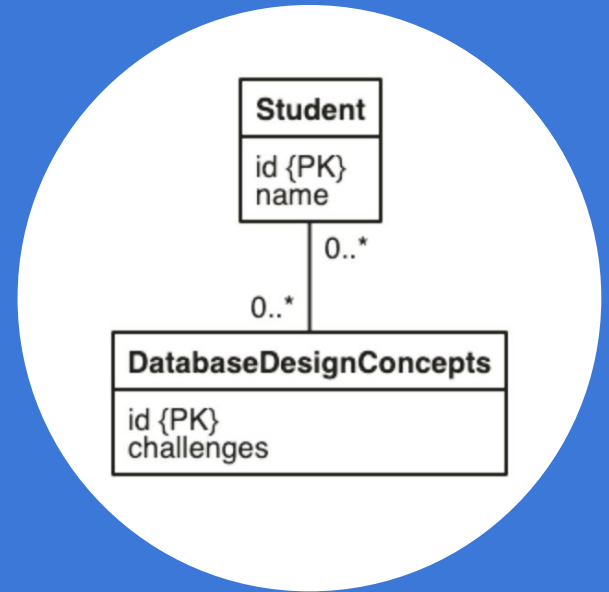


# AutoER:

Automated Question Generation and Evaluation to Assist Students Learning UML Database Design



A Thesis Submitted by Tatiana Urazova For The Degree of B.S. Computer  
Science Honours in Faculty of Science  
Supervisor: Dr. Ramon Lawrence



# Motivation & Background

1.

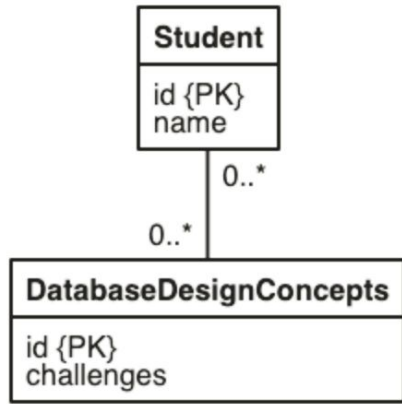


Significant increase of **Online Learning** over the past years



## **Automated Online Question Systems**

- Support **continuous practice and learning**
- Provide **real-time feedback**



## Database Design Questions

- Key concepts, require practice to master
- Time-consuming to create, mark and produce feedback

## Using Autograding systems:

- Potential to increase engagement and allow for formative learning
- Challenging to use with design questions, due to:

high variability of answers, and interpreting the semantics of the diagram and tested concepts

# The AutoER System

## ◆ **Autograding**

String representation,  
restricting naming:  
removes ambiguity of  
marking results

## ◆ **Immediate Feedback**

Immediately generates  
Marker & Student  
Feedback

## ◆ **Question Generation**

Supports both  
instructor-generated  
questions and automatic  
question generation

**2.**

**User Interface**

# Question Format

Construct a database design in UML for a fish store where:

A fish store maintains a number of aquaria tanks, each with a number, name, volume and color.

Each tank contains a number of fish, each with an id, name, color, and weight.

Each fish is of a particular species, which has a id, name, and preferred food.

Each  number of events in its life, involving a date and a note relating to the event.

Add entity

Add attribute

Your ER Diagram is empty

- Students interact directly with question text
- Reduces the variability of student answers
- Removes ambiguities in marking results by restricting the names of the elements that the students can add to the diagram

