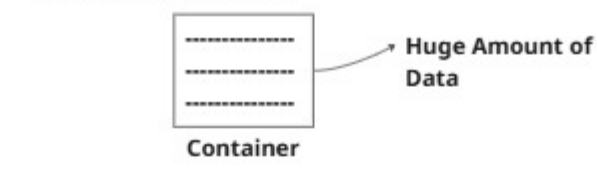
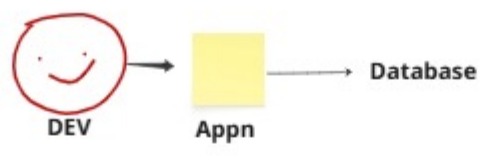
MySQL

Data was stored by the people(Dev) inside a FILE

Storing that HUGE amount of data inside a file was not a good approach

Solution ==> DATABASE





1) Dev should create a new database
2) Store the data
3) Delete that database

Developer ==> SOFTWARES ==>
Internet ==> create any no of databases, delete the database, ...operation

Oracle, MySQL, MongoDB .....(Database Management Softwares) ==> DBMS
==> Softwares to manage databases

DBMS ==> MySQL ==> create database, store the data, read that data, updated the data, delete the data, ...

Database is Ready ==> create a table(rows and columns) ==> store the data

Africa, London, Berlin, London

Africa, Berlin

If multuple customers have same city, then i want to print those details in the descending order of the customer name.

```
select * from Customers order by City asc, CustomerName desc;
```

Print the name of the top 3 products that has lowest price

```
select ProductName from Products order by Price limit 3;
select ProductName from Products order by price desc limit 1 offset 1;
```

```
select * from Products where Price = 10;
select * from Products where Price > 20;
select * from Products where Price < 20;
select * from Products where Price != 20;
select * from Products where Price between 10 and 50;(ranges)
select * from Products where Price not between 10 and 50;(ranges)
SELECT count(ProductName) as products_count FROM Products;
SELECT count(*) as products_count FROM Products;
select sum(Price) as total_price from Products;
select max(Price) from Products;
select min(Price) from Products;
select avg(Price) from Products;
```

id	name	department	salary
1	Raju	IT	50000
2	Mary	HR	40000
3	John	IT	60000
4	Alex	HR	45000
5	Cook	IT	55000

IT HR

1	Raju	IT	50000
3	John	IT	60000
5	Cook	IT	55000

2	Mary	HR	40000
4	Alex	HR	45000

1️⃣ Total number of employees in each department
IT 3
HR 2

```
select department, count(*) as department_count
from employees group by department;
```

2️⃣ Average salary for each department
IT 55000
HR 42500

3️⃣ Maximum salary in each department
IT 60000
HR 45000

```
select department, max(salary) as maximum_salary
from employees group by department;
```

4️⃣ Minimum salary in each department
IT 50000
HR 40000

5️⃣ Total salary paid per department
IT 165000
HR 85000

```
select department,
sum(salary) as total_salary from employees group by department;
```

1️⃣ Departments having more than 2 employees
IT 3

```
select department, count(*) as total_count
from employees group by department having total_Count > 2;
```

2️⃣ Departments where average salary > 50000
IT 55000

3️⃣ Departments whose total salary exceeds 100000
IT 165000

5️⃣ Departments having maximum salary > 55000
IT 60000

unique + not null ==> primary key

primary key is a column that always stores unique values with not null feature.

Data ==> Table

students.m_id = marks.marks_id;

Students

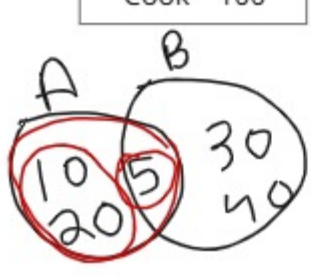
rollno	name	m_id
1	Raju	100
2	Ramu	100
3	Rani	300
4	Cook	400
5	Sam	

Marks

marks_id	marks
100	54
200	78
300	90
400	100
500	47

```
select students.name, marks.marks from students
left join marks
on students.m_id = marks.marks_id;
```

Raju	54
Ramu	54
Rani	90
Cook	100



(A − B) ∪ (A ∩ B)
10, 20 ∪ 5
10, 20 and 5

A n B ==> 5 ==> INNER JOIN
A - B ==> 10, 20
B - A ==> 30, 40

A ∪ B
outer join

A n B ==> INNER JOIN
(A − B) ∪ (A ∩ B) ==> 10,20 and 5 ==> LJ
(B − A) ∪ (A ∩ B) ==> 30,40 and 5 ==> RJ
A ∪ B ==> 10, 20, 30, 40 and 5