Relational Databases

Content Creator 2, 2018

Continuous Assesment

Business Academy Aarhus

Supervisor:

Murat Kilic

Submitted by:

Raimonds Korzenevskis

25.April. 2018

https://github.com/Raipis/databaseexamraimonds

Table of Contents

Introduction

What is the purpose of this synopsis?

The goal of this continuous assessment to test our knowledge in Front-End and Back-end development focusing on SQL, HTML and CSS. The case of choice is IKEA catalogue website database replication.

To understand the website's functionality and being able to recreate it's database, we have to reverse engineer it. To assist me in that I am following the "5 steps" of database designing and that is going to be what this synopsis structure is following.

Step 1

Conceptual Database Model

Requirements:

In order to plan a database layout, what data it will hold and how the end-users are supposed to interact with it, first we have to find out the requirements of the database.

Requirements were given by teacher in a case description document which are as follows: Visitor should be able to order a IKEA catalogue and fulfil the form with personal information. Webmaster should be able to add new product catalogues and get an overview to see how many customers have ordered which IKEA catalogue.

Most of the ordering fields were copied from the original IKEA website:

- First Name
- Last Name
- Address
- Address 2
- Suburb
- state
- Postcode

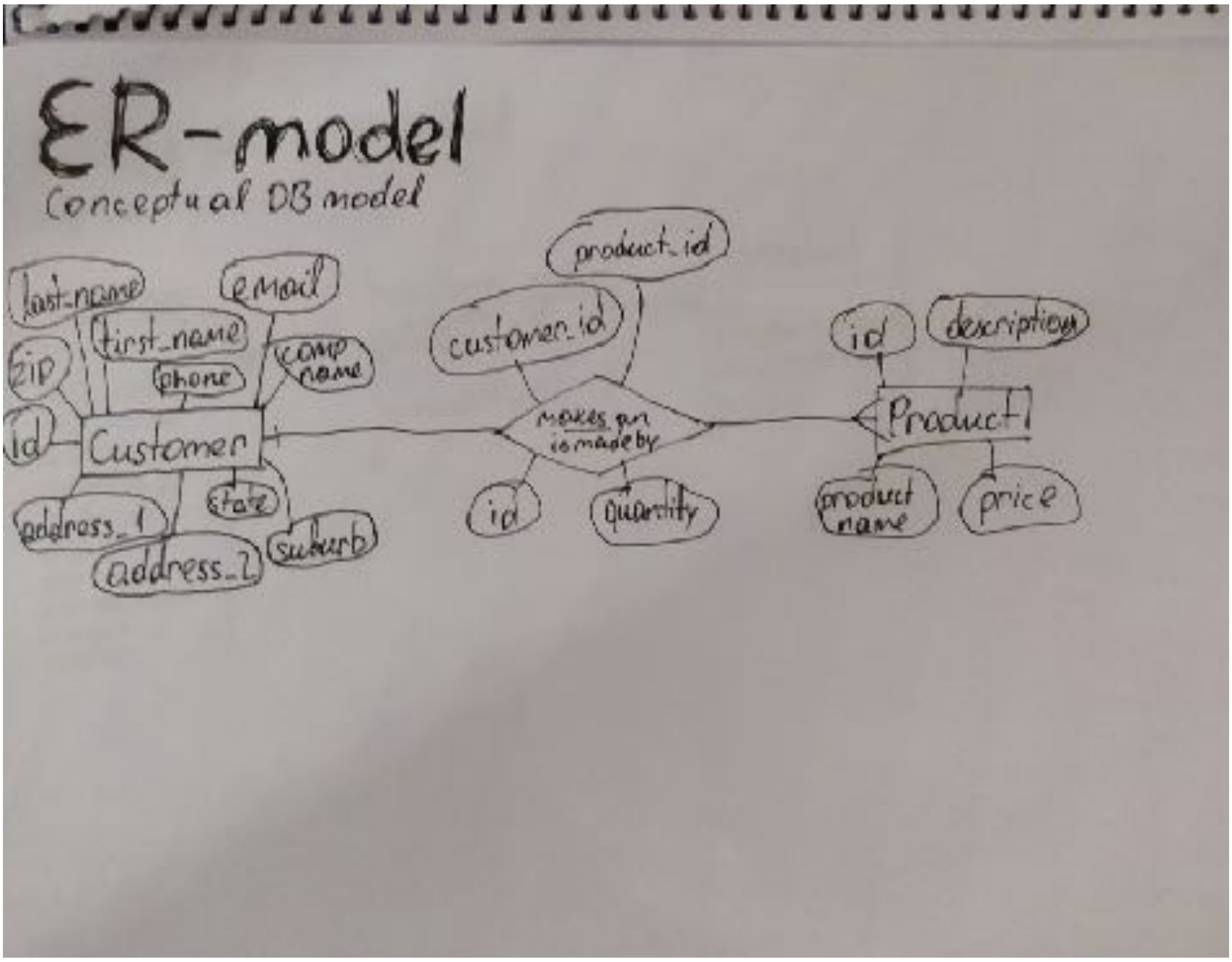
Slight alterations in adding/editing the fields were made, those include

- email
- optional company name
- phone
- Manual region input.

