
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

UNIVERSITÉ DE LIMOGES


# Advanced Databases
**Winter Semester 2022-2023**

## Project, Part 2
**"Data Modeling and Database Design"**

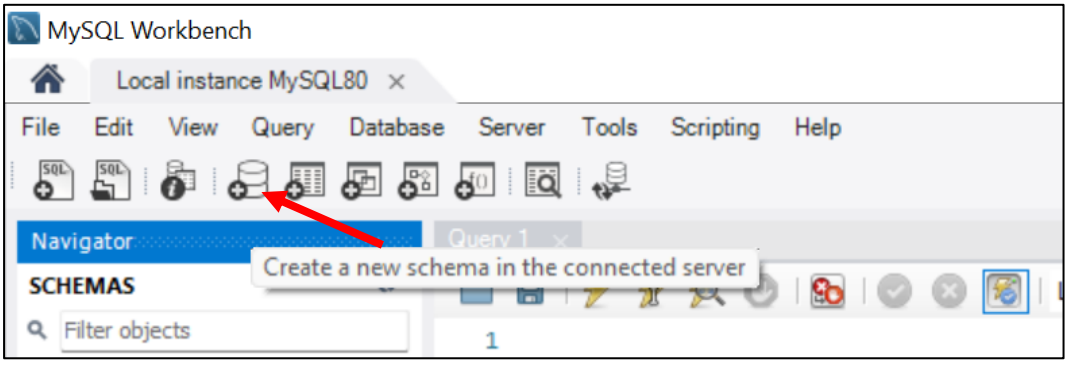
**Ονοματεπώνυμο: Χριστοδουλίδης Αναστάσιος**
**A.M.: aivc22019**

