People

• Gokhan Kul (gokhanku@buffalo.edu)

• William Spoth (wmjspoth@buffalo.edu)

• Carl Nuesse (carlnues@buffalo.edu)

• Alexander Stachnik (ajstachn@buffalo.edu)
Projects

- **Goal**: Build a Relational Query Engine
- **Teams**: Up to 3-person Groups (3 strongly recommended)
- Setup (Checkpoint 0) + 4 Project Checkpoints
  - Evaluation Criteria
    - **Correctness**: Produce correct results or get an F.
    - **Performance**: Meet or beat ref implementation for an A.
  - Resubmit *as many times as desired* until deadline.
Project Objectives

- Designed to make you…
  - understand how real database systems work
  - design an important component of a database system
- Shows what kind of problems you need to solve in the real world
Project Outline

Checkpoint 0
Hello World

Hello, World Java program

Submission System
Project Outline

A relational query processor

SQL Query → Parser & Translator → Relational Algebra → Optimizer → Execution Plan → Evaluation Engine → Query Result

Statistics
Project Outline

SQL Query → Parser & Translator → Relational Algebra (Only Select & Project)

Checkpoint 1
Intro to CSV

Optimizer → Statistics

Query Result → Evaluation Engine → Execution Plan
Checkpoint 1

- Sanity check
- How do you implement a given schema?
- How can you parse a query?
- Can you read from a CSV file and report results correctly?
- How do you use the least memory possible?
Project Outline

SQL Query → Parser & Translator → Relational Algebra (Query Plans)

Checkpoint 2
Real SQL

Optimizer

Statistics

Query Result

Evaluation Engine

Execution Plan
Checkpoint 2

- How do you join multiple tables, efficiently?
- How do you create a query plan?
- How do you deal with nested queries?
- Can you sort data? Just choose top-k rows?
Project Outline

1. SQL Query
2. Parser & Translator
3. Relational Algebra
4. Checkpoint 3 Optimization
5. Optimizer
6. Statistics
7. Execution Plan
8. Evaluation Engine
9. Query Result
Checkpoint 3

- How do make your system faster?
- Programming efficiency?
- Choosing a strategy?
- More efficient operators?
- How can you deal with aggregation?
Project Outline

SQL Query → Parser & Translator → Relational Algebra

Checkpoint 4
The Real World

Optimizer → Execution Plan → Evaluation Engine → Query Result

Statistics
Checkpoint 4

- What is the effect of the data you have on the query planning?
- What do you do when there is not enough memory for certain operators?
Libraries

- **JSqlParser (Forked version)**
  - Text to SQL Parse Tree

- **EvalLib**
  - Arithmetic Expression Evaluator

- **Apache Commons CSV**
  - CSV Format Support / Parsing CSV
Checkpoint 0

- Team formations
  - Form your team
  - The teams are **final**, you cannot switch teams
  - Create a **private** git repository on github or bitbucket
  - Do **NOT** share the link to your repository with **anyone**
  - Make sure all of you can push and pull code
Checkpoint 0

- Repository structure
  - Repository Name: CSE4562SP18
  - Main File: edu.buffalo.www.cse4562.Main.java
  - System.out.println("Hello, World");
  - Add .gitignore file for java before your first commit
Checkpoint 0

**Timeline**

- Email Will and Gokhan until **Monday, February 5th 11:59 pm**
  - Names, UBIT names, UB ID #
  - The link to your repository
- We will provide you with your team number and a deploy key
- You will add the deploy key to your repository
- Submission will open on **Tuesday, February 6th 12:00 pm (noon)**
- Deadline is **Friday, February 9th 11:59 pm**
Checkpoint 0

- Submission
  - https://autograder.cse.buffalo.edu
  - Create a text file (submission.txt)
  - First line: The **secret key** we sent to your group and **nothing else**
  - Submit the file and wait for your grade
Checkpoint 0

- **Problems**
  - Come to office hours
  - Try asking your questions on Piazza
  - If absolutely necessary, email us
Tips

• Github (for students), gitlab, and bitbucket provide free private repositories

• Learn how to use git commands
  • add
  • checkout
  • push
  • pull, and more…
Questions?