**Databases and SQL for Data Science with Python**

**RDBMS (Relational Database Management System):**

The relational model is the most used data model for databases because this model allows for data independence. Data is stored in a simple data structure such as Tables.

**ERM (Entity Relationship Model):**

An entity relationship data model, or ER data model, is an alternative to a relational data model.

ERD that represents entities called tables and their relationships. In the library example, we have books. A book can be written by one or many authors. The library can have one or many copies of a book. Each copy can be borrowed by only one borrower at a time.

An entity relationship model proposes thinking of a database as a collection of entities rather than being used as a model on its own.

The ER model is used as a tool to design relational databases. In the ER model, entities are objects that exist independently of any other entities in the database.

The building blocks of an ER diagram are entities and attributes. An entity can be a noun: person, place, or thing.

In an ER diagram, an entity is drawn as a **rectangle**.

Entities have attributes which are the data elements that characterize the entity.

**Attributes:**

Attributes tell us more about the entity.

In an ER diagram, attributes are drawn as **ovals**. Using a simplified library as an example, the book is an example of an entity.

Attributes are certain properties or characteristics of an entity and tell us more about the entity.

The entity book has attributes such as book title, the edition of the book, the year the book was written, etc

Attributes are connected to exactly one entity. The entity book becomes a table in the database and the attributes become the columns in a table.

A table is a combination of rows and columns. While mapping, the entity becomes the table.

Having said that, the table has not yet taken the form of rows and columns. The attributes get translated into columns in a table providing the actual table form of rows and columns. Later, we add some data values to each of the columns, which completes the table form.

Each attribute stores data values of different formats, characters, numbers, dates, currency, and many more besides. In the book table example, the title is made up of characters.

**Week 4: Accessing Databases using Python**

The MySQL Connector/Python library is used to connect to a Compose for MySQL database.

The psycopg2 library is used to connect to a Compose from PostgreSQL database. And finally, the PyMongo library is used to connect to a Compose for MongoDB database.

The two main concepts in the Python DB-API are connection objects and query/cursor objects.

You use connection objects to connect to a database and manage your transactions.

Cursor objects are used to run queries. You open a cursor object and then run queries. The cursor works similar to a cursor in a text processing system where you scroll down in your result set and get your data into the application. Cursors are used to scan through the results of a database.

**Connection Methods:**

The DB\_API includes a connect constructor for creating a connection to the database. It returns a Connection Object, which is then used by the various connection methods. These connection methods are:

**The cursor()** method, which returns a new cursor object using the connection.

**The commit()** method, which is used to commit any pending transaction to the database.

**The rollback()** method, which causes the database to roll back to the start of any pending transaction.

**The close()** method, which is used to close a database connection.

These objects represent a database cursor, which is used to manage the content of a fetch operation.

Cursors created from the same connection are not isolated, that is, any changes done to the database by a cursor are immediately visible by the other cursors. Cursors created from different connections can or cannot be isolated depending on how the transaction support is implemented.

**What are the cursor methods?**

* **.callproc()**
* **.execute()**
* **.executemany()**
* **.fetchone()**
* **.fetchall()**
* **.fetchmany()**
* **.nextset()**
* **.arrarysize()**
* **.close()**

**Database Cursor:**

A database cursor is a control structure that enables traversal over the records in a database. It behaves like a file name or file handle in a programming language.

Just as a program opens a file to access its contents, it opens a cursor to gain access to the query results. Similarly, the program closes a file to end its access and closes a cursor to end access to the query results

**Week 5:**