**Ονοματεπώνυμο: Κάππος Ιωάννης**
**A.M.: aivc22005**

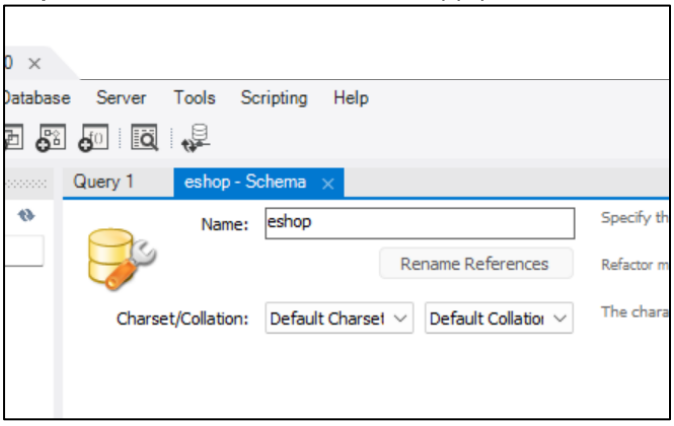
# How to write, load, export SQL

## 1. Create new Schema

**Step 1:** Create New Schema

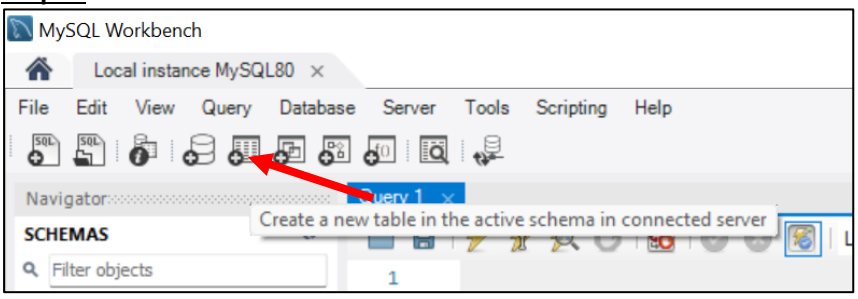



**Step 2:** Name the Schema and click apply

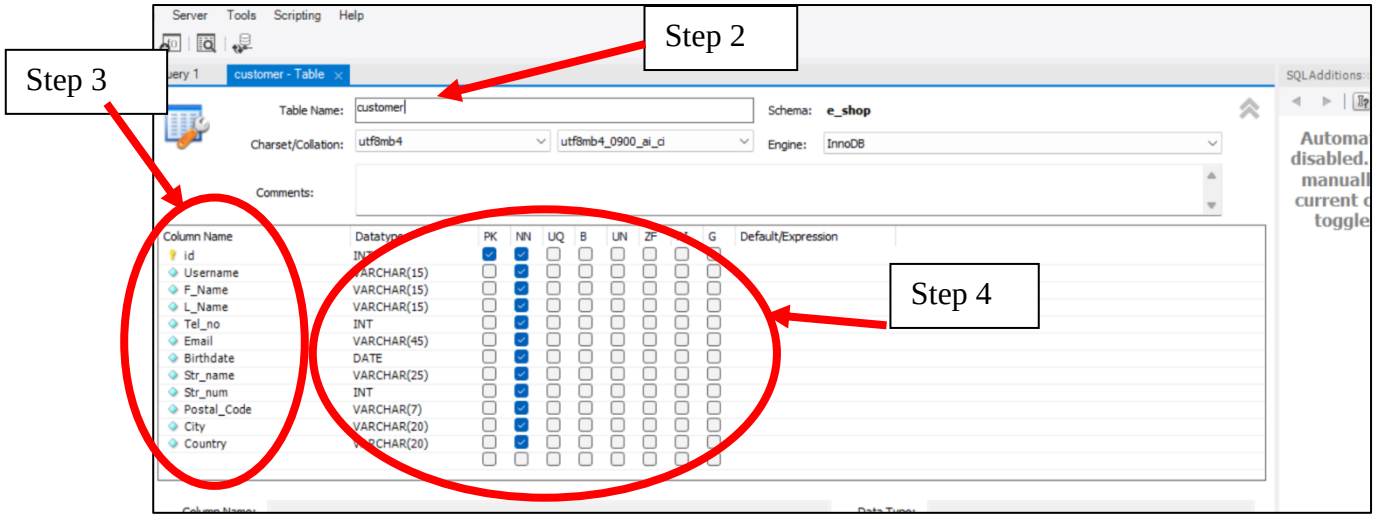



## 2. Create Table

**Step 1:** Create New table

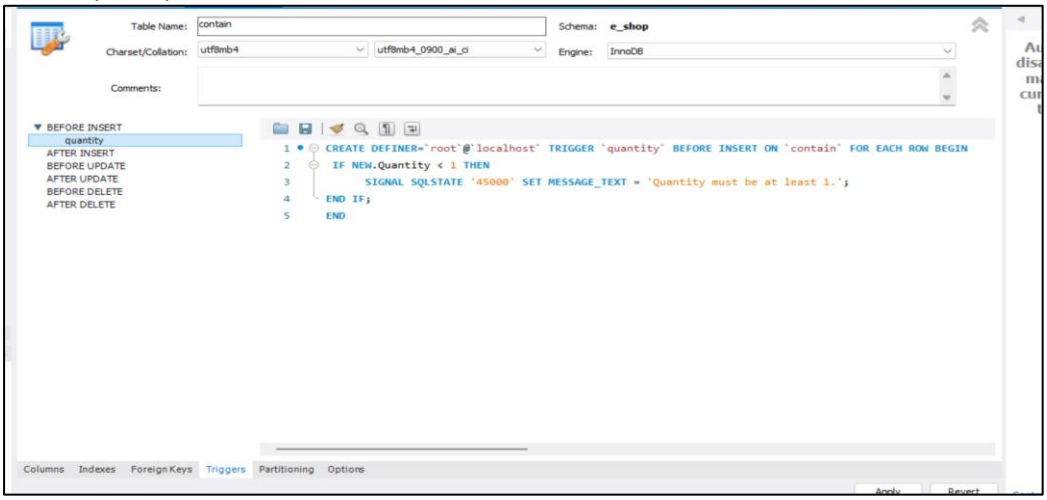


**Step 2:** Name Table
**Step 3:** Add attributes
**Step 4:** Set Data types/Constraints

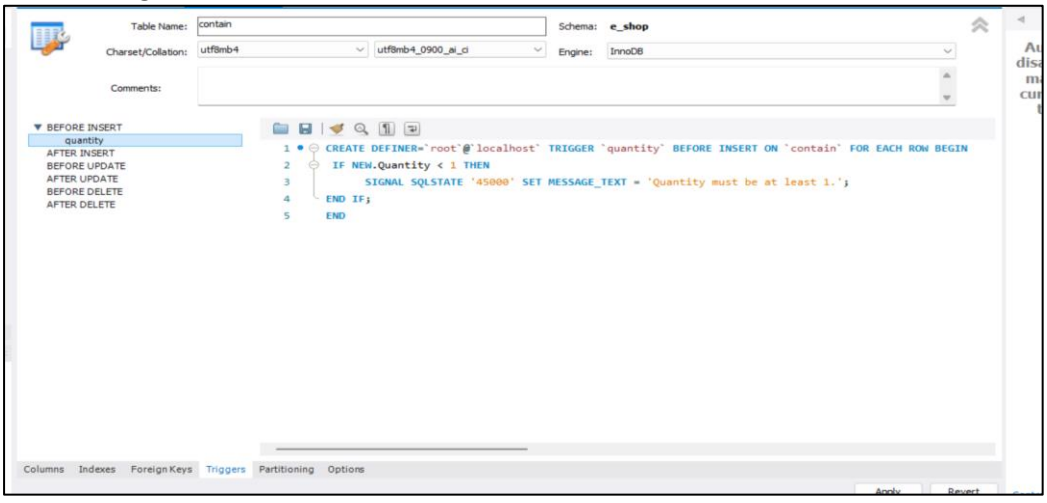



**3.** If additional constraints are needed, we can add **triggers**.
Below we see 2 examples that are used in our e_shop database.

a) Quantity of a product in the Contain table must be at least 1.

