**CST-236 Activity 3 Guide**

Contents

[Part 1: Database Design 2](#_Toc530474453)

[Part 2: MySQL Imports 2](#_Toc530474454)

[Part 3: Inner Joins 4](#_Toc530474455)

[Part 4: Table Indexing 5](#_Toc530474456)

[Activity 3 Overall Submissions 7](#_Toc530474457)

Part 1: Database Design

**Overview**

In this activity, students will create an ER Diagram using MySQL Workbench and also create a Data Dictionary using a XLS Template. Reference "How To - Create an ER Diagram in MySQL Workbench," located within the Required Readings, as needed.

**Execution**

Execute this activity according to the following guidelines:

1. Open MySQL Workbench, connect to your local MySQL database, and make the schema you are using for Topic 1 CLC Assignment active.
2. Select the Database->Reverse Engineer menu items.
3. Select your local MySQL database from the Stored Connection dropdown. Click the Continue button. Workbench will connect to the database to retrieve all the metadata (schemas, etc.). Click the Continue button.
4. Select the Milestone 1 Project schema that you want to create an ER diagram for. Click the Continue button.
5. Workbench will connect to the database to retrieve all the metadata (table names, etc.). Click the Continue button.
6. Filter the desired tables (they are all selected by default) and make sure that the 'Place imported objects on diagram' is checked. Click the Execute button. Click the Continue and Close buttons to display your database design.
7. Drag the tables around on the diagram to arrange the tables as desired on your ER Diagram.
8. Select the File->Export->Export as PNG menu items to save your ER Diagram as an image file.
9. Save your Database Design for future reference by selecting the File->Save Model menu items.
10. Complete the Data Dictionary (using the provided "Data Dictionary Template," located within the Course Materials) for all tables and all columns for your database design completed for the Milestone 1 Project.

**Documentation**

All documentation will be submitted at the end of the activity to the learning management system. Ensure documentation of the following:

1. ER Diagram exported in Step 8.
2. Data Dictionary completed in Step 10.

Part 2: MySQL Imports

**Overview**

In this activity, students will bulk data from a file into a database table using MySQL Workbench.

**Execution**

Execute this activity according to the following guidelines:

1. Open MySQL Workbench and connect to your local MySQL database.
2. Create a new schema named *ica8*. Make this schema active by double clicking on this schema name. Expand the schema to display the Tables, Views, Stored Procedures, and Functions.
3. Create a Customer Table:
   1. Right click on the Table and select the Create Table context menu item.
   2. Name the table *Customer*.
   3. Add the following columns to the *Customer* table.
      1. ID of type INT with AI option
      2. FIRST\_NAME of type VARCHAR(50) with a NN option
      3. LAST\_NAME of type VARCHAR(50) with a NN option
      4. AGE of type INT with a NN option
   4. Click the Apply button and execute the SQL statement to create the table.
4. Create an import data file:
   1. Using any standard text editor create a new file called *customers.csv*.
   2. Add a list of at least 10 customer records into the import data file specifying the customer's first name, last name, and age. Each record should be terminated with a carriage return/line feed and each of the customer fields should be separated with a pipe (|) character.
5. Import the data file:
   1. Right click on the *Customer* table and select the Table Data Import Wizard context menu item. This will display the Data Import Wizard.
   2. Browse to the *customer.csv* file. Click the next button.
   3. Select the ica8.customer table. Click the next button.
   4. Map the data file fields to the proper table columns. Click the next button.
   5. Click the next button and validate that the import was successful.
   6. Click the next button to finish the wizard.
   7. Click the finish button.
6. Validate data import:
   1. Right click on the *Customer* table and select the Select Rows – Limit 1000 context menu item. This will display the Data Import Wizard.
   2. Take a screenshot of the data imported (should be at least 10 rows).
   3. Create an ER Diagram for this schema (see instructions from In-Class Activity #7). Export the ER Diagram as a PNG image file.
   4. Complete a Data Dictionary (using the provided Data Dictionary XLS Template) for this schema.
7. Export the data:
   1. Right click on the *Customer* table and select the Table Data Export Wizard context menu item. This will display the Data Export Wizard.
   2. Select all columns to export. Click the next button.
   3. Select a file named *export.csv*, select CSV type, LF line separator, and semi-colon (;) as a field separator. Click the next button.
   4. Click the next button and validate the export was successful.
   5. Click the next button.
   6. Click the finish button.
   7. Open the data export file and take a screenshot of the file contents.
   8. Repeat steps 7a thru 7f except select a file *export.json* and JSON type.
   9. Open the data export file and take a screenshot of the file contents.

**Documentation**

All documentation will be submitted at the end of the activity to the learning management system. Ensure documentation of the following:

1. Screenshot of the data imported from Step 6b
2. ER Diagram exported in Step 6c
3. Data Dictionary completed in Step 6d
4. Screenshot of the data exported from Step 7g
5. Screenshot of the data exported from Step 7i

## Part 3: Inner Joins

**Overview**

In this activity, students will learn how to use SQL Inner Joins to query for data across two tables. Students will create a table and bulk load data into the table, then write SQL Inner Joins in MySQL Workbench.

