

Introduction to SQL

BA770 Lab Session

Questrom School of Business, Boston University

July 31, 2019

Keywords and Functions

- SELECT
- FROM
- DISTINCT
- COUNT
- WHERE
- AND/OR/NOT
- BETWEEN
- IN
- IS (NOT) NULL
- (NOT) LIKE
- MAX/MIN/SUM/AVG
- AS
- ORDER BY (DESC)
- GROUP BY
- HAVING
- LIMIT

Selecting Columns: SELECT, FROM

- **SELECT** expression
SELECT 'SQL'
- **SELECT** arithmetic operation
SELECT 4/3
- **SELECT** column1, column2, ...
FROM table
SELECT id, year
FROM people

Selecting Columns: DISTINCT, COUNT

- **SELECT DISTINCT** column
FROM table
SELECT DISTINCT year
FROM people
- **SELECT COUNT** (column)
FROM table
SELECT COUNT (*)
FROM people
- **SELECT COUNT** (**DISTINCT** column)
FROM table
SELECT COUNT (**DISTINCT** year)
FROM people

Filtering Rows: WHERE, AND/OR/NOT

- **SELECT** column **FROM** table
WHERE condition
SELECT id **FROM** people
WHERE age \geq 50
- **SELECT** column **FROM** table
WHERE conditions
SELECT id **FROM** people
WHERE (year = 2010 **OR** year = 2015)
AND (age $<>$ 40)
- Use \geq (\leq) to denote 'not less (larger) than', and $<>$ or \neq to denote 'equal'.
- Wrap conditions properly with parentheses.
- Do not forget to quote text values.

Filtering Rows: BETWEEN/IN

- **SELECT** column **FROM** table
WHERE expression **BETWEEN** num1 **AND** num2
SELECT id **FROM** people
WHERE age **BETWEEN** 30 **AND** 35
- **SELECT** column **FROM** table
WHERE expression **IN** (list_of_numbers)
SELECT id **FROM** people
WHERE age **IN** (30, 31, 32, 33, 34, 35)
- **BETWEEN...AND** is inclusive.
- You can negate the above result by prefacing **BETWEEN** or **IN** with **NOT** to exclude specified values.

Filtering Rows: IS (NOT) NULL, (NOT) LIKE

- **SELECT** column **FROM** table
WHERE column **IS (NOT) NULL**
SELECT id **FROM** people
WHERE birthdate **IS NOT NULL**
- **SELECT** column **FROM** table
WHERE column **(NOT) LIKE** 'pattern'
SELECT id **FROM** people
WHERE name **LIKE** 'B%'
- The '_' wildcard matches a single character.
- The '%' wildcard matches zero, one, or many characters.
- A missing value is not necessarily a **NULL** value. For example, some databases use extreme values like 0 or 99.9 to denote 'missingness'. Refer to database descriptions and properly handle missing values.