# Database Overview

Sakshi Gautam, Pallavi Bahl

**Tables:**

1. quiz_content
2. quiz_results
3. credentials
4. constraints

**quiz_content:**

- Overview: This table will contain all the data that is “produced” by the “professor” user. This will also be the content that’ll be ultimately displayed on the system.
- Columns:
    - 

| | |
|---|---|
| quiz_id | The ID for a quiz. |
| ques_id | The ID for a question for a quiz. |
| ques_type | “SA/MAQ/TEXT” |
| ques_desc | Question text. |
| ans_id | The ID for an answer for a question. |
| ans_desc | The text value of available answer options. |
| is_correct | Is this the correct answer choice. |
| max_score | Max score possible for a question. |

**credentials:**

- Overview: This table will contain all login related data.
- Columns:
    - 

| | |
|---|---|
| username | Username |
| password | Hashed Password |
| type | Type of user (to control access and available features). |

    - 

**quiz_results:**

- Overview: This table will contain all the data that is “produced” by the “student” user.
- Columns:
    - 

| | |
|---|---|
| student_id | The ID for a student. |
| quiz_id | The ID for a quiz (FK). |
| ques_id | The ID for a question for a quiz. (FK) |
| ans_id | The ID for an answer for a question for a quiz. (FK) |
| attempt_id | The ID for a quiz attempt. |
| total_score | Total score achieved in the quiz. |
| attempted_on | Date-Time value for when the quiz was completed on. |
| time_taken | Time taken to complete the quiz. |
| is_final | True if no partial attempts are allowed. |

**constraints:**

- Overview**:** This table will contain all the constraints (time, #attempts allowed etc) for a quiz.
- Columns:
    - 

| | |
|---|---|
| quiz_id | |
| time_allowed | Time allowed to complete a quiz (Integer value in seconds). |
| take_by | Date by which the quiz is to be completed. |
| attempts_allowed | The number of attempts allowed for a quiz. |

**Advantages of the above schema:**

1. Easy to implement and query.
2. Easy to perform aggregation queries (stats feature) on this.
3. Eliminates (mostly) the need for joins.

**Disadvantages of the above schema:**

1. Redundant data (most of the data would be repeated multiple times).
2. Not normalized.
3. Maybe difficult to extend

**Since:**

1. we are not being graded on the database design (have sent an email to the TA to confirm this, currently awaiting a reply).
2. the volume of data is going to be very low
3. Performance is not an issue/constraint/important

**We think that this will be the best approach.**