 7    | Eve Black     | 789-012-3456 |
| 8    | Frank Gray    | 890-123-4567 |
| 9    | Grace Lee     | 901-234-5678 |
| 10   | Hank Martin   | 012-345-6789 |
| NULL | NULL          | NULL         |

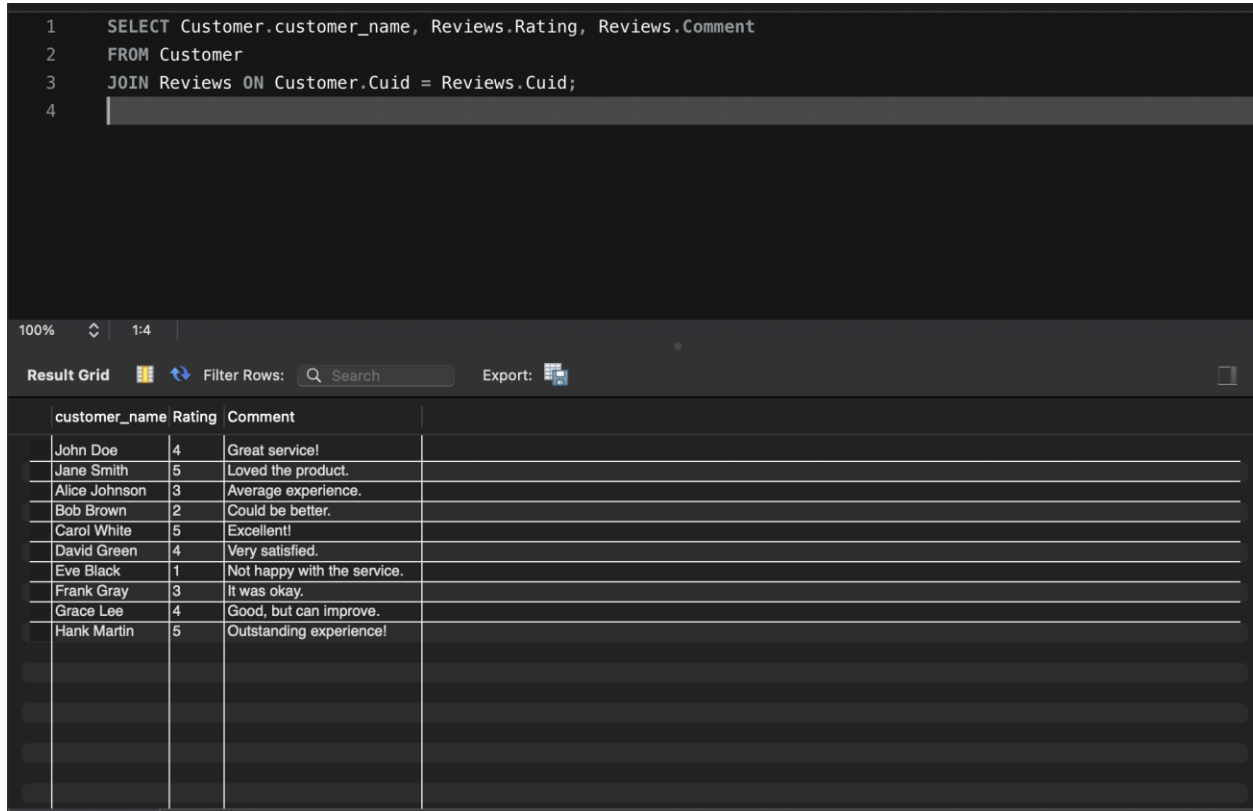
### STEP 5:

Retrieve the customer names and their corresponding ratings and comments from the "Customer" and "Reviews" tables, matching them based on the customer ID (Cuid) in both tables.

$$\pi \text{ customer\_name, Rating, Comment } (\sigma \text{ Customer.Cuid} = \text{Reviews.Cuid } (\text{Customer} \bowtie \text{Reviews}))$$

STEP 6:

```
SELECT Customer.customer_name, Reviews.Rating, Reviews.Comment
FROM Customer
JOIN Reviews ON Customer.Cuid = Reviews.Cuid;
```



The screenshot shows a SQL query editor with the following code:

```
1 SELECT Customer.customer_name, Reviews.Rating, Reviews.Comment
2 FROM Customer
3 JOIN Reviews ON Customer.Cuid = Reviews.Cuid;
4
```

Below the editor is a 'Result Grid' showing the results of the query. The grid has columns for 'customer\_name', 'Rating', and 'Comment'. The results are as follows:

| customer_name | Rating | Comment                     |
|---------------|--------|-----------------------------|
| John Doe      | 4      | Great service!              |
| Jane Smith    | 5      | Loved the product.          |
| Alice Johnson | 3      | Average experience.         |
| Bob Brown     | 2      | Could be better.            |
| Carol White   | 5      | Excellent!                  |
| David Green   | 4      | Very satisfied.             |
| Eve Black     | 1      | Not happy with the service. |
| Frank Gray    | 3      | It was okay.                |
| Grace Lee     | 4      | Good, but can improve.      |
| Hank Martin   | 5      | Outstanding experience!     |

