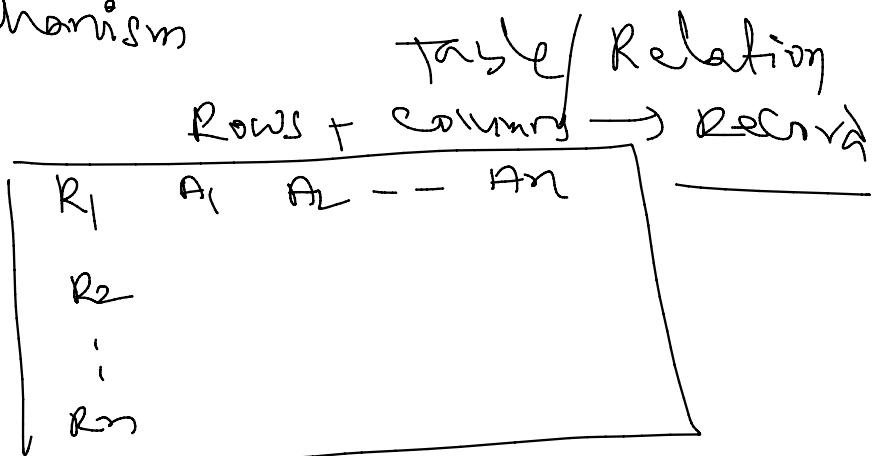


- ① Database  
 ② Table  
 ③ Attributes
- Components

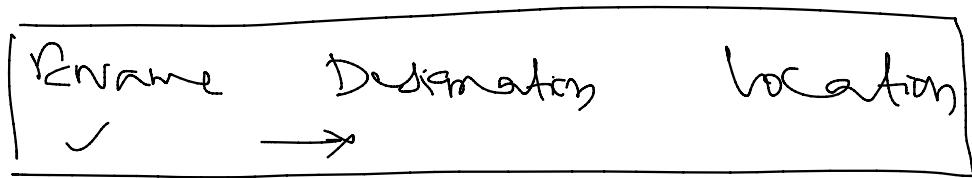
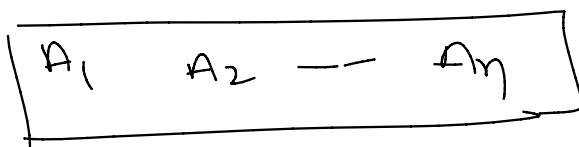
DB : Storage (Bulk Data)

Access mechanism

Record →



✓ Relation →



Dr.Uma	Professor	Bangalore
ramaiah	Pr. in. M.	Chennai

Vivek  
Pranav

Professor  
Scientist

Vivek  
Chennai

## Functional Dependency \*

MySQL

Schema →

DB → ① Schema → the overall Design of the Database  
② Instance → the Database at a particular moment  
③ Data Dictionary → Details of DB objects

a) Object : Having property/state/  
Database object

Priyanka

Afshan

(EmpId → 17262 ✓ 17363)

Designation → Data Engineer  
Speciality → Python

Data  
Python

Inyanca

Specialization → Prospects

purpose

Entity → Properties

✓ State / ✓ Unique / ✓ Properties

Object

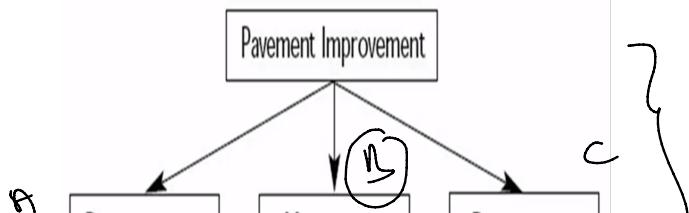
Database Object

Table → DB Object → Row x Col  
 View  
 Index  
 Tracing  
 Function  
 Stored procedures  
 Cursors  
 Trigger

