SQL

Structured Query Language

```
class A:
    def __init__(self, a):
        self.a = a
    def __sub__(self, other):
        return self.a - other.a
class B(A):
    def __init__(self, a):
        super().__init__(a)
    def __sub__(self, other):
        return self.a - other.a - 3
class C(B):
    def __init__(self, a):
        super().__init__(a)
    def __add__(self, other):
        return self.a + other.a

obj1 = C(20)
obj2 = C(30)
print(obj1 - obj2)
```

Question from MRO (Method Resolution Order)

obj1 = C(20)

--sub-- in C ✗

--sub-- in B ✓

```
def __sub__(self,other):
    return self.a - other.a - 3
```

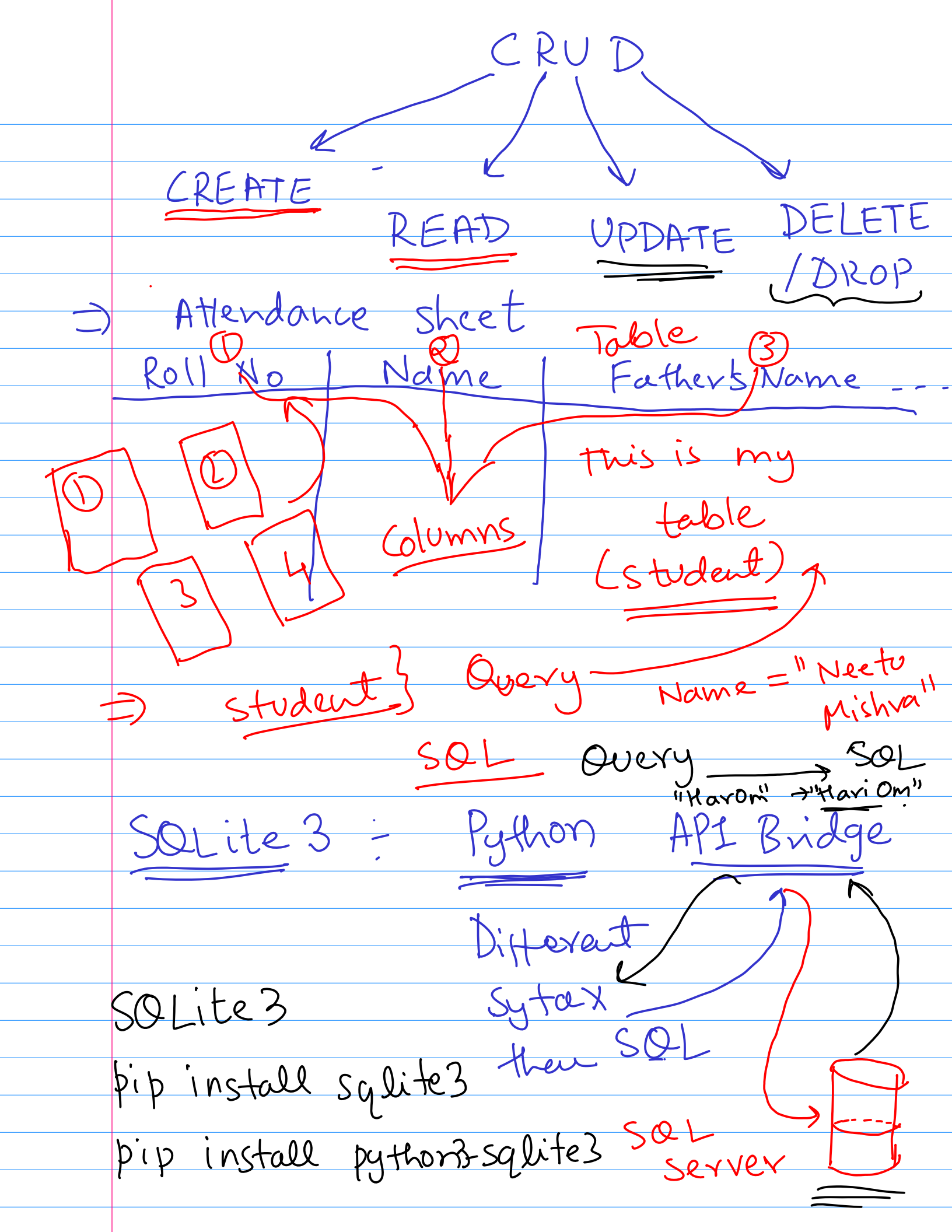
>> -13 ✓

Data → .csv files → Banking systems

We require Database Management System

RDBMS (Relational)

