Practicum in Database Systems

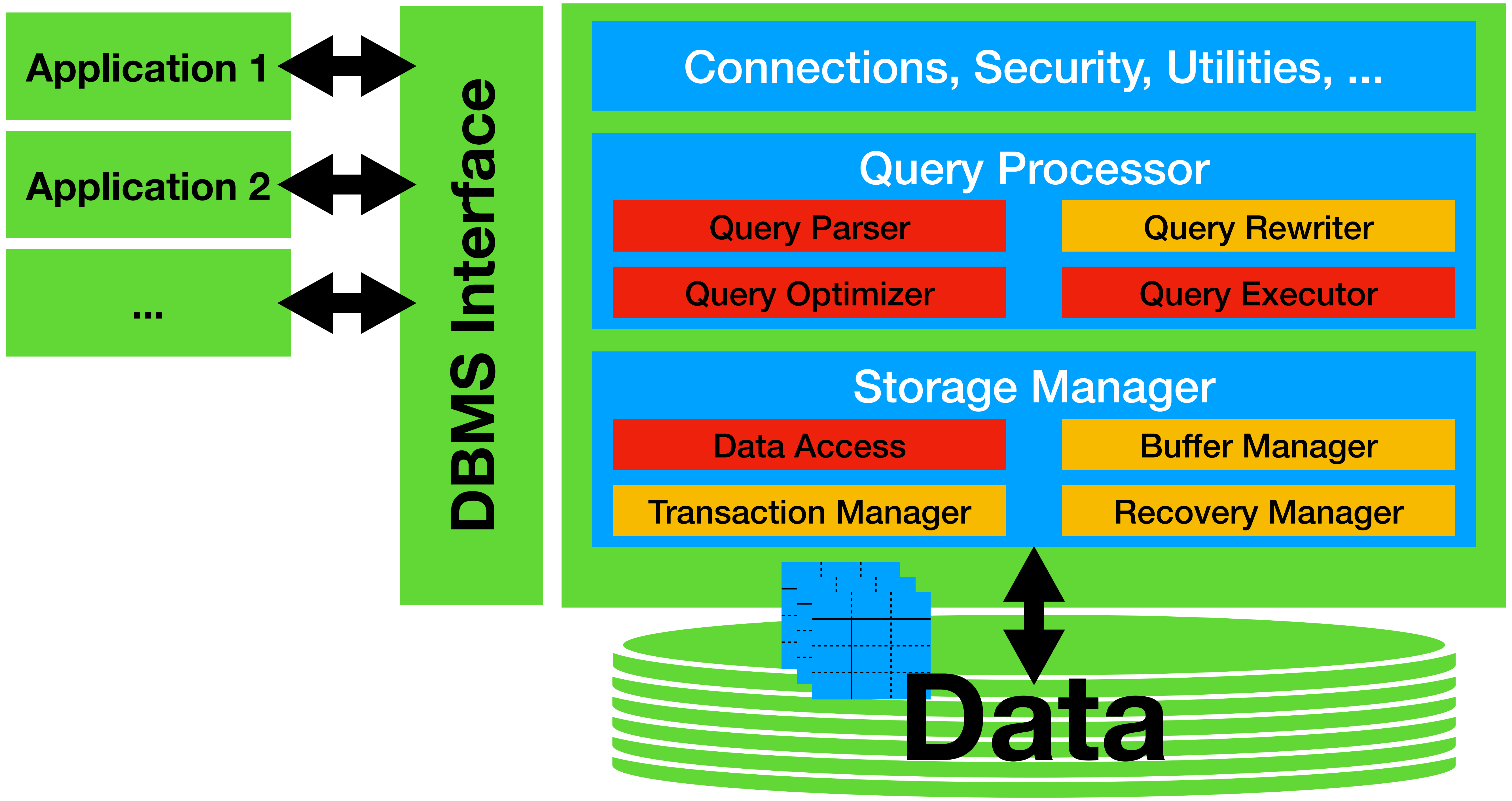
Immanuel Trummer (itrummer@cornell.edu)

Course Logistics

Practicum Overview

- Work in teams of up to **four students**
- Develop **simple database management system**
    - Processes SELECT-FROM-WHERE queries
    - Implements different operator versions
    - Optimizes plans via query optimization

Practicum Scope



Practicum Organization

- Weekly meetings on **Fridays (G01 at 11:15 AM)**
  - Introduces new project phase or Q&A
- Five **project phases**
  - Details provided as Canvas assignments
  - Submit code on Canvas for each phase
- Teams start with an **empty code repository**
  - No solution code is provided!

Web and Other Resources

- **Canvas Web site** for assignments and slides
- Associated **Ed forum** for discussions
- **Background**: www.databaselecture.com
- Will offer **office hours**
  - Regular hours posted on Canvas
  - Exceptions posted on Ed

Project Phases Overview

- **P1**: SQL interpreter on small data
  - In-memory evaluation
- **P2**: Scaling to bigger files
- **P3**: Data indexing
- **P4**: Query optimization
- **P5**: Extensions (multiple options)

Prerequisites

- Mastery of **Java and data structures** at CS 2110 level
- Being able to work **independently**
- Being able to **debug** code
  - Limited help for debugging from staff ("10 minute debugging policy")

Development Tools

- Must submit **Eclipse project** for each project phase
  - Additionally, must submit runnable .jar file
- Can use **different IDE** for development!

Grading

- Grading is based on the **quality of code** submitted
  - No exams or tests
- **Three criteria** for grading:
  - Code satisfies constraints outlined in assignments
  - Code passes various automated test cases
  - Some points for code style and comments

Alignment with CS 4320

- If taking CS 4320/5320 **this semester**:
  - Read up on material before it appears in CS 4320/5320
  - Use text book or references in instructions

Project 1

Project 1 Overview

- Implement system that **processes simple SQL queries**
  - In-memory processing
- **Input**
  - SQL query
  - Database
- **Output**: query result
  - Make sure to follow output format!

Processing Steps

- **Parse** SQL query
  - Use JsqlParser library
- **Translate** query into query plan
- **Evaluate** query to generate result
- **Write** result to output file

SQL Scope

- Only support **limited subset of SQL**
  - See details in the instructions
- Supported **SQL clauses**:
  - SELECT
  - FROM
  - WHERE
  - ORDER-BY
  - DISTINCT

SQL Operators

- Need to implement **SQL standard operators**
    - E.g., filtering data, joining data from different sources
- Implement the **iterator model**
    - Each operator extends abstract operator class
    - Provides getNextTuple() and reset() methods

Suggested Steps (1/2)

- Familiarize yourself with **JsqlParser**
- Implement **database catalogue**
- Implement the **scan operator**
  - Must decide on tuple representation
- Implement **filtering**
  - Must evaluate Boolean expressions on tuples
  - Use expression visitor interface by JsqlParser

Suggested Steps (2/2)

- Implement **join operator**
  - Avoid computing cross product!
  - Consider "pushing down" predicates
- Implement **table aliases**
- ORDER BY
- DISTINCT

?