STEP 7:

Retrieve the average rating given by customers for each customer's reviews. This will involve grouping the reviews by customer, calculating the average rating for each customer, and joining the result with the "Customer" table to display customer names alongside their average ratings.

```
SELECT Customer.customer_name, AVG(Reviews.Rating) AS AverageRating
FROM Customer
JOIN Reviews ON Customer.Cuid = Reviews.Cuid
```

GROUP BY Customer.customer\_name;

```
1 SELECT Customer.customer_name, AVG(Reviews.Rating) AS AverageRating
2 FROM Customer
3 JOIN Reviews ON Customer.Cuid = Reviews.Cuid
4 GROUP BY Customer.customer_name;
```

100% 1:5

Result Grid Filter Rows: Search Export:

| customer_name | AverageRating |
|---------------|---------------|
| John Doe      | 4.0000        |
| Jane Smith    | 5.0000        |
| Alice Johnson | 3.0000        |
| Bob Brown     | 2.0000        |
| Carol White   | 5.0000        |
| David Green   | 4.0000        |
| Eve Black     | 1.0000        |
| Frank Gray    | 3.0000        |
| Grace Lee     | 4.0000        |
| Hank Martin   | 5.0000        |

STEP 8:

The first SQL statement adds a constraint to the "Reviews" table, ensuring that the "Rating" column must be less than or equal to 5. The second SQL statement attempts to insert a record with a rating of 8, which violates the constraint and will result in an error.

```
ALTER TABLE Reviews ADD CONSTRAINT CheckRating CHECK (Rating <= 5);
INSERT INTO Reviews (Rid, Rating, Comment, Cuid) VALUES (7, 8, 'This is test comment', 1);
```

1 SELECT \*

2 FROM Reviews

3

100%

1:3

Result Grid

Filter Rows: Search

Edit: Export/Import:

| Rid  | Rating | Comment                     | Cuid |
|------|--------|-----------------------------|------|
| 1    | 4      | Great service!              | 1    |
| 2    | 5      | Loved the product.          | 2    |
| 3    | 3      | Average experience.         | 3    |
| 4    | 2      | Could be better.            | 4    |
| 5    | 5      | Excellent!                  | 5    |
| 6    | 4      | Very satisfied.             | 6    |
| 7    | 1      | Not happy with the service. | 7    |
| 8    | 3      | It was okay.                | 8    |
| 9    | 4      | Good, but can improve.      | 9    |
| 10   | 5      | Outstanding experience!     | 10   |
| NULL | NULL   | NULL                        | NULL |

Reviews 5

Apply Revert

Action Output

|    | Time     | Action   | Response   | Duration / Fetch Time     |
|----|----------|--|--|---------------------------|
| 16 | 15:13:39 | SELECT * FROM Reviews ORDER BY Rating  | 10 row(s) returned                                       | 0.00044 sec / 0.00044 sec |
| 13 | 15:13:39 | ALTER TABLE Reviews ADD CONSTRAINT CheckRating CHECK (Rating <= 5)             | 10 row(s) affected Records: 10 Duplicates: 0 Warnin...   | 0.023 sec                 |
| 14 | 15:13:39 | INSERT INTO Reviews (Rid, Rating, Comment, Cuid) VALUES (7, 8, 'This is tes... | Error Code: 3819. Check constraint 'CheckRating' is v... | 0.00035 sec               |
| 15 | 15:14:49 | SELECT * FROM Reviews LIMIT 0, 1000  | 10 row(s) returned                                       | 0.00044 sec / 0.00044 sec |

1 ALTER TABLE Reviews ADD CONSTRAINT CheckRating CHECK (Rating <= 5);

2 INSERT INTO Reviews (Rid, Rating, Comment, Cuid) VALUES (7, 8, 'This is test comment', 1);

3

100%

1:3

Action Output

|    | Time     | Action   | Response   | Duration / Fetch Time     |
|----|----------|--|--|---------------------------|
| 11 | 15:09:07 | SELECT * FROM Reviews ORDER BY Rating  | 10 row(s) returned                                       | 0.00044 sec / 0.00044 sec |
| 12 | 15:09:10 | SELECT Customer.customer_name, AVG(Reviews.Rating) AS AverageRating...         | 10 row(s) returned                                       | 0.0012 sec / 0.00050 sec  |
| 13 | 15:13:39 | ALTER TABLE Reviews ADD CONSTRAINT CheckRating CHECK (Rating <= 5)             | 10 row(s) affected Records: 10 Duplicates: 0 Warnin...   | 0.023 sec                 |
| 14 | 15:13:39 | INSERT INTO Reviews (Rid, Rating, Comment, Cuid) VALUES (7, 8, 'This is tes... | Error Code: 3819. Check constraint 'CheckRating' is v... | 0.00035 sec               |