Course 6: Databases and SQL for Data Science with Python

Week 1:

Lesson 1: Basic SQL

Welcome to SQL for Data Science

- Why?
  - Demand is high
  - Median salary: $110,000
  - SQL is one of the top 3 skills for data scientists
- Big data
- Boost professional profile
- Given good understanding of relational databases

Introduction to Databases

- What is SQL
  - Language used for relational databases
  - Query data
- What is data?
  - Facts (words, numbers)
  - Pictures
  - One of the most critical assets of any business
  - Needs to be secure
- What is a database?
  - Repository of data
  - Program that stores data
  - Provides functionality for adding, modifying, and querying that data
  - Different kinds of databases store data in different forms
- Relational database
  - Data stored in tabular form
  - Columns contain item properties
  - Table is collection of related things
  - Relationships can exist between tables
- DBMS
  - Database: repository of data
  - DBMS: Database Management System - software to manage databases
  - Database, database server, database system, data server, dbms - often used interchangeably
- RDBMS

        - Relational database management system
        - Set of software tools that controls the data
            - Access, organization, storage
        - Examples:
            - MySQL
            - Oracle Database
            - IBM Db2
            - Etc.
- Basic SQL Commands
    - Create a table
    - Insert
    - Select
    - Update
    - Delete

SELECT Statement

- Retrieving rows from a table
- After creating a table and inserting data into the table, we want to see the data
- SELECT
    - A data manipulation language (DML) statement used to read and modify data
- SELECT * FROM <tablename>
- Retrieve a subset of columns
    - SELECT <column 1>, <column 2> FROM <table>
- Restricting the Result: WHERE Clause
    - Restricts the result set
    - Always requires a predicate:
        - Evaluates to:
            - True, False, or Unknown
        - Used in search condition of the Where clause

COUNT, DISTINCT, LIMIT

- Count
    - Built-in function that retrieves the number of rows matching the query criteria
- Distinct
    - Remove duplicate values from a result set
    - select DISTINCT country from medals
- Limit
    - Restricting number of rows retrieved from database

INSERT

- Populate table with data
- DML statement
- INSERT INTO [TableNam]
    - <(ColumnName,...)>

- VALUES (Value,...)
- Can insert multiple rows

UPDATE and DELETE Statements
- UPDATE
    - DML statement
    - Can alter data
    - UPDATE tableName
    - SET coulmnName = value
    - WHERE condition
    - If not specifying where, all rows will be affected
- DELETE
    - Remove 1 or more rows from the table
    - DML statement
    - DELETE FROM tableName
    - WHERE condition

Week 2:

Lesson 1: Introduction to Relational Databases and Tables

Relational Database Concepts
- Relational model
    - Most used
    - Data independence
    - Data is stored in tables
- ER model
    - Entity relationship model
    - Collection of entities
    - Used as a tool to design relational databases
- Mapping entity diagrams to tables
    - Entities become tables
    - Attributes get translated into columns
- Primary and foreign keys
    - Primary
        - Uniquely identifies each tuple / row in a table
        - Prevents duplication
    - Foreign
        - Primary keys from other tables

How to Create a Database Instance on Cloud
- Ease of use and access

        - Api
        - Web interface
        - Cloud or remote applications
    - Scalability & economics
        - sexpand/shrink storage & compute resources
        - Pay per use
    - Disaster Recovery
        - Cloud backups and geographical distribution
    - Examples
        - IBM Db2
        - Databases for PostgreSQL
        - Oracle Database Cloud Service
        - Microsoft Azure SQL Database
        - Amazon Relational Database Services (RDS)
    - Database Service Instances
        - DBaaS provides users with access to database resources in cloud without setting up hardware and installing software
        - Database service instance holds data in data objects/tables
        - Once data is loaded, it can be queried using web interfaces and applications
    - Creating a database instance on IBM Db2 on Cloud

Types of SQL Statements (DDL vs. DML)
- Data Definition Language vs. Data Manipulation Language
- DDL
    - Define, change, or drop data
    - CREATE, ALTER, TRUNCATE, DROP
- DML
    - Read and modify data
    - INSERT, SELECT, UPDATE, DELETE

