

# CSE 4020/5260 Database Systems

Spring 2023

## In-Class Activity 1: DDL

Total Points: 25

**Date Assigned:** Monday, Feb 20, 2023

**Due Date:** Friday, Feb 24, 2023

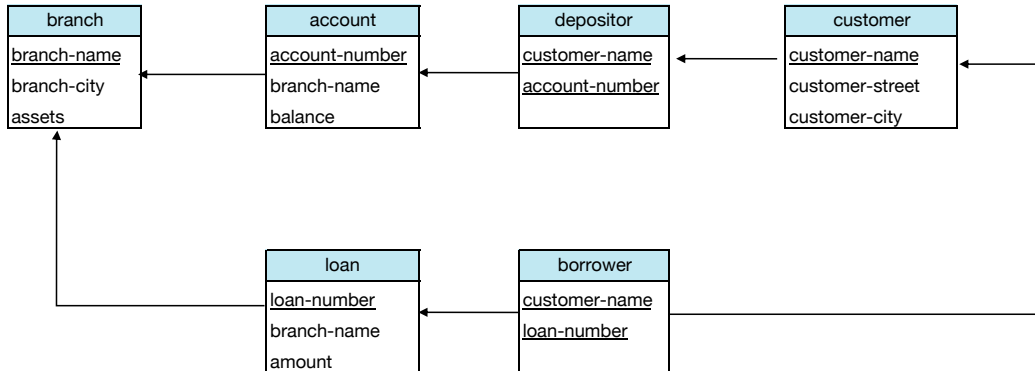
---

**Instructions:** Submit your work on Canvas as a Jupyter Notebook ipynb file named `cse4020_yourname_activity1.ipynb`. Make sure to add headings and question numbers to your notebook using Markdown. Include in your notebook the contents of your DDL file, your code, and the corresponding output.

### Key Concepts Demonstrated

- Executing DDL Statements on a database
  - Reducing an ER Diagram into tables
  - Creating a DDL File based on a set of tables
  - Enforcing integrity constraints using DDL
  - Coding in a Jupyter notebook
  - Writing program code that reads a DDL file and executes the statements
  - Issuing DDL statements on a cloud-based database (AWS RDS)

- (5 points) Provide evidence to show that AWS RDS was setup correctly.  
**Evidence 1:** A screenshot from MySQL Workbench that shows a successful connection to your endpoint.  
**Evidence 2:** Make a connection to your RDS database using code and test your connection using the statement `connection.is_connected()` (You may read your password from a file when connecting to your database.)
- (5 points) Create a DDL file based on the following schema diagram (You may use dbdiagram.io as demonstrated in class or you may create the tables manually). Add integrity constraints to your DDL that ensures that the *balance* in the *account* table and the *amount* in the *loan* table cannot be negative. See the slides on relation algebra for example data types. Add your DDL file as Markdown color-coded SQL statements to your notebook.



- (5 points) In a Jupyter Notebook, write code either in Python, Java or C that connects to your AWS RDS instance and creates each of the tables in your DDL file.
- (10 points) Verify that your tables were created by using code to issue the following command for each table in your list.

SHOW CREATE TABLE TABLE\_NAME

Given that Markdown can be used to display and syntax-highlight\* code, use Markdown to display each of the resulting DDL statement from your query. For example, the result of the command `SHOW CREATE TABLE 'classroom'` would return the following:

```
[ ('classroom',
'CREATE TABLE 'classroom' (\n
'building' varchar(15) NOT NULL,\n
'room_number' varchar(7) NOT NULL,\n
'capacity' decimal(4,0) DEFAULT NULL,\n
PRIMARY KEY ('building','room_number')\n)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci')]
```

This can be displayed in your notebook using Markdown as follows:

#### Python:

```
from IPython.display import display, Markdown
display(Markdown("""
a variable or code block containing your SQL DDL
"""))
```

#### Java:

```
String data = """
a variable or code block containing your SQL DDL
"""
render(data, "text/markdown"); //the built-in render function
```

#### Output:

```
CREATE TABLE 'classroom' (
'building' varchar(15) NOT NULL,
'room_number' varchar(7) NOT NULL,
'capacity' decimal(4,0) DEFAULT NULL,
PRIMARY KEY ('building','room_number')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

\*Read more here <https://www.markdownguide.org/extended-syntax/> to learn about syntax highlighting