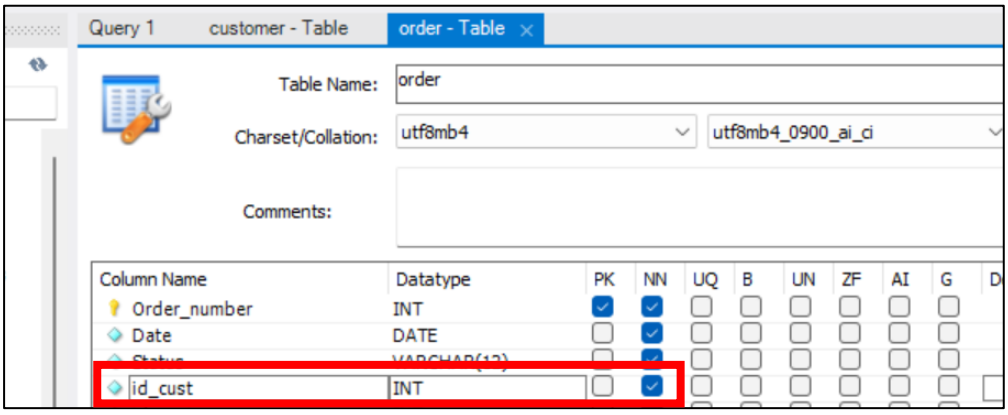



b) Customer age must be over 18

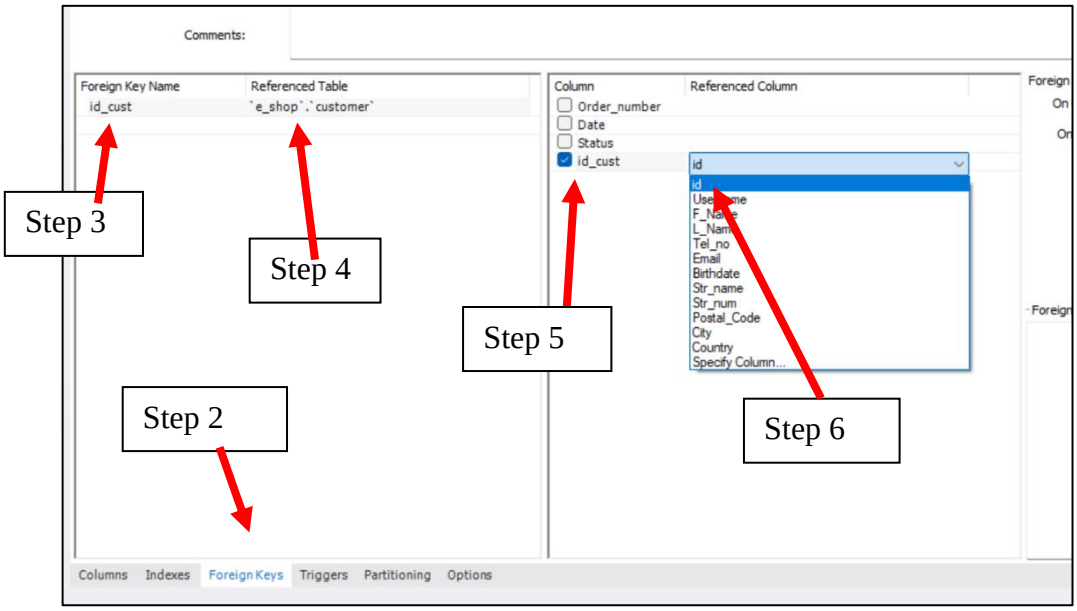


4. After Creating all the tables of the database we need to set the **Foreign Keys**.

**Step 1:** Add the foreign key as an attribute in the table as we did in the previous step with the same datatype of the attribute it's referring to (in the below example we added the attribute 'id_cust' which will be connected to the 'id' attribute of the customer table and become a foreign key)