CREATE TABLE Statement
- Syntax
    - CREATE TABLE table_name
        - {
        - column_name_1 datatype optional parameters,
        - column_name_2 datatype,
        - …
        - column_name_n datatype
        - }
- Example
    - CREATE TABLE provines {
        - id char(2) PRIMARY KEY NOT NULL,
        - name varchar(24)
        - }

ALTER, DROP, and Truncate Tables

- ALTER TABLE … ADD COLUMN
    - Add or remove columns
    - Modify the data type of columns
    - Add or remove keys
    - Add or remove constraints
- SET DATA TYPE changes column type
- DROP COLUMN
- DROP TABLE
    - Deletes all data along with table
- TRUNCATE TABLE
    - Deletes data (all rows) in table without deleting table itself
    - TRUNCATE TABLE table_name
    - IMMEDIATE;

Week 3:

Lesson 1: Refining your Results

Using String Patterns and Ranges

- WHERE requires a predicate
    - True, False, Unknown
- Use the LIKE predicate with string patterns for the search
    - WHERE columnName LIKE stringPattern
    - WHERE firstname LIKE R%
- Using a range
    - WHERE pages >= 290 AND pages <= 300
    - WHERE … OR …
    - WHERE … IN (first value, second value, …)

Sorting Result Sets

- ORDER BY …
    - By default sorted in ascending order
- ORDER BY … DESC
- ORDER BY 2 (2 refers to the column to order by)

Grouping Result Sets

- DISTINCT(value)
- GROUP BY value
- HAVING
    - Only works with GROUP BY clause

Lesson 2: Functions, Multiple Tables, and Sub-queries

Built-in Database Functions

- Most databases come with built-in SQL functions
- Built-in functions can be included as part of SQL statements
- Database functions can significantly reduce the amount of data that needs to be retrieved
- Can speed up data processing
- Aggregate or Column Functions
    - INPUT: Collection of values (ex. Entire column)
    - Output: Single Value
    - Ex: SUM(), MIN(), MAX(), AVG(), etc.
- SUM
    - Add up all values in a column
    - SUM(column_name)
- MIN, MAX
    - Returns minimum, returns maximum
    - MIN(column_name), MAX(column_name)
- AVG
    - Return average value
    - AVG(column_name)
- SCALAR and STRING FUNCTIONS
    - ROUND(), LENGTH(), UCASE, LCASE

Date and Time Built-in Functions

- DATE
    - 8 digits
- TIME
    - 6 digits
- TIMESTAMP
    - 20 digits (includes microseconds
- Functions
    - YEAR(), MONTH(), DAY(), DAYOFMONTH(), DAYOFWEEK(), DAYOFYEAR(), WEEK(), HOUR(), MINUTE(), SECOND()

Sub-Queries and Nested Selects

- Query inside another query
- Example situation
    - Retrieve a list of employees who earn more than the average salary
    - select EMP_ID, F_NAME, L_NAME, SALARY
    - from employees
    - where SALARY <
        - (select AVG(salary) from employees
- Column expression

    - Substitute column name with a sub-query
- Derived Tables / Table Expressions
    - Substitute table name with sub-query

Working with Multiple Tables
- Access multiple tables in same query
    - Sub-queries
    - Implicit JOIN
    - JOIN operators (INNER, OUTER, etc.)

Week 4:

Lesson 1: Accessing Databases Using Python

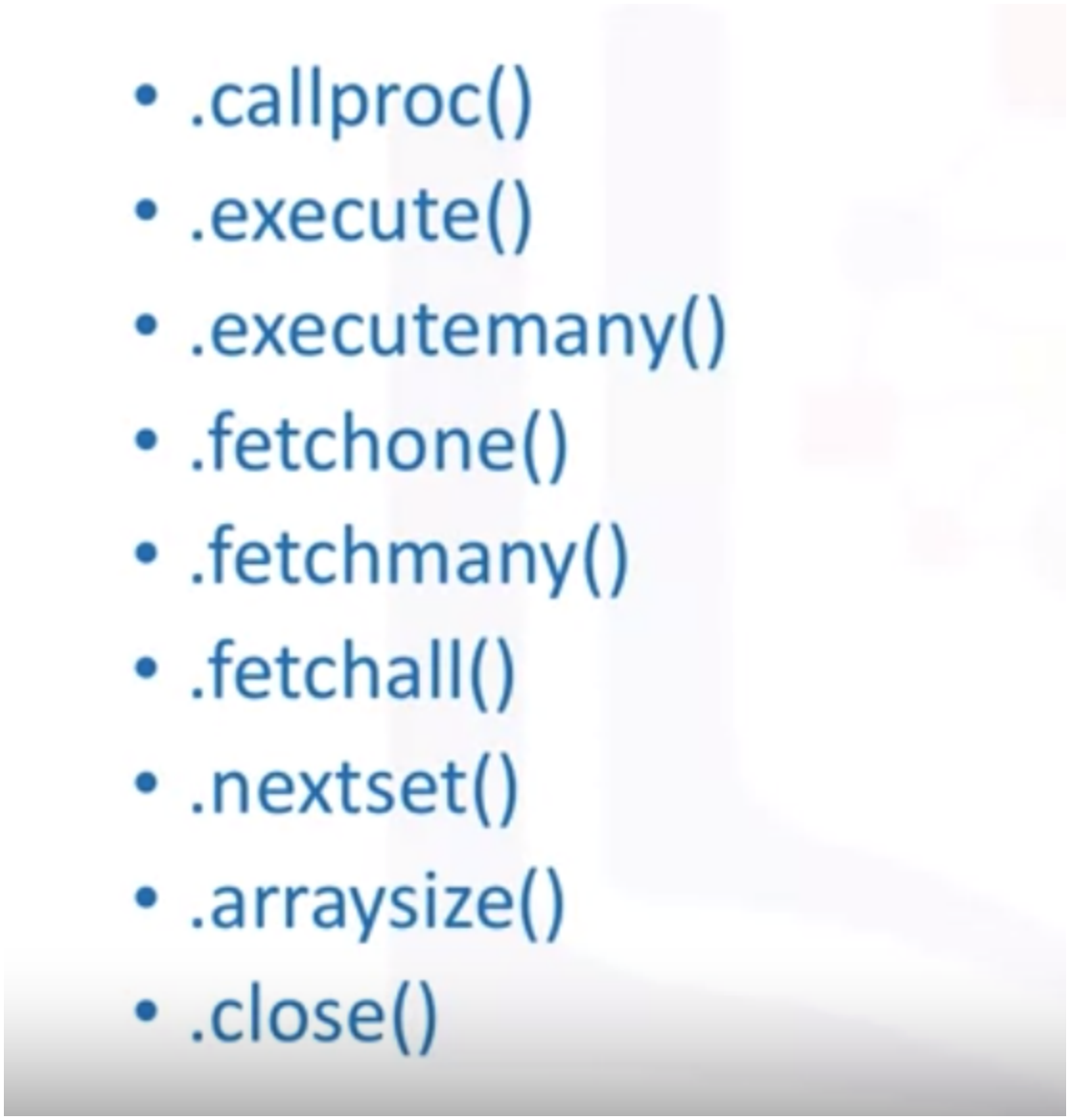
How to Access Databases Using Python
- Benefits of using Python
    - Ecosystem: NumPy, pandas, matplotlib, SciPy
    - Easy to use
    - Portable
    - Supports relational database systems
    - Database API (DB-API)
    - Documentation is easily available
- Notebooks
    - Matlab
    - Jupyter
        - Open source web app to share data science stuff
        - Language of choice
        - Share notebooks
        - Interactive output
        - Big data integration
- SQL API
    - CONNECT
    - SEND
    - EXECUTE
    - STATUS_CHECK
    - OK
    - DISCONNECT
- APIS used by popular SQL-based DBMS systems

