# CSC-430(001) / 530 : DATABASE MANAGEMENT SYSTEMS / DATABASE THEORY

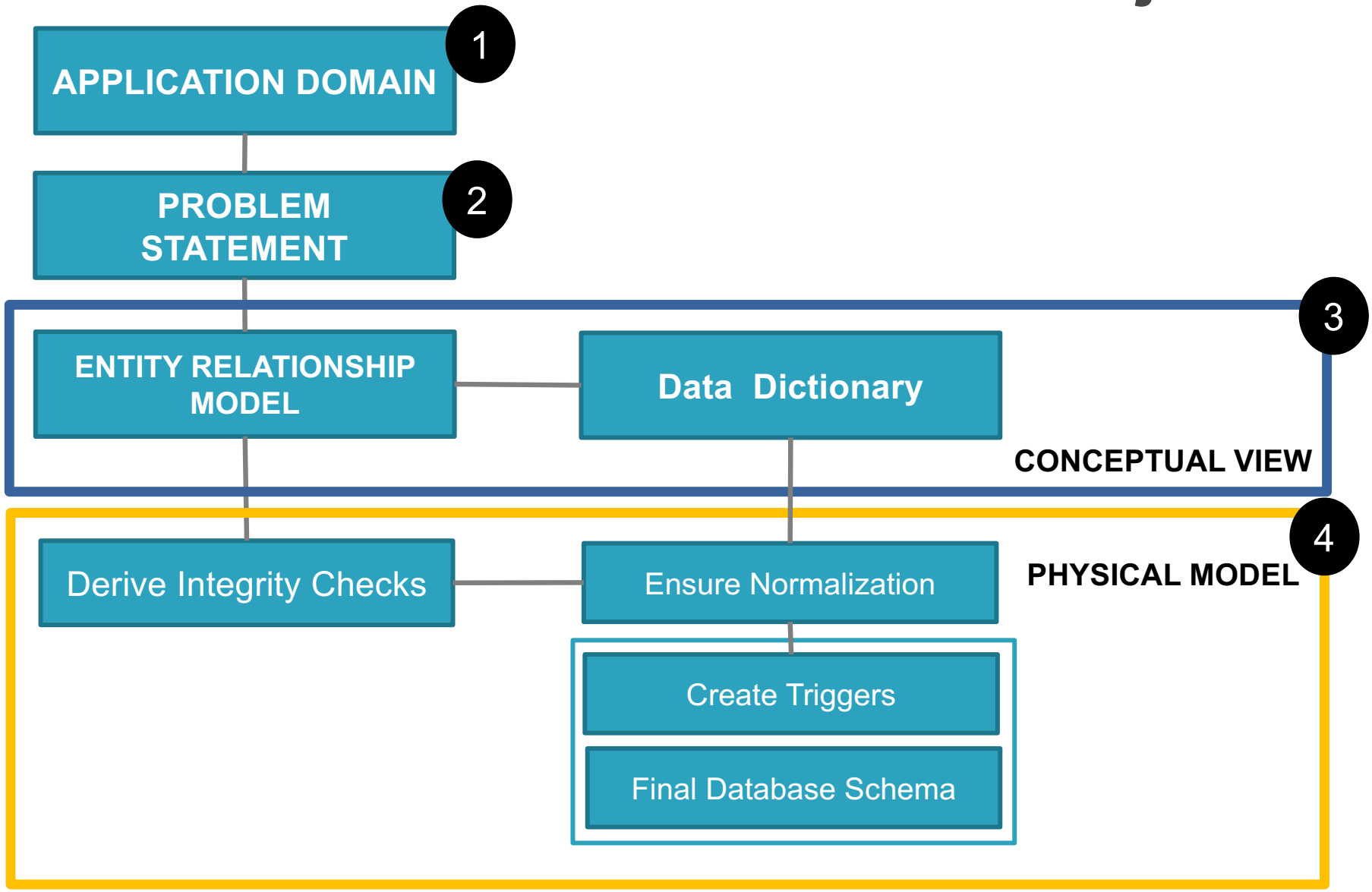
Winter 2022

Final Project Presentation - Instruction Guide

And

Example

# Overview: A Relational Database Project



# Project Proposals

- **What is expected in Final Report Submission?**
  1. **A consolidated deck of slides that captures the following:**
     - **Provide a clear description of the Application Domain (2-3 slides)**
     - Problem Statement (1 slide)
     - The ER from the proposal, and Updated ER. (1-2 slides)
  2. **Enumerate the integrity constraints (1-2 slides)**
     - Enumerate and describe the different constraints enforced (1-2 slides)
     - Do not forget to include CASCADES for Anomalous DDL Queries (in demo)
  3. **For each relation in the DB, explain which Normal Form was achieved with tables in your relationship (2 slides)**
  4. **Explain one trigger and one view built in the application ?**
     1. **In both (2-slides) and in demo**
  5. **Acknowledgement of Team by roles played by each member.**
     1. **In both (1-slide) and in demo**

# What you will submit

- One Submission per group; to include:
  - The Final PPT presentation.
  - A 5-minute recording of project (demo)
  - The Final SQL CODE to create the Database.
    - Examples for Integrity constrains (with DDL anomalies)
    - Examples of Triggers and Views.
  - A clear demarcation of roles.

- Final Date of Submission: **Feb 19th, 2022 (@5:00 pm).**

- **NO EXTENSIONS WILL BE POSSIBLE**

# DBMS FINAL PROJECT

EXAMPLE:

# Application Domain - Area of Application

Purpose: Make it easier for teachers to conduct Evaluations on Students within a school

Our database will…

- Keep track of every student in a class along with unique information about them, including their answers to an evaluation completed multiple times throughout the year
- Come with a user-friendly web-based interface for accessing and changing information
- Make it easier for teachers to conduct the evaluations and compare/analyze evaluation results by consolidating all the data in one place

# Application Domain - Challenges

This project requires a database because…

1. The high number of relationships and the complexity of those relationships between the entities
2. Multiple users (teachers, students)
3. Centralized control of data