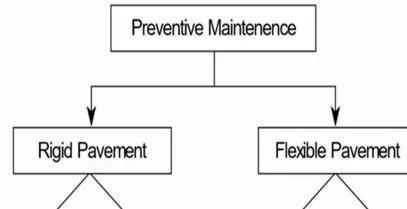
Records  
 attributes

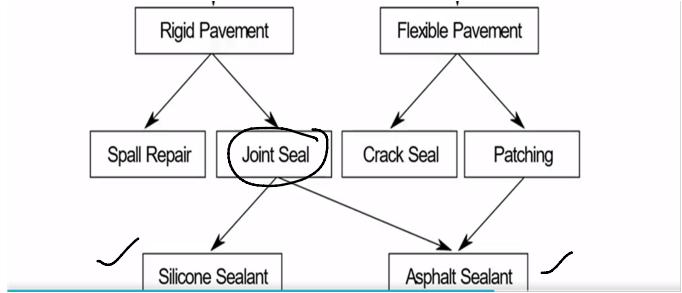
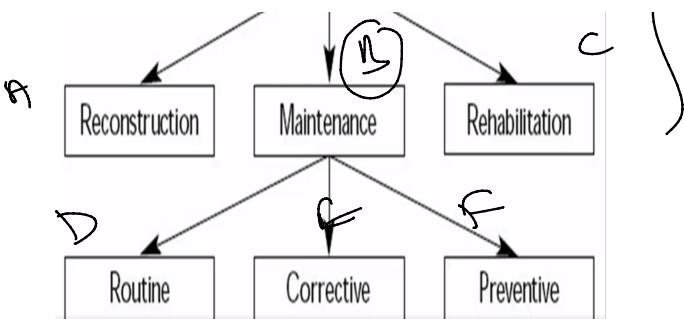
✓ Hierarchical Model

Rm



✓ Network Model





## Relational model:

Diagram illustrating the Relational model with three tables:

- Activity Table:** Shows unique activities with their codes and names.
- Task Table:** Shows tasks with dates, activity codes, and route numbers.
- Log Table:** Shows log entries with dates, activity codes, and route numbers.

Relationships:

- The Activity Table is linked to the Task Table via the Activity Code column.
- The Task Table is linked to the Log Table via the Activity Code column.
- A key value of 24 is highlighted in the Activity Table, which serves as a primary key for the Task Table.

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Unique

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495

Unique

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Key = 24

Repeat

Day 2:

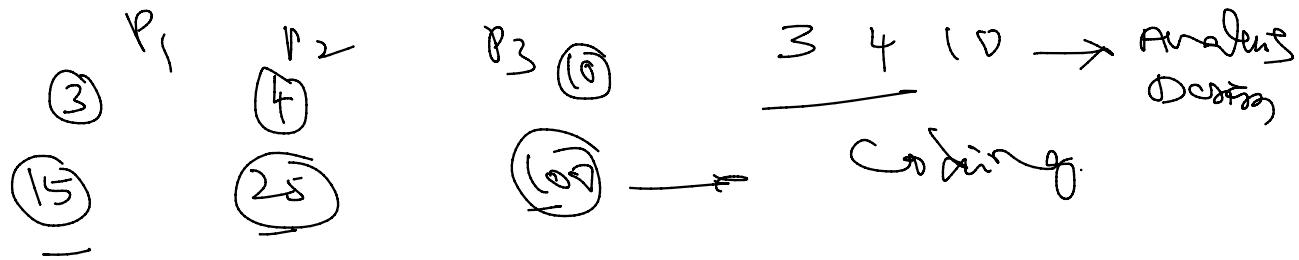


- (a) Distinguish b/w Hierarchical, Network & Relational model in the context of storage, access