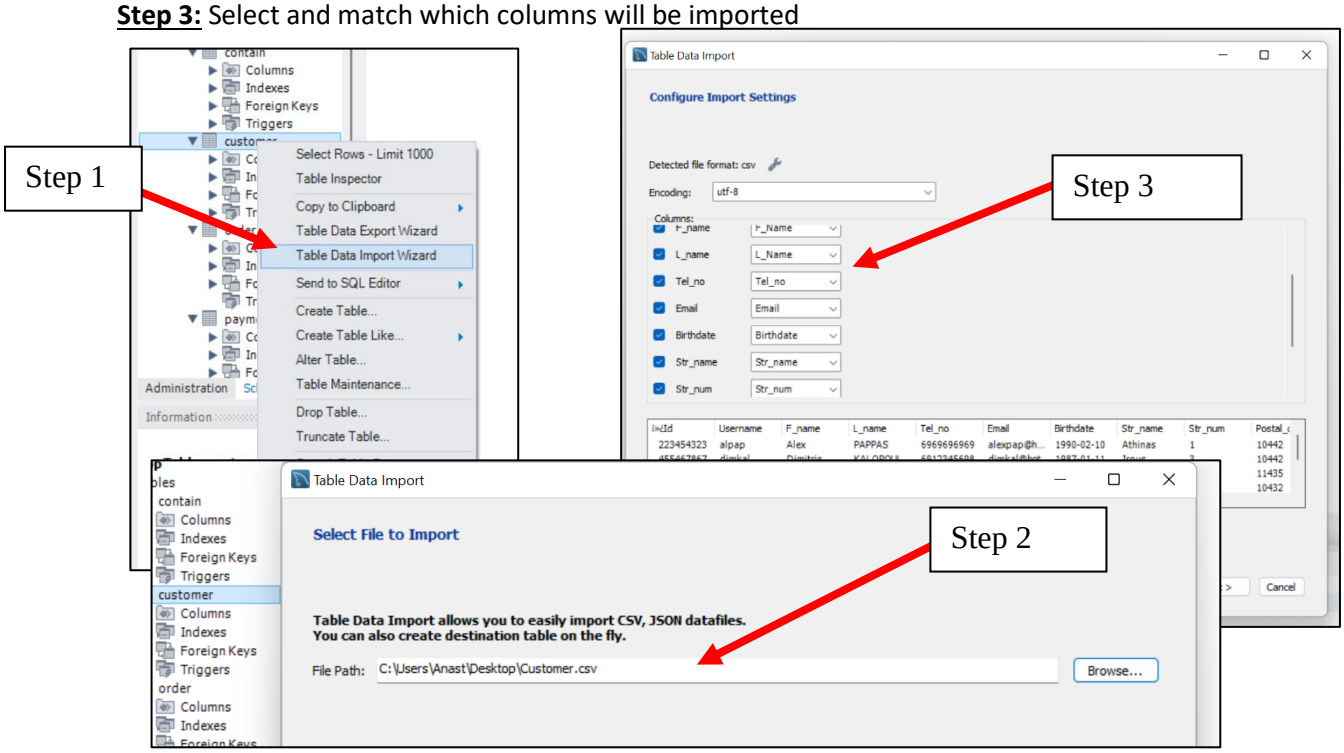



**Step 2:** Go to the 'Foreign Keys' tab
**Step 3:** Name the foreign key
**Step 4:** Choose the table the foreign key is referring to
**Step 5:** Choose the column of the current table that will be set as a foreign key
**Step 6:** Choose the attribute of the reference table that will be connected to the foreign key



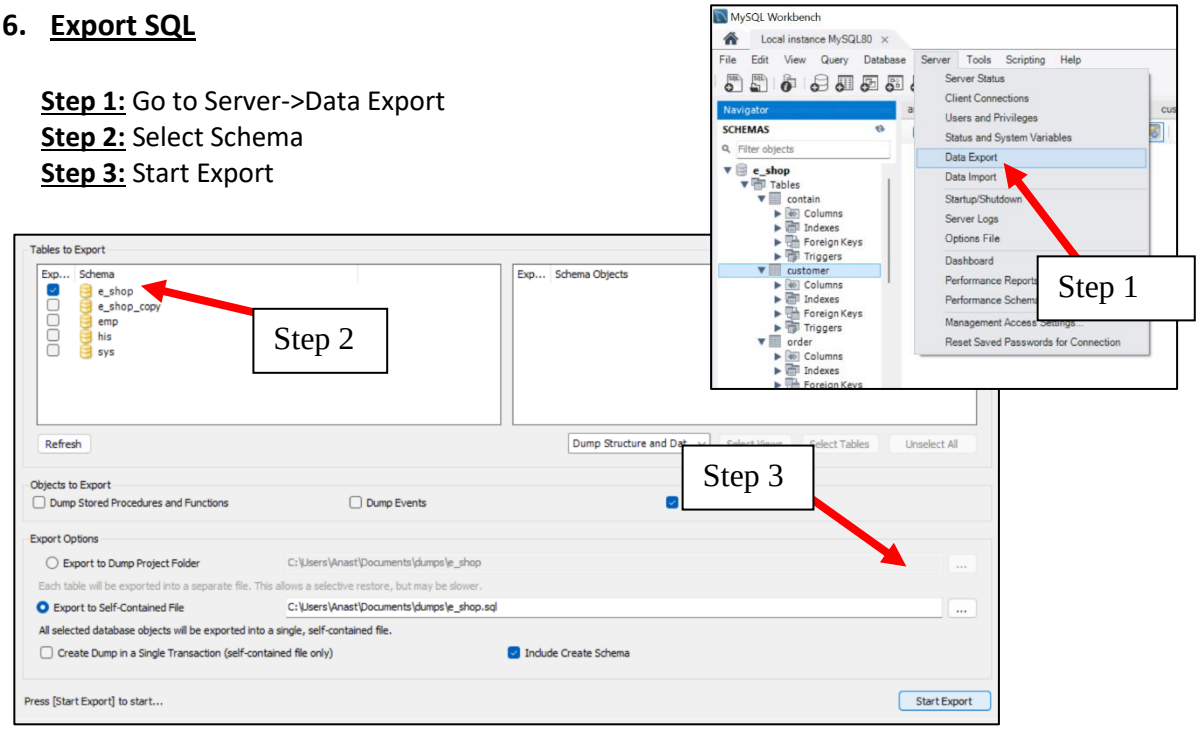
## 5. Populating the database

**Step 1:** Go to Table Data Import Wizard
**Step 2:** Select CSV file to import
**Step 3:** Select and match which columns will be imported




## 6. Export SQL

**Step 1:** Go to Server->Data Export
**Step 2:** Select Schema
**Step 3:** Start Export



***This query is used to update the Amount column in Payment when a new order is made.***

```
update payment set Amount =
(select sum(contain.Quantity*product.price) as total
from contain, product
where contain.Order_number=Order_no and contain.Pr_id=product.Pr_id);
```

## Queries and Results

### 1. Find Suppliers name and Tin who supply women hats

```
Select distinct S.Tin , S.`Name`
from supplier as S, product
where S.Tin=product.Supplier_tin and product.Category='hats' and product.Gender='F';
```

**Result:**

| Tin | Name |
|---|---|
| 123456789 | TSAKIROGLOU Panagiotis |
| 135799753 | DEPP Johnny |
| 579135790 | SPARROW Jack |