These alterations were made to fulfil the intended concept idea and enhance the database with more accurate information about customers.

Entity Relationship Modeling:

After analyzing the requirements of Database, there is all information needed to come up with a Conceptual Database Diagram by using an Entity-Relationship Model. It is used to visualize the database and plan out relationships between entities to eliminate any mistakes while creating the real database. It consists of entities, attributes and relationship



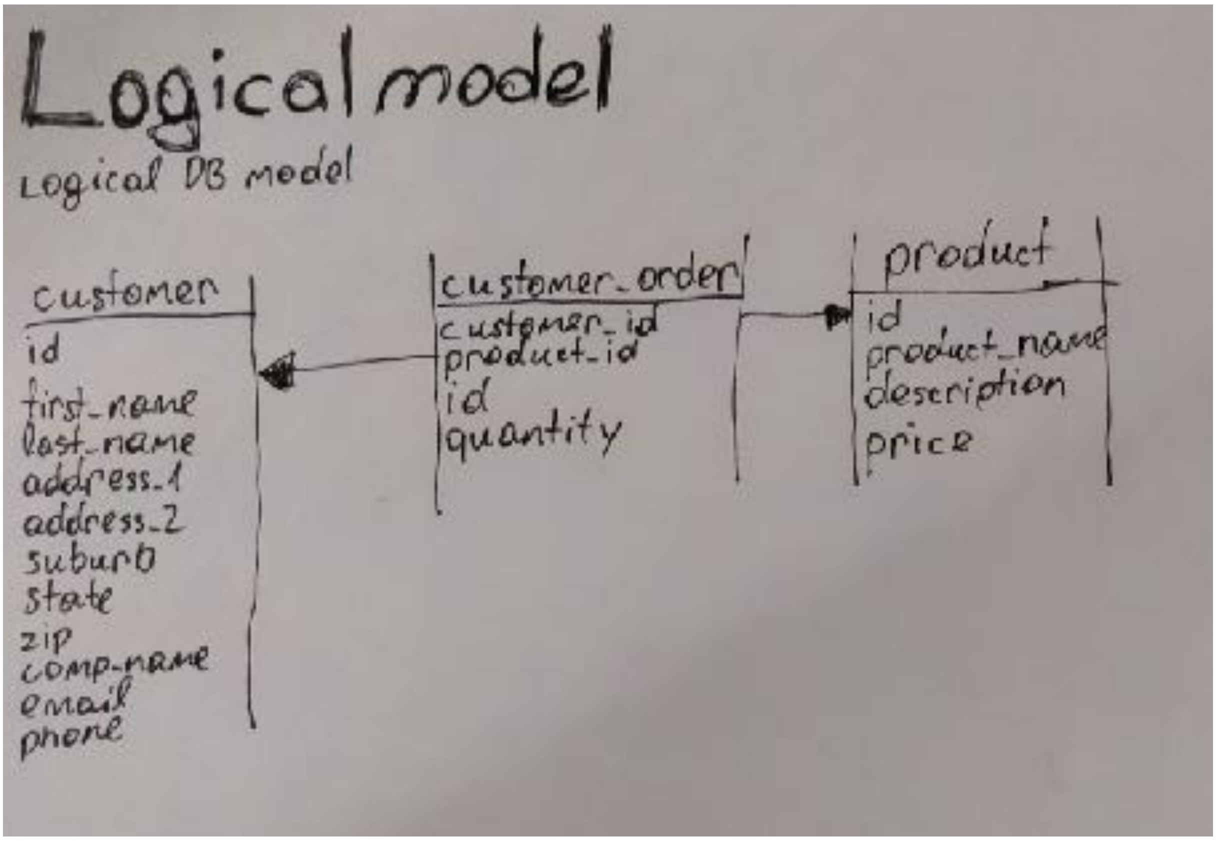
In the result of planing this entity relation model, the model is structured this way - Entity “Customer” has a one-to-many relation with Entity “Product”. Those entities are interconnected with a relation “product_order“ that has an attribute dayofpurchase and 2 attributes which are id’s of previous 2 entities.

Step 2

Logical Database Model

After the Entity Relationship Model is created, it can be converted to Logical Database Model, where I have specify primary and foreign keys and resolve cardinality, convert entities into tables with relations and attributes into column names.

