

CSCI 403 Database Management

3

An Informal Introduction to Queries
in SQL

CS@Mines

Tables

- Data in a relational database lives in tables:
 - Tables, also called *relations*, usually have a name
 - Rows, also called *tuples*, contain one data point
 - Each row is divided into columns giving different *attributes* of the data point
 - Each column has its own data type and name
 - All rows have the same set of columns
- Tables are just one kind of *object* that lives in a relational database.

CS@Mines

2

Table Example

- In the “csci403” database (the one we’ll use for this class), there is a table named **mines_cs_faculty**

Here is some sample data from **mines_cs_faculty**:

name	office	email	← Column names
Mehta, Dinesh	BB 280J	dmehta@mines.edu	} Rows
Paone, Jeffrey	BB 280C	jpaone@mines.edu	
Fisher, Wendy	BB 280D	wfisher@mines.edu	
Rader, Cynthia		crader@mines.edu	

Columns

CS@Mines

3

Getting Data with SQL: SELECT

To get the contents of a table using SQL, you use a **SELECT** query:

```
SELECT * FROM mines_cs_faculty;
```

We’ll demonstrate this now using **psql**.

Note, the semicolon at the end of the query is required in **psql**, but not in **Squirrel**.

CS@Mines

4

Output From psql

```
csci403=# SELECT * FROM mines_cs_faculty;
 name | office | email
-----+-----+-----
 Han, Qi | BB 280H | qhan@mines.edu
 Painter-Wakefield, Christopher | BB 280T | cpainter@mines.edu
 Paone, Jeffrey | BB 280C | jpaone@mines.edu
 Wang, Hua | BB 280F | huawang@mines.edu
 Yang, Dejun | BB 280G | djyang@mines.edu
 Yue, Chuan | BB 248 | chuanyue@mines.edu
 Camp, Tracy | BB 280E | tcamp@mines.edu
 Dantam, Neil | BB 249 | ndantam@mines.edu
 Fisher, Wendy | BB 280D | wfisher@mines.edu
 Mehta, Dinesh | BB 280J | dmehta@mines.edu
 Rader, Cynthia | BB 280J | crader@mines.edu
 Williams, Thomas | BB 254A | twilliams@mines.edu
 Wu, Bo | BB 247 | bwu@mines.edu
 Zhang, Hao | BB 250 | hzhang@mines.edu
 Baldwin, Mark | BB 312 | baldwin@mines.edu
 Dave, Vibhuti | BB 314C | vdave@mines.edu
 Hoff, William | BB 249 | whoff@mines.edu
(17 rows)
```

CS@Mines

5

Ordering

- In the relational model, there is no intrinsic *ordering* of rows in a table.
 - Unless you specify an order, the database is allowed to give you rows in any order
 - Typically order has to do with how the data is stored, and won’t change until the data is modified
- You can specify an order with the **ORDER BY** clause:


```
SELECT * FROM mines_cs_faculty
ORDER BY name;
```

CS@Mines

6

Filtering in SQL: WHERE

Often we only want one or a selection of rows, not all rows from a table.

In SQL, this is achieved using a WHERE clause.

Suppose we only want Dr. Mehta's information:

```
SELECT * FROM mines_cs_faculty
WHERE name = 'Mehta, Dinesh';
```

name	office	email
Mehta, Dinesh	BB 280J	dmehta@mines.edu

(1 row)

CS@Mines

7

WHERE Condition

- The WHERE condition can be any logical expression on column names:

```
name != 'Han, Qi'
name <> 'Han, Qi' (same as above)
office >= 'BB 300'
1 = 1
```

plenty more to come soon...

- Conditions can also be compound using AND, OR, NOT:

```
office >= 'BB 300' AND name < 'D'
```

CS@Mines

8

Projecting in SQL

Often we only want some columns, not all.

The SELECT clause lets us list what we want:

```
SELECT name, email
FROM mines_cs_faculty;
```

name	email
Han, Qi	qhan@mines.edu
Painter-Wakefield, Christopher	cpainter@mines.edu
Paone, Jeffrey	jpaone@mines.edu
Wang, Hua	huawang@mines.edu
Yang, Dejun	djyang@mines.edu
...	

CS@Mines

9

Putting It Together

We can use SELECT, WHERE, ORDER BY, and more all in one query:

```
SELECT name
FROM mines_cs_faculty
WHERE office >= 'BB 300'
AND name < 'D'
ORDER BY name;
```

name	office	email
Baldwin, Mark	BB 312	baldwin@mines.edu

(1 row)

CS@Mines

10

NULL

- NULL is a special value in SQL:
 - It represents the absence of any value
 - It can symbolize many things:
 - The data is unknown
 - The data is missing
 - The data is irrelevant
- In `mines_cs_faculty`, Dr. Rader's office entry is NULL:

name	office	email
Rader, Cynthia		crader@mines.edu

Note: unfortunately, NULL looks like an empty string in most query tool results.

CS@Mines

11

More About Nothing (NULL)

- NULL cannot be compared, e.g.


```
WHERE office = NULL
```

 yields NULL rather than true or false
- Note that NULL is neither true nor false
 - True AND NULL is NULL
 - True OR NULL is True
 - False AND NULL is False
 - False OR NULL is NULL
 - NULL AND/OR NULL is NULL
- NULL in most expressions gives back NULL:
 - $X + \text{NULL} \rightarrow \text{NULL}$
- To detect in a condition, use IS NULL or IS NOT NULL:


```
SELECT * FROM mines_cs_faculty WHERE office IS NULL;
```

CS@Mines

12

Other Tables To Explore

- **mines_courses, mines_courses_meetings** – data about courses at Mines in a previous semester
- **department, project, employee**, etc. – data used in the textbook examples

Up Next

- Next lecture:
SQL functions and operators