# Restricting Submissions

## Max Attempts



Limiting the number of submissions

## Regression Penalty

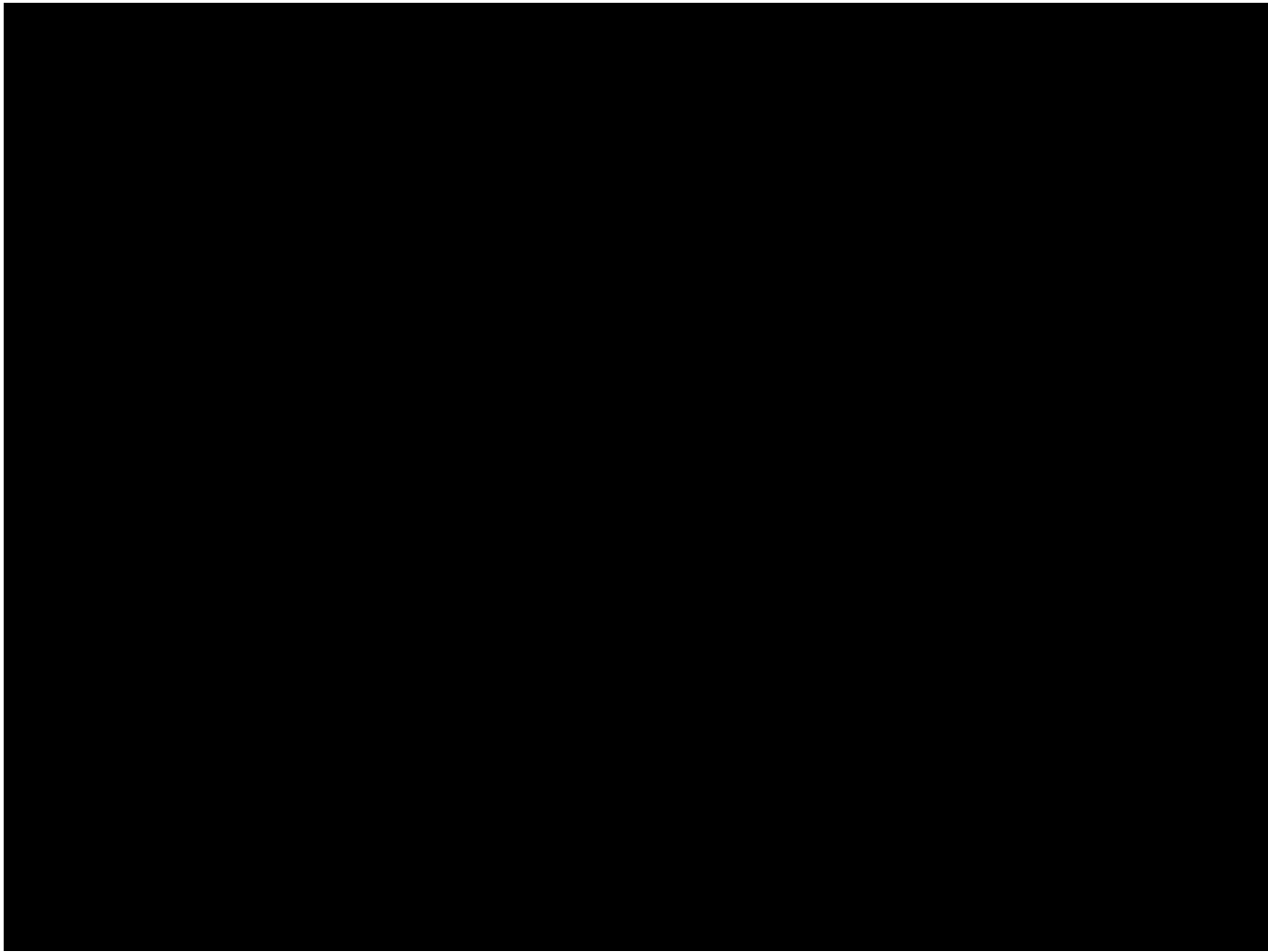


Unlimited submissions, but every time the student's mark goes down, the regression penalty is applied

$$\text{penalty} = -\text{abs}(\text{higher\_mark} - \text{lower\_mark}) * 0.5$$

Penalty type is completely configurable by the instructor





# 1. First submission under Regression



3.41/10.0 ID: 25213 2022-03-14 17:40:00 IP: 24.67.108.72 Attempts: 1/Infinity Regression Penalty: 0.00

Construct a database design in UML for an app store described below.

A **Publisher** where each **publisher** is identified by an **id** and has a **name**.

# 2. Second submission: mark goes down



1.11/10.0 ID: 25214 2022-03-14 17:40:00 IP: 24.67.108.72 Attempts: 2/Infinity Regression Penalty: -0.77

Construct a database design in UML for an app store described below.

A **Publisher** where each **pub**

A **Category** where each **cate**

An **App** storing each **app** tha

Publisher. A **Publisher** may p

A **AppVersion** stores each ve

versions of the same **App**. Ea

A **AppVersionReview** stores

identified by **reviewer** attrib

Answer feedback:

Entity name marks: 1.0/1.0  
Entity attribute marks: 0.4/0.5  
Entity primary key marks: 0.2/1.0  
Weak entity key marks: 0/1.0  
Extra entities: 0.0  
Total entity marks: 1.6/3.5  
Relationship entity marks: 0/2.5  
Relationship cardinalities marks: 0/2.5  
Extra relationships: 0.0  
Total relationship marks: 0.0/5.0  
Total marks: 1.6/8.5  
Total scaled marks: 1.88/10.0  
Regression Penalty: 0.77

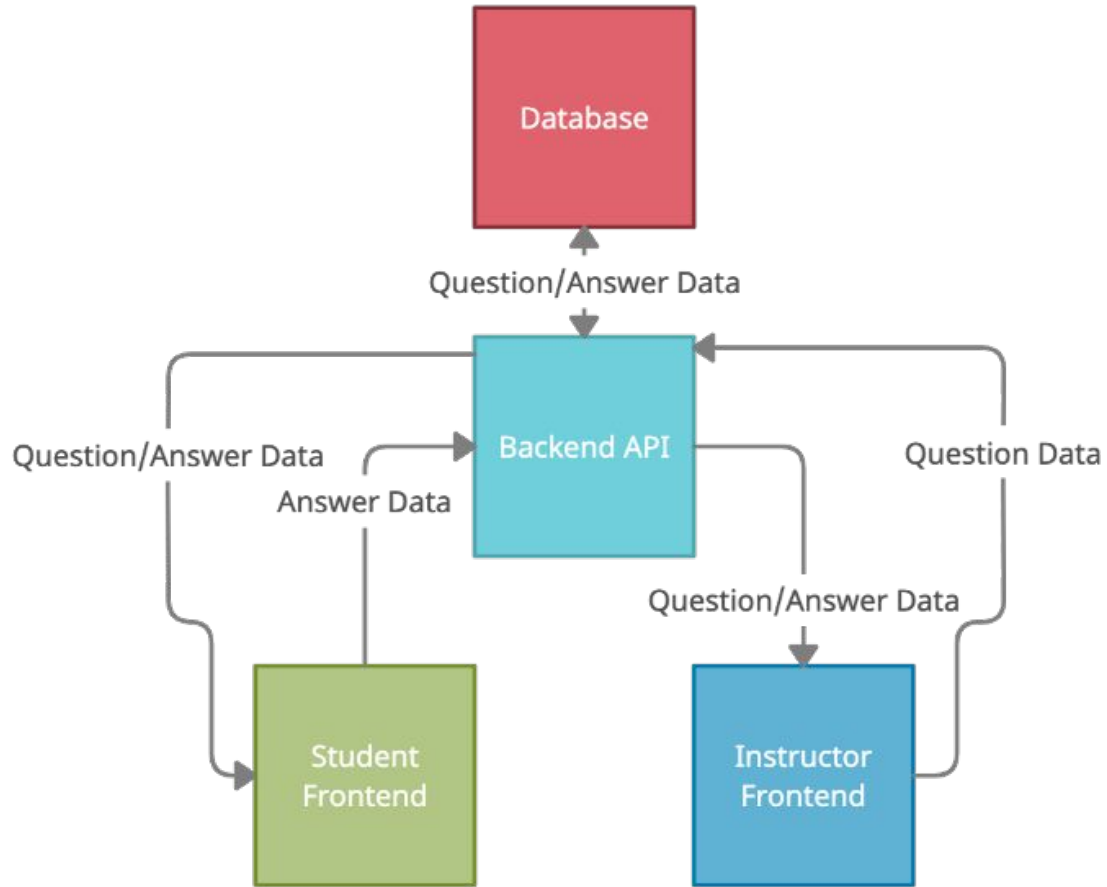
**Publish**  
id  
name

**Review**

# AutoER Server

3.

