**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, a 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:**

**Week 6:**

**Views:**

A view is an alternative way of representing data that exists in one or more tables. Just like a real table, it contains rows and columns. The fields in a view are fields from one or more real tables in the database. Though views can be queried like a table, views are dynamic; only the definition of the view is stored, not the data.

**Stored Procedures:**

A stored procedure is a set of SQL statements that are stored and executed on the database server.

So instead of sending multiple SQL statements from the client to server, you encapsulate them in a stored procedure on the server and send one statement from the client to execute them.

You can write stored procedures in many different languages. For example, for Db2 on Cloud and DB2 you can write in SQL PL, PL/SQL, Java, C, or other languages.

They can accept information as parameters, perform create, read, update, and delete (CRUD) operations, and return results to the client application.

**The benefits of stored procedures include:**

Reduction in network traffic because only one call is needed to execute multiple statements.

1. Improvement in performance because the processing happens on the server where the data is stored, with just the final result being passed back to the client.
2. Reuse of code because multiple applications can use the same stored procedure for the same job.

3- Increase in security because

a) you do not need to expose all of your table and column information to client-side developers and

b) you can use server-side logic to validate data before accepting it into the system.

**ACID:**

A = Atomic - All changes must be performed successfully or not at all

C = Consistent - Database must be consistent in state before and after the transaction

I = Isolated – No other process can change the data while the transaction is running

D = Durable – The change made by the transaction must persist.

ACID is an acronym that refers to the set of 4 key properties that define a transaction: Atomicity, Consistency, Isolation, and Durability. If a database operation has these ACID properties, it can be called an ACID transaction, and data storage systems that apply these operations are called transactional systems.

A diagram of a diagram

Description automatically generated

A transaction is simply a sequence of operations performed using one or more SQL statements as a single logical unit of work. A database transaction must be ACID (Atomic, Consistent, Isolated and Durable). The effects of all the SQL statements in a transaction can either be applied to the database using the COMMIT command or undone from the database using the ROLLBACK command.

**Summary & Highlights**

Views are a dynamic mechanism for presenting data from one or more tables.A transaction represents a complete unit of work, which can be one or more SQL statements.

* An ACID transaction is one where all the SQL statements must complete successfully, or none at all.
* A stored procedure is a set of SQL statements that are stored and executed on the database server, allowing you to send one statement as an alternative to sending multiple statements.
* You can write stored procedures in many different languages like SQL PL, PL/SQL, Java, and C.