**INTRODUCTION TO SQL**

**Ch1: Relational databases**

Rows = ‘records’

Columns = ‘fields’

Table names and field names should be lowercase and not include spaces (use ‘\_’). If they do have a space, you need to enclose them in double quotes.

Field names should be singular

Field names should not share a name with the table name

There are different data types for strings, integers and floats, depending on the length.

DB schema: a db design with the tables included, their relationships, their fields and data types

**Ch2: Querying**

**Select fields from a table**

*SELECT name*

*FROM table;*

(It is best practice to end the query with a semi-colon to indicate that the query is complete)

*SELECT field1, field2*

*FROM table;*

*SELECT \**

*FROM table;*

**Select fields from a table and change their name**

*SELECT field1 AS my\_field, field2*

*FROM table;*

**Select unique values from a field**

*SELECT DISTINCT field1,*

*FROM table;*

**Select unique values from a combination of fields**

*SELECT DISTINCT field1, field*

*FROM table;*

**View**: a table that is the result of a saved SQL SELECT statement

When accessed, view automatically update in response to updates in the underlying data.

*CREATE VIEW view\_name AS*

*SELECT field2, field2*

*FROM table;*

Once a view is created, you can query it just as a normal table

*SELECT field2*

*FROM view\_name*

**SQL Flavors:** different SQL versions, dialects of the same language

* **PostgreSQL:** free, open-source, relational database system
  + Limit the number of records returned / return the first N records

*SELECT id, field1*

*FROM table*

*LIMIT 2;*

* **SQL Server:** has free and paid (T-SQL) versions, created by Microsoft
  + Limit the number of records returned / return the first N records

*SELECT id, field1*

*FROM table*

*TOP 2;*

**-- INTERMEDIATE SQL**

**-- Ch1: Selecting data**

**-- Return the number of records with a value in a field**

*SELECT COUNT(field1) AS count\_field1*

*FROM table;*

**Count multiple fields**

*SELECT COUNT(field1) AS count\_field1, COUNT(field2) AS count\_field2*

*FROM table;*

**-- Count the total number of records in a table**

*SELECT COUNT(\*) AS total\_records*

*FROM table;*

**-- Count the number of unique values of a field**

*SELECT COUNT(DISTINCT field2) AS count\_distinct\_field1*

*FROM table;*

**-- Order of execution:** SQL code is NOT processed in the order in which it is written.

-- First, the “FROM” statement, then “WHERE”, then the “SELECT” statement (before that the aliasing statements), then the “LIMIT” statement

**-- Debugging SQL code:** most common errors are misspelling, incorrect capitalization, and incorrect or missing punctuation, specially commas.

**-- SQL style:** formatting (new lines, capitalization, indentation) are NOT required, but there are style standards.

**-- Ch2: Filtering records**

**-- Filtering numbers**

*SELECT title*

*FROM films*

*WHERE release\_year > 2010;*

*SELECT title*

*FROM films*

*WHERE release\_year = 2010;*

*SELECT title*

*FROM films*

*WHERE release\_year <> 2010;*

**-- Filtering strings:** use single quotation marks

*SELECT title*

*FROM films*

*WHERE country = ‘Japan’;*

**-- Multiple criteria**

*SELECT \**

*FROM coats*

*WHERE color = ‘yellow’*

*OR color = ‘black;*

*SELECT \**

*FROM coats*

*WHERE color = ‘yellow’*

*AND length = ‘short’;*

*SELECT \**

*FROM coats*

*WHERE buttons*

*BETWEEN 1 AND 5;*

(between is INCLUSIVE)

*SELECT \**

*FROM coats*

*WHERE (color = ‘yellow’ OR color=’black’)*

*AND (length = ‘short’ OR length = ‘medium’);*

*Multiple OR conditions*

*SELECT \**

*FROM coats*

*WHERE color IN (‘red‘, ’black’, ‘blue’);*

**-- More string filtering techniques**

* **LIKE:** search for a pattern in a field
  + Wild card “%”: match zero, one or many characters

*SELECT name*

*FROM people*

*WHERE name LIKE ‘Jua%’*

Matches names like: Juan, Juan Bautista, Juan Alfredo. It is case sensitive.

* + Wild card “\_”: match a single character

*SELECT name*

*FROM people*

*WHERE name LIKE ‘Jua%’*

Matches names like: Juan, Juas

* **NOT LIKE:** match records that do not include a pattern

*SELECT name*

*FROM people*

*WHERE name LIKE ‘Juan%’*

**-- Find strings that end with ‘r’:**

*SELECT name*

*FROM people*

*WHERE name LIKE ‘%r’*

**-- Find strings that start with ‘B’:**

*SELECT name*

*FROM people*

*WHERE name LIKE 'B%'*

**-- Find strings that have ‘r’ as a second letter:**

*SELECT name*

*FROM people*

*WHERE name LIKE '\_r%'*