4/4/2021

DBMS LAB ASSIGNMENT

Triggers, views and index

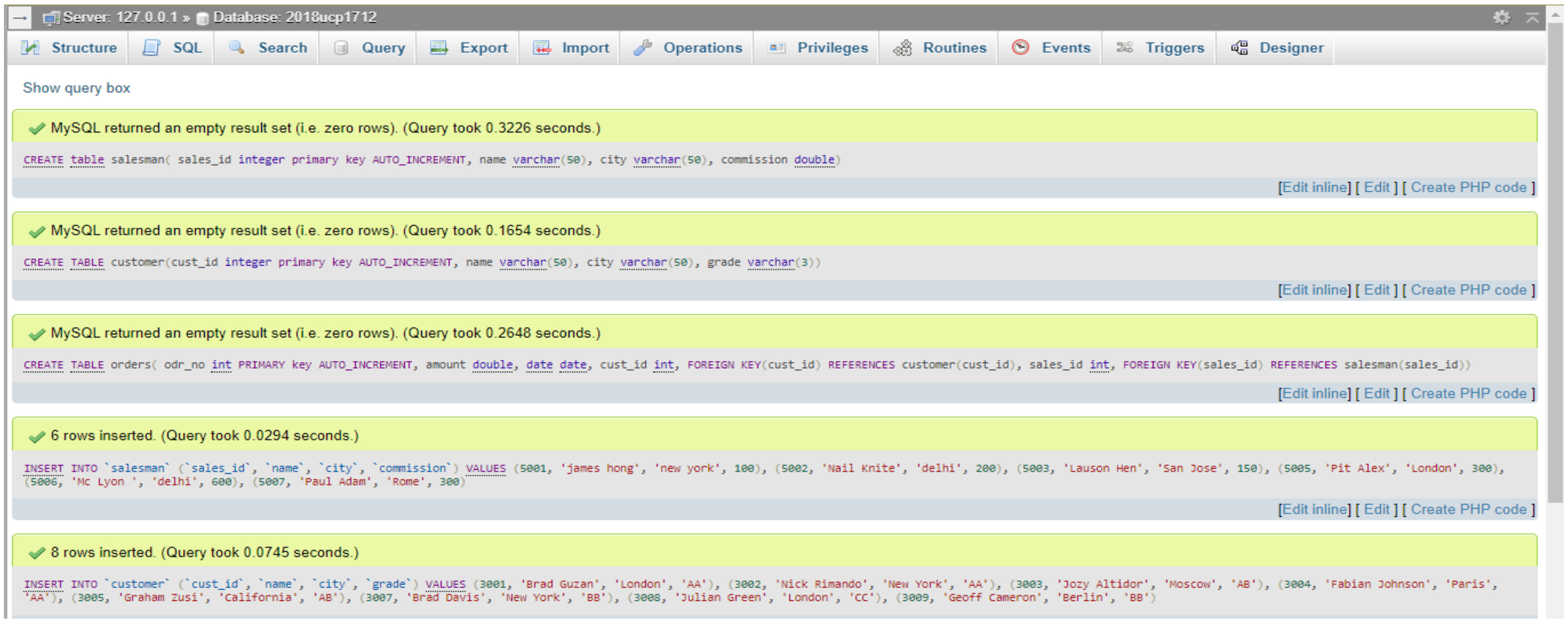
-Abdullah Jamal 2018UCP1712

Q1) create following tables :
Salesman(sales_id, name, city, commission);
Customer(cust_id, name, city, grade);
Orders(odr_no, amount, date, cust_id, sales_id);

```
CREATE table salesman( sales_id integer primary key AUTO_INCREMENT,
                    name varchar(50),
                    city varchar(50),
                    commission double);

CREATE TABLE customer(cust_id integer primary key AUTO_INCREMENT,
                    name varchar(50),
                    city varchar(50),
                    grade varchar(3));

CREATE TABLE orders( odr_no int PRIMARY key AUTO_INCREMENT,
                  amount double,
                 date date,
                  cust_id int,
                  FOREIGN KEY(cust_id) REFERENCES customer(cust_id),
                  sales_id int,
                  FOREIGN KEY(sales_id) REFERENCES salesman(sales_id));

INSERT INTO `salesman` (`sales_id`, `name`, `city`, `commission`) VALUES (5001,
'james hong', 'new york', 100), (5002, 'Nail Knite', 'delhi', 200), (5003, 'Lauson Hen', 'San
Jose', 150), (5005, 'Pit Alex', 'London', 300), (5006, 'Mc Lyon ', 'delhi', 600), (5007, 'Paul
Adam', 'Rome', 300);

INSERT INTO `customer` (`cust_id`, `name`, `city`, `grade`) VALUES (3001, 'Brad
Guzan', 'London', 'AA'), (3002, 'Nick Rimando', 'New York', 'AA'), (3003, 'Jozy Altidor',
'Moscow', 'AB'), (3004, 'Fabian Johnson', 'Paris', 'AA'), (3005, 'Graham Zusi', 'California',