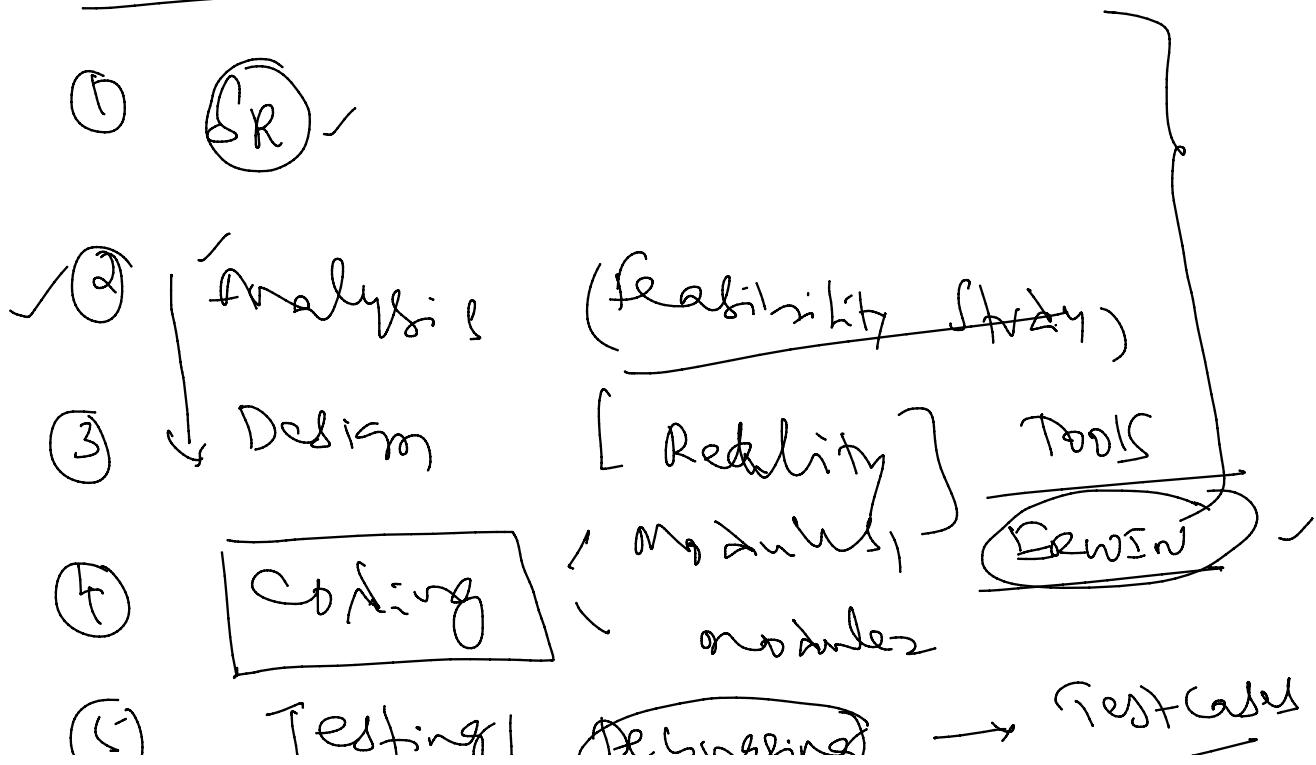
- (b) Explain the advantages of Relational model over the Hierarchical & Network model.

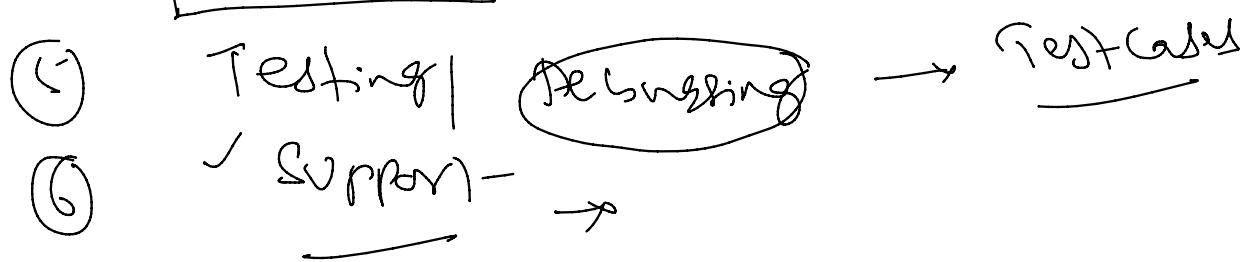
Relational DB }  
 Schema  
 Instance  
 DB objects