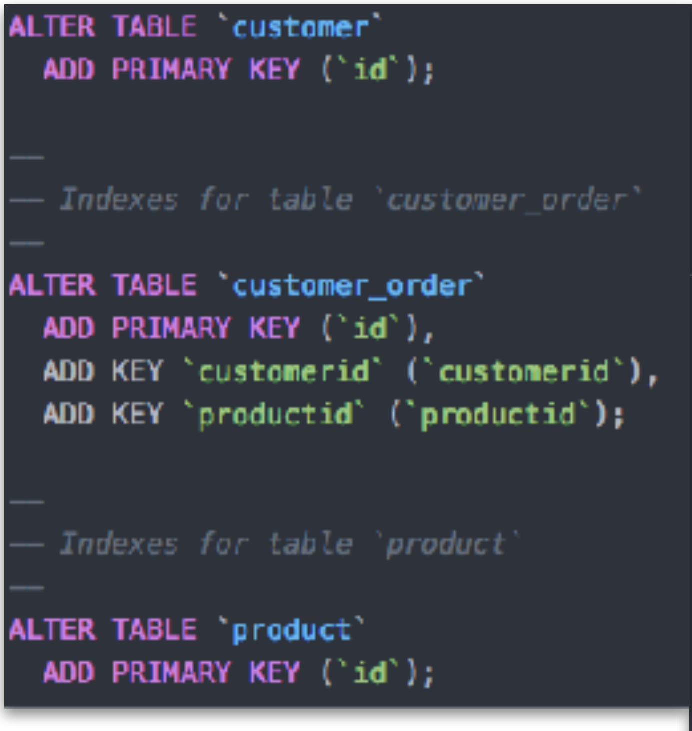
In this Logical Database Model I have aapped a one to many relation with foreign keys(customerid and productid) on the relationship “customer_order”)



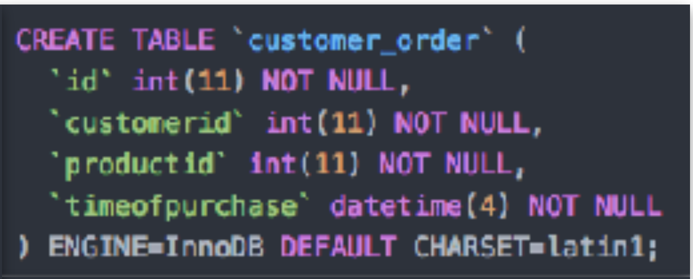
Step 3

Physical Database Implementation

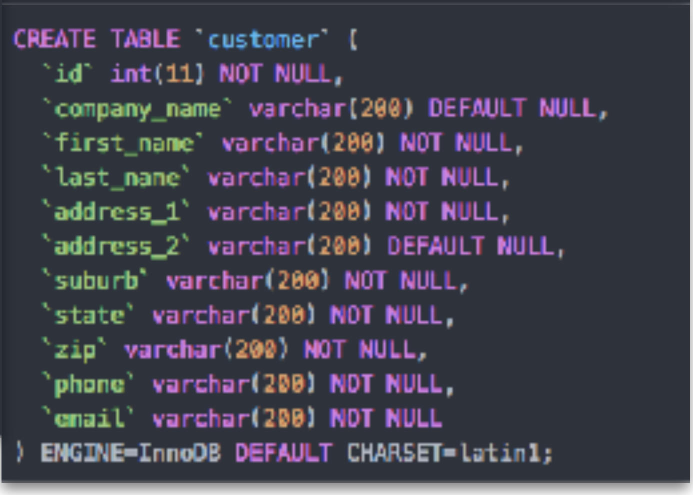
Now since the Logical Database Model is created, it is an accurate layout of a real database and it is ready to be realized in SQL.



```
ALTER TABLE `customer`
  ADD PRIMARY KEY (`id`);


--
-- Indexes for table `customer_order`
--
ALTER TABLE `customer_order`
  ADD PRIMARY KEY (`id`),
  ADD KEY `customerid` (`customerid`),
  ADD KEY `productid` (`productid`);


--
-- Indexes for table `product`
--
ALTER TABLE `product`
  ADD PRIMARY KEY (`id`);
```



