

ITC 6000 - Database Management System
Signature Assignment #4
SQL Complex Query Practice

Devi Somalinga Bhuvanesh
NUID – 001428225

Signature Assignment #4

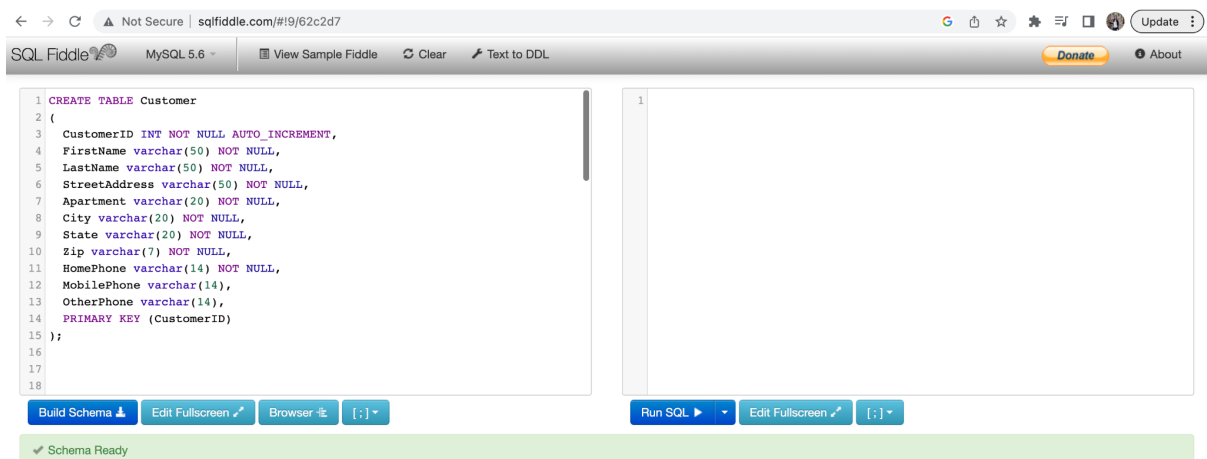
SQL Complex Query Practice

Based on the ER model created for Donut Shop, the following SQL Complex Query Practice has been written in SQL Fiddle platform.

A. Provide the SQL code that inserts data into all of the tables

CUSTOMER TABLE

```
CREATE TABLE Customer (CustomerID INT NOT NULL AUTO_INCREMENT,  
FirstName varchar(50) NOT NULL, LastName varchar(50) NOT NULL, StreetAddress  
varchar(50) NOT NULL, Apartment varchar(20) NOT NULL, City varchar(20) NOT NULL,  
State varchar(20) NOT NULL, Zip varchar(7) NOT NULL, HomePhone varchar(14) NOT  
NULL, MobilePhone varchar(14), OtherPhone varchar(14), PRIMARY KEY (CustomerID));
```



INSERT VALUES INTO CUSTOMER TABLE

```
INSERT INTO Customer (FirstName,LastName,StreetAddress,Apartment,City,State,Zip,  
HomePhone,MobliePhone,OtherPhone)  
VALUES  
("Devi","Bhuvanesh","695-4588 Ac Av.","#417","Chicago","Illinois","44027",  
"456-000-3982","91-8778588654","91-9840624903"),  
("Shubhakar","Reddy","1007 MLK Jrdr","#500","Seattle","Washington","98122",  
"206-000-3982","098-111325646","098-3234532145");
```