it works on CRUD Principle


CRUD
CREATE
READ
UPDATE
DELETE
/DROP
Attendance sheet
Table
Roll No
Name
Father's Name
1
2
3
4
Columns
this is my table (student)
student
Query
Name = "Neetu Mishra"
SQL
Query
SQL
"Harom" → "Hari Om"
SQLite3 : Python API Bridge
Different Sytax then SQL
SQLite3
pip install sqlite3
pip install python3-sqlite3
SQL server

pointing to some table

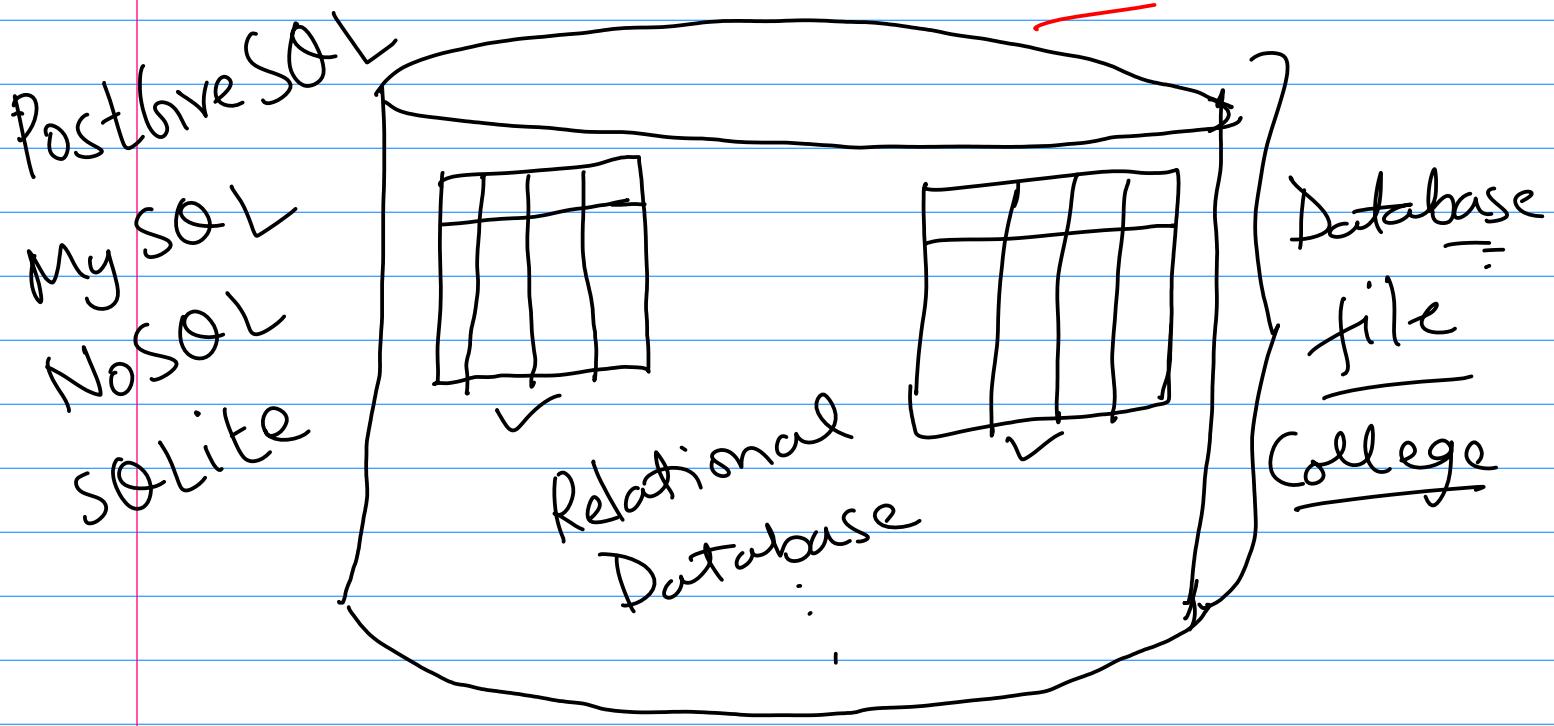
```
select * from tracks;
```

Read Command — Every Column — table the information being queried from.

pointing to a table

```
Select Name from tracks;
```

Reading — Column — Table.



SQL :- Query Language