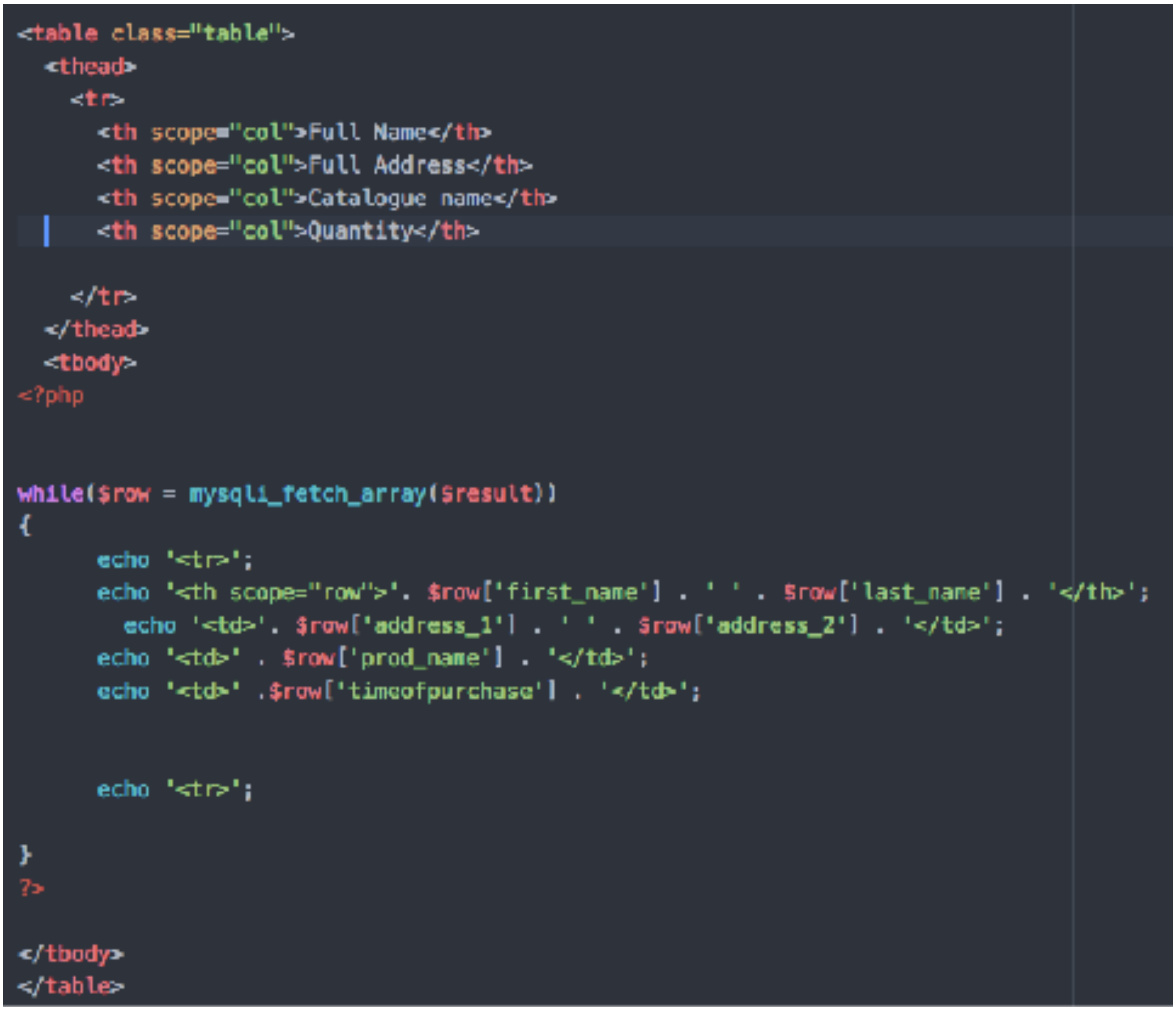
```
CREATE TABLE `customer_order` (
  `id` int(11) NOT NULL,
  `customerid` int(11) NOT NULL,
  `productid` int(11) NOT NULL,
  `timeofpurchase` datetime(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



```
CREATE TABLE `customer` (
  `id` int(11) NOT NULL,
  `company_name` varchar(200) DEFAULT NULL,
  `first_name` varchar(200) NOT NULL,
  `last_name` varchar(200) NOT NULL,
  `address_1` varchar(200) NOT NULL,
  `address_2` varchar(200) DEFAULT NULL,
  `suburb` varchar(200) NOT NULL,
  `state` varchar(200) NOT NULL,
  `zip` varchar(200) NOT NULL,
  `phone` varchar(200) NOT NULL,
  `email` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

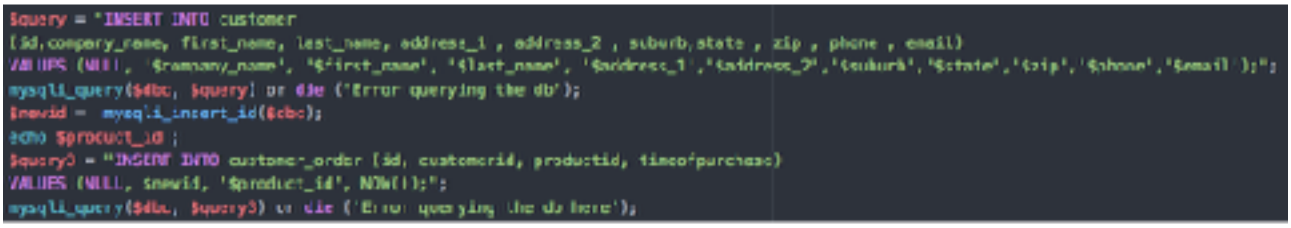
This is an example of DDL(**Data Definition Language**) which is a sublanguage of SQL used to create, alter or drop table structures. I have used DDL to create new tables(CREATE TABLE) and primary keys using PRIMARY KEY (`pk_id`) as well as adding foreign keys using ADD KEY `fk`(`fk`);

Data Control Language, also known as DCL is responsible for controling privileges in a Database. It includes commands GRANT and REVOKE and two privilege types : System and Object. It can be executed in DBMS to provide user with space for data and tables, access to use DDL language and give/take back permissions from users or system on executing commands, for example, REVOKE CREATE TABLE FROM username.**Data**