← → ↻ Not Secure | sqlfiddle.com/#19/17c35d

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
7 Apartment varchar(20) NOT NULL,
8 City varchar(20) NOT NULL,
9 State varchar(20) NOT NULL,
10 Zip varchar(7) NOT NULL,
11 HomePhone varchar(14) NOT NULL,
12 MobilePhone varchar(14),
13 OtherPhone varchar(14),
14 PRIMARY KEY (CustomerID)
15 );
16
17 INSERT INTO Customer
18 (FirstName,LastName,StreetAddress,Apartment,City,State,Zip,HomePhone,MobilePh
19 VALUES
20 ("Devi","Bhuvanesh","695-4588 Ac Av.","#417","Chicago","Illinois","44027",
21 ("Shubhakar","Reddy","1007 MLK Jr dr","#500","Seattle","Washington","98122"
22
23
24
```

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

Schema Ready

DONUT TABLE

CREATE TABLE Donut (DonutID INT NOT NULL AUTO_INCREMENT, DonutName varchar(15) NOT NULL, Description varchar(35) NOT NULL, UnitPrice decimal(5,2) NOT NULL, PRIMARY KEY (DonutID));

← → ↻ Not Secure | sqlfiddle.com/#19/6e3d5b

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
19 CREATE TABLE Donut
20 (
21 DonutID INT NOT NULL AUTO_INCREMENT,
22 DonutName varchar(15) NOT NULL,
23 Description varchar(35) NOT NULL,
24 UnitPrice decimal(5,2) NOT NULL,
25 PRIMARY KEY (DonutID)
26 );
27
28
29
30
31
32
33
34
35
36
```

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

Schema Ready

INSERT VALUES INTO DONUT TABLE

INSERT INTO Donut (DonutName,Description,UnitPrice)
VALUES

```
("Healthy Donut","Uses organic ingredients","10"),  
("Chocolate Donut","Filled with chocolate","10"),  
("Glazed Donut","Filled with powdered sugar and milk","15");
```

The screenshot shows the SQL Fiddle interface with the following SQL code in the editor:

```
24 CREATE TABLE Donut  
25 (  
26   DonutID INT NOT NULL AUTO_INCREMENT,  
27   DonutName varchar(15) NOT NULL,  
28   Description varchar(35) NOT NULL,  
29   UnitPrice decimal(5,2) NOT NULL,  
30   PRIMARY KEY (DonutID)  
31 );  
32  
33 CREATE INDEX DonutName ON Donut(DonutName);  
34  
35 INSERT INTO Donut  
36 (DonutName,Description,UnitPrice)  
37 VALUES  
38 ("Healthy Donut","Uses organic ingredients","10"),  
39 ("Chocolate Donut","Filled with chocolate","10"),  
40 ("Glazed Donut","Filled with powdered sugar and milk","15");  
41
```

The interface includes buttons for "Build Schema", "Edit Fullscreen", "Browser", "Run SQL", "Edit Fullscreen", and "Schema Ready" status is shown at the bottom.

INVOICE TABLE

CREATE TABLE Invoice (DonutOrderID INT NOT NULL AUTO_INCREMENT,
CustomerID INT NOT NULL, OrderDate Date NOT NULL, Notes varchar(255), PRIMARY
KEY (DonutOrderID), FOREIGN KEY (CustomerID) REFERENCES Customer
(CustomerID));

The screenshot shows the SQL Fiddle interface with the following SQL code in the editor:

```
71 CREATE TABLE INVOICE_LINE_ITEM  
72 (  
73   DonutOrderID INT,  
74   DonutID INT,  
75   Quantity INT,  
76   PRIMARY KEY(DonutOrderID,DonutID),  
77   FOREIGN KEY(DonutID) REFERENCES Donut(DonutID) ON UPDATE CASCADE,  
78   FOREIGN KEY(DonutOrderID) REFERENCES INVOICE(DonutOrderID) ON UPDATE CASCAD  
79 );  
80  
81  
82  
83  
84  
85  
86  
87  
88
```

The interface includes buttons for "Build Schema", "Edit Fullscreen", "Browser", "Run SQL", "Edit Fullscreen", and "Schema Ready" status is shown at the bottom. A "Query Panel" is visible on the right side.

INSERT VALUES INTO INVOICE TABLE

INSERT INTO Invoice (CustomerID,OrderDate,Notes)
VALUES

("002","2023-03-03","No sugar"),

("001","2023-04-02","Medium chocolate"),

("002","2023-03-03","None");

The screenshot shows the SQL Fiddle web application. The left pane contains the following SQL code:

```
49 CREATE TABLE Invoice
50 (
51     DonutOrderID INT NOT NULL AUTO_INCREMENT,
52     CustomerID INT NOT NULL,
53     OrderDate Date NOT NULL,
54     Notes varchar(255),
55     PRIMARY KEY (DonutOrderID),
56     FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
57 );
58
59 INSERT INTO Invoice
60 (CustomerID,OrderDate,Notes)
61 VALUES
62 ("002","2023-03-03","No sugar"),
63 ("001","2023-04-02","Medium chocolate"),
64 ("002","2023-03-03","None");
65
66
```

The right pane shows the results of the SQL execution, which is currently empty. The interface includes a toolbar with buttons for 'Build Schema', 'Edit Fullscreen', 'Browser', and 'Run SQL'. A status bar at the bottom indicates 'Schema Ready'.

