Goals:

Alter a table
Create Foreign Key constraints
Practice `SELECT` with predicates
Insert into a table

0. Review in class code examples and the definitions.

1. ALTER a table

- Add a column to `ClassStudent` : **`SignUpDate datetime`**.
  - Include a constraint that defaults the value to CURRENT_TIMESTAMP.
- Add a foreign key in the `ClassStudent` table on column `ClassID` that references `Class.ID` *(as we did in class)*
- Add a foreign key in the `ClassStudent` table on `StudentID` that references `Student.ID`
  Reference: https://dev.mysql.com/doc/refman/5.6/en/create-table-foreign-keys.html

Deliverable 1: send your code

Note:

- Ensure foreign keys are created with correct columns and references
- Column SignupDate name and datatype must be correct

2. Insert data into ClassStudent

- Create one `INSERT` instruction for `ClassStudent` that inserts data into the columns `ClassID` and `StudentID` and uses a `SELECT` statement *with a predicate* to enter many result set records based on the `Student` table.
  - Try not to run the statements twice. Having duplicated data is messy.
  - The `INSERT` statement should put `Student.ID` into ClassStudent's `StudentID` column.
  - Where the Student has an "a" or "e" anywhere in `Student.FirstName`.
  - The `INSERT` statement **must** reference the class that represents the **CS-HU310** record. (Use that `Class.ID` value in the column for `ClassStudent.ClassID`) (You will want to find the ID for this class by using a separate `SELECT` statement before writing the `INSERT` instruction.)
  - The insert will ignore the `SignUpDate` column and allow the constraint to default the data.

Deliverable 2: send your code and the result set

3. Insert more data into `ClassStudent`

- Create one `INSERT` instruction for `ClassStudent` that inserts data into the columns `ClassID` and `StudentID` and uses a SELECT statement to enter many result set records.
  - The `INSERT` statement should put `Student.ID` into ClassStudent's `StudentID` column, use each `Student.ID` value where the `Student` has a `FirstName` with "o" in the name.
  - The `INSERT` statement should use the number for the ClassStudent's `ClassID` column that represents the COMM113 class. (You will want to find the ID for this class by using a separate `SELECT` statement before writing the `INSERT` instruction.)
  - Run the code to create the data

Deliverable 3: send your code

4. Insert more data into `ClassStudent`

- Create one `INSERT` instruction for `ClassStudent` that inserts data into the columns `ClassID` and `StudentID` and uses a SELECT statement to enter many result set records.
  - The `INSERT` statement should put `Student.ID` into ClassStudent's `StudentID` column, use each `Student.ID` value where the `Student` has a `FirstName` longer

than 4 characters. *Hint:* `WHERE LENGTH(FirstName) > 4 --` you will read more about this in the next homework
    - The `INSERT` statement should use the number for the ClassStudent's `ClassID` column that represents the ENG101 class. (You will want to find the ID for this class by using a separate `SELECT` statement before writing the `INSERT` instruction.)
    - Run the code to create the data

Deliverable 4: send your code

5. Write a query to return the Student first and last names combined into one column, but only include the records if the FirstName has an "h" in the name.
Deliverable 5: send your code and the result set

6. Answer these questions
Deliverable 6:

Question 1: What does the DEFAULT constraint do? Can you skip naming the column when running an insert instruction (and also skip the corresponding value) when the column has a DEFAULT constraint?

Question 2: What is a "scalar function" in sql? For each distinct set of inputs, how many values can it return?

Question 3: Which scalar function can show you the number of characters in the Class.Code? Send code to support your answer.

Question 4: Conditional operators can evaluate to 3 different results, what are they?

Question 5: What error appears when you ignore a NOT NULL column name (and value) when running an insert instruction?

7. Select the data from each of the tables: `ClassStudent`, `Class` and `Student`
Deliverable 7: send your result sets
/ place the result set into a multi-line comment */*