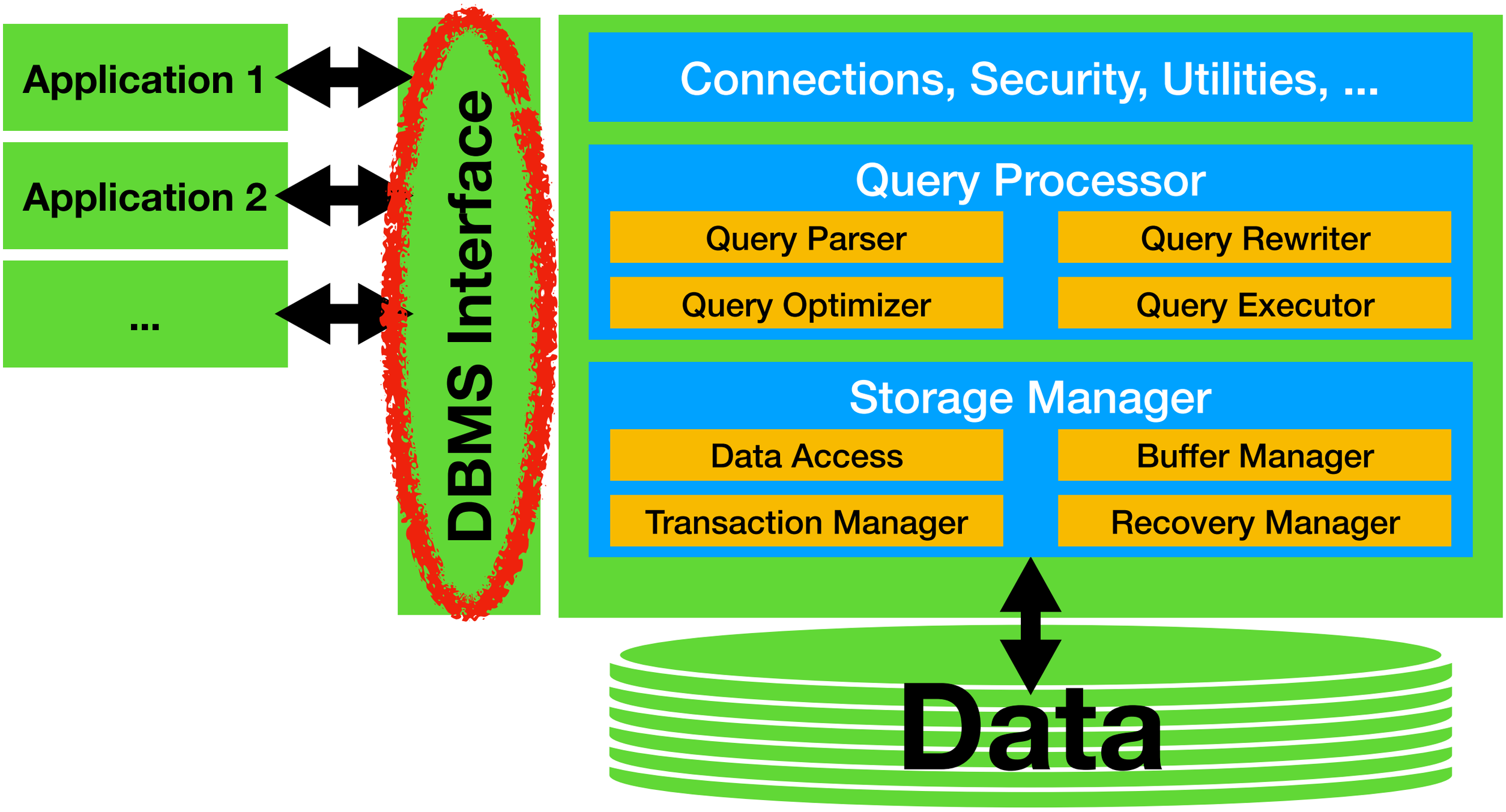
# SQL: Simple Analysis

Immanuel Trummer
itrummer@cornell.edu
www.itrummer.org

# Database Management Systems (DBMS)



# Reminder

- Learning SQL (Structured Query Language)
- Have seen how to define **database schema**
- Have seen how to **insert/update/delete** data
- Now: how can we **analyze** data via SQL queries?

# Simple SQL Queries

- An SQL query describes a **new relation** to generate
- Simple SQL queries consist of **three parts**
  - **SELECT**: describes **columns** of relation to generate
  - **FROM**: describes **source** relations and how to match
  - **WHERE**: defines **conditions** result rows must satisfy

# Simple Query Format

- SELECT **<columns>**
  FROM **<table1>** JOIN **<table2>** ON (**<join-pred>**) ...
  WHERE **<where-pred>**

- **<columns>** is comma-separated list of columns

- **<table1>** and **<table2>** are database relations

- **<join-pred>** is condition defining matching tuples pairs

- **<where-pred>** are additional conditions

# Example Query

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

# Example Query

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

Find pairs of students and enrollment tuples where Sid (i.e., student ID) is the same ...

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

# Example Query

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

... pair that with courses where Cid (i.e., course ID) matches the one in enrollment ...

# Example Query

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

... filter to rows where Cname (course name) is 'CS4320' ...

# Example Query

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

... then discard all columns except for Sname (student name).

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

# Exercise (5 Minutes)

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

- For the same database, find all names of courses for which student "John Smith" is enrolled!