# System Overview





# Docker Containers

## 1. **django**

Communicates with frontend services to form the Backend API for AutoER

## 2. **postgres**

Database for AutoER, not generally accessible outside of Docker

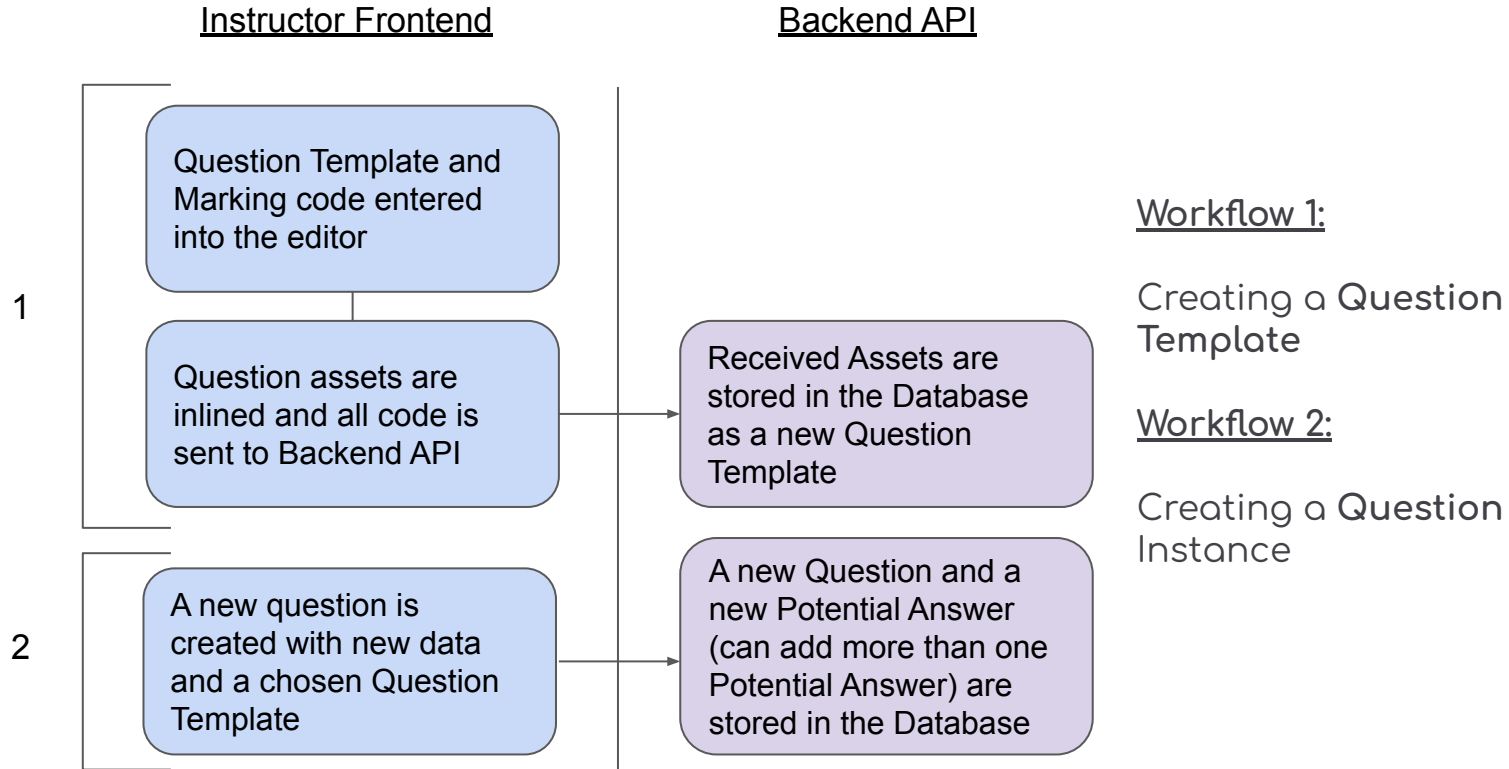
## 3. **student**

Hosts a static website for students to complete questions

## 4. **reverse\_proxy**

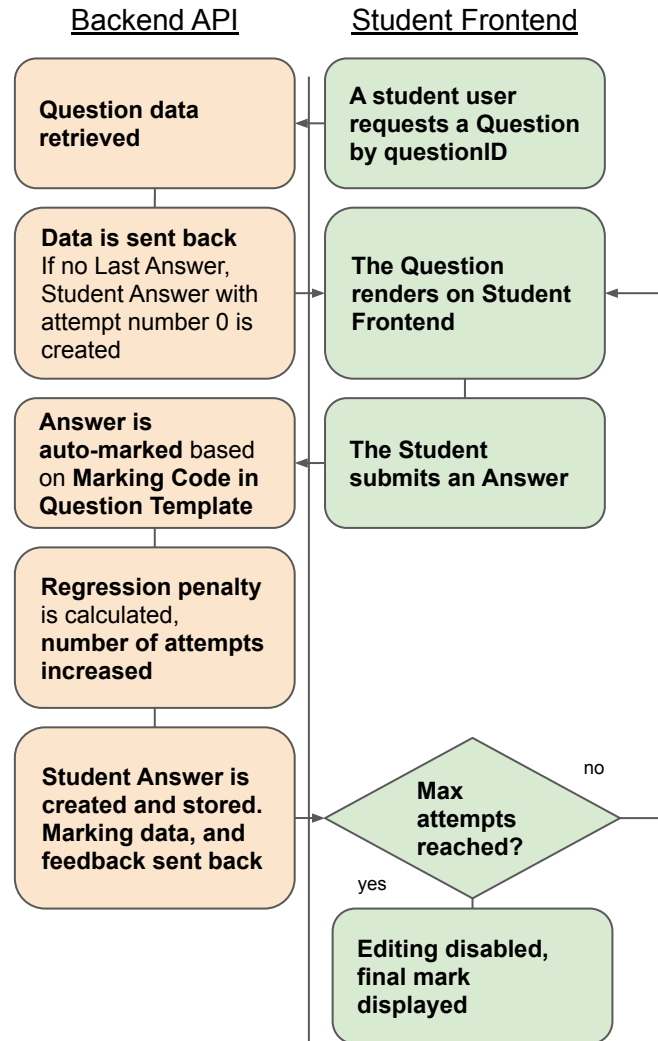
Maps external URLs to internal Docker container services.