### 2. Find Customers with total payments over 1000

```
SELECT p.id_cust, SUM(p.amount) as total_payments
FROM payment as p
GROUP BY p.id_cust
Having total_payments>1000;
```

**Result:**

| id_cust | total_payments |
|---|---|
| 345646773 | 1205 |
| 475662564 | 1012 |
| 757473734 | 1700 |

### 3. Find total payments of all customers

```
SELECT SUM(p.amount) as total_payments
FROM payment as p;
```

**Result:**

| total_payments |
|---|
| 7868 |

**4. Find available female shoe ids of size 36 and colour red or black between prices 40 & 70**

```
Select P.Pr_id
from product as P
where(stock>0 and size=36 and colour='black' or 'red');
```

**Result:**

| Pr_id |
|---|
| 1212409 |
| 1212419 |
| 1212451 |

**5. Find most sold product id and quantity**

```
select T.Pr_id, T.total
from(Select C.Pr_id, sum(C.quantity) as total
        from contain as C
        group by C.Pr_id) as T
        having T.total=(select max(T.total)
                        from(Select C.Pr_id, sum(C.quantity) as total
                                from contain as C
                                group by C.Pr_id) as T);
```

**Result:**

| Pr_id | total |
|---|---|
| 1212425 | 6 |
| 1212438 | 6 |
| 1212461 | 6 |
| 1212467 | 6 |

**6. Find items with stock under 3 pieces**

```
select P.pr_id
from product as p
where(stock<3);
```

**Result:**

| pr_id |
|---|
| 1212347 |
| 1212350 |
| 1212351 |
| 1212352 |
| …… |

**7. Find shoes with size 42 and price less than 100 that are included in orders with quantity=2**

```
SELECT *
FROM product P, contain C
WHERE Category = "Shoes" and Size = 42 and Price < 100
and C.Quantity=2 and C.Pr_id = P.Pr_id;
```

**Result:**

| Pr_id | Pr_name | Category | Colour | Size | Description | Gender | Brand | Price | ……. |
|---|---|---|---|---|---|---|---|---|---|
| 1212366 | Casual | Shoes | Green | 42 | Athletic Shoe | M | Pollo | 57 | ……. |
| 1212382 | Casual | Shoes | Black | 42 | Casual Shoe | M | Boxxer | 57 | ……. |
| 1212394 | Casual | Shoes | Black | 42 | Casual Shoe | M | Galt | 57 | ……. |

| ……. | Stock | Reg_date | Supplier_tin | Order_number | Pr_id | Quantity |
|---|---|---|---|---|---|---|
| ……. | 4 | 15/10/2022 | 123456789 | 13584780 | 1212366 | 2 |
| ……. | 3 | 19/10/2022 | 135799753 | 13584769 | 1212382 | 2 |
| ……. | 2 | 27/10/2022 | 135799753 | 13584776 | 1212394 | 2 |

**8. Find supplier tin with most sold products (total value)**

```
select max(x.s),x.Supplier_tin
from (select sum(t.total*p.price) as s, p.supplier_tin
      from (Select C.Pr_id, sum(C.quantity) as total
            from contain as C
            group by C.Pr_id) as t
            natural join product as p
      group by p.supplier_tin) as x;
```

**Result:**

| max(x.s) | Supplier_tin |
|---|---|
| 1937 | 123456789 |

**9. Find most sold shoe size and quantity sold of male shoes**

```
select size, sold
from (select size, sum(sold) as sold
        from (select Pr_id, sold, category, size, gender
                from (Select C.Pr_id, sum(C.quantity) as sold
                        from contain as C
                        group by C.Pr_id) as q
                        natural join product as p
                where category = 'shoes' and gender='m') as k
        group by size) as w
where sold = (select max(sold)
              from (select size, sum(sold) as sold
                    from(select Pr_id, sold, category, size, gender
                            from (Select C.Pr_id, sum(C.quantity) as sold
                                    from contain as C
                                    group by C.Pr_id) as q
                                    natural join product as p
                            where category = 'shoes' and gender='m') as k
                    group by size) as w);
```

**Result:**

| size | sold |
|---|---|
| 42 | 22 |

**10. Find products id, category, colour, size gender and stock, that are not contained in orders and are not out of stock. Ordered by gender and product id.**

```
SELECT p.pr_id, p.category, p.colour, p.size, p.gender, p.stock
FROM product p
WHERE NOT EXISTS ( SELECT pr_id
                FROM product
                where pr_id IN(
                SELECT DISTINCT c.pr_id
                FROM contain c
                WHERE c.pr_id = p.pr_id)) AND stock<>0
ORDER BY gender,pr_id;
```

