Type something

attribute ==> not null
rollno ==> not null
name ==> not null
age ==> not null

rollno column ==> unique + not null
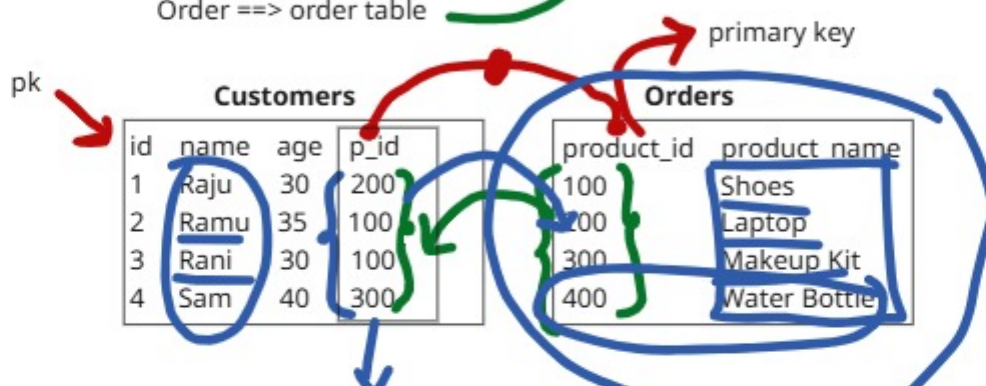rollno column ==> primary key
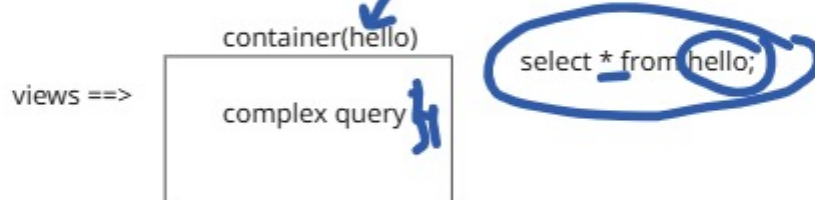
primary key = unique + not null

Project ==> multiple table

Customer ==> customer table
Order ==> order table

primary key

pk

Customers

id	name	age	p_id
1	Raju	30	200
2	Ramu	35	100
3	Rani	30	100
4	Sam	40	300

Orders

product_id	product_name
100	Shoes
200	Laptop
300	Makeup Kit
400	Water Bottle

Foreign Key ==> column taht connects to the primary key of the anoother table.

container(hello)

complex query

views ==>

select * from hello;

index   details

name   email
--------   j@gmail.com
--------   s@gmail.com
--------- t@gmail.com
Raju   r@gmail.com
....
1 lakh records

select * from details;
select * from details where email = 'g@gmail.com'

Which ever the column we use to read the data,
in order to speed up the reading process for
that particular column, we create index.

creating index for email column

create index email_index on details(email)

MySQL ==> takes all the details in the email column
and arrange them in sorted order

details table

name   email
--------   j@gmail.com
--------   s@gmail.com
--------- t@gmail.com
Raju   r@gmail.com
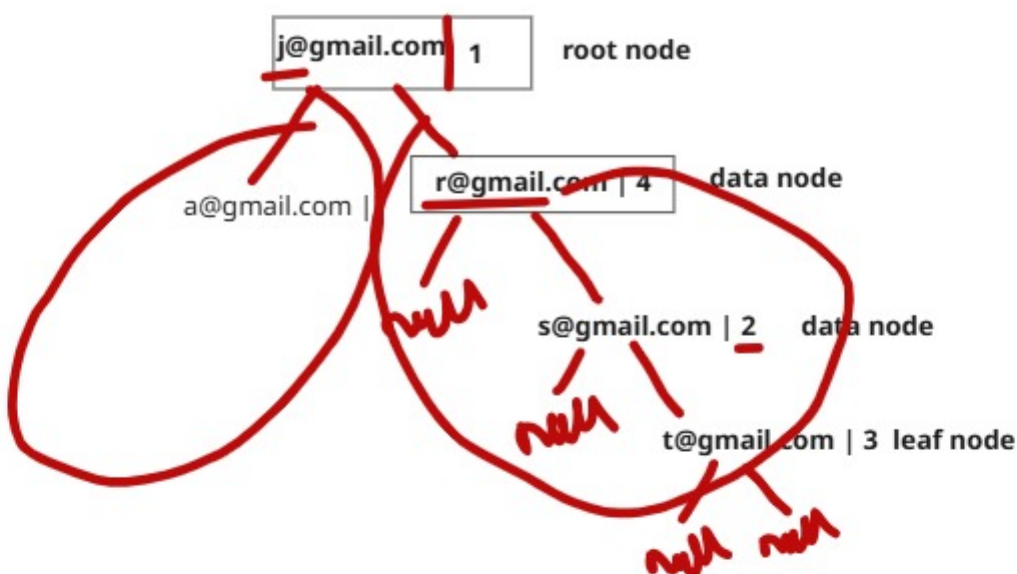...
1 lakh records

j@gmail.com ==> Row No 1
r@gmail.com ==> Row No 4
s@gmail.com ==> Row No 2
t@gmail.com ==> Row No 3

B Tree(Data Structure) ==> Balanced Tree

j@gmail.com   1   root node

r@gmail.com | 4   data node

a@gmail.com |

s@gmail.com | 2   data node

t@gmail.com | 3 leaf node

select * from details where email = 's@gmail.com'

100 tables

Project ==> one DB(MySQL) ==> So many tables............
Development   CRUD
Devops ==> R
Ops(3) ==> U
....

grant ==> grant will give permission and revoke
will remove the permissions

OPS ==> 3 ==> Update on MySQL employeeDB.employees

For each person one separate account will get created(username and
password)
3 people ==> 3 username and 3 password ==> database admin

OPS ==> 3 ==> 1 ==> update access on customers

root user ==> parent most user

grant update, select, delete on
customer_order_db.customers
to "Raju";

revoke select, update on customer_order_db.customers
from "Raju";

E(Entity)-R(Relationship) diagram

Entity ==> Stores data(table, views,
stored prodecures)

distinct keyword ==> unique values from that column
SELECT distinct City FROM Customers;