# Simplification by Alias

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

```
SELECT S.Sname
FROM Students S
  JOIN Enrollment E ON (S.sid = E.sid)
  JOIN Courses C ON (E.cid = C.cid)
WHERE C.Cname = 'CS4320'
```

# Simplification by Alias

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

Assign alias S for student relation ...

```
SELECT S.Sname
FROM Students S
  JOIN Enrollment E ON (S.sid = E.sid)
  JOIN Courses C ON (E.cid = C.cid)
WHERE C.Cname = 'CS4320'
```

# Simplification by Alias

```
SELECT Students.Sname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

... use alias instead of full table name

```
SELECT S.Sname
FROM Students S
  JOIN Enrollment E ON (S.sid = E.sid)
  JOIN Courses C ON (E.cid = C.cid)
WHERE C.Cname = 'CS4320'
```

# Omitting Table Names

```
SELECT S.Sname
FROM Students S
  JOIN Enrollment E ON (S.sid = E.sid)
  JOIN Courses C ON (E.cid = C.cid)
WHERE C.Cname = 'CS4320'
```

```
SELECT Sname
FROM Students S
  JOIN Enrollment E ON (S.sid = E.sid)
  JOIN Courses C ON (E.cid = C.cid)
WHERE Cname = 'CS4320'
```

Can omit table names if no ambiguity exists.

# More Diverse Predicates

- Can use inequalities (>, >=): **Students.scores > 70**
- Writing "not equal": **Courses.Cname <> 'CS4320'**
- Check if value in list: **Cname IN ('CS4320', 'CS5320')**
- Regular expressions: **Cname LIKE 'CS_320%'**
  - **%** stands for zero or more arbitrary characters
  - **_** stands for one arbitrary character

# Composite Predicates

- Logical conjunction via **AND** keyword
- Logical disjunction via **OR** keyword
- Negation via **NOT** keyword
- E.g., **Cname = 'CS4320' OR Cname = 'CS5320'**

# Diverse Select Clauses

- Shortcuts for selecting multiple columns
  - * selects all columns
  - **<table>.*** selects all columns from <table>
- Can use arithmetic expressions in select clause
  - E.g., **SELECT  3 * (<column1> + <column2>)**
- Can assign new names for output columns
  - E.g., **SELECT Sname as StudentName**

# Join Syntax Alternatives

- Simply specify names of columns that appear in multiple tables
  - <table1> JOIN <table2> **USING (<column>)**
  - Abbreviates <table1> JOIN <table2>
    **ON (<table1>.<column> = <table2>.<column>)**
- "Natural joins" match values in columns with same name
  - <table1> **NATURAL JOIN** <table2>
  - Introduces equality conditions between columns of same name
- No join keyword: **FROM <table1>, <table2> WHERE <join-condition>**

# Distinct Results

- **SELECT** ... may generate the same row multiple times

- Use **SELECT DISTINCT** instead to eliminate duplicates

# Aggregation Queries

- Can calculate **aggregates** over all rows of result relation
- SQL Aggregates: **COUNT, SUM, AVG, MIN, MAX**
  - **SUM, AVG, MIN, MAX**: numerical expression parameter
  - **COUNT(*)** for counting rows in result relation
  - **COUNT(<column>)** counts rows with value in <column>
  - **COUNT(DISTINCT <column>)** counts number of distinct values in <column> in result relation

# Aggregation Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*)
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

# Aggregation Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*)
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

Find pairs of students and enrollment tuples where Sid (i.e., student ID) is the same ...

# Aggregation Example

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

```
SELECT Count(*)
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

... pair that with courses where Cid (i.e., course ID) matches the one in enrollment ...

# Aggregation Example

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

```
SELECT Count(*)
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

... filter to rows where Cname (course name) is 'CS4320' ...

# Aggregation Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

... then count all remaining rows.

```
SELECT Count(*)
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Courses.Cname = 'CS4320'
```

# Aggregation by Group

- Common: want aggregates for **multiple** data subsets
- Use SQL **GROUP-BY** clause to define data subsets
  - **GROUP BY <column-list>** - distinguish data subsets based on their values in specified columns

# Grouping Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
```

# Grouping Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
```

Find pairs of students and enrollment tuples where Sid (i.e., student ID) is the same ...

# Grouping Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
```

... pair that with courses where Cid (i.e., course ID) matches the one in enrollment ...

# Grouping Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
```

... filter to rows where Cname (course name) is 'CS4320' or 'CS5320' ...

# Grouping Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
```

... group remaining rows by Cname (Course name) ...

# Grouping Example

Database Relations:

Students(Sid, Sname)
Enrollment(Sid, Cid)
Courses(Cid, Cname)

... count rows in each group and report count with course name (unique per group).

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
```

# Grouping Details

- Grouping is applied **after** pairing data sources (FROM) and filtering rows (WHERE)
- Result contains **one row per group**
  - Implies **restrictions** on SELECT clause!
  - Only expressions with **unique** value per group
  - This includes **aggregates** and **grouping** columns

# Predicates on Groups

- Condition in **WHERE** clause applies to single rows (evaluated **before** grouping)

- **HAVING** clause specifies conditions on groups (evaluated **after** grouping)

# Having Example

**Database Relations:**

**Students(Sid, Sname)**
**Enrollment(Sid, Cid)**
**Courses(Cid, Cname)**

```
SELECT Count(*), Cname
FROM Students
  JOIN Enrollment ON (Students.sid = Enrollment.sid)
  JOIN Courses ON (Enrollment.cid = Courses.cid)
WHERE Cname IN ('CS4320', 'CS5320')
GROUP BY Cname
HAVING Count(*) >= 100
```

Only keep groups with at least 100 rows!

# Exercise

- Download a data set matching your interests
- Load it into your database
- Find fun facts about the data via SQL!