SQL ↗ Emp → Design / Analysis  
 ↗ Moderate/ Brit → Coding



S E process



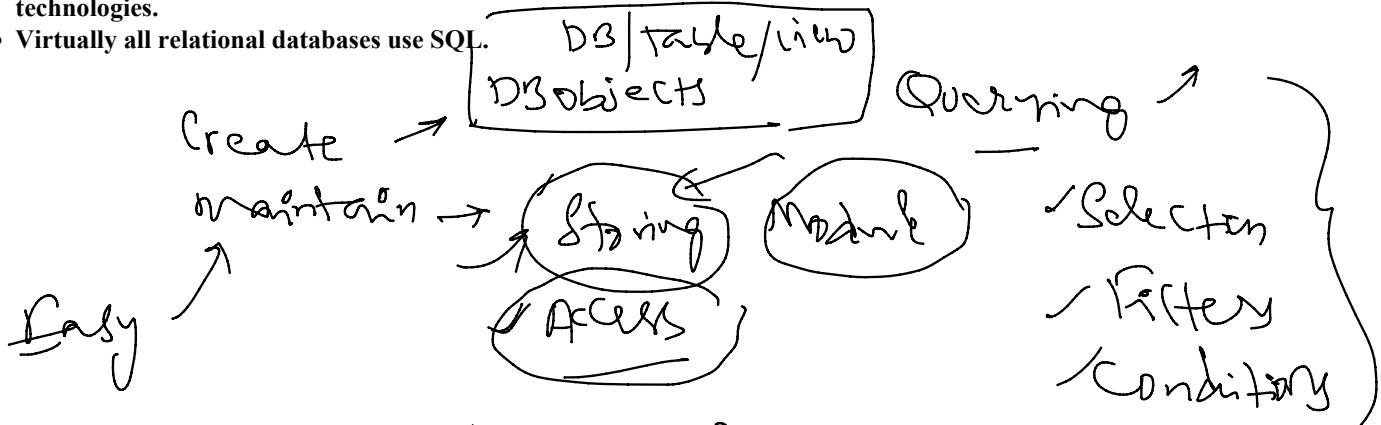


SQL

MySQL

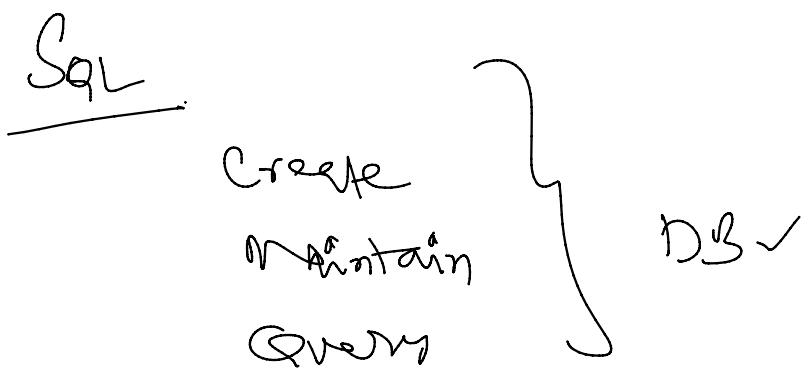
What is SQL:

- A computer Language that was created by IBM decades ago.
- Can be used to create, Maintain, and query relational databases.
- SQL was originally named as SEQUEL (All Caps).
- Structured English QUERy Language.
- You can use SQL to craft Queries.
- The syntax of SQL is like the syntax of English, but more structured.
- When IBM want to release the product to market there was a copyright problem with SEQUEL.
- IBM dropped the vowels and called it SQL.
- SQL is standardized by ISO (International Organization for Standardization)/IEC (International Electrotechnical Commission) 9075.
- ISO/IEC 9075 to specify the grammar of SQL and the result of processing statements in that language by an SQL-implementation.
- ISO focuses on standards for quality management systems, product-, material and construction standards, whereas IEC mainly focuses on all the international standards for electrical, electronic, and related technologies.
- Virtually all relational databases use SQL.