```

'AB'), (3007, 'Brad Davis', 'New York', 'BB'), (3008, 'Julian Green', 'London', 'CC'), (3009, 'Geoff Cameron', 'Berlin', 'BB')

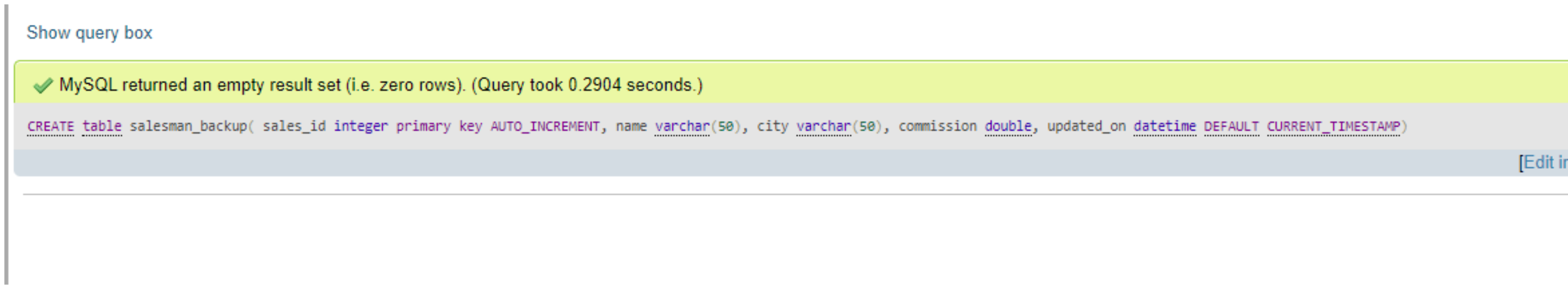
INSERT INTO `orders` (`odr_no`, `amount`, `date`, `cust_id`, `sales_id`) VALUES (70001, 150.5, '2020-10-05', 3005, 5001), (70002, 65.26, '2020-10-05', 3003, 5006), (70004, 110.5, '2020-08-17', 3005, 5006), (70005, 2400.6, '2020-09-10', 3003, 5007), (70007, 948.5, '2020-07-27', 3003, 5002), (70008, 5760, '2020-10-08', 3005, 5003), (70009, 270.65, '2020-10-31', 3001, 5002), (70010, 1983.43, '2020-10-10', 3004, 5003), (70011, 250.45, '2020-08-17', 3008, 5002), (70012, 2480.4, '2020-06-30', 3008, 5003), (70013, 75.29, '2020-04-25', 3008, 5007);



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.3226 seconds.)
CREATE table salesman( sales_id integer primary key AUTO_INCREMENT, name varchar(50), city varchar(50), commission double)
[Edit inline] [ Edit ] [ Create PHP code ]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.1654 seconds.)
CREATE TABLE customer(cust_id integer primary key AUTO_INCREMENT, name varchar(50), city varchar(50), grade varchar(3))
[Edit inline] [ Edit ] [ Create PHP code ]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.2648 seconds.)
CREATE TABLE orders( odr_no int PRIMARY key AUTO_INCREMENT, amount double, date date, cust_id int, FOREIGN KEY(cust_id) REFERENCES customer(cust_id), sales_id int, FOREIGN KEY(sales_id) REFERENCES salesman(sales_id))
[Edit inline] [ Edit ] [ Create PHP code ]

