# SQL For Data Science – Week One:
## Basics and Retrieving Data with SQL

## Objectives

- Distinguish between use of SQL for data science applications and SQL for more common data management operations.
- Use an Entity Relationship diagram, describing the data elements, their relationships, and inter-dependencies and determine if the existent data is sufficient to address a business question.
- Retrieve one or more columns of data from a table that relates to the research topic.
- Identify a subset of data needed from a column or set of columns and write an SQL query to limit to those results.
- Create an analysis environment and use INSERT to put data into a table.
- Add effective comments in your queries so that one, you can remember what you're doing, and two, so others can review your work.

## What is SQL anyway?

- SQL – Structured Query Language for relational database management and data manipulation
- Standard language for data manipulation
  - Used to query, insert, update, and modify data
- Non-procedural language
  - Cannon write complete applications
- Three purposes
  - Read/Retrieve Data
  - Write Data
  - Update Data

## Data Models: Thinking about your data

- What is the problem you are trying to solve?
- Understand your data
  - Know the business process
  - Know the business rules
  - Understand the structure of your data
- Database
  - A container to store organized data/ a set of related information
- Tables
  - A structure list of data or a specific type
- Column
  - A single field in a table
- Row
  - A single record

Evolution of Data Models

- Organizes and structures information into multiple, related columns
- Can represent a business process or show relationships between business processes

Types of Data Models

- Prediction models
- Data relational models

NoSQL

- Not Only SQL
- A mechanism for storage and retrieval of **unstructured** data modeled by means other than tabular relations in relational databases

Relational vs Transactional Models

- Relational
    - Allows for easy querying and data manipulation in an easy, logical and intuitive way
- Operational Database
    - E.g. Insurance claims within a healthcare database
    - May need to be changed for relational model for analysis

Building Blocks

- Entity
    - An event/thing/person
- Attribute
    - A characteristic of an entity
- Relationship
    - Describes association among entities
- ER diagrams
    - https://www.smartdraw.com/entity-relationship-diagram/
- Primary Key
    - Column(s) that uniquely identify every row in table
- Foreign Key
    - Column(s) that can be used together to identify a single row in another table
- Basic Notation
    - 1:M
        - One to many
        - Painter can paint many walls
    - M:N
        - Many to Many
        - Employees can learn many skills
    - 1:1

    - One to one
    - A manager manages one store
  - Three types of notation
    - Chen notation
      - Use 1 and M
    - Crow’s foot
      - Uses crows feet for many and crow’s feet for one
    - UML Class Diagram Notation
      - Use * for Many

Select Statement

- SELECT prod_name FROM Products;
- SELECT * FROM Products
  - Selects all columns
- SELECT * FROM Products LIMIT 10;
  - How to limit results
  - **Syntax changes across databases**

Creating Tables

- When to create new tables
  - Create dashboards
  - Visualize in other tools
  - Use tables to make models
  - Extract data from other sources
- CREATE TABLE SHOES (
  ID char (10) PRIMARY KEY,
  BRAND char(10) NOT NULL);
- Primary keys cannot be null
- Null values are not empty strings
- Null values are the absence of everything

Insert into Tables

- INSERT INTO SHOES (Id, Brand)
  VALUES ( ‘123343’, ‘Gucci’);

Temporary Tables

- Will be deleted when current session is terminated
- Helpful for complex queries using subsets and joins
- Faster than creating a real table
- Statement
  - CREATE TEMPORARY TABLE Sandals AS
    ( SELECT * from shoes where shoe_type = ‘sandles’);
- Research how to update and delete tables

Adding Comments in SQL

- Single line
    - –
- Whole section
    - /* blah blah */

Star Schema vs Snowflake Schema

- Star Schema
    - Dimension tables are not normalized
    - Used in data marts
        - Subsets of data warehouses
- Snowflake
    - Dimensions are normalized
    - Used in data warehouses
        - Process of normalizing dimension tables is called snowflaking
    - Use less space
    - Fewer redundant
    - Can require more complex queries
        - Normalization requires deeper digs to get information
- How can we speed up reporting
    - Aggregations
    - Build a central storage area for all company’s aggregated data, not just sales data

SQL vs NoSQL

- SQL
    - Fixed Schemas
    - Scaling is vertical
        - Bigger servers
    - Data Structure isn’t changing
- NoSQL
    - Schemas are dynamic
    - Horizontal
        - Across servers
    - High throughput to handle viral data
    - NOT ACID complient