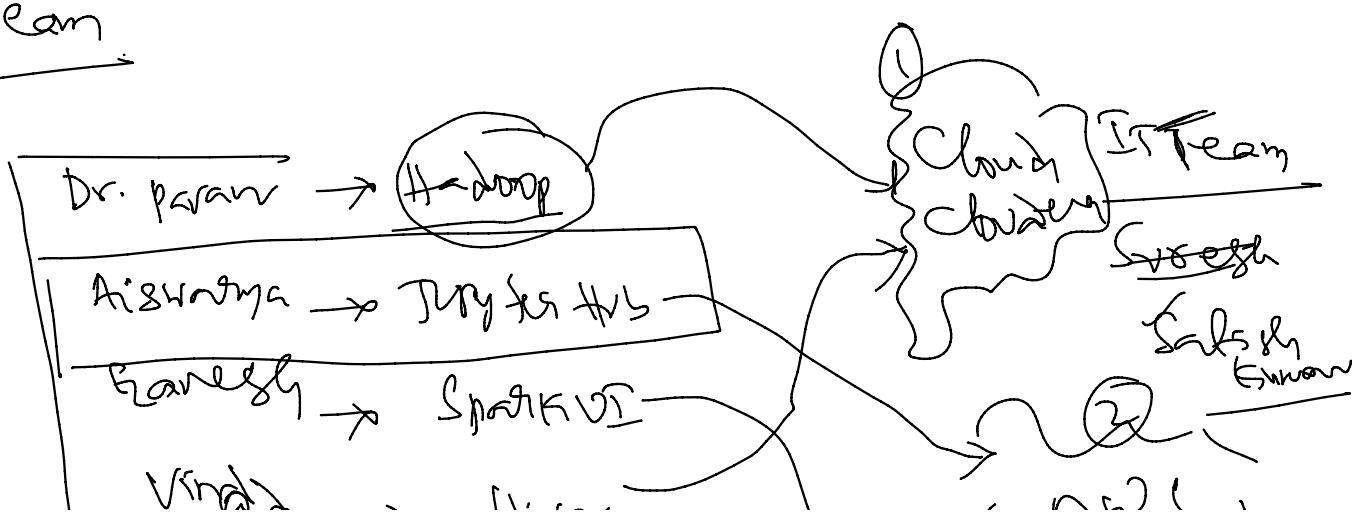
Process / logical Connect

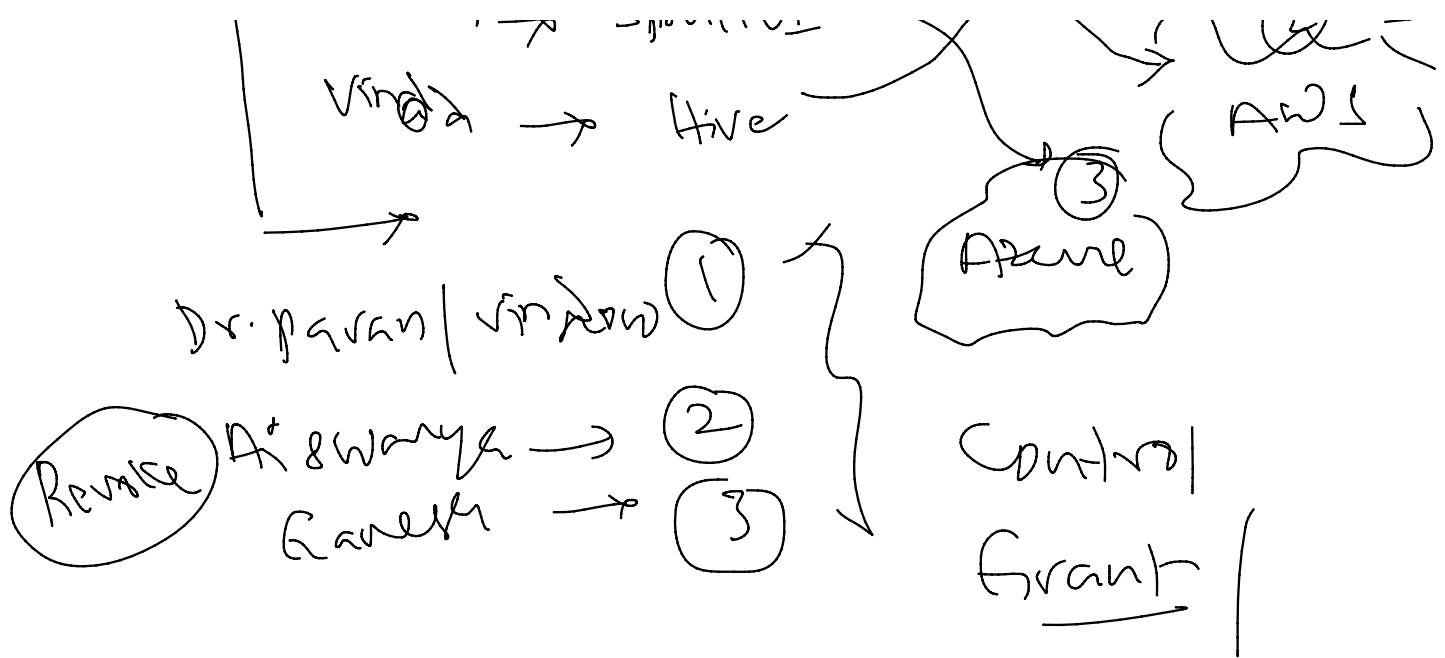




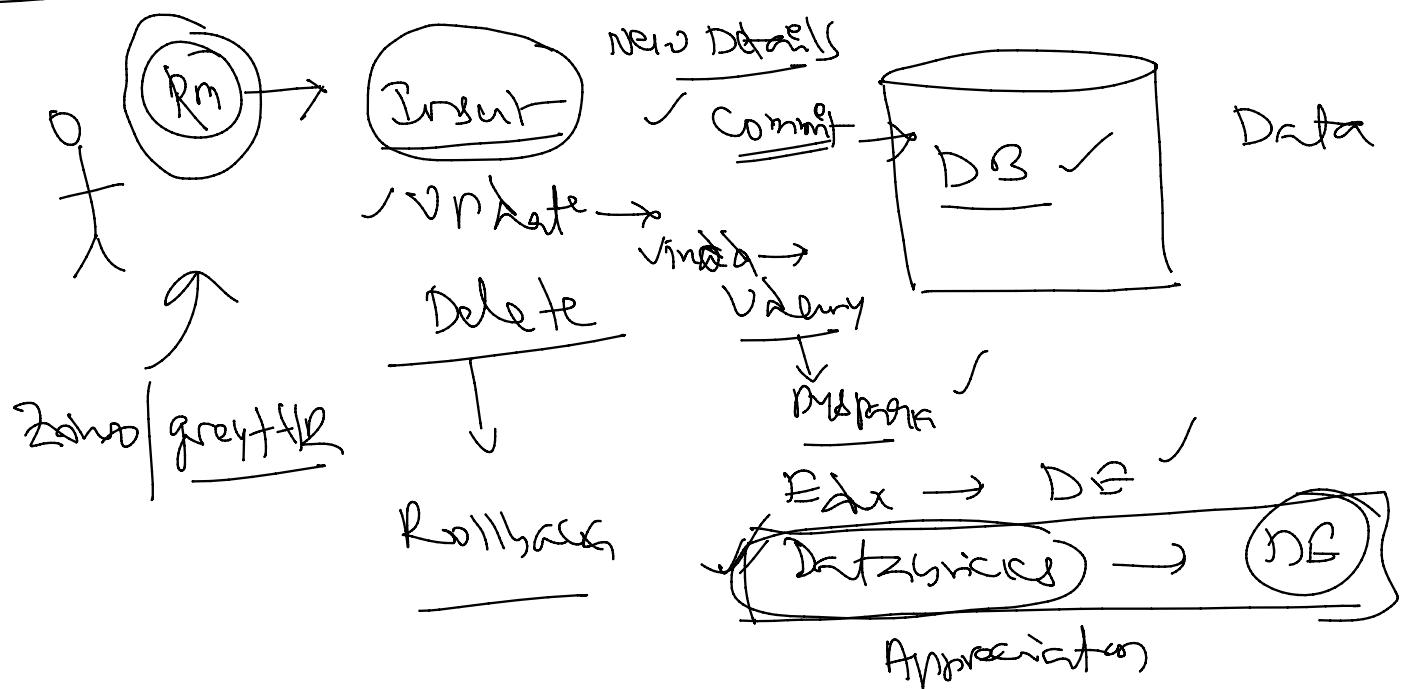
- ✓ a) Data Definition Language [Create]
- ✓ b) Data Manipulation Language [Query]
- c) Data Control Language [Grant-Revocation]
- transactional Control Language