INVOICE_LINE_ITEM TABLE

CREATE TABLE INVOICE_LINE_ITEM (DonutOrderID INT, DonutID INT, Quantity INT,
PRIMARY KEY(DonutOrderID,DonutID), FOREIGN KEY(DonutID) REFERENCES
Donut(DonutID) ON UPDATE CASCADE, FOREIGN KEY(DonutOrderID)
REFERENCES INVOICE(DonutOrderID) ON UPDATE CASCADE);

The screenshot shows the SQL Fiddle web application. The left pane contains the following SQL code:

```
69 CREATE TABLE INVOICE_LINE_ITEM
70 (
71     DonutOrderID INT,
72     DonutID INT,
73     Quantity INT,
74     PRIMARY KEY(DonutOrderID,DonutID),
75     FOREIGN KEY(DonutID) REFERENCES Donut(DonutID) ON UPDATE CASCADE,
76     FOREIGN KEY(DonutOrderID) REFERENCES INVOICE(DonutOrderID) ON UPDATE CASCADE
77 );
78
79
80
81
82
83
84
85
86
```

The right pane shows the results of the SQL execution, which is currently empty. The interface includes a toolbar with buttons for 'Build Schema', 'Edit Fullscreen', 'Browser', and 'Run SQL'. A status bar at the bottom indicates 'Schema Ready'.

INSERT VALUES INTO INVOICE_LINE_ITEM

INSERT INTO INVOICE_LINE_ITEM (DonutOrderID,DonutID,Quantity)

VALUES

("1","1","2"),

("2","1","1"),

("2","2","2");

The screenshot shows the SQL Fiddle interface with the following SQL code in the left pane:

```
71 CREATE TABLE INVOICE_LINE_ITEM
72 (
73     DonutOrderID INT,
74     DonutID INT,
75     Quantity INT,
76     PRIMARY KEY(DonutOrderID,DonutID),
77     FOREIGN KEY(DonutID) REFERENCES Donut(DonutID) ON UPDATE CASCADE,
78     FOREIGN KEY(DonutOrderID) REFERENCES INVOICE(DonutOrderID) ON UPDATE CASCADE
79 );
80
81 INSERT INTO INVOICE_LINE_ITEM
82 (DonutOrderID,DonutID,Quantity)
83 VALUES
84 ("1","1","2"),
85 ("2","1","1"),
86 ("2","2","2");
87
88
```

The right pane shows the result of the execution, which is empty. The interface includes buttons for 'Build Schema', 'Edit Fullscreen', 'Browser', 'Run SQL', 'Edit Fullscreen', and 'Schema Ready' status.

B. Provide SQL Code for simple queries to display all data in each table created

To view Customer Table

SELECT * FROM Customer

The screenshot shows the SQL Fiddle interface with the following SQL code in the left pane:

```
4  FirstName varchar(50) NOT NULL,
5  LastName varchar(50) NOT NULL,
6  StreetAddress varchar(50) NOT NULL,
7  Apartment varchar(20) NOT NULL,
8  City varchar(20) NOT NULL,
9  State varchar(20) NOT NULL,
10 Zip varchar(7) NOT NULL,
11 HomePhone varchar(14) NOT NULL,
12 MobilePhone varchar(14),
13 OtherPhone varchar(14),
14 PRIMARY KEY (CustomerID)
15 );
16 INSERT INTO Customer
17 (FirstName,LastName,StreetAddress,Apartment,City,State,Zip,HomePhone,MobilePh
18 VALUES
19 ("Devi","Bhuvanesh","695-4588 Ac Av.","#417","Chicago","Illinois","44027",
20 ("Shubhakar","Reddy","1007 MLK Jr dr","#500","Seattle","Washington","98122"
21
```