**Result:**

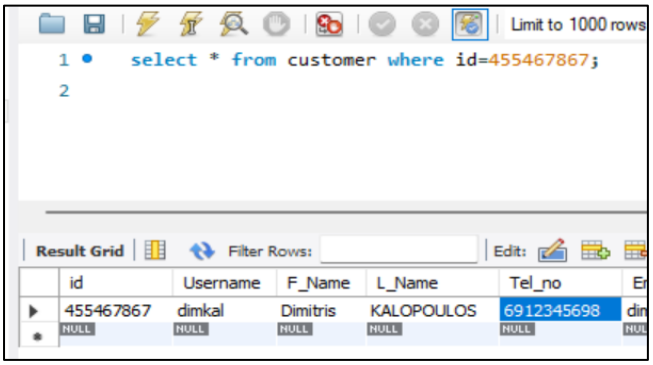
| pr_id | category | colour | size | gender | stock |
|---|---|---|---|---|---|
| 1212348 | Shoes | Black | 41 | M | 4 |
| 1212349 | Shoes | Black | 42 | M | 8 |
| 1212350 | Shoes | Black | 43 | M | 2 |
| 1212362 | Shoes | Green | 38 | M | 5 |
| 1212368 | Shoes | Green | 44 | M | 2 |
| 1212369 | Shoes | Green | 45 | M | 3 |
| …… | …….. | …….. | …… | …… | …… |
| 1212464 | Shoes | Green | 39 | F | 3 |
| 1212465 | Shoes | Green | 40 | F | 2 |
| 1212471 | Hats | Black | M | F | 3 |
| 1212473 | Hats | Black | M | F | 2 |

## UPDATE QUERIES:

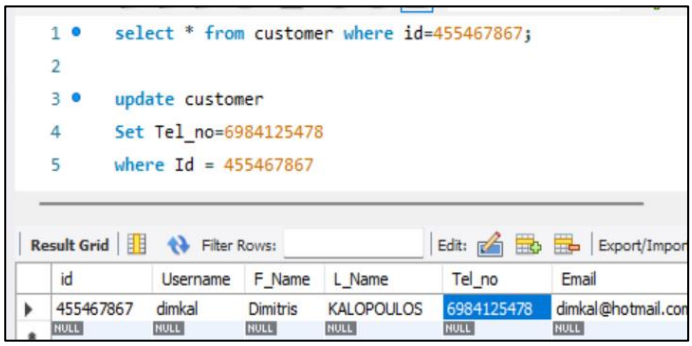
1. **Change the phone number of customer with id 455467867 to 6984125478**

```
update customer
Set Tel_no=6984125478
where Id = 455467867;
```

**Before**:



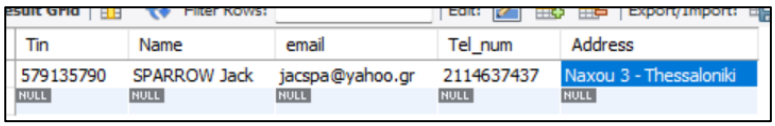

**After**:



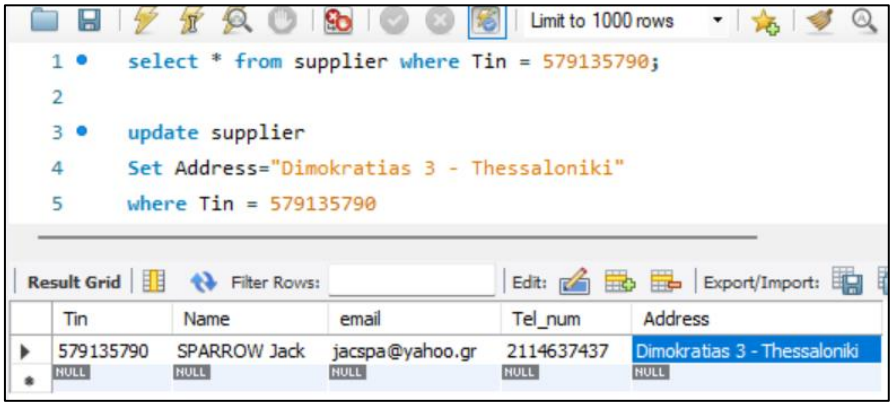

2. **Change the address of supplier with Tin 579135790 to Dimokratias 3 – Thessaloniki.**

```
update supplier
Set Address="Dimokratias 3 - Thessaloniki"
where Tin = 579135790;
```

**Before**:




**After**:

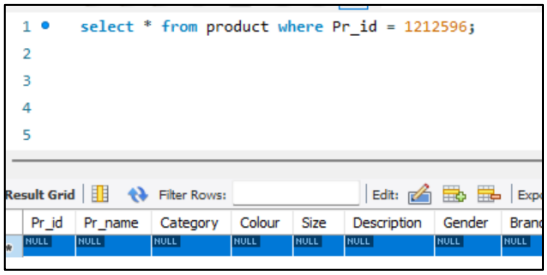


## INSERT QUERIES:

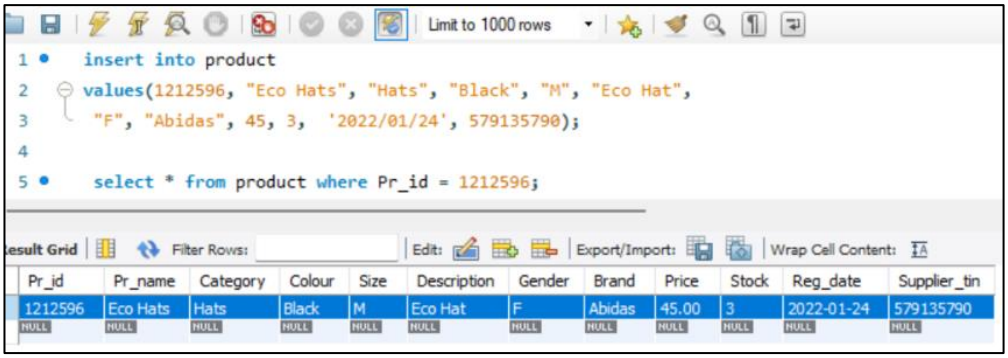
### 1. Insert new product

```
insert into product
values(1212596, "Eco Hats", "Hats", "Black", "M", "Eco Hat",
 "F", "Abidas", 45, 3, "2022-01-24", 579135790);
```

