Code Review 9

Good morning students,

Welcome to your 9th CodeReview.

You will need to achieve more than 60 points to pass this CodeReview successfully.

You will be graded accordingly to the results achieved. Please see the number of points for each task below.

The grading system has the following rules:

Up to 60 points = red

Between 61 and 79 = yellow

Over 80 = green

You can submit your solution **(as GitHub project link) until Saturday 18:00:00.**

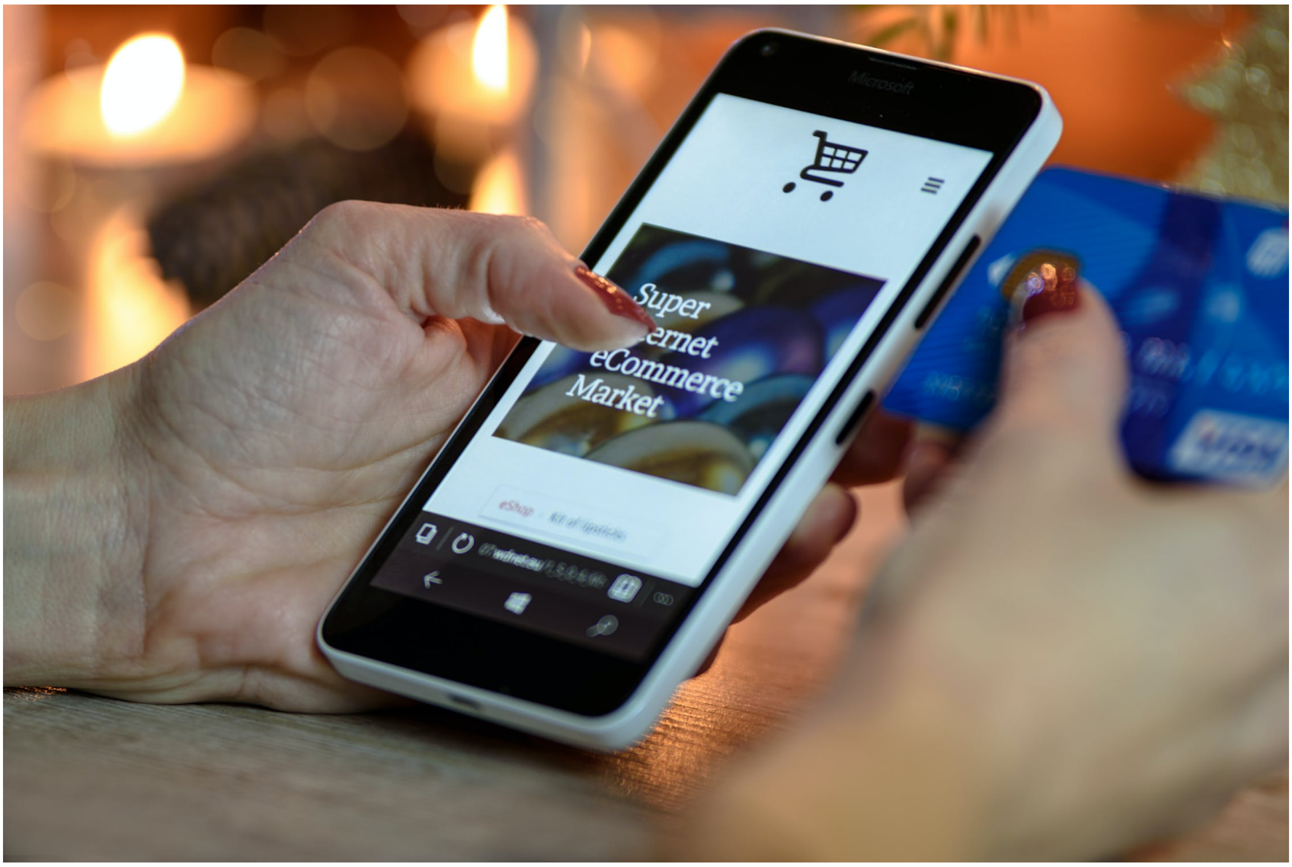
Task description: E-Commerce Site

We will create a database for the e-commerce site, Famazon.