**Execution**

Execute this activity according to the following guidelines:

1. Initialize database:
   1. Using MySQL Workbench, create a database schema named *ica18*.
   2. Create a table *users* that has 5 columns:
      1. ID of type int that is auto-incremented.
      2. FIRST\_NAME of type varchar(100)
      3. LAST\_NAME of type varchar(100)
      4. USERNAME of type varchar(100)
      5. PASSWORD of type varchar(100)
   3. Create a data load text file with 100 users and use MySQL Workbench to bulk load the users into the *users* table.
   4. Create a table *address* that has 3 columns:
      1. ID of type int that is auto-incremented.
      2. USER\_ID of type int that is a FK constraint to the PK ID column of *users* table
      3. ADDRESS of type varchar(100)
   5. Create a data load text file with an address for each of the 100 users and use MySQL Workbench to bulk load the users into the *address* table.
2. Write INNER JOINS MySQL Workbench:
   1. Write SQL Inner Joins and test PK/FK constraint using the SQL statements provided in the course materials.
   2. Take a screenshot of your users displayed in the table for desired search results and when no users are found from Step 2a.

**Documentation**

All documentation will be submitted at the end of the activity to the learning management system. Ensure documentation of the following:

1. Upload a screenshot of the SQL queries from Step 2a.

## Part 4: Table Indexing

**Overview**

In this activity, students will learn how run a MySQL Explain Plan and add an Index to a desired column in a table. Students will use the customers table from a previous activity and MySQL Workbench.

**Execution**

Execute this activity according to the following guidelines:

1. Prepare your Database Schema and Tables:
2. Refer to Activity 1 Part 8 and Activity 2 Part 1 to create a new schema, create a *Customers* table, and bulk load some data into the *Customers* table.
3. Run an Explain Plan on the *Customer* tables (NOTE, you will need to provide a query that does not return every row in the table):
4. Run the following query to execute an Explain Plan:

EXPLAIN SELECT \* FROM CUSTOMER WHERE AGE > 20.

1. Take a screenshot of the Results Grid and Query Stats.
2. Type the following query to execute in the SQL Editor and select the Query->Explain Current Statement menu options to display a Visual Explain Plan (NOTE, you will need to provide a query that does not return every row in the table):

SELECT \* FROM CUSTOMER WHERE AGE > 20.

1. Take a screenshot of the Visual Query Plan.
2. Add an Index to the Age column:
3. Right click on the Age column and select the Create Index menu option. Click the Create button to create the Index.
4. Run an Explain Plan on the *Customer* tables:
5. Run the following query to execute an Explain Plan (NOTE, you will need to provide a query that does not return every row in the table):

EXPLAIN SELECT \* FROM CUSTOMER WHERE AGE > 20.

1. Take a screenshot of the Results Grid and Query Stats.
2. Type the following query to execute in the SQL Editor and select the Query->Explain Current Statement menu options to display a Visual Explain Plan (NOTE, you will need to provide a query that does not return every row in the table):

SELECT \* FROM CUSTOMER WHERE AGE > 20.

1. Take a screen hot of the Visual Query Plan.
2. Run misc. statements:
3. Display Indexes: SHOW INDEX FROM Customer;
4. Take a screenshot of output.
5. Rebuild Indexes: ANALYZE TABLE Customer;
6. Take a screenshot of output.

**Documentation**

All documentation will be submitted at the end of the activity to the learning management system. Ensure documentation of the following:

1. Screenshot of the Results Grid and Query Stats with no Indexes
2. Screenshot of the Results Grid and Query Stats with Indexes
3. Screenshot of the Display Indexes output
4. Screenshot of the Rebuild Indexes output
5. 50-100 words explaining how the Explain Plan tools (both using EXPLAIN keyword and Visual Query Plan can help you find if indexes are required and being used for a particular SQL query

Activity 3 Overall Submissions

Submit the following to the learning management system:

**Part 1: Database Design**

ER Diagram exported in Step 8

Data Dictionary completed in Step 10

**Part 2: MySQL Imports**

Screenshot of the data imported from Step 6b

ER Diagram exported in Step 6c.

Data Dictionary completed in Step 6d

Screenshot of the data exported from Step 7g

Screenshot of the data exported from Step 7i

**Part 3: Inner Joins**

Screenshot of the SQL queries from Step 2a

**Part 4: Table Indexing**

Screenshot of the Results Grid and Query Stats with no Indexes

Screenshot of the Results Grid and Query Stats with Indexes

Screenshot of the Display Indexes output

Screenshot of the Rebuild Indexes output

50-100 words explaining how the Explain Plan tools (both using EXPLAIN keyword and Visual Query Plan can help you find if indexes are required and being used for a particular SQL query.