# Question Creation

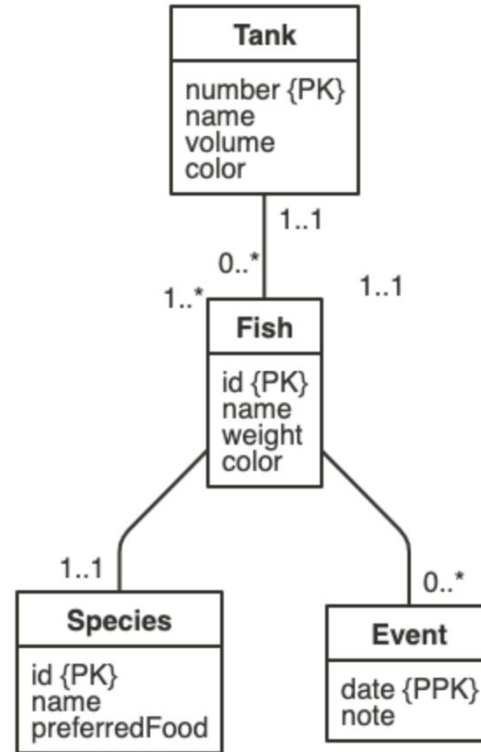


# Answering and Automarking a Question

- Process initiated with the Student Frontend:  
`/questions/<questionID>`
- If no previous answers are found, a new Student Answer with attempt number 0 is created. Otherwise, the latest answer data is loaded.
- This ensures that a student does not lose the progress & keeps track of the regression penalties accumulated and number of submission attempts.



```
[Tank|number {PK};name;volume;color]
[Fish|id {PK};name;weight;color]
[Species|id {PK};name;preferredFood]
[Event|date {PPK};note]
[Tank]1..1 - 0..*[Fish]
[Fish]1..* - 1..1[Species]
[Fish]1..1 - 0..*[Event]
```



## nomnoml library

translates simple syntax strings into visual representations of UML diagrams



# Automarking and Feedback Steps

```
[Tank]0@1 - *[Fish]  
[Fish]1..* - 1..1[Species]  
[Fish]1..1 - *[Event]
```

Flexible Cardinalities: And Example

1. Match entities in the student answer with entities in instructor solution by name.
2. Match the attributes within each entity.
3. Match the relationships in the student answer with instructor solution, using entity names and cardinality constraints (0..1, 1..1, 0..\*, 1..\*).
4. Compare the student's answer to all the provided correct answers,
5. Throughout each step, appropriate feedback is generated. Aggregate overall feedback.

## Instructor Frontend

Instructor enters the generation code for a chosen Question Template to create a new Generated Question Type

## Backend API

Received data is stored in a new Generated Question Type

First Load?

Generate seed for the generation code in Generated Question Type

Execute generation code, store resulting assets as a new Question

Create new Question Details to map a Question to a Student

Load Question corresponding to Question Details, send data to student

## Student Frontend

A student user requests a Generated Question

The Question renders on student frontend

## Generated Questions

- Each user accesses the same URL but gets a unique version of the question
- Provide additional practice & discourage academic misconduct

# Gibberish Module

Reard has key schluol, schmoid has field struerly.

Qoants has key criopp and has attributes schreaub, mcgaueff.

Bluils is identified by wow, chesh has fields spraalt, kaarts.

Hruell has key neently has field stuelly.

Thenn is identified by its association with Bluils and has identifying attribute schrauez. Thenn has fields rand, wangly. Thenn has at most one connection with Hruell, and Hruell is connected with one Thenn.

Joiant is identified by its association with Thenn and has identifying attribute strintly. Joiant has field cleef.

Bluils has zero or more connections with Thenn, and Thenn is connected with one Bluils.

Reard has multiple relationships with Bluils, and Bluils is connected with one Reard.

Qoants has zero or more connections with Thenn, and Thenn must be related to exactly one Qoants.

Qoants may be related to many Hruell, and Hruell must be related to exactly one Qoants.

Thenn has zero or more connections with Joiant, and Joiant is connected with one Thenn.

Reard has multiple relationships with Joiant, and Joiant is connected with one Reard.