Famazon is a site where customers can browse through and purchase products. These products are supplied by different companies. Famazon is the platform from where these companies can sell their products, but Famazon itself does not produce any of the products sold on its site.

In order for a customer to purchase a product, they will need to create an account, containing their personal data (e.g. name, address, username, email). They would also have to supply a form of payment (e.g. creditcard, paypal etc.). Each payment will have its own invoice.

Once the payment has been successful, the product will need to be shipped off to the customer’s provided address. There might be one company used to ship off products or several companies used for this.



For this CodeReview you are expected to ensure that your database is in 3rd Normal Form. All elements in the task description must be covered in your database design. (hint: use draw.io/app.diagrams tool, but try to draw the purchasing process first on paper before creating any diagrams in digital form). The brainstorming process is vital in order to design a correct Database. Please save your diagram(s) as pdf or drawio files. There should be one folder for the diagrams with 4 different pdf files or one drawio file with all 4 diagrams. When choosing pdf files, the diagrams should occupy only one page each. There should be another folder for the database file and this folder should have **2 different SQL files** (the **DB file** and the **queries file**). Feel free to add a README file if you consider it necessary, where you can provide the thought process behind your database construction.

The following criteria will be graded:

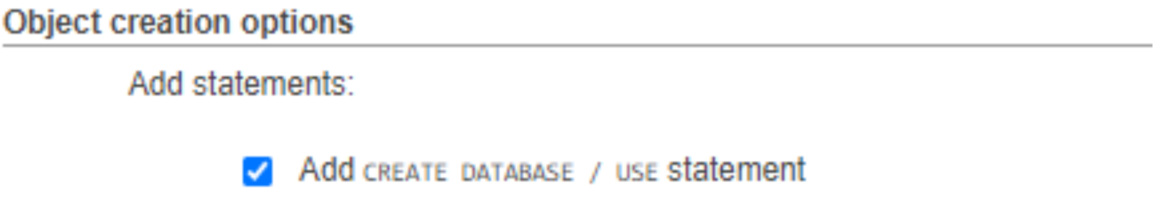
Basic points:

1. **(05 points)** Create a GitHub Repository named: BE16_CR9_Name. Set it to private. Collaborate with **codefactorygit**. Push the files into it and send the link through

the learning management system (lms). See an example of a GitHub link below:

https://github.com/JohnDoe/repositoryname.git.

2. **(25 points)** Use the 4 steps for a database development process:
    1. Define the purpose - **Cloud diagram**
    2. The Entity model - **ER diagram**
    3. The Unified Modeling Language - **Classes Diagram**
    4. **Refine and Normalize** the **Design**
3. **(40 points)** Recreate your database design in MySQL. Name your database as BE16_CR9_famazon_johndoe (use your name instead of "johndoe"). The name must be as the task requests. Export your database as BE16_CR9_famazon_johndoe.sql and then upload the .sql file with the project to GitHub. Please remember to use the option “**add create database**” in the custom options.



Object creation options

Add statements:

☑ Add CREATE DATABASE / USE statement

4. **(10 points)** Insert test data into the database (insert some dummy data in your database, at least 3 sets of data per table)

5. **(20 points)** Create at least **6 different SQL queries** to show the power of your database (save these queries to a BE16_CR9_famazon_johndoe_queries.sql file and upload it to the project) Note: The use of **JOINS** is required for at least **three queries**). Examples of queries:

- how many products were bought from a specific company
- who purchased products on this date
- what products were sent between this and that date,

- how many products were sent to a specific city etc.

Bonus points:

1. **(10 points)** Expand initial specification with **more** than 10 tables
2. **(10 points)** Create at least 2 extra JOINS in your SQL queries that connect more than 3 tables*

*the queries with JOINS made previously on the normal requirement will not be accepted in this task, you need to create new queries.