6 rows inserted. (Query took 0.0294 seconds.)
INSERT INTO `salesman` (`sales_id`, `name`, `city`, `commission`) VALUES (5001, 'james hong', 'new york', 100), (5002, 'Nail Knite', 'delhi', 200), (5003, 'Lauson Hen', 'San Jose', 150), (5005, 'Pit Alex', 'London', 300),
(5006, 'Mc Lyon ', 'delhi', 600), (5007, 'Paul Adam', 'Rome', 300)
[Edit inline] [ Edit ] [ Create PHP code ]

8 rows inserted. (Query took 0.0745 seconds.)
INSERT INTO `customer` (`cust_id`, `name`, `city`, `grade`) VALUES (3001, 'Brad Guzan', 'London', 'AA'), (3002, 'Nick Rimando', 'New York', 'AA'), (3003, 'Jozy Altidor', 'Moscow', 'AB'), (3004, 'Fabian Johnson', 'Paris',
'AA'), (3005, 'Graham Zusi', 'California', 'AB'), (3007, 'Brad Davis', 'New York', 'BB'), (3008, 'Julian Green', 'London', 'CC'), (3009, 'Geoff Cameron', 'Berlin', 'BB')
```

Q2) create a before update trigger that is invoked before any change is made to the salesman table and store the changes in a backup table.

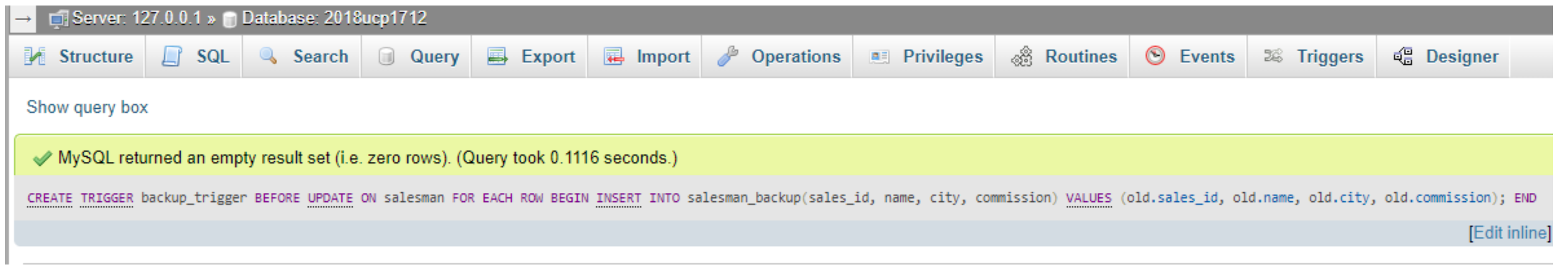
```
CREATE table salesman_backup( sales_id integer primary key AUTO_INCREMENT,
                name varchar(50),
                city varchar(50),
                commission double,
                updated_on datetime DEFAULT CURRENT_TIMESTAMP);
```


Show query box

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2904 seconds.)

```
CREATE table salesman_backup( sales_id integer primary key AUTO_INCREMENT, name varchar(50), city varchar(50), commission double, updated_on datetime DEFAULT CURRENT_TIMESTAMP)
```

```
DELIMITER $$
CREATE TRIGGER backup_trigger
   BEFORE UPDATE
   ON salesman FOR EACH ROW
BEGIN
   INSERT INTO salesman_backup(sales_id, name, city, commission) VALUES
(old.sales_id, old.name, old.city, old.commission);
END $$
DELIMITER ;
```



✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1116 seconds.)