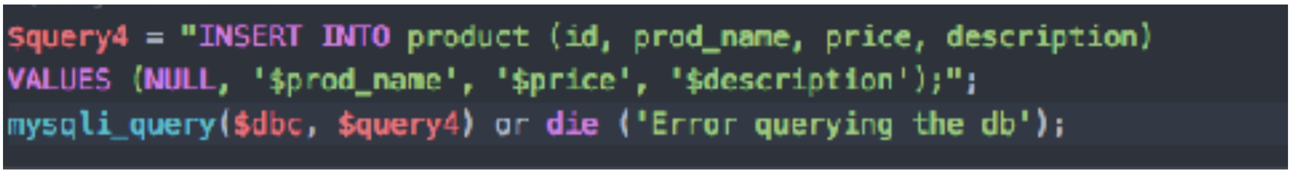


```
<table class="table">
  <thead>
    <tr>
      <th scope="col">Full Name</th>
      <th scope="col">Full Address</th>
      <th scope="col">Catalogue name</th>
      <th scope="col">Quantity</th>

    </tr>
  </thead>
  <tbody>
<?php


while($row = mysqli_fetch_array($result))
{
      echo '<tr>';
      echo '<th scope="row">'. $row['first_name'] . ' ' . $row['last_name'] . '</th>';
        echo '<td>'. $row['address_1'] . ' ' . $row['address_2'] . '</td>';
      echo '<td>' . $row['prod_name'] . '</td>';
      echo '<td>' .$row['timeofpurchase'] . '</td>';


      echo '<tr>';

}
?>

</tbody>
</table>
```

Manipulation Language, also known as DML is responsible for displaying, adding, deleting, and modifying data in a database. DML commands include SELECT, INSERT, UPDATE, DELETE. In my examples I have used DML to query out a string of values from all 3 of my tables.

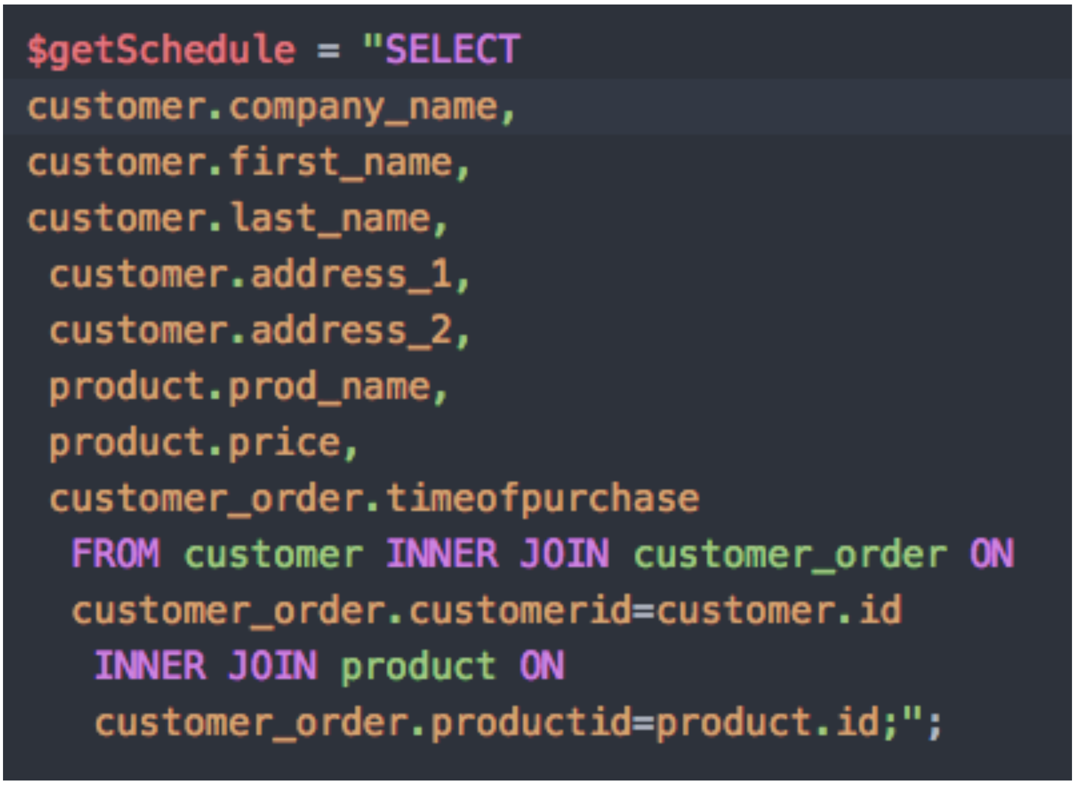


```
$query = "INSERT INTO customer
(id,company_name, first_name, last_name, address_1 , address_2 , suburb,state , zip , phone , email)
VALUES (NULL, '$company_name', '$first_name', '$last_name', '$address_1','$address_2','$suburb','$state','$zip','$phone','$email');";
mysqli_query($dbc, $query) or die ('Error querying the db');
$newid = mysqli_insert_id($dbc);
echo $product_id ;
$query3 = "INSERT INTO customer_order (id, customerid, productid, timeofpurchase)
VALUES (NULL, $newid, '$product_id', NOW());";
mysqli_query($dbc, $query3) or die ('Error querying the db here');
```



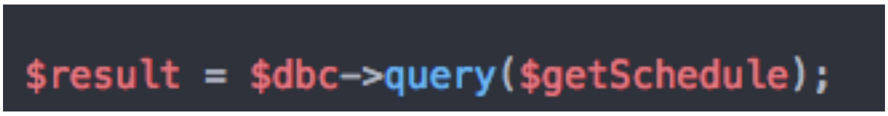
```
$query4 = "INSERT INTO product (id, prod_name, price, description)
VALUES (NULL, '$prod_name', '$price', '$description');";
mysqli_query($dbc, $query4) or die ('Error querying the db');
```

Webmaster overview table is outputted by a combination of 3 tables using **INNER JOIN**. The INNER JOIN predicate compares the rows of table 1 and table 2 to find the pairs that it is compatible with and it outputs compatilbe rows into a result row.