Team





## Scenario 3



Appreciation

Savepoint ✓

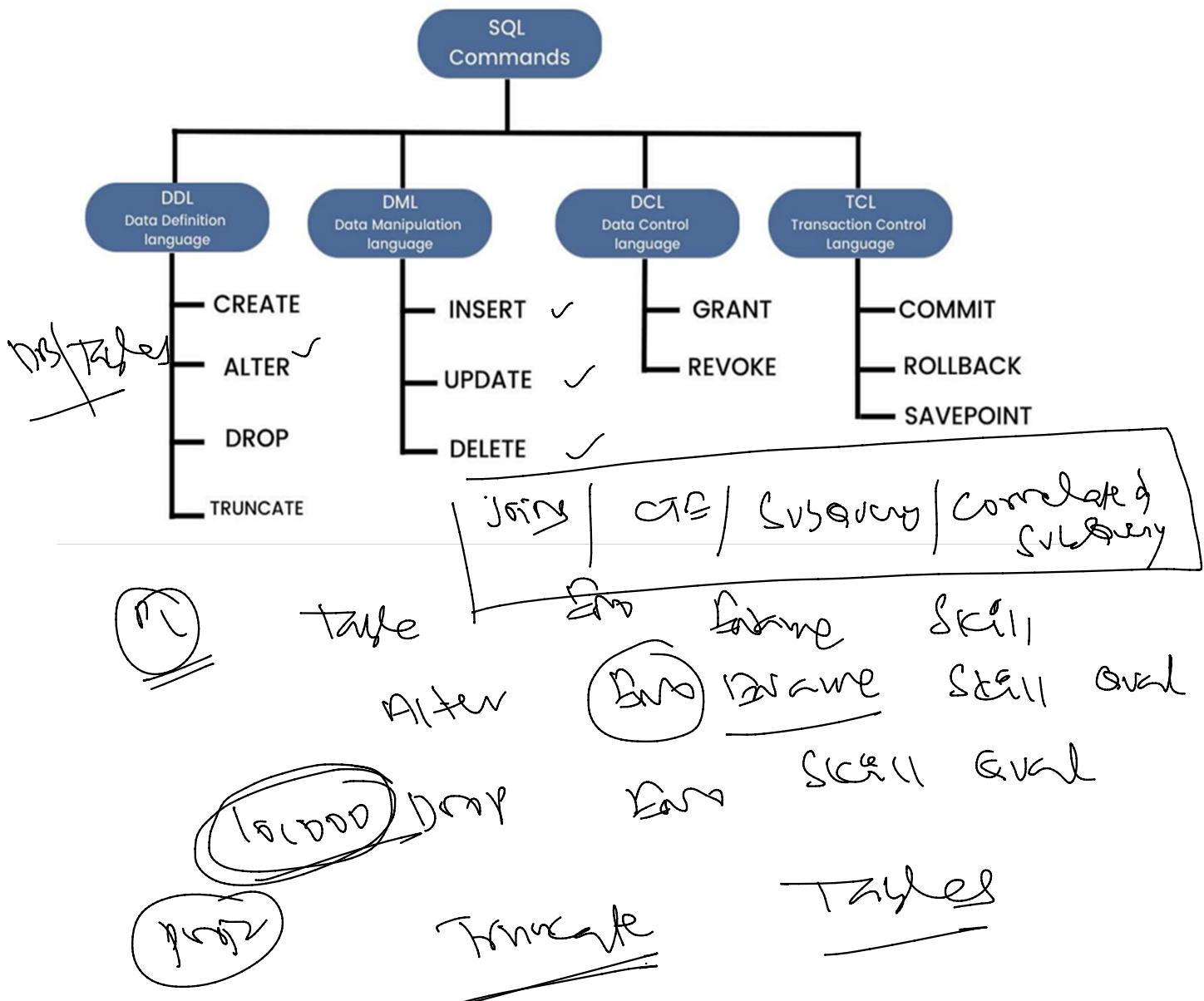
a) Commit

b) Savepoint

TCL

- a) Commit
- b) Savepoint
- c) Rollback

TCL



DCL

DBA

✓ Monitor the DB

Create user }

Access

Revoke

USER-DA-SCALA

Indium-DE-Databases

USER-DE-Pyspark

Create a task



✓  
USER-DE-Pyspark  
↓

DE\_Emp\_Data ✓

DE\_Emp\_Data.

DB-name • Table-name

✓  
USER-DE-Pyspark • DE\_Emp\_Data

↓  
Table:

①

Every time the long

DB name should be given.

To point only a specific DB

by mysql

> USE USER\_DB\_pyspark;

> Create table → Store in

USER\_DB\_pyspark

Database



DB-objects [Tables, views, Index, functions]



Table  $[A_1, A_2, \dots, A_n]$

①

Documentation / Comments

②

Naming convention

③

Explicitly mention the current Database that you want to use

use DB-name

Name

CREATE TABLE employee\_table(  
C1 → id int NOT NULL AUTO\_INCREMENT,  
C2 → name varchar(45) NOT NULL,  
C3 → occupation varchar(35) NOT NULL,  
C4 → age int NOT NULL,  
PRIMARY KEY (id)  
);