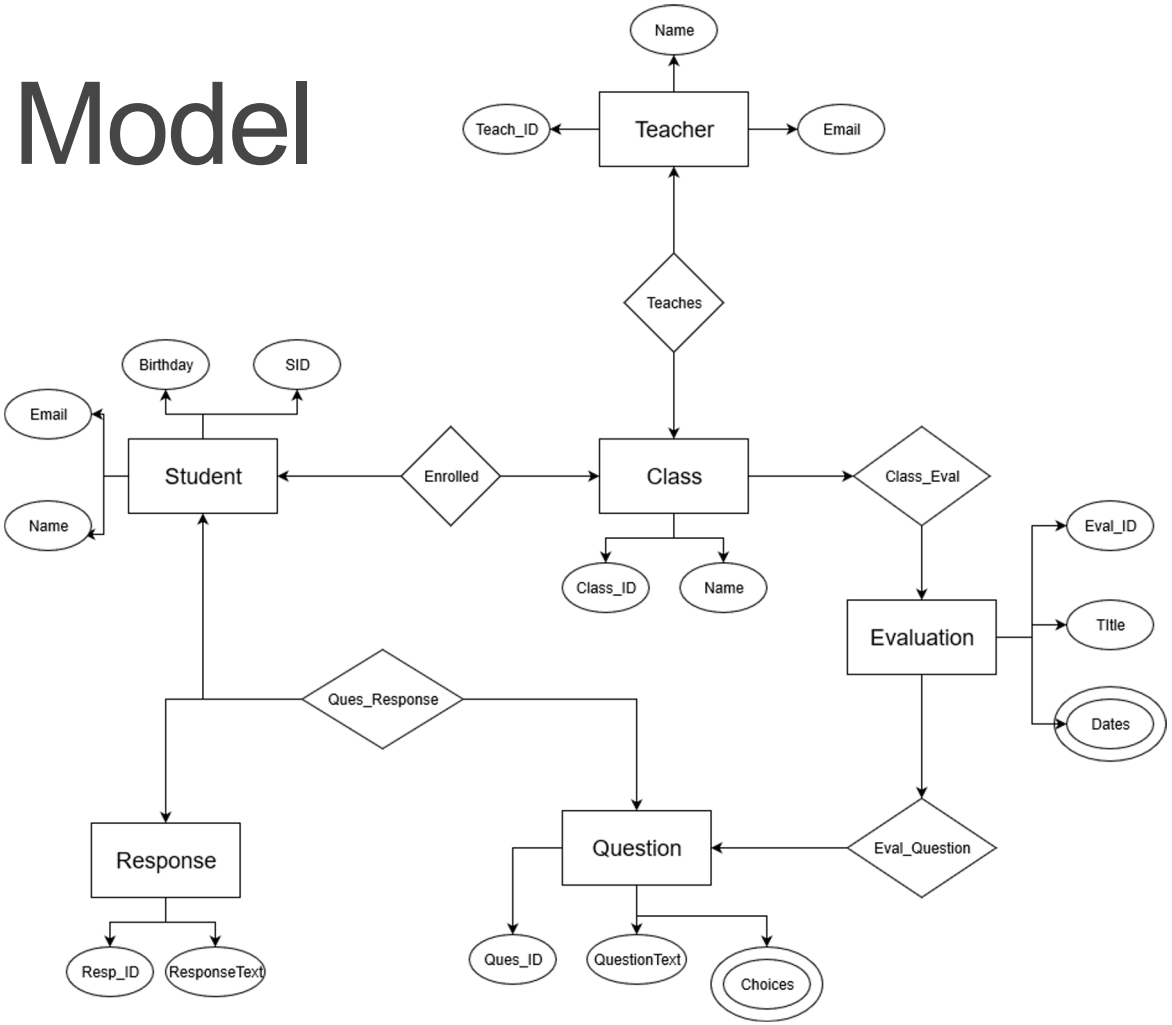
# Problem Statement

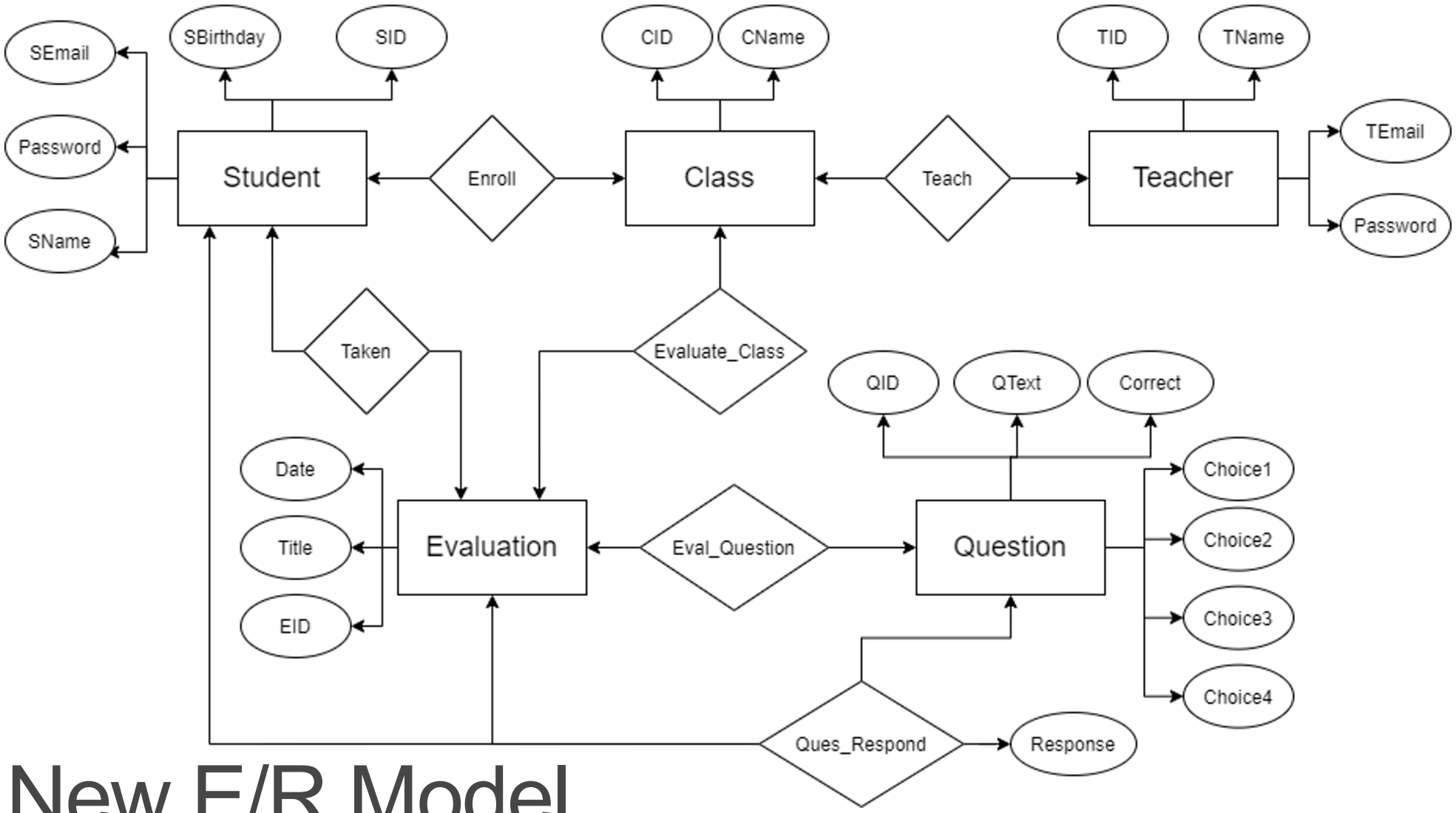
- Entities
  - Student (SID, SName, SEmail, Password, SBirthday)
  - Teacher(TID, TName, TEmail, Password)
  - Class (CID, CName)
  - Evaluation (EID, Title, Date)
  - Question(QID, QText, Correct, Choice1, Choice2, Choice3, Choice4)
- Relationships
  - Enroll (SID, CID)
  - Teach (TID, CID)
  - Evaluate_Class (EID, CID)
  - Eval_Question (EID, QID)
  - Ques_Respond (SID, QID, EID, Response)
  - Taken (SID, EID)

# Problem Statement

- Entity Relationship Constraints
  - A class will have a single teacher but a teacher can teach multiple classes
  - A student can be enrolled in many classes and a class can have many student enrolled in it
  - A class can have zero or more evaluations and an evaluation can belong to multiple classes
  - A student can have 0 or 1 response for each question
  - A student can only take an evaluation once

# Old E/R Model


Name
Teacher
Teach_ID
Email
Teaches
Birthday
SID
Email
Student
Enrolled
Class
Class_Eval
Name
Class_ID
Name
Eval_ID
Evaluation
TItle
Ques_Response
Dates
Response
Question
Eval_Question
Resp_ID
ResponseText
Ques_ID
QuestionText
Choices


SEmail
SBirthday
SID
CID
CName
TID
TName
Password
Student
Enroll
Class
Teach
Teacher
TEmail
SName
Password
Taken
Evaluate_Class
QID
QText
Correct
Date
Choice1
Title
Evaluation
Eval_Question
Question
Choice2
EID
Choice3
Choice4
Ques_Respond
Response

# New E/R Model

# Integrity Constraints

Primary Keys are  NOT NULL and UNIQUE

Relationships: PRIMARY KEY (*ID, *ID)

(*ID is FOREIGN KEY referencing entity)