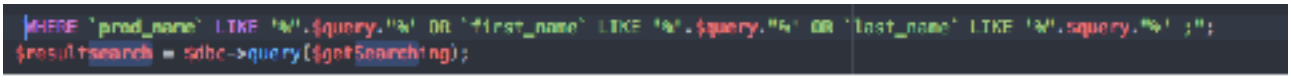
```
$getSchedule = "SELECT
customer.company_name,
customer.first_name,
customer.last_name,
 customer.address_1,
 customer.address_2,
 product.prod_name,
 product.price,
 customer_order.timeofpurchase
  FROM customer INNER JOIN customer_order ON
  customer_order.customerid=customer.id
   INNER JOIN product ON
   customer_order.productid=product.id;";
```

I am using this query in a table of overview for webmaster, inserting the rows into a while loop to output combined strings of the output.



```
$result = $dbc->query($getSchedule);
```

As an advanced SQL query I have created a search function for Webmaster to filter results by magazines or people's names. It is achieved by combining previous inner join and a new LIKE query with a request on a search form value.



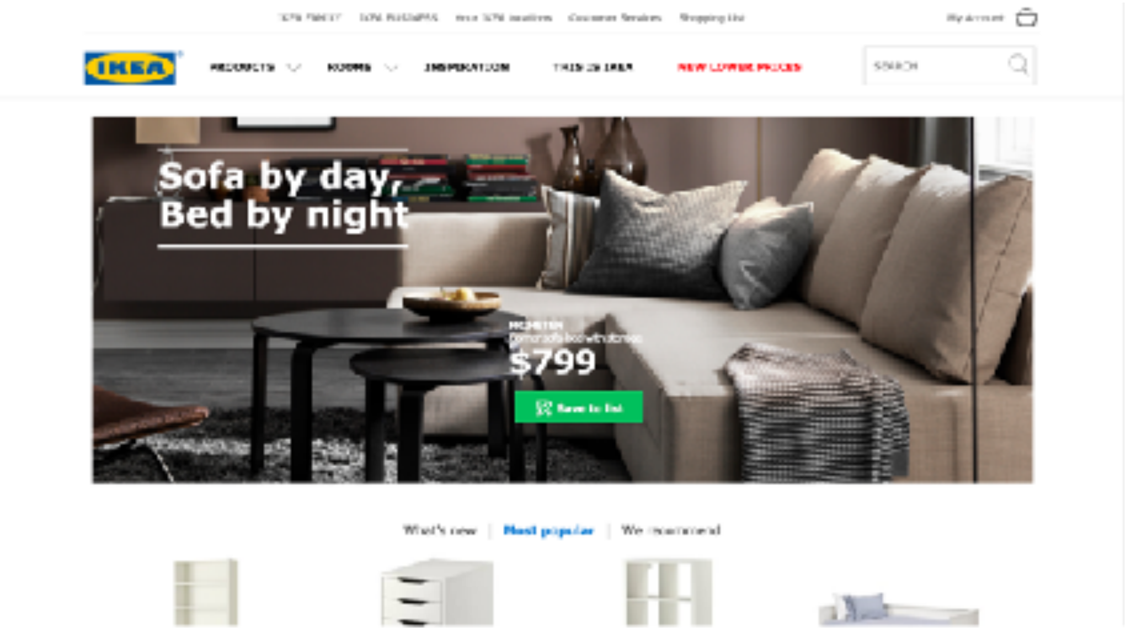
```
WHERE `prod_name` LIKE '%".$query."%' OR `first_name` LIKE '%".$query."%' OR `last_name` LIKE '%".$query."%' ;";
$resultsearch = $dbc->query($getSearching);
```