Reard
schluol {PK} schmoid struerly

Qoants
criopp {PK} schreaub mcgaueff

Bluils
wow chesh spraalt kaarts

# 4.

## Results

Winter 2021 offering of  
COSC 304 Database  
course, UBC Okanagan

180 students

# Student Survey Results

## Student Feedback:

- Easy to use & Saves time compared to drawing diagrams by hand
- Appreciated autograding & immediate feedback
- Areas of improvement included the display of the diagrams, especially positioning of cardinalities on relationships, displaying, recursive relationships, and the ability to rearrange diagrams

**SUS Score**

76.95

# AutoER Evaluation in COSC 304 Fall 2021

## Midterm Exam

- Random Generated Question
- Students were **randomly assigned** either Restricted Attempts or Regression Penalty

	<b>Max Attempts</b>	<b>Regression</b>
Max submissions	7	52
Avg submissions	2.32	3.57
% Students	<b>51%</b>	<b>49%</b>
Avg Grade	<b>73%</b>	<b>81%</b>

## Final Exam

- Instructor Generated Question
- Students were **able to choose** between Restricted Attempts or Regression Penalty

	<b>Max Attempts</b>	<b>Regression</b>
Max submissions	7	54
Avg submissions	4.63	13.97
% Students	<b>80%</b>	<b>20%</b>
Avg Grade	<b>70%</b>	<b>63%</b>

# Student Profiles

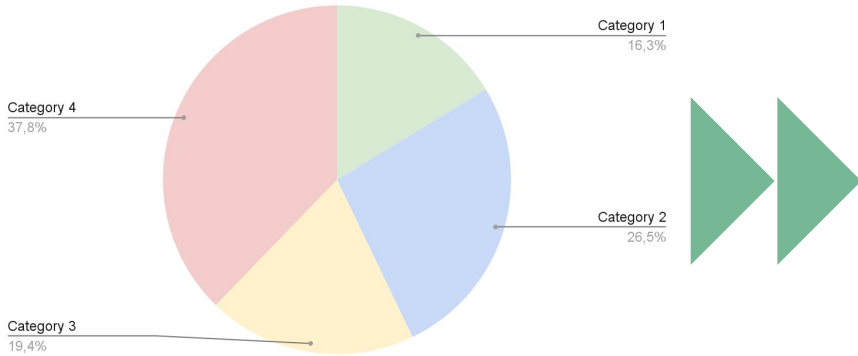
Category 1	No regressions, 80%+ first submit
Category 2	0 to 2 regressions, iterative development
Category 3	2-4 regressions
Category 4	5+ regressions, end thrashing

## Summer 2021 Distribution:

Category	% Stu
1	16%
2	26%
3	19%
4	37%

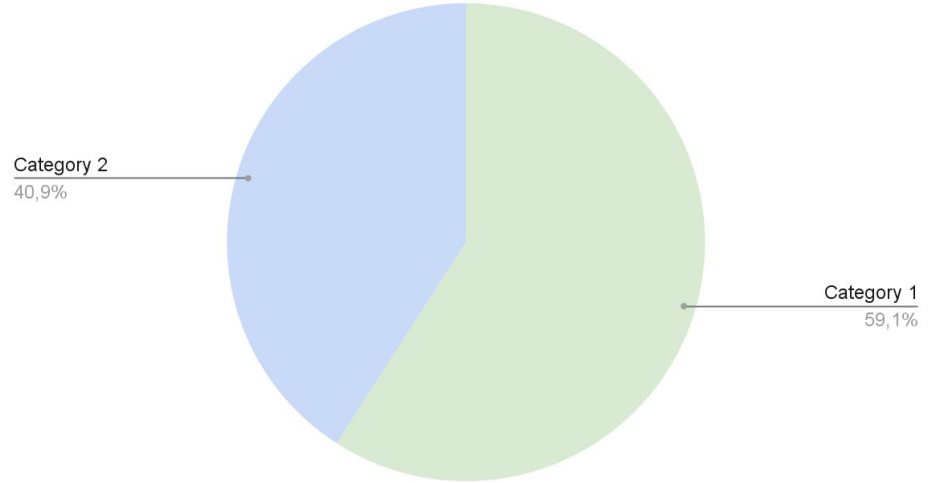
# Performance on the Midterm: Restricted Attempts