# BCNF - Old Model

Our Violations:

- 1NF: multivalued attributes
  - Question.Choices
  - Evaluation.Dates

Our Solutions:

- Question.Choice1, Question.Choice2, Question.Choice3, Question.Choice4
- New Evaluation entity for each date

# BCNF - New Model

Functional Dependencies:
{SID → SName, SEmail, Password, SBirthday}
{CID → CName}
{TID → TName, TEmail, Password}
{QID → QText, Correct, Choice1, Choice2, Choice3, Choice4}
{EID → Title, Date}
{SID, QID, EID → Response}

1NF: no multivalued attributes ✅

2NF: no partial dependencies ✅

3NF: no transitivity ✅

BCNF: every determinant is a candidate key ✅

# Triggers

## Deletes all Questions and responses associated with an Evaluation if the Evaluation is deleted

```
DELIMITER $$
CREATE TRIGGER DeleteEval
BEFORE DELETE ON Evaluation
FOR EACH ROW
BEGIN
                DELETE FROM Taken WHERE EID=OLD.EID;
                DELETE FROM Evaluate_Class WHERE EID=OLD.EID;
        DELETE FROM Eval_Question WHERE EID=OLD.EID;
        DELETE FROM Ques_Respond WHERE EID=OLD.EID;
END$$
DELIMITER ;
```

# Views

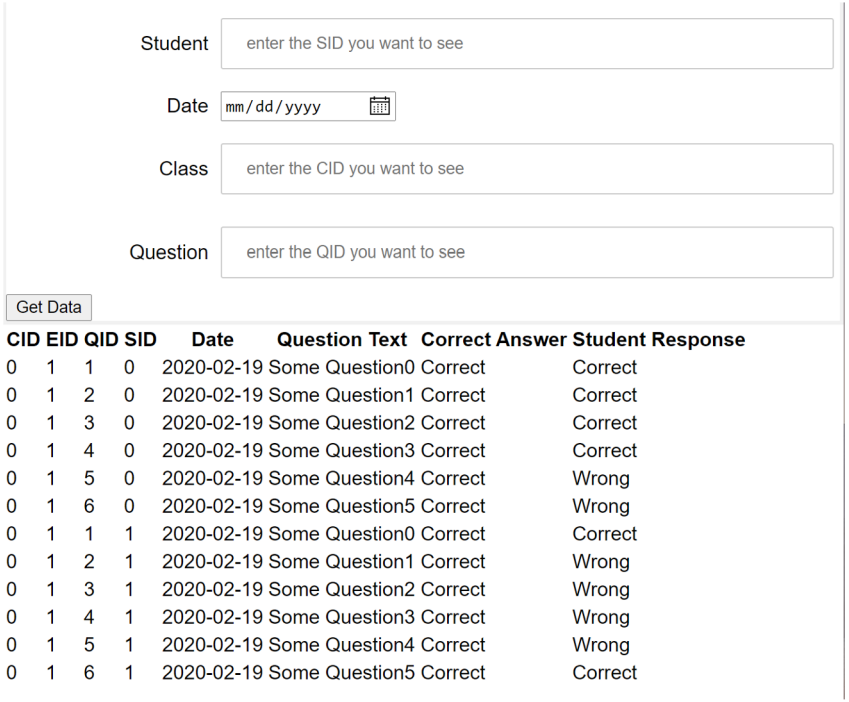
- AllResults: Used to return all responses associated with a specific teacher
  - Can then easily be filtered by Question, Student, or Evaluation