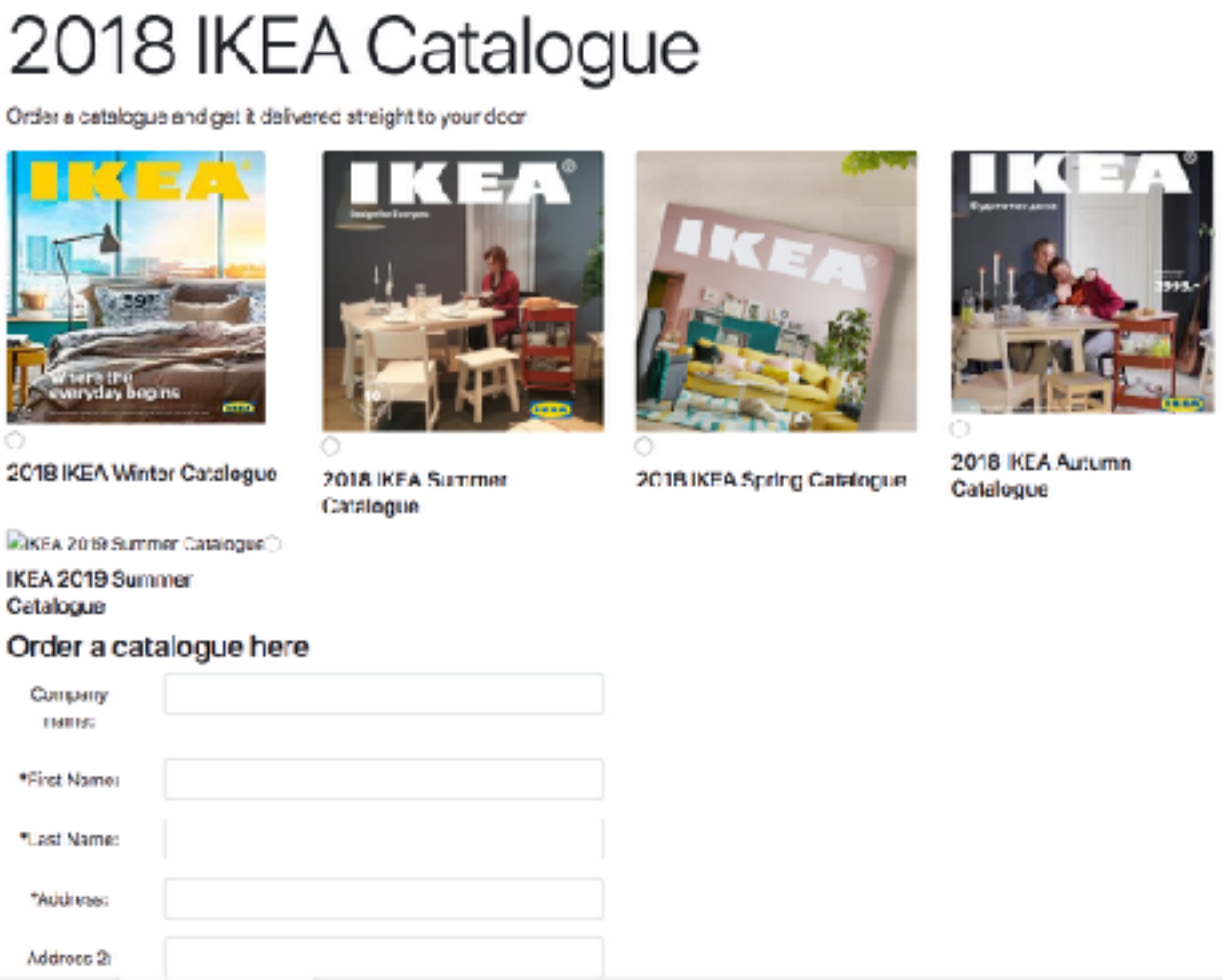
Step 4

Front-End Development

After previous step, the database is working and now as a step 4 it is ready for styling. I am using only Bootstrap 4 as a front-end development base. I have tried to match the look of original website as close as possible.

As a visual I have made the catalogue choice visual with actual magazine pictures and the forms are regular HTML forms.





2018 IKEA Catalogue

Order a catalogue and get it delivered straight to your door

○ 2018 IKEA Winter Catalogue

○ 2018 IKEA Summer Catalogue

○ 2018 IKEA Spring Catalogue

○ 2018 IKEA Autumn Catalogue

IKEA 2019 Summer Catalogue ○

IKEA 2019 Summer Catalogue

Order a catalogue here

Company name:

*First Name:

*Last Name:

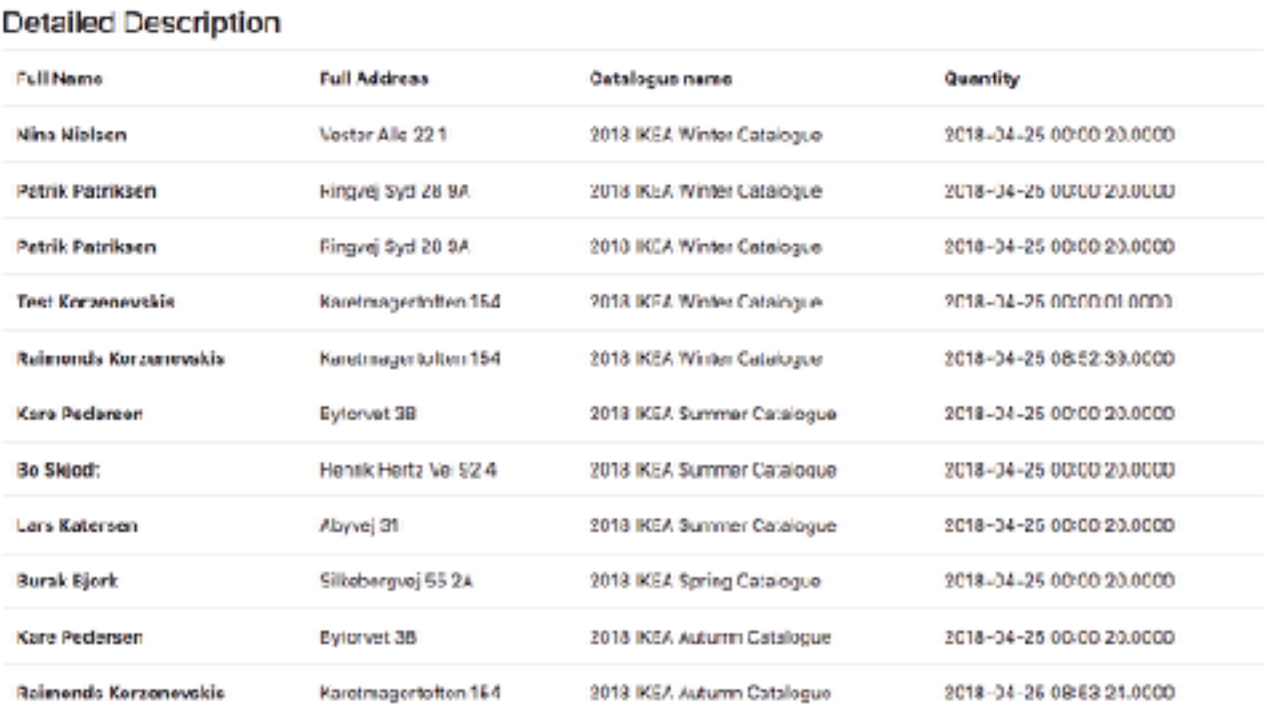
*Address:

Address 2:

Step 5

Database Deployment

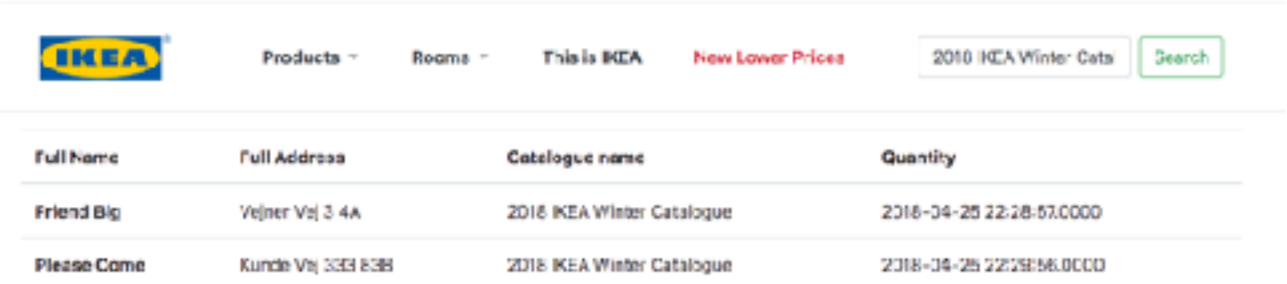
The last step in creating a database is deploying it with some dummy data to make sure it works and display the functionality in the Front-End.



Detailed Description

Full Name	Full Address	Catalogue name	Quantity
Nina Nielsen	Vester Alle 22 1	2018 IKEA Winter Catalogue	2018-04-25 00:00:20.0000
Petrik Patriksen	Ringvej Syd 28 9A	2018 IKEA Winter Catalogue	2018-04-25 00:00:20.0000
Petrik Patriksen	Ringvej Syd 28 9A	2018 IKEA Winter Catalogue	2018-04-25 00:00:20.0000
Test Korzenevskis	Karetmagertoften 154	2018 IKEA Winter Catalogue	2018-04-25 00:00:01.0000
Raimonds Korzenevskis	Karetmagertoften 154	2018 IKEA Winter Catalogue	2018-04-25 08:52:39.0000
Kare Pedersen	Eyltorvet 3B	2018 IKEA Summer Catalogue	2018-04-25 00:00:20.0000
Bo Skjodt	Henrik Hertz Vej 52 4	2018 IKEA Summer Catalogue	2018-04-25 00:00:20.0000
Lars Katersen	Abyvej 31	2018 IKEA Summer Catalogue	2018-04-25 00:00:20.0000
Burak Bjork	Silkeborgvej 55 2A	2018 IKEA Spring Catalogue	2018-04-25 00:00:20.0000
Kare Pedersen	Eyltorvet 3B	2018 IKEA Autumn Catalogue	2018-04-25 00:00:20.0000
Raimonds Korzenevskis	Karetmagertoften 154	2018 IKEA Autumn Catalogue	2018-04-26 08:53:21.0000

After making some test queries on it working, all the intended functionality including search forms, we can deploy it on the server as a functioning solution.



Full Name	Full Address	Catalogue name	Quantity
Friend Big	Vejner Vej 3 4A	2018 IKEA Winter Catalogue	2018-04-25 22:28:57.0000
Please Come	Kunde Vej 333 83B	2018 IKEA Winter Catalogue	2018-04-25 22:29:56.0000

Sources:

https://getbootstrap.com
https://github.com/muratkilic1978/contentcreator2spring2018
https://secure.ikea.com/au/en/customerservices/catalogues