Points scored



Summer 2021: No restriction

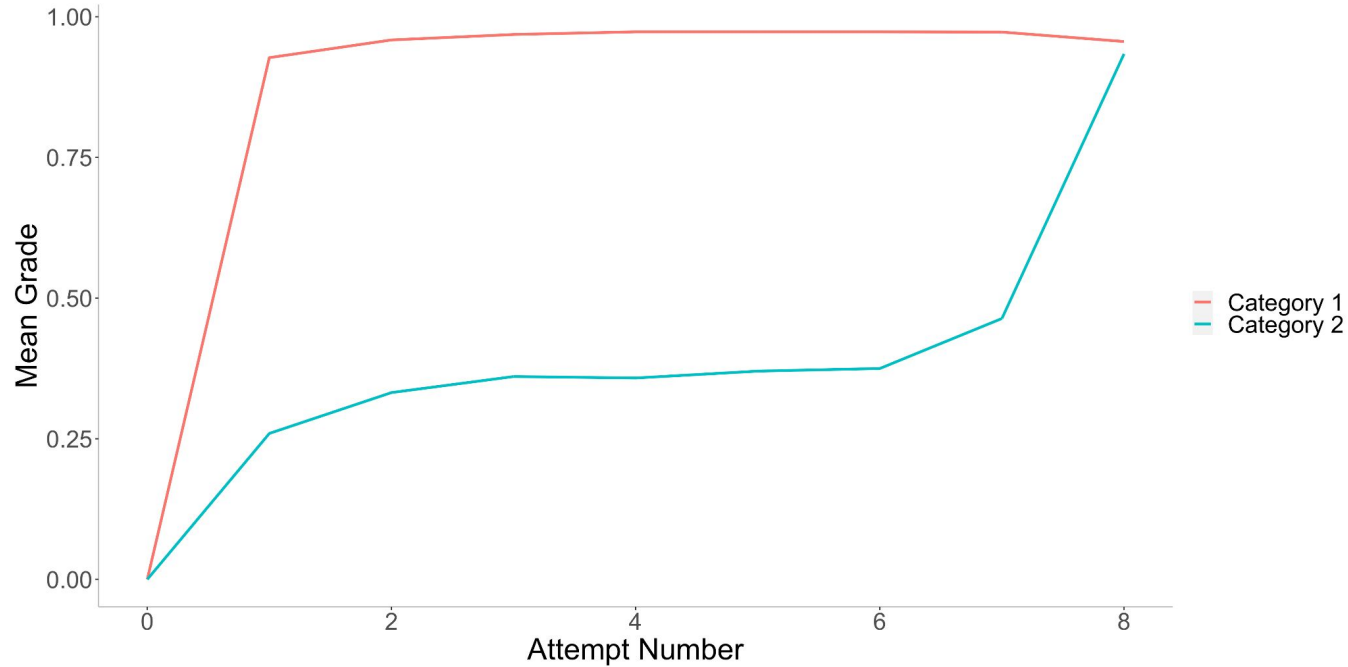
% Students



Fall 2021: Restricted Attempts



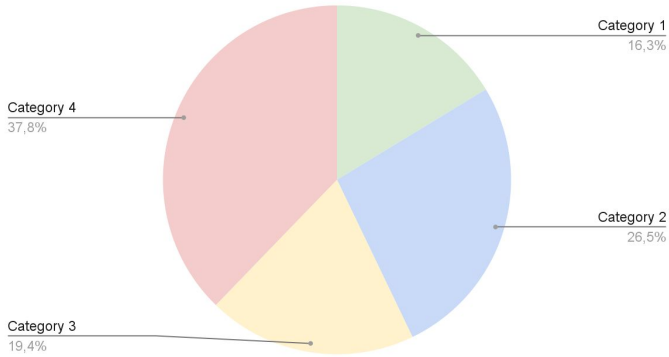
# Performance on the Midterm: Restricted Attempts



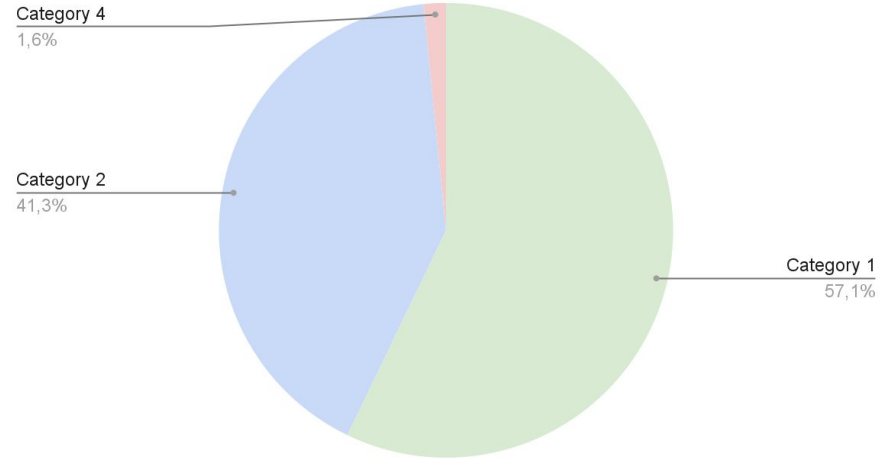
Restricted Attempts Performance on the Midterm Grouped by Category