The right pane shows the result of the query: `SELECT * FROM Customer`. Below the panes, a table displays the data:

| CustomerID | FirstName | LastName | StreetAddress | Apartment | City | State | Zip | HomePhone | MobilePhone | OtherPhone |
|------------|-----------|-----------|-----------------|-----------|---------|------------|-------|--------------|---------------|----------------|
| 1 | Devi | Bhuvanesh | 695-4588 Ac Av. | #417 | Chicago | Illinois | 44027 | 456-000-3982 | 91-8778588654 | 91-9840624903 |
| 2 | Shubhakar | Reddy | 1007 MLK Jr dr | #500 | Seattle | Washington | 98122 | 206-000-3982 | 098-111325646 | 098-3234532145 |

The interface includes buttons for 'Build Schema', 'Edit Fullscreen', 'Browser', 'Run SQL', 'Edit Fullscreen', and a status bar showing 'Record Count: 2; Execution Time: 1ms'.

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!

To view Donut Table

SELECT * FROM Donut

← → ↺ Not Secure | sqlfiddle.com/#19/caec0f/1

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
40 CREATE TABLE Donut
41 (
42   DonutID INT NOT NULL AUTO_INCREMENT,
43   DonutName varchar(15) NOT NULL,
44   Description varchar(35) NOT NULL,
45   UnitPrice decimal(5,2) NOT NULL,
46   PRIMARY KEY (DonutID)
47 );
48
49
50 INSERT INTO Donut
51 (DonutName,Description,UnitPrice)
52 VALUES
53 ("Healthy Donut","Uses organic ingredients","10"),
54 ("Chocolate Donut","Filled with chocolate","10"),
55 ("Glazed Donut","Filled with powdered sugar and milk","15");
56
57
```

```
1 SELECT * FROM Donut
```

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

| DonutID | DonutName | Description | UnitPrice |
|---------|-----------------|-------------------------------------|-----------|
| 1 | Healthy Donut | Uses organic ingredients | 10 |
| 2 | Chocolate Donut | Filled with chocolate | 10 |
| 3 | Glazed Donut | Filled with powdered sugar and milk | 15 |

✔ Record Count: 3; Execution Time: 1ms View Execution Plan link

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!

To view Invoice Table

SELECT * FROM Invoice

← → ↺ Not Secure | sqlfiddle.com/#19/e64e01/1

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
61 CREATE TABLE Invoice
62 (
63   DonutOrderID INT NOT NULL AUTO_INCREMENT,
64   CustomerID INT NOT NULL,
65   OrderDate Date NOT NULL,
66   Notes varchar(255),
67   PRIMARY KEY (DonutOrderID),
68   FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
69 );
70
71
72 INSERT INTO Invoice
73 (CustomerID,OrderDate,Notes)
74 VALUES
75 ("002","2023-03-03","No sugar"),
76 ("001","2023-04-02","Medium chocolate"),
77 ("002","2023-03-03","None");
78
79
```

```
1 SELECT * FROM Invoice
```

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

| DonutOrderID | CustomerID | OrderDate | Notes |
|--------------|------------|------------|------------------|
| 1 | 2 | 2023-03-03 | No sugar |
| 2 | 1 | 2023-04-02 | Medium chocolate |
| 3 | 2 | 2023-03-03 | None |

✔ Record Count: 3; Execution Time: 1ms View Execution Plan link

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!

To view INVOICE_LINE_ITEM Table

SELECT * FROM INVOICE_LINE_ITEM

The screenshot shows the SQL Fiddle interface. The left pane contains the following SQL code:

```

71 CREATE TABLE INVOICE_LINE_ITEM
72 (
73     DonutOrderID INT,
74     DonutID INT,
75     Quantity INT,
76     PRIMARY KEY(DonutOrderID,DonutID),
77     FOREIGN KEY(DonutID) REFERENCES Donut(DonutID) ON UPDATE CASCADE,
78     FOREIGN KEY(DonutOrderID) REFERENCES INVOICE(DonutOrderID) ON UPDATE CASCADE
79 );
80
81 INSERT INTO INVOICE_LINE_ITEM
82 (DonutOrderID,DonutID,Quantity)
83 VALUES
84 ("1","1","2"),
85 ("2","1","1"),
86 ("2","2","2");
87
88

