

# Esercitazione di preparazione all'esame per il corso di "Data Base & Data Analytics"

## (SQL, SQLAlchemy Core e ORM )

[ In questo documento ci sono soltanto considerazioni generiche, che non riguardano i quesiti posti ]

Come dovrebbe essere ormai evidente, a valle delle esercitazioni svolte sulle varie parti, in SQLAlchemy ci sono molti modi di "colloquiare" con una base di dati.

Come già evidenziato a lezione, la gerarchia è la seguente:

<b>Livello di astrazione</b>	<b>Nota</b>
SQLAlchemy ORM con relazioni	Object Relation Model con relazioni
SQLAlchemy ORM senza relazioni	Object Relation Model
SQLAlchemy Core	Usa SQL Core language
SQL	Si usa direttamente SQL query language mediante stringhe o istanze di 'text'

Il Core di SQLAlchemy consente nella sostanza di realizzare una mappatura diretta tra SQL e istruzioni in Python. Per le interrogazioni qui si usa tipicamente il costrutto `select` e le query vengono eseguite mediante una `connection` (vedere `engine.connect`).

La parte ORM invece consente di dialogare con il database tramite classi e oggetti che rappresentano righe di tabelle del database. La sincronizzazione tra le "righe" di tabelle presenti nel programma e quelle delle tabelle viene gestita completamente da SQLAlchemy ORM. Le interrogazioni qui vengono eseguite mediante una `session` (vedere la classe `Session`).

### NOTA BENE

Le relazioni ORM non costituiscono la soluzione di ogni problema. Possono essere molto utili se le principali query sono in accordo con il contenuto di una o più relazioni, inutili o addirittura "dannose" a livello di efficienza. Il loro uso deve quindi essere subordinato al fatto che siano effettivamente utili. Va ricordato, inoltre, che –al di là dei meccanismi automatici implementati da SQLAlchemy per la loro gestione– le relazioni sono a tutti gli effetti delle liste Python.