# Performance on the Midterm: Regression Penalty

Points scored



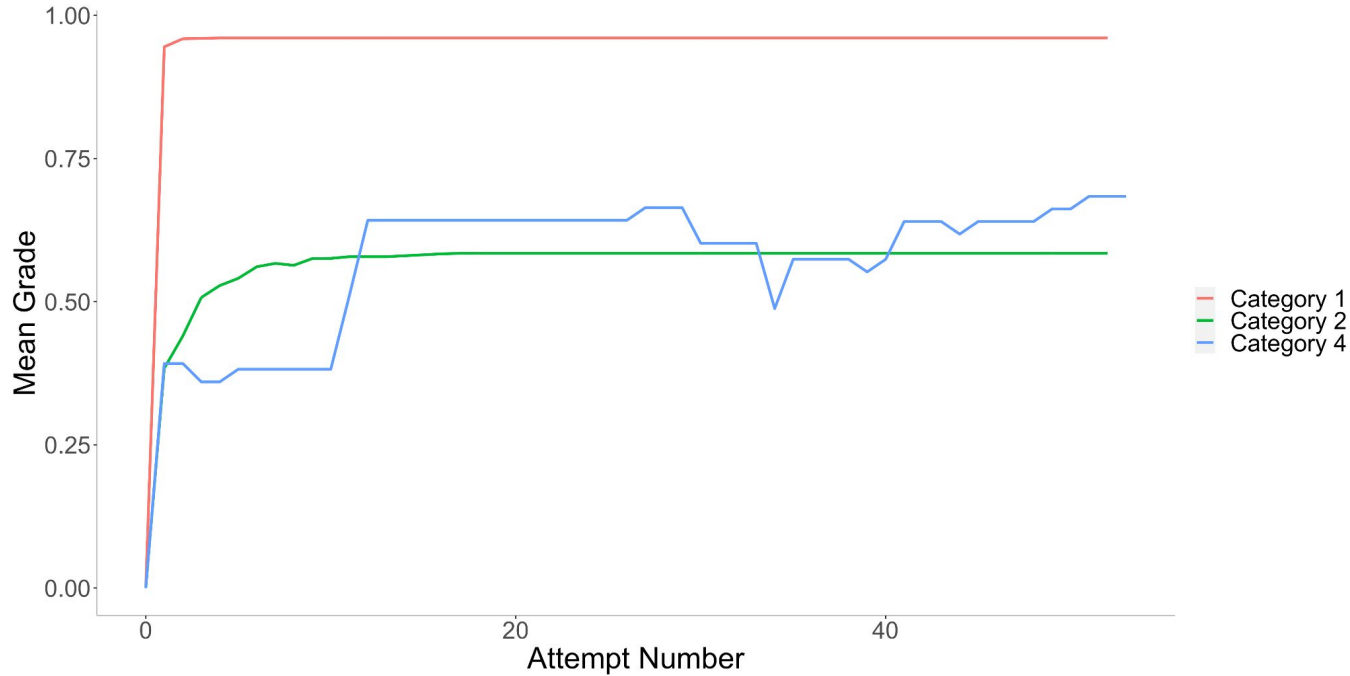
% Students



Summer 2021: No restriction

Fall 2021: Regression

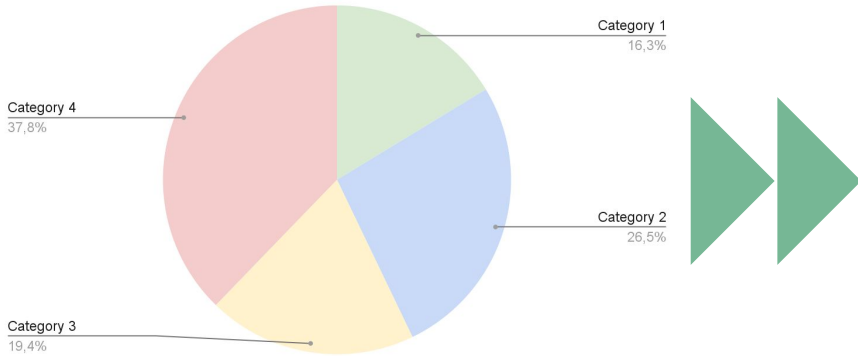
# Performance on the Midterm: Regression Penalty



Regression Penalty Performance on the Midterm Grouped by Category

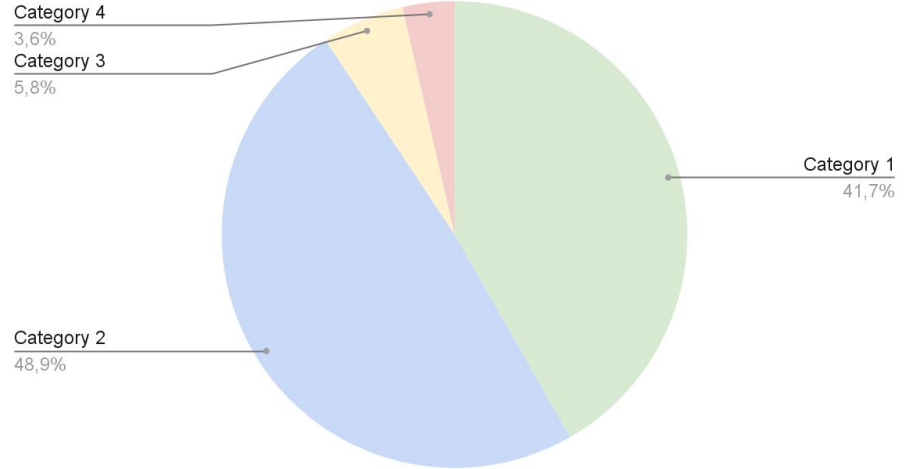
# Performance on the Final: Restricted Attempts