```

The right pane contains the query:

```

1 SELECT * FROM INVOICE_LINE_ITEM

```

Below the code panes, the 'Run SQL' button is clicked, and the results are displayed in a table:

| DonutOrderID | DonutID | Quantity |
|--------------|---------|----------|
| 1 | 1 | 2 |
| 2 | 1 | 1 |
| 2 | 2 | 2 |

At the bottom, a green status bar indicates: Record Count: 3; Execution Time: 18ms. Below this, a small text link says: [View Execution Plan](#) [link](#). At the very bottom, a small disclaimer reads: "Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!"

C. Provide the SQL code for a complex join query to display all of the information contained in the attached “Sales Order Form.”

```

SELECT Invoice.`OrderDate` AS
`DATE`,INVOICE_LINE_ITEM.`DonutOrderID` AS `DONUT ORDER
ID`,CONCAT(Customer.`FirstName`,``,CUSTOMER.`LastName`) AS NAME,
Customer.`StreetAddress`, Customer.`City`, Customer.`Zip`,
Customer.`HomePhone`, Customer.`MobilePhone`,
Customer.`OtherPhone`,INVOICE_LINE_ITEM.`Quantity`,
INVOICE_LINE_ITEM.`DonutID`,Donut.`DonutName`,Donut.`Description`,D
onut.`UnitPrice` FROM Customer
JOIN Invoice ON Customer.`CustomerID`=Invoice.`CustomerID` JOIN
`INVOICE_LINE_ITEM` ON Invoice.`DonutOrderID` =
INVOICE_LINE_ITEM.`DonutOrderID`
JOIN Donut ON INVOICE_LINE_ITEM.`DonutID`=Donut.`DonutID`
WHERE INVOICE_LINE_ITEM.`DonutOrderID`=2;

```


7

Apartment varchar(20) NOT NULL,

8

City varchar(20) NOT NULL,

9

State varchar(20) NOT NULL,

10

Zip varchar(7) NOT NULL,

11

HomePhone varchar(14) NOT NULL,

12

MobilePhone varchar(14),

13

OtherPhone varchar(14),

14

PRIMARY KEY (CustomerID)

15

);

16

17

CREATE VIEW CUSTOMER_DETAILS AS SELECT

18

CustomerID,

19

CONCAT(FirstName, " ", LastName) AS Name,

20

StreetAddress,

21

Apartment,

22

City,

1

SELECT Invoice.`OrderDate` AS `DATE`,INVOICE_LINE_ITEM.`DonutOrderID` AS `DON

2

JOIN Invoice ON Customer.`CustomerID`=Invoice.`CustomerID` JOIN `INVOICE_LINE

3

JOIN Donut ON INVOICE_LINE_ITEM.`DonutID`=Donut.`DonutID`

4

WHERE INVOICE_LINE_ITEM.`DonutOrderID`=2;

Build Schema

Edit Fullscreen

Browser

[;]

Run SQL

Edit Fullscreen

[;]

| DATE | DONUT ORDER ID | NAME | StreetAddress | City | Zip | HomePhone | MobilePhone | OtherPhone | Quantity | DonutID | DonutName | Description | UnitPrice |
|------------|----------------|----------------|-----------------|---------|-------|--------------|---------------|---------------|----------|---------|-----------------|--------------------------|-----------|
| 2023-04-02 | 2 | Devi Bhuvanesh | 695-4588 Ac Av. | Chicago | 44027 | 456-000-3982 | 91-8778588654 | 91-9840624903 | 1 | 1 | Healthy Donut | Uses organic ingredients | 10 |
| 2023-04-02 | 2 | Devi Bhuvanesh | 695-4588 Ac Av. | Chicago | 44027 | 456-000-3982 | 91-8778588654 | 91-9840624903 | 2 | 2 | Chocolate Donut | Filled with chocolate | 10 |

Record Count: 2; Execution Time: 2msView Execution Planlink

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!