# DBAS 1007 - DDL Assignment

## Setup

In MariaDB, create a new database named **Products\_DDL**, using the root user. Ensure that all your scripts are set to explicitly use the Products\_DDL database, by using the **USE** keyword.

There is an ERD for this database at the end of this document. You will be creating four separate SQL scripts for this assignment. Make sure you comment them well, with your name, W number and a description at the top of each file. (ex. **John Doe – A1 – Part 1**).

## Part One

#### Scenario

Your development Team Lead has given you a small database project. One of your company’s clients runs a clothing store and wishes to expand their business with an online e-commerce website. They have provided a list of tables/fields they wish to track in their database. You are expected to build a SQL script that will be used to create the database structure that will be used to store product-related data.

#### Task

Create a SQL script file (named **DDL\_PartOne\_Tables.sql**) to define and build the following basic table structures. It is recommended that you open and edit the file in DataGrip. Use datatypes appropriate for the sample data, but apply no constraints at this time.

Table: **Product**

|  |  |
| --- | --- |
| Field | Sample Data |
| ProductID | 101 |
| ProductCode | GLF10034 |
| ProductName | Men’s Golf Shirt |
| Description | Short-sleeved golf shirt, 100% cotton. |
| RetailPrice | $39.99 |
| SalePrice | $31.99 |
| ProductRating | 4 |
| NumberInStock | 48 |
| IsActive | 1 |
| IsDiscontinued | 0 |

Table: **Supplier**

|  |  |
| --- | --- |
| Field | Sample Data |
| SupplierID | 45 |
| SupplierName | Wakefield Clothing & Fabrics, Ltd. |
| Address | Suite 400, 248 Oak Street, Halifax NS |
| ContactName | Robert Owens |
| ContactPhone | (902) 555-8394 |
| ContactEmail | Robert@WCF.com |
| IsActive | 1 |

Table: **ClothingCategory**

|  |  |
| --- | --- |
| Field | Sample Data |
| CategoryID | 10 |
| CategoryName | Pants |
| IsActive | 1 |

Table: **ClothingSize**

|  |  |
| --- | --- |
| Field | Sample Data |
| SizeID | 6 |
| SizeName | Extra Large |
| SizeAbbreviation | XL |
| IsActive | 1 |

Table: **Colour**

|  |  |
| --- | --- |
| Field | Sample Data |
| ColourID | 14 |
| ColourName | White |
| ColourCode | WHT |
| IsActive | 1 |

Once you believe you are finished, test that your script is complete by running it in MariaDB in the **Products\_DDL** database. A successful test will create all the table objects, without any unexpected errors. Once your script works, save it as **A1\_DDL\_PartOne\_Tables.sql**.

## Part Two

#### Scenario

The client has sent you a list of business rules that will be used to validate data in the database.

#### Task

Make a copy of your PartOne script, and call it A1\_**DDL\_PartTwo\_TableAndConstraints.sql**. This should give you two identical SQL scripts.

In the second SQL script, make the appropriate changes to your original PartOne script to incorporate the changes listed below entirely within the CREATE TABLE statements. **Note:** The constraints can be done at either the table or column level.

Table: **Product**

|  |  |
| --- | --- |
| Field(s) | Modification |
| ProductID | Mark as key identifier field for the table, with auto-numbering |
| ProductName, ProductCode, RetailPrice, ProductRating, NumberInStock, IsActive, IsDiscontinued | Mandatory fields |
| ProductCode | Cannot have same code as any other product in the table |
| RetailPrice, SalePrice | Values must be greater than zero |
| ProductRating | Can only be a rating value between 1-5 |
| NumberInStock | Initial value of zero for new records |
| IsActive | New records should automatically be set as active |
| IsDiscontinued | New records should not be marked as discontinued |

Table: **Supplier**

|  |  |
| --- | --- |
| Field(s) | Modification |
| SupplierID | Mark as key identifier field for the table, with auto-numbering |
| SupplierName, Address, IsActive | Mandatory fields |
| IsActive | New supplier records should automatically be set as active |

Table: **ClothingCategory**

|  |  |
| --- | --- |
| Field(s) | Modification |
| CategoryID | Mark as key identifier field for the table, with auto-numbering |
| CategoryName, IsActive | Mandatory field |
| IsActive | New records should automatically be set as active |

Table: **ClothingSize**

|  |  |
| --- | --- |
| Field(s) | Modification |
| SizeID | Mark as key identifier field for the table, with auto-numbering |
| SizeName, SizeAbbreviation, IsActive | Mandatory fields |
| SizeAbbreviation | Must be one of the following values: XS, S, M, L, XL, XXL |
| IsActive | New records should automatically be set as active |

Table: **Colour**

|  |  |
| --- | --- |
| Field(s) | Modification |
| ColourID | Mark as key identifier field for the table, with auto-numbering |
| ColourName, ColourCode, IsActive | Mandatory field |
| ColourCode | Cannot have same code as any other colour in the table |
| IsActive | New records should automatically be set as active |

Once you believe you are finished, test that your script is complete by running it in DataGrip in the **Products\_DDL** database. A successful test will create all the table objects, with constraints, without any unexpected errors. Once your script works, save it as **A1\_DDL\_PartTwo\_ TableAndConstraints.sql**.

## Part Three

#### Scenario

You present your finished SQL script to your team lead. You are informed the client has a sizeable amount of sample data that will be loaded into your database once it is complete. However, since the sample data has not been tested or validated, it would cause too many issues to try a data load into a database with pre-existing constraints.

You are instructed to use two separate SQL scripts, one that will be used to create the basic table structure (which you already made in PartOne), the other to apply the data constraints at a later date. This will allow the sample data to be loaded and checked by the QA team before applying the database constraints.

#### Task

Re-create the constraints listed in Part Two in a separate script file, named **DDL\_PartThree\_ConstraintsOnly.sql**. It should not attempt to recreate the database. Also, this script should contain no CREATE TABLE or DROP TABLE statements, only ALTER TABLE statements to add the constraints to a pre-existing set of database tables. Make sure that all constraints have a defined name.

Once you believe you are finished, test that BOTH your scripts (Parts One & Three) are complete by running them in DataGrip in the **Products\_DDL** database. A successful test will create all the table objects from the PartOne script, and then apply the constraints using the PartThree script, without any unexpected errors.

## Part Four

#### Scenario

Now that the basic database scripts are tested and working, your Team Lead instructs you to finalize the database design and make any modifications to the scripts that are required to make this a working relational database.

#### Task

Analyze the existing table structure and determine the relationships between the table entities. You will then add the foreign key fields to the existing tables and create any additional derived table entities required to make this a complete relational database design.

Work for Part Four should be done in a new SQL script named **DDL\_PartFour\_FKs.sql**. This script should:

* Use ALTER TABLE statements to add any foreign key fields to existing tables,
* Use CREATE TABLE statements to create any required junction tables,
* Add FOREIGN KEY constraints to applicable fields to create the database relationships.

**Please Note:** In some situations, the order in which SQL statements are run is mandated by the design. Keep this in mind as you build your script.

## Final Outcome

Once all tasks are complete, you should have four SQL scripts that can be used to create a full, working relational database structure for the scenario solution. If done correctly, you should be able to create the full database by running the applicable scripts in the proper order.

Products Database ERD











