INFO-UB 23: Introduction to Programming and Data Science

Katherine Hoffmann Pham

July 2018

NYU Stern, Department of Information Systems

SQL Setup

Before we get started . . .

Three important differences between MySQL and Python:

- Semicolon required at the end of each line ;
- Consequently, even more flexibility in use of whitespace
- Code is not caps-sensitive (convention = capitalize); table names are caps-sensitive

Getting started

`SHOW DATABASES;`	List available databases
`USE db;`	Choose which database to use
`SHOW TABLES;`	List tables in active database
`DESCRIBE table;`	List attributes in table

Creating Tables

```
CREATE TABLE newtable
(
id INT,
foreign_id INT NOT NULL, → minimum cardinality 1
numvar DECIMAL(n), → n=precision, e.g. 5
datevar DATE,
stringvar VARCHAR(n), → n=size, e.g. 20
PRIMARY KEY (id), → define primary key
UNIQUE (foreign_id), → maximum cardinality 1
FOREIGN KEY (foreign_id) → link to another table
    REFERENCES foreign_table(id)
);
```

SQL Selecting Data

The Query Framework

`SELECT attributes`	Choose columns to select
`FROM relation`	Choose table to select from
`WHERE condition`	Filter selection
`ORDER BY attributes`	Sort selection
`LIMIT n;`	Restrict number of results

Selecting Data

`SELECT *`	Select all columns
`SELECT attr`	Select specific column
`SELECT attr AS alias`	Select and rename in result
`SELECT attr1, attr2`	Select multiple columns
`SELECT DISTINCT attr`	Select unique values in column

Sorting and Limiting Results

`ORDER BY attr ASC`	Sort ascending
`ORDER BY attr DESC`	Sort descending
`ORDER BY attr1 ASC, attr2 ASC`	Sort by multiple cols
`LIMIT n`	Limit # rows

Notes:

- `ORDER BY` attributes don't need to appear in the results

Filtering Results with `WHERE`

We can test relationships with the following syntax:

`attr = 'text'`	Test equality (text)
`attr = val`	Test equality (numeric)
`attr in (val1, val2, ...)`	Check if in list
`attr not in (val1, val2, ...)`	Check if not in list
`cond1 AND cond2`	Apply both conditions
`cond1 OR cond2`	Apply either condition

As in Python, we can use comparison operators:

=	Equal (not ==)
<>	Not equal
!=	Not equal

>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Filtering Results with `WHERE`

We can also search for approximate matches:

- `%` Match any number of characters
- `_` Match one character

For example:

`attr LIKE 'str%'`	Starts with `str`
`attr LIKE '%str'`	Ends with `str`
`attr LIKE '%str%'`	Contains `str`
`attr LIKE 'str_'`	`str` + 1 character
`attr LIKE '_str'`	1 character + `str`

Bonus: We can use `REGEXP` for more flexible queries; e.g.

`attr REGEXP '[0-9]+'` returns all records containing a digit

Null values

Columns without a value are assigned to `NULL` values

`attr IS NULL`	Select null results
`attr IS NOT NULL`	Select non-null results

Notes:

- `NULL` is not the same as an empty string `''`
- We cannot test null values with the '=' sign

SQL Joins

Joins

`table1 INNER JOIN table2`	All records in both tables
`table1 LEFT OUTER JOIN table2`	All records in table 1
`table1 RIGHT OUTER JOIN table2`	All records in table 2

Joins