```
CREATE TRIGGER backup_trigger BEFORE UPDATE ON salesman FOR EACH ROW BEGIN INSERT INTO salesman_backup(sales_id, name, city, commission) VALUES (old.sales_id, old.name, old.city, old.commission); END
```

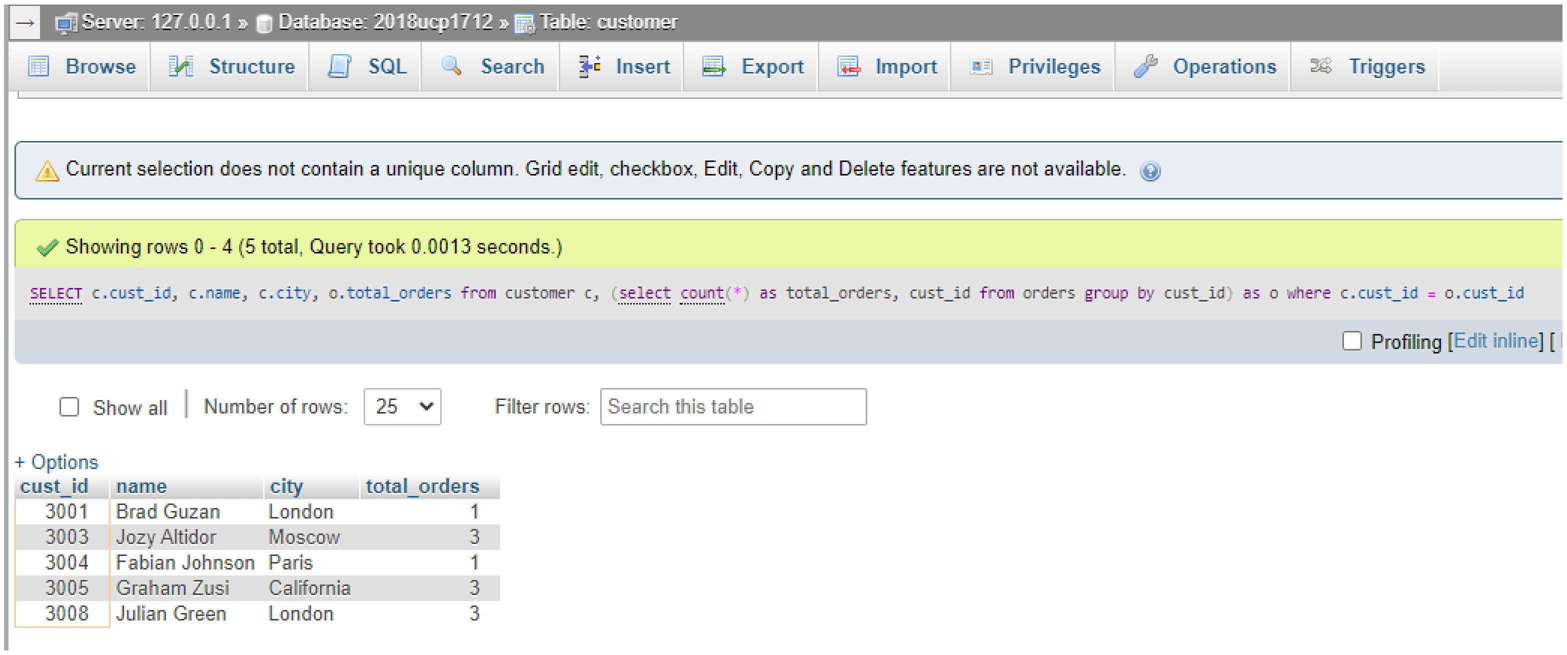
Q3) write a view to list customer id, name, city and total number of orders made by him.

Using sub-query

Create view customer_view as SELECT c.cust_id, c.name, c.city, o.total_orders from customer c, (select count(*) as total_orders, cust_id from orders group by cust_id) as o where c.cust_id = o.cust_id;

Using join

Create view customer_view as SELECT c.cust_id, c.name, c.city, count(*) as total from customer c join orders o on c.cust_id = o.cust_id group by c.cust_id


Server: 127.0.0.1 » Database: 2018ucp1712 » Table: customer
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0013 seconds.)

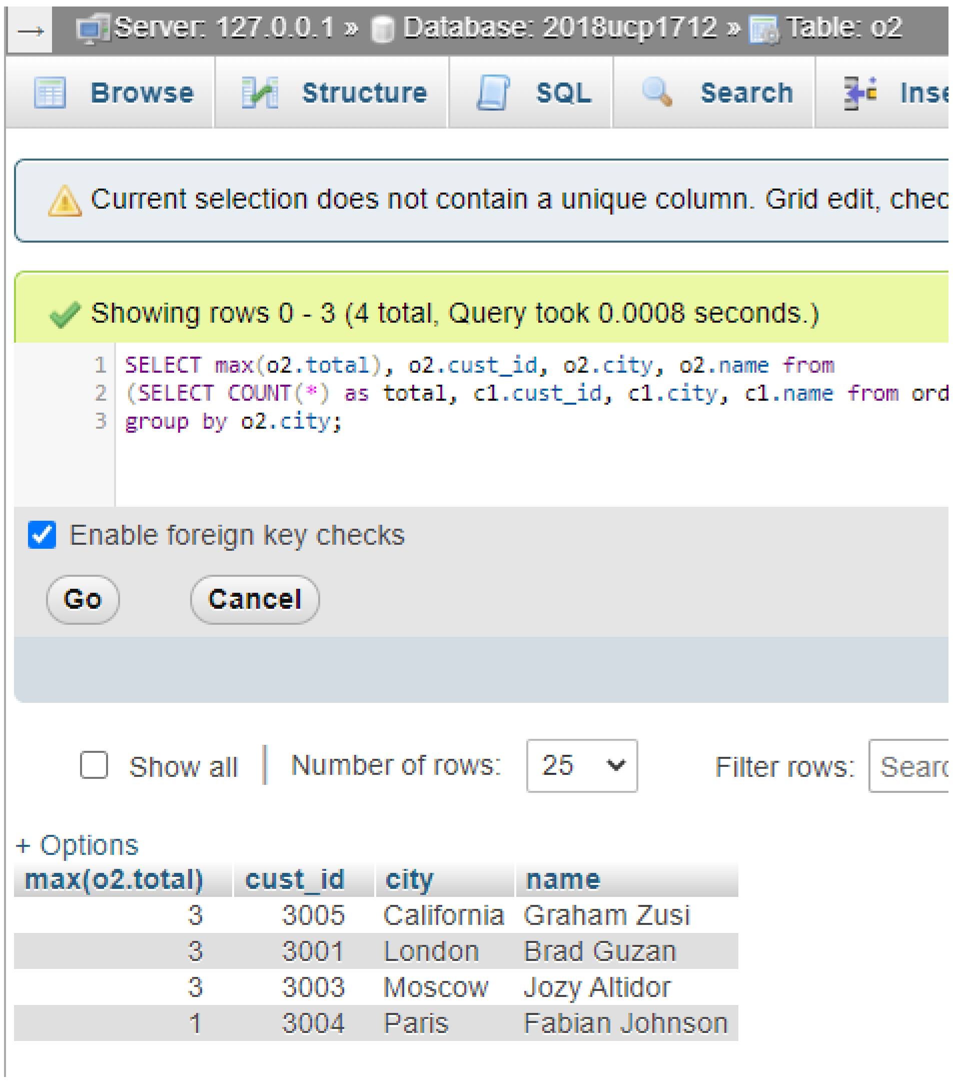
```
SELECT c.cust_id, c.name, c.city, o.total_orders from customer c, (select count(*) as total_orders, cust_id from orders group by cust_id) as o where c.cust_id = o.cust_id
```

Profiling [Edit inline] [
Show all
Number of rows: 25
Filter rows: Search this table
+ Options

cust_id	name	city	total_orders
3001	Brad Guzan	London	1
3003	Jozy Altidor	Moscow	3
3004	Fabian Johnson	Paris	1
3005	Graham Zusi	California	3
3008	Julian Green	London	3

Q4) write a query to find the customer with the highest number of orders in each city.

SELECT max(o2.total), o2.cust_id, o2.city, o2.name from
(SELECT COUNT(*) as total, c1.cust_id, c1.city, c1.name from orders o1, customer c1
WHERE o1.cust_id = c1.cust_id group by c1.cust_id) as o2
group by o2.city;


Server: 127.0.0.1 » Database: 2018ucp1712 » Table: o2

Browse | Structure | SQL | Search | Inse

Current selection does not contain a unique column. Grid edit, chec

Showing rows 0 - 3 (4 total, Query took 0.0008 seconds.)