Points scored



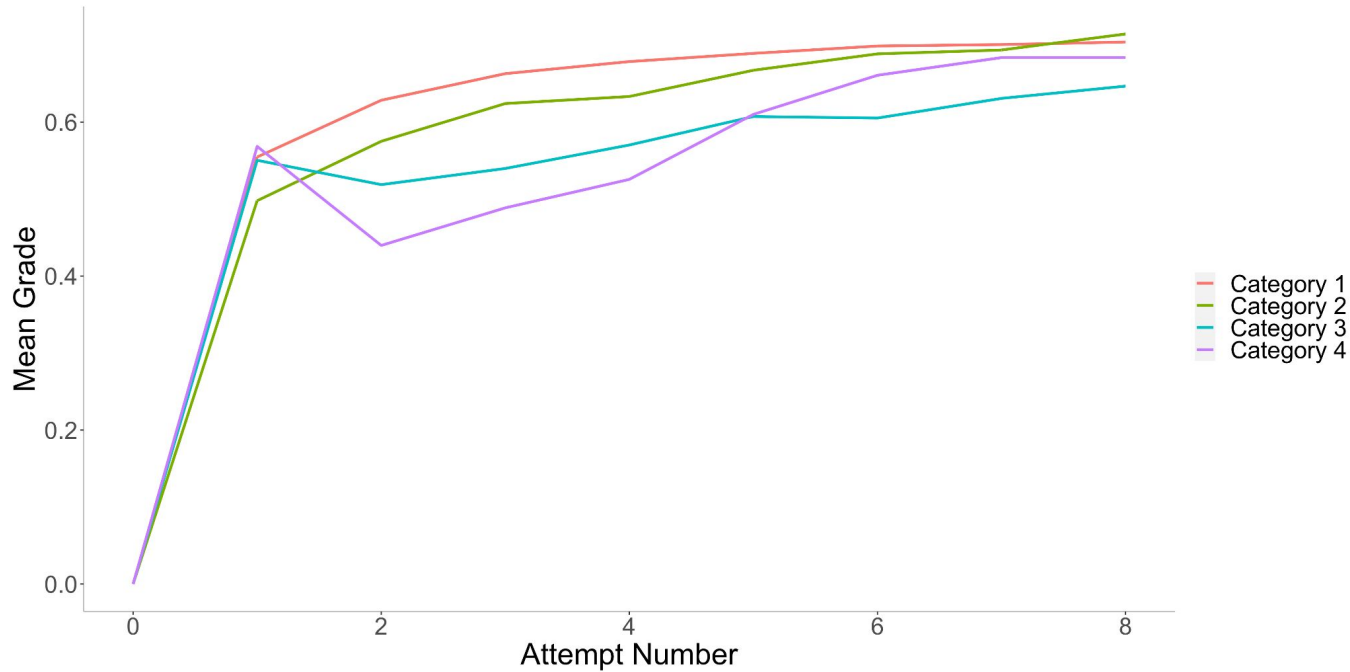
Summer 2021: No restriction

% Students



Fall 2021: Restricted Attempts

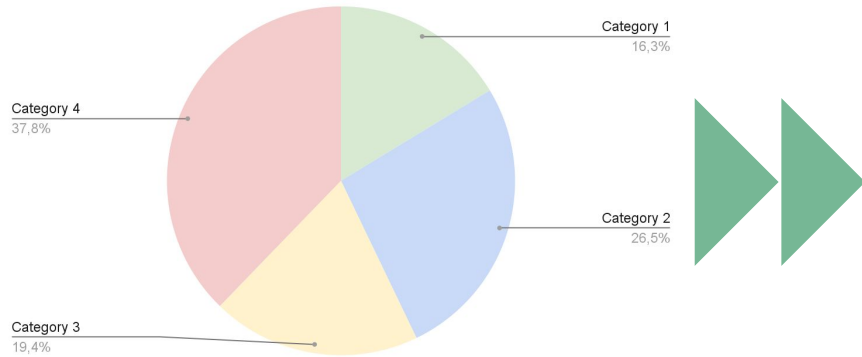
# Performance on the Final: Restricted Attempts



Restricted Attempts Performance on the Final Grouped by Category

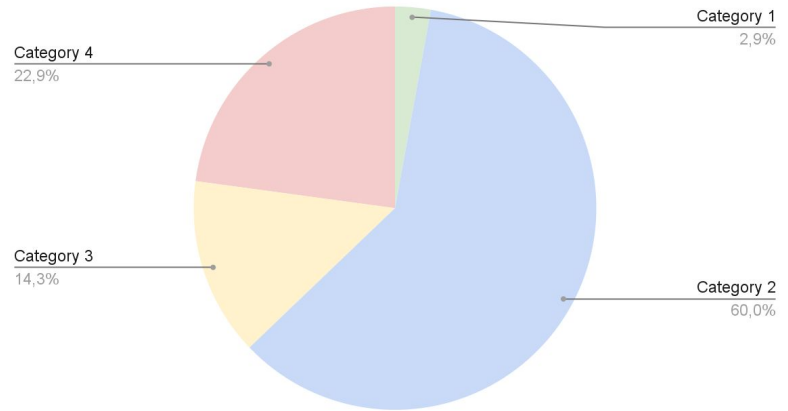
# Performance on the Final: Regression Penalty

Points scored



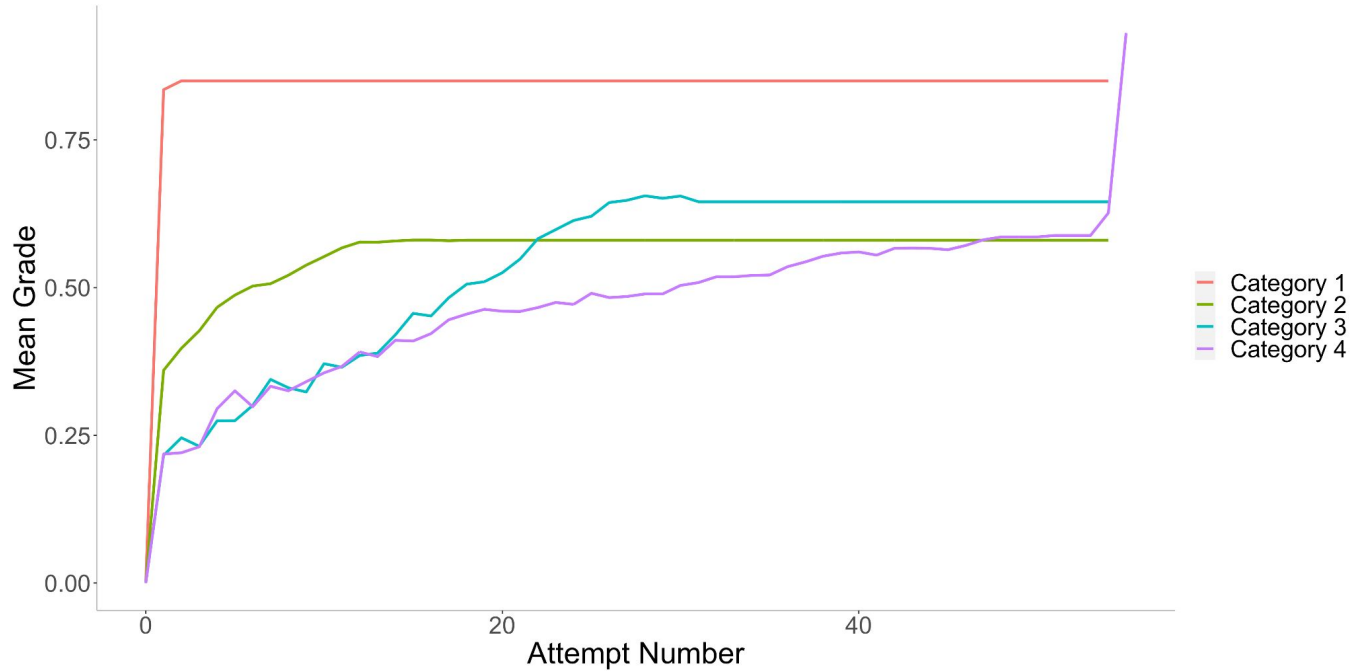
Summer 2021: No restriction

% Students



Fall 2021: Regression Penalty

# Performance on the Final: Regression Penalty



Regression Penalty Performance on the Final Grouped by Category

# Future work

1. Improving Visual Representation, allowing interaction with the diagram itself
2. Developing further strategies to prevent system exploitation and undesired user behaviour
3. Integrating into a learning management system: PrairieLearn
4. Evaluating the system on future Database course offerings



**Thank you!**

**Dr. Ramon Lawrence**

**Sarah Foss**

