**Datenbankanbindung in Python & Bioinformatische Anwendungen**

Arbeiten mit Datenbanken ist ein zentraler Aspekt der Bioinformatik. SQL- Datenbankmanagementsysteme werden häufig verwendet und können mit Python angesteuert werden, um **CRUD Operationen** (Create, Read, Update, Delete) durchzuführen.

**MySQL** ist ein Client-Server-Datenbanksystem und eignet sich besonders für Webanwendungen sowie größere Projekte, die gleichzeitige Datenbankzugriffe von mehreren Benutzern oder Anwendungen erfordern. Die Daten werden in mehreren Dateien und Tabellenstrukturen organisiert, die von der Datenbank-Engine verwaltet werden. Die Verbindung mit Python erfolgt über Module wie **mysql-connector** oder **PyMySQL**, die die Ausführung von SQL-Befehlen ermöglichen. Das Modul **SQLAlchemy** bietet zusätzlich die Möglichkeit des Object-Relational Mapping (ORM).

**SQLite** ist eine eingebettete Datenbank, die alle Daten in einer einzigen Datei speichert, die lokal abgelegt ist. Die Verbindung mit Python erfolgt direkt über das **sqlite3**-Modul oder ebenfalls über **SQLAlchemy**.

**Object-Relational Mapping:** Die Tabellen der Datenbank werden als Python-Klassen behandelt, die die Struktur der Datenbanktabelle widerspiegeln. Eine Klasse repräsentiert eine Tabelle in der Datenbank. Die Attribute der Klassen entsprechen den Spalten, die Instanzen der Klasse entsprechen den Zeilen.

**SQLite3** ist eine leichtgewichtige, serverlose SQL-Datenbank-Engine, die sich besonders für eingebettete Anwendungen und lokale Datenspeicherung eignet. Die komplette Datenbank wird in einer einzelnen Datei gespeichert. Hauptmerkmale:

* Keine separate Server-Installation nötig
* Teil der Python-Standardbibliothek über das **sqlite3**-Modul
* Unterstützt ACID-Transaktionen (Atomicity, Consistency, Isolation, Durability)
* Ideal für kleinere bis mittlere Datensätze und Einzelbenutzer-Anwendungen

**Entrez** ist das Programmierschnittstellen-System (API) des NCBI (National Center for Biotechnology Information). In Python wird es über die **Bio.Entrez**-Bibliothek aus BioPython genutzt und ermöglicht den Zugriff auf wichtige biologische Datenbanken:

* PubMed (Literatur)
* GenBank (Sequenzen)
* Protein und Nucleotide Datenbanken
* Die E-utilities erlauben automatisierte Abfragen und das Herunterladen von Datensätzen (esearch, efetch, esummary, elink)

**BioSQL** ist ein Projekt zur Standardisierung biologischer Datenbank-Schemata. Es ermöglicht:

* Speicherung von Sequenzen und Annotationen
* Integration mit BioPython über das **Bio.BioSQL**-Modul
* Unterstützung verschiedener SQL-Datenbanken (PostgreSQL, MySQL, SQLite)
* Konsistente Datenhaltung für biologische Sequenzdaten BioSQL kann als Backend für lokale Sequenz-Datenbanken verwendet werden und erlaubt die effiziente Verwaltung von GenBank/SwissProt-Einträgen.

Die Kombination dieser Tools ermöglicht die systematische Verwaltung und Analyse biologischer Daten: Entrez zum Abrufen von Online-Ressourcen, SQLite3 für lokale Datenspeicherung und BioSQL als standardisiertes Schema für biologische Datenbanken.