APIs used by popular SQL-based DBMS systems

Application or Database	SQL API
MySQL	MySQL C API
PostgreSQL	psycopg2
IBM DB2	ibm_db
SQL Server	dblib API
Database access for Microsoft Windows OS	ODBC
Oracle	OCI
Java	JDBC

Writing Code Using DB-API

- Python Programs <-> DB API calls <-> DBMS
- Benefits
    - Easy to implement and understand
    - Encourages similarity between python modules
    - Achieves consistency
    - Portable across databases
    - Broad reach of database connectivity from Python
- Concepts
    - Connection objects
        - Database connections
        - Manage transactions
    - Cursor objects
        - Database queries
        - Scroll through result set
        - Retrieve results
- Methods
    - .cursor()
    - .commit()
    - .rollback()
    - .close()



- Database cursor
    - Control structure which allows transversal over a database
    - Keeps track of programs current position in query results
- from dbmodule import connect
- # Create Connection Object
- Connection = connect('databasename', 'username', 'pswd')

- # Create a Cursor Object
- Cursor=connection.cursor()

- # Run Queries
- Cursor.execute('select * from mytable')
- Results = cursor.fetchall()

- # Free Resources
- Cursor.close()
- Connection.close()

Connecting to a Database Using ibm_db API
- Provides a variety of useful Python functions for accessing and manipulating data in an IBM data server Database
- Uses IBM Data Server Driver for ODBC and CLI APIs to connect to IBM DB2 and Informix

Creating Tables, Loading Data, and Querying Data
- Connect to database
- Creating tables
    - .exec_immediate()
        - Connection
        - Statement
        - Options
    - Ex.
        - stmt = ibm_db.exec_immediate(conn, "…sql statement…")

SQL Magic Commands
- Cell magics
    - Start with a double %% sign and apply to the entire cell
- Line magics
    - Start with a single % sign and apply to a particular line in a cell
- Ex.
    - %sql select * from tablename

Analyzing Data with Python
- Load CSV file into DB2 on cloud
    - Source
        - Load spreadsheet using console
    - Target
        - schema
    - Define
        - Create new table, specify table name and define columns
    - Finalize
        - Finish selections
- df.describe(include='all')

Week 5:

Lesson 1: Assignment Preparation

Working with Real World Datasets

- Many are .CSV files
    - Comma Separated Values
- DOGS.CSV
- First row: Headers (not always)
- Specify within “” for mixed case column names
- \ splits queries into multiple lines

Getting Table and Column Details

- select * from syscat.tables
- CREATE_TIME
- select * from syscat.columns

Week 6:

Lesson 1: Views, Stored Procedures, and Transactions

Views

- Alternative way of representing data
- Can include specific columns from multiple base tables and existing views
- Shows selection of data for a given table
- Combine two or more tables
- CREATE VIEW viewname (columns)
    - AS SELECT
    - FROM

Stored Procedures

- Set of sql statements stored and executed on the database server
    - Can write in many different languages
    - Accept information as parameters
    - Return results to parent application
- Reduction in network traffic
- Improvement in performance
- Reuse of code
- Increase in security
- CREATE PROCEDURE procedureName (
    - LANGUAGE SQL
    - BEGIN

        - IF
        - ELSE
        - END IF
    - END

ACID Transactions
- Transaction
    - Indivisible unit of work
    - Consists of one or more sql statements
    - Either all happens or none
- Atomic
    - All changes must be performed successfully or not at all
- Consistent
    - Data must be in a consistent state before and after the transaction
- Isolated
    - No other process can change the data while the transaction is running
- Durable
    - The changes made by the transaction must persist

- BEGIN
    - Commands here are part of the transaction
- COMMIT
- ROLLBACK

Lesson 2: JOIN Statements

Join Overview
- Combines rows from two or more tables
- Based on a relationship
- Primary key
    - Uniquely identifies each row in a table
- Foreign Key
    - Refers to primary keys of other tables

Inner Join
- Most common type
- Displays only the rows from two tables which have a matching value in a common column
- Rows that do not have a matching value do not appear

Outer Join
- Left
    - All rows from left table are included, only matching ones from the right are included

- Right
    - Vice versa
- Full
    - All rows from both tables