```
SELECT max(o2.total), o2.cust_id, o2.city, o2.name from
(SELECT COUNT(*) as total, c1.cust_id, c1.city, c1.name from ord
group by o2.city;
```

Enable foreign key checks

Go Cancel

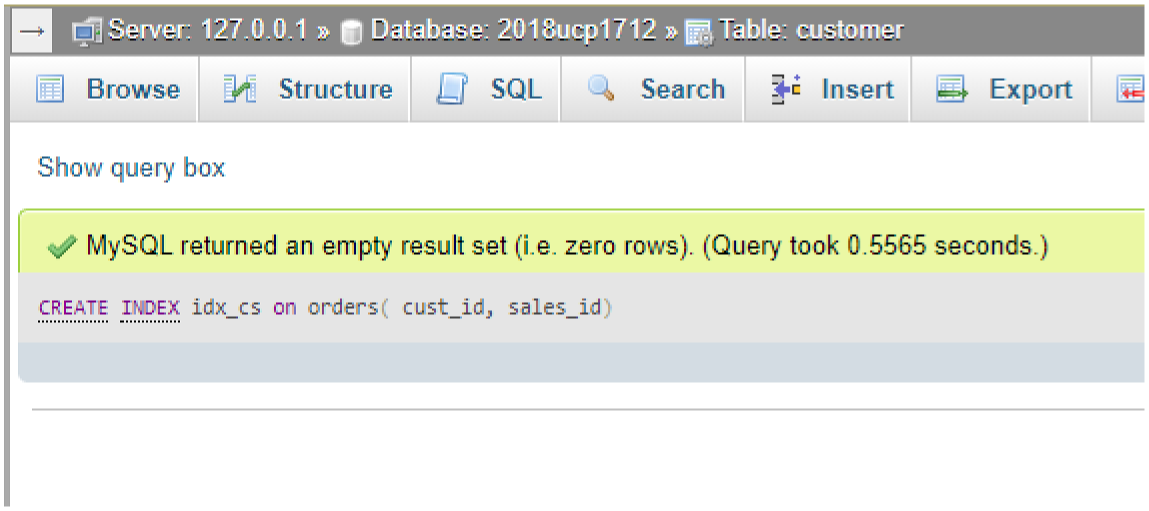
Show all | Number of rows: 25 Filter rows: Searc

+ Options

max(o2.total)	cust_id	city	name
3	3005	California	Graham Zusi
3	3001	London	Brad Guzan
3	3003	Moscow	Jozy Altidor
1	3004	Paris	Fabian Johnson

Q5) Write a query to create index using columns “cust_id” and sales_id” on the orders tables and name the index as “idx_cs”

CREATE INDEX idx_cs on orders(cust_id, sales_id);

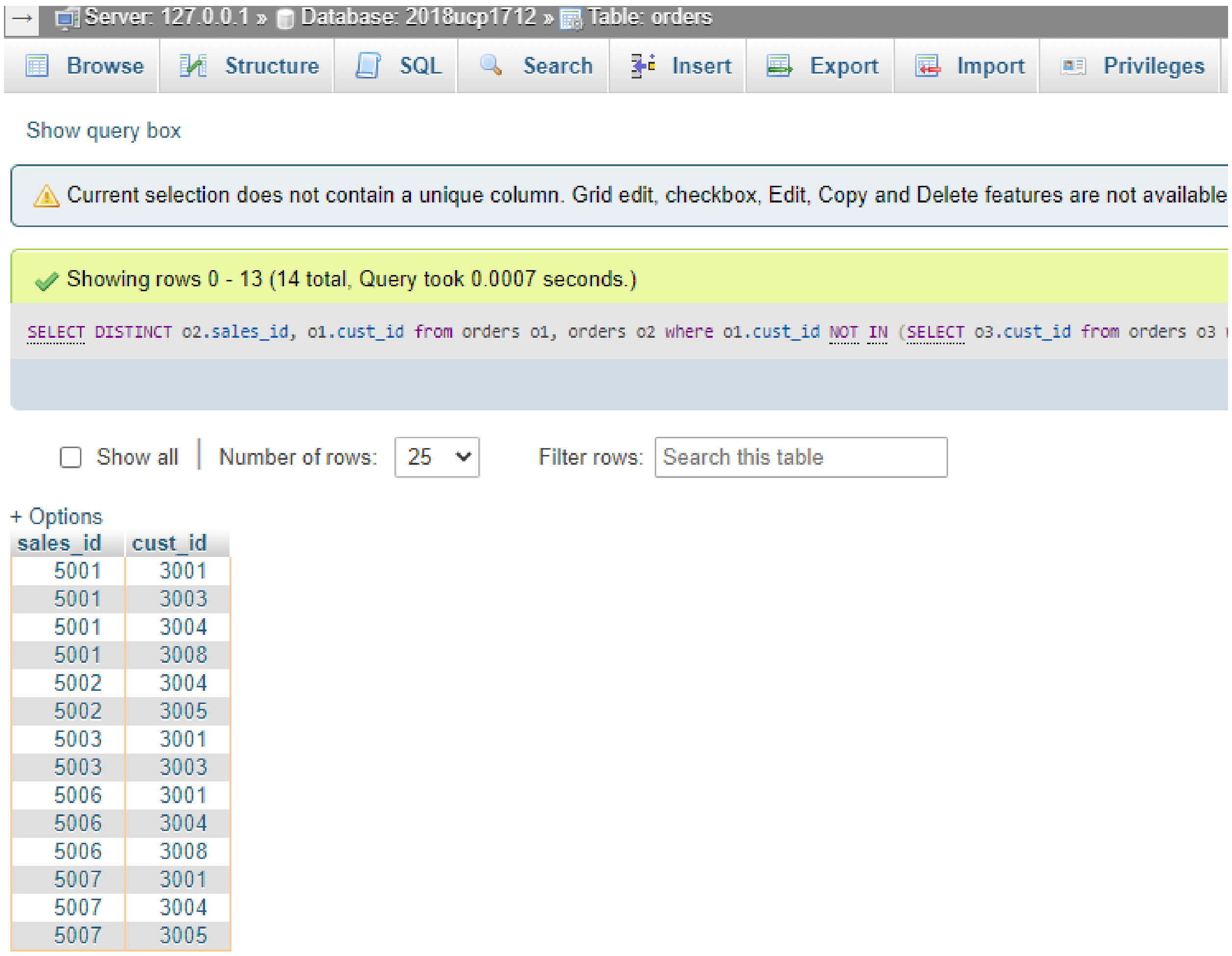


MySQL returned an empty result set (i.e. zero rows). (Query took 0.5565 seconds.)

```
CREATE INDEX idx_cs on orders( cust_id, sales_id)
```

Q6) Write a query to list for each salesman, the customer who has not ordered from him.

```
SELECT DISTINCT o2.sales_id, o1.cust_id
from orders o1, orders o2
where o1.cust_id NOT IN
(SELECT o3.cust_id from orders o3 where o3.sales_id = o2.sales_id);
```



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available

Showing rows 0 - 13 (14 total, Query took 0.0007 seconds.)

```
SELECT DISTINCT o2.sales_id, o1.cust_id from orders o1, orders o2 where o1.cust_id NOT IN (SELECT o3.cust_id from orders o3
```

sales_id	cust_id
5001	3001
5001	3003
5001	3004
5001	3008
5002	3004
5002	3005
5003	3001
5003	3003
5006	3001
5006	3004
5006	3008
5007	3001
5007	3004
5007	3005