INF115-Assignment-1

Deadline 24.02.2023 at 18:00

Contents

1 About the assignment

1.1 Assignment Files

The assignment contains the questions file, this file, a file for creating tables and inserting data into them that you will work with them, and a file containing data for the table `Ratings`.

1.2 Generator File

The file, called `assignment_1.sql`, includes queries for creating a database, called `assignment_1`, and queries to create and insert data into tables `Users`, `Products`, `Orders`, and `OrdersProducts`. You should run `assignment_1.sql` on your SQL server.

1.3 Submission Files

You should submit a zip file containing the four following files:

1. `Task1.sql`: All queries asked in Task 1.
2. `Task2.sql`: All queries asked in Task 2.
3. `Task3.sql`: All queries asked in Task 3.
4. `Task4.sql`: All queries asked in Task 4.

There is a file called `TemplateForAnswers.sql`, and you can use that to write your answers.

1.4 Grading

For each question, you should write a query or several queries. Task 1 and Task 4 each have 20 points and Task 2 and Task 3 have 25 points for each. The `.zip` file, which should contain four SQL files, has 10 points. All queries should run without any errors.

Late submissions will be penalized with a 10% point deduction for each late day.

1.5 Tips

Are you struggling with the tasks?

- Stay Calm!
- Recheck your notes and the slides.
- Go to your group sessions and ask your questions there.
- Ask on Discord server.
- Search the internet as all programmers do.

For each query you are writing, test on phpMyAdmin or MySQL workbench to check that the result is the same as the one asked in the question.

2 Introduction

The University of Bergen is launching an online-shop startup called Uibay, to compete with big companies in this business [a]. For the first phase, the university plans to run the startup with limited options for products and services. At UiB, there is a database course called INF115 which you are now enrolling in. To cut down on initial costs, the managers plan to give the students some database work. Their tasks are listed below, but first, let's start with some details about the Uibay database.

[a] This is just a scenario for the assignment, and at the moment, the university does not have the intention to launch the startup.

2.1 Tables

The Uibay database has five tables: Users, Products, Orders, OrdersProducts, and Ratings. Below, you can find out more about the structure of the tables.

2.1.1 Users

- `used_id`: a unique integer assigned to each registered user, primary key.
- `name`: user's name.
- `email`: user's email address, unique.
- `username`: username, unique.
- `password`: a hash of a user's password.
- `registered_date`: registration date.

2.1.2 Products

- `product_id`: a unique integer assigned to each product, primary key.
- `brand`: brand of products.
- `category`: category of products.
- `model`: model of products.
- `price`: price of products.
- `quantity`: an integer showing how many of a product exists in the warehouse.

The `brand` stores the brand of products, for instance, Apple, Samsung, etc. The `model` is for the model of products. For example, if it is an iPad, then we have *"iPad Pro"*.

2.1.3 Orders

- `order_id`: a unique integer assigned to each order, primary key.
- `user_id`: a user ID of the user that placed the order, foreign key.
- `registered_date`: date of orders.

This table only stores the `order_id` and `user_id`, and the order details can be found in the `OrdersProducts` table, and information about users can be extracted from the Users table.

2.1.4 OrdersProducts

- `order_id`: an integer for order ID, primary key, foreign key.
- `product_id`: an integer for product ID, primary key, foreign key.
- `quantity`: an integer greater than 0.
- `price`: price of a sold product.

This table may have rows with the same `order_id` since each order can have several items. But there are no rows with the same pair of (`order_id`, `product_id`). The `price` is the price of one item. For example, if somebody bought three "iPad 10,2" (2021) 64 GB WiFi (stellargrå)", in the `price`, the price of one of them is written since it is obvious that the total price is three times of the price in `price`.

2.1.5 Ratings

- `user_id`: the ID of a user, primary key, foreign key.
- `product_id`: the product ID, primary key, foreign key.
- `rate`: an integer between 0 and 5.
- `comment`: a string of size at most 1000 characters about a product.

3 Tasks

In each of the following questions, you are asked to write a query or several queries and use all commands in the queries that are taught in the course.

3.1 Task 0: Creating Given Tables.

(0 points) Run the queries that are provided for you.

3.2 Task 1: Creating and modifying tables.

(20 points)

a) Write a query to create the table called Ratings. Pay attention to the restrictions.

b) Write queries to insert the values from `Ratings_data.sql`.

c) Increase the price of tablets in the Products table by 20%.

d) Decreases the number of smartphones in stock by one, and for those that are out of stock, it does not change.

3.3 Task 2: Single table queries.

(25 points)

a) Write a query to show all registered users between '2010-03-02' and '2020-03-02'.

b) Write a query to show all products in ascending price order.

c) Write a query to show all out-of-stock products.

d) Write a query to show the total price for each `order_id`.

e) Write a query to show the average price of in-stock smartphones.

3.4 Task 3: Multiple tables queries.

(25 points)

a) Write a query to show all products with a rating strictly greater than 2.

b) Write a query to show all products sold on 2010-02-03.

c) Write a query to show all users who placed orders more than two times.

d) Write a query to display all products where there is an order containing exactly two units of the product for each of the products.

e) Write a query to show all orders placed by a user with username `justin81`.

3.5 Task 4: Advance queries.

(20 points)

a) Write a query to show all products that have not been rated.

b) Write a query to display all products with price changes since the product's last order.

c) Write a query to show the most favorite products, i.e., products with the highest ratings.

d) Write a query to determine the average daily sales of each product from "2005-04-08" to "2023-02-24". (Hint: first, find out how many of each product have been sold from "2010-04-08" to "2023-02-24", then calculate the average. Functions `DATEDIFF(date1,date2)` calculates the number of days between two dates.)