**Before:**



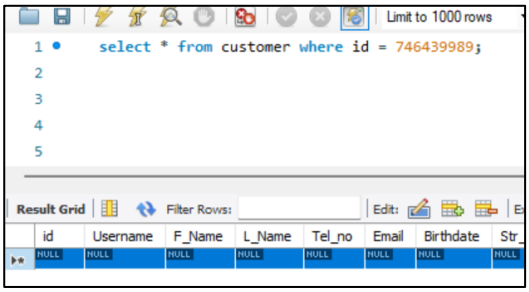

**After**:



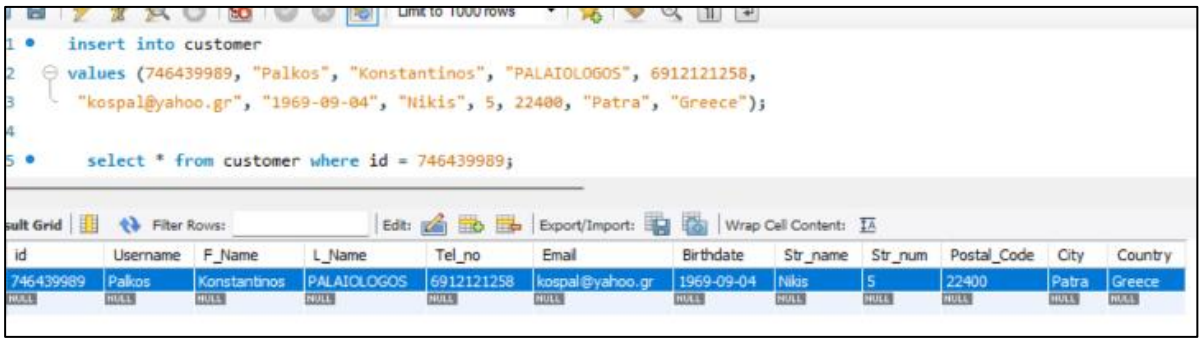

### 2. Insert new customer

```
insert into customer
values (746439989, "Palkos", "Konstantinos", "PALAIOLOGOS", 6912121258,
"kospal@yahoo.gr", "1969-09-04", "Nikis", 5, 22400, "Patra", "Greece");
```

**Before:**




**After**:

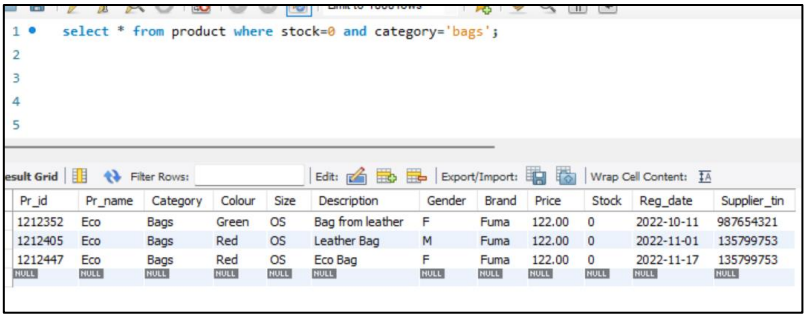


## DELETE QUERIES:

1. **Delete out of stock products with category bags**

```
delete from product where stock=0 and category='bags';
```

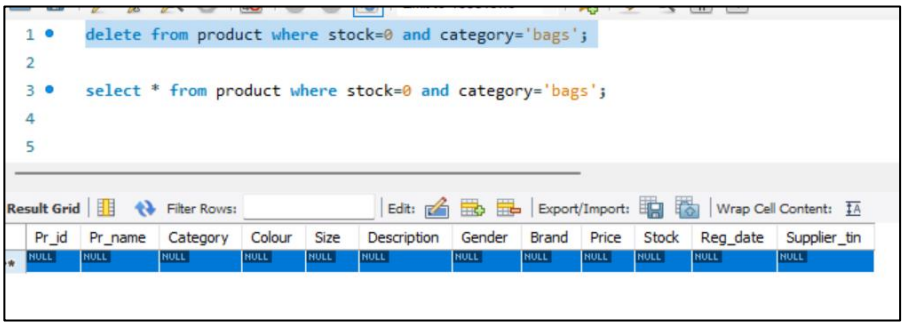
**Before:**



```
select * from product where stock=0 and category='bags';
```

| Pr_id | Pr_name | Category | Colour | Size | Description | Gender | Brand | Price | Stock | Reg_date | Supplier_tin |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1212352 | Eco | Bags | Green | OS | Bag from leather | F | Fuma | 122.00 | 0 | 2022-10-11 | 987654321 |
| 1212405 | Eco | Bags | Red | OS | Leather Bag | M | Fuma | 122.00 | 0 | 2022-11-01 | 135799753 |
| 1212447 | Eco | Bags | Red | OS | Eco Bag | F | Fuma | 122.00 | 0 | 2022-11-17 | 135799753 |