SELECT * FROM AllResults;

| CID | SID | QID | EID | Title | Date | Response | TID | QText | Correct | Choice1 | Choice2 | Choice3 | Choice4 | SName | CName |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | Eval1 | 2020-02-19 | Correct | 0 | Some Question0 | Correct | Correct | Wrong | Wrong | Wrong | Jim | DBMS |
| 0 | 0 | 2 | 1 | Eval1 | 2020-02-19 | Correct | 0 | Some Question1 | Correct | Correct | Wrong | Wrong | Wrong | Jim | DBMS |
| 0 | 0 | 3 | 1 | Eval1 | 2020-02-19 | Correct | 0 | Some Question2 | Correct | Correct | Wrong | Wrong | Wrong | Jim | DBMS |
| 0 | 0 | 4 | 1 | Eval1 | 2020-02-19 | Correct | 0 | Some Question3 | Correct | Correct | Wrong | Wrong | Wrong | Jim | DBMS |
| 0 | 0 | 5 | 1 | Eval1 | 2020-02-19 | Wrong | 0 | Some Question4 | Correct | Correct | Wrong | Wrong | Wrong | Jim | DBMS |
| 0 | 0 | 6 | 1 | Eval1 | 2020-02-19 | Wrong | 0 | Some Question5 | Correct | Correct | Wrong | Wrong | Wrong | Jim | DBMS |
| 0 | 1 | 1 | 1 | Eval1 | 2020-02-19 | Correct | 0 | Some Question0 | Correct | Correct | Wrong | Wrong | Wrong | Mary | DBMS |
| 0 | 1 | 2 | 1 | Eval1 | 2020-02-19 | Wrong | 0 | Some Question1 | Correct | Correct | Wrong | Wrong | Wrong | Mary | DBMS |
| 0 | 1 | 3 | 1 | Eval1 | 2020-02-19 | Wrong | 0 | Some Question2 | Correct | Correct | Wrong | Wrong | Wrong | Mary | DBMS |
| 0 | 1 | 4 | 1 | Eval1 | 2020-02-19 | Wrong | 0 | Some Question3 | Correct | Correct | Wrong | Wrong | Wrong | Mary | DBMS |
| 0 | 1 | 5 | 1 | Eval1 | 2020-02-19 | Wrong | 0 | Some Question4 | Correct | Correct | Wrong | Wrong | Wrong | Mary | DBMS |
| 0 | 1 | 6 | 1 | Eval1 | 2020-02-19 | Correct | 0 | Some Question5 | Correct | Correct | Wrong | Wrong | Wrong | Mary | DBMS |

# Views

## Can limit what responses are shown using the frontend website




| CID | EID | QID | SID | Date | Question Text | Correct Answer | Student Response |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 2020-02-19 | Some Question0 | Correct | Correct |
| 0 | 1 | 2 | 0 | 2020-02-19 | Some Question1 | Correct | Correct |
| 0 | 1 | 3 | 0 | 2020-02-19 | Some Question2 | Correct | Correct |
| 0 | 1 | 4 | 0 | 2020-02-19 | Some Question3 | Correct | Correct |
| 0 | 1 | 5 | 0 | 2020-02-19 | Some Question4 | Correct | Wrong |
| 0 | 1 | 6 | 0 | 2020-02-19 | Some Question5 | Correct | Wrong |
| 0 | 1 | 1 | 1 | 2020-02-19 | Some Question0 | Correct | Correct |
| 0 | 1 | 2 | 1 | 2020-02-19 | Some Question1 | Correct | Wrong |
| 0 | 1 | 3 | 1 | 2020-02-19 | Some Question2 | Correct | Wrong |
| 0 | 1 | 4 | 1 | 2020-02-19 | Some Question3 | Correct | Wrong |
| 0 | 1 | 5 | 1 | 2020-02-19 | Some Question4 | Correct | Wrong |
| 0 | 1 | 6 | 1 | 2020-02-19 | Some Question5 | Correct | Correct |


| CID | EID | QID | SID | Date | Question Text | Correct Answer | Student Response |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 6 | 0 | 2020-02-19 | Some Question5 | Correct | Wrong |

# Acknowledgements

XXXX

- HTML/Front End
- Foundational SQL database

XXXXXX

- Connecting database to front end with Python
- Hosting local server

* Both team members contributed to every part of the project, these acknowledgments just represent where the majority of the work was completed