**After:**



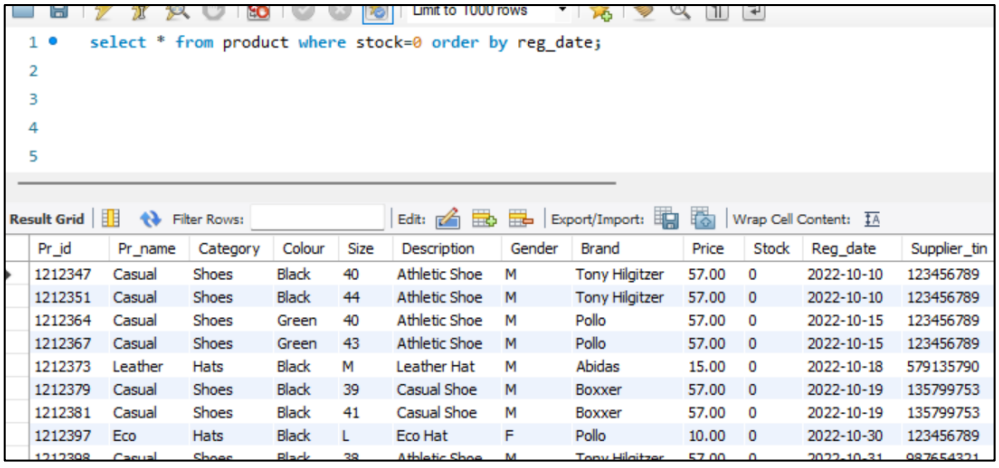
```
delete from product where stock=0 and category='bags';

select * from product where stock=0 and category='bags';
```

| Pr_id | Pr_name | Category | Colour | Size | Description | Gender | Brand | Price | Stock | Reg_date | Supplier_tin |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

2. **Delete out of stock products, registered before the 16th of October 2022**

```
delete from product where stock=0 and Reg_date<="2022-10-15";
```

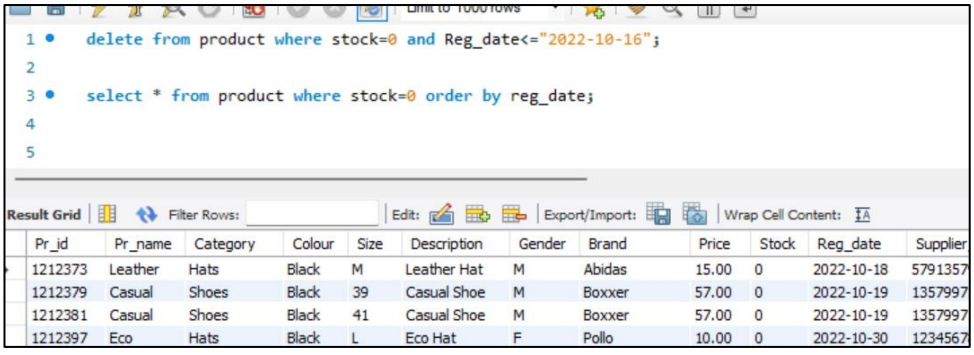
**Before:**



```
select * from product where stock=0 order by reg_date;
```

| Pr_id | Pr_name | Category | Colour | Size | Description | Gender | Brand | Price | Stock | Reg_date | Supplier_tin |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1212347 | Casual | Shoes | Black | 40 | Athletic Shoe | M | Tony Hilgitzer | 57.00 | 0 | 2022-10-10 | 123456789 |
| 1212351 | Casual | Shoes | Black | 44 | Athletic Shoe | M | Tony Hilgitzer | 57.00 | 0 | 2022-10-10 | 123456789 |
| 1212364 | Casual | Shoes | Green | 40 | Athletic Shoe | M | Pollo | 57.00 | 0 | 2022-10-15 | 123456789 |
| 1212367 | Casual | Shoes | Green | 43 | Athletic Shoe | M | Pollo | 57.00 | 0 | 2022-10-15 | 123456789 |
| 1212373 | Leather | Hats | Black | M | Leather Hat | M | Abidas | 15.00 | 0 | 2022-10-18 | 579135790 |
| 1212379 | Casual | Shoes | Black | 39 | Casual Shoe | M | Boxxer | 57.00 | 0 | 2022-10-19 | 135799753 |
| 1212381 | Casual | Shoes | Black | 41 | Casual Shoe | M | Boxxer | 57.00 | 0 | 2022-10-19 | 135799753 |
| 1212397 | Eco | Hats | Black | L | Eco Hat | F | Pollo | 10.00 | 0 | 2022-10-30 | 123456789 |

**After:**



```
delete from product where stock=0 and Reg_date<="2022-10-16";

select * from product where stock=0 order by reg_date;
```

| Pr_id | Pr_name | Category | Colour | Size | Description | Gender | Brand | Price | Stock | Reg_date | Supplier |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1212373 | Leather | Hats | Black | M | Leather Hat | M | Abidas | 15.00 | 0 | 2022-10-18 | 5791357 |
| 1212379 | Casual | Shoes | Black | 39 | Casual Shoe | M | Boxxer | 57.00 | 0 | 2022-10-19 | 1357997 |
| 1212381 | Casual | Shoes | Black | 41 | Casual Shoe | M | Boxxer | 57.00 | 0 | 2022-10-19 | 1357997 |
| 1212397 | Eco | Hats | Black | L | Eco Hat | F | Pollo | 10.00 | 0 | 2022-10-30 | 1234567 |