→ DDL

Table

1  
2  
3  
4  
5  
6



NOT NULL

AUTO\_INCREMENT

PRIMARY KEY

Create table

→ (



var

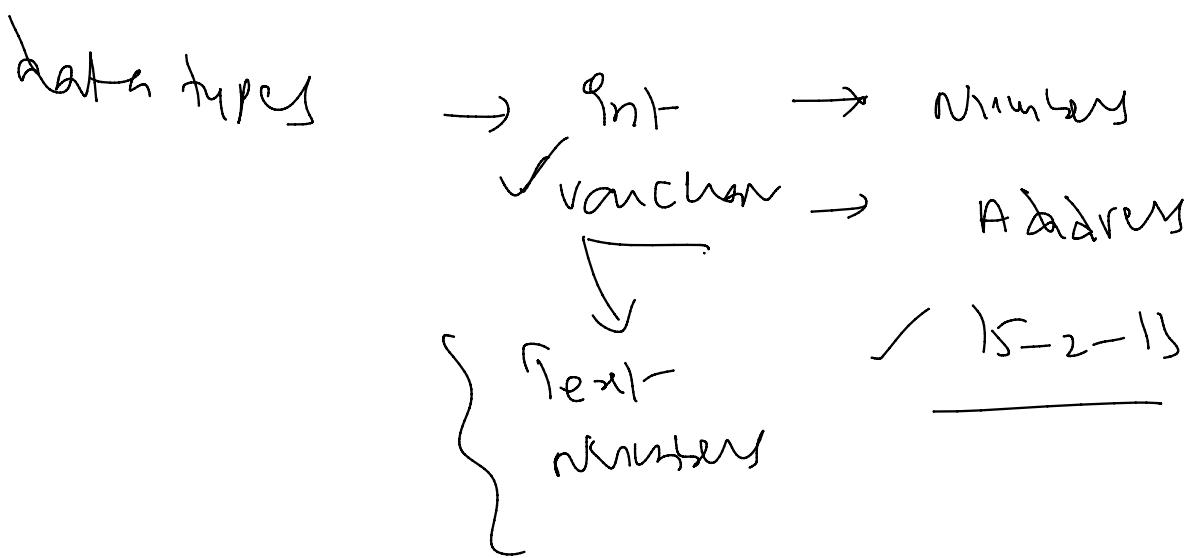
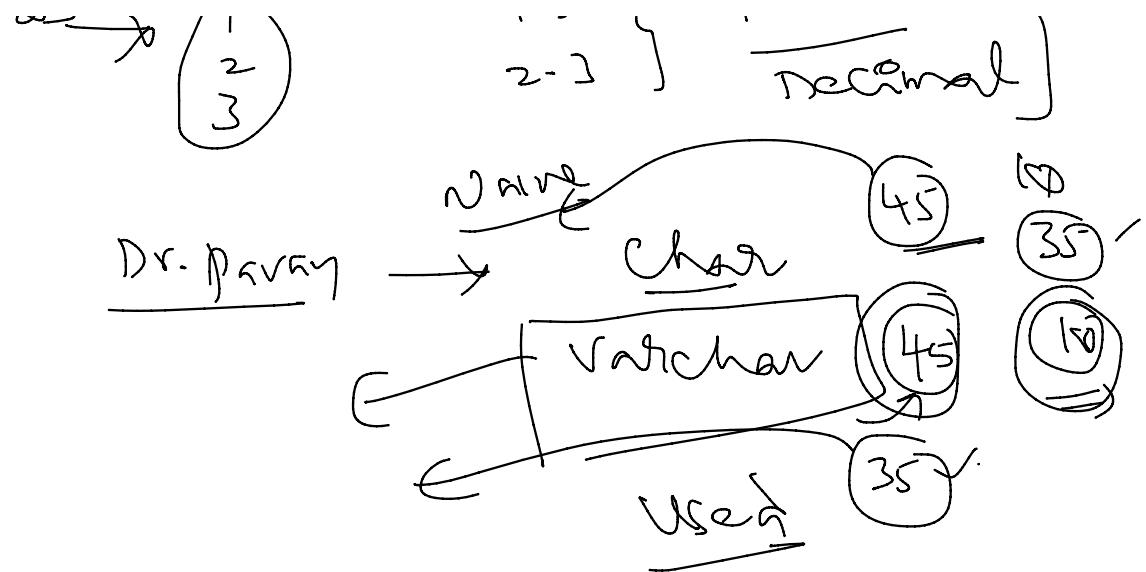
employee\_table [ id, name, occupation, age ]



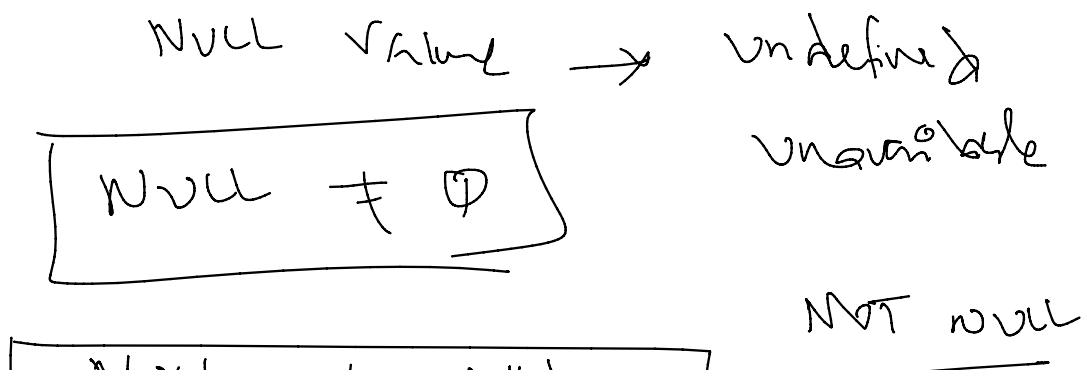
int  
varchar  
int

1  
2  
3  
4

1.5  
2.3  
float  
decimal



## Constraints



Null ≠ null

abc ≠ xyz  
false

Not null

carous

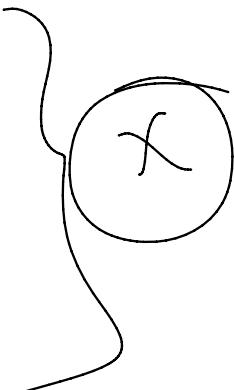
→ driving

passenger

window

→ ~~IS~~ → driving  
IS 5262

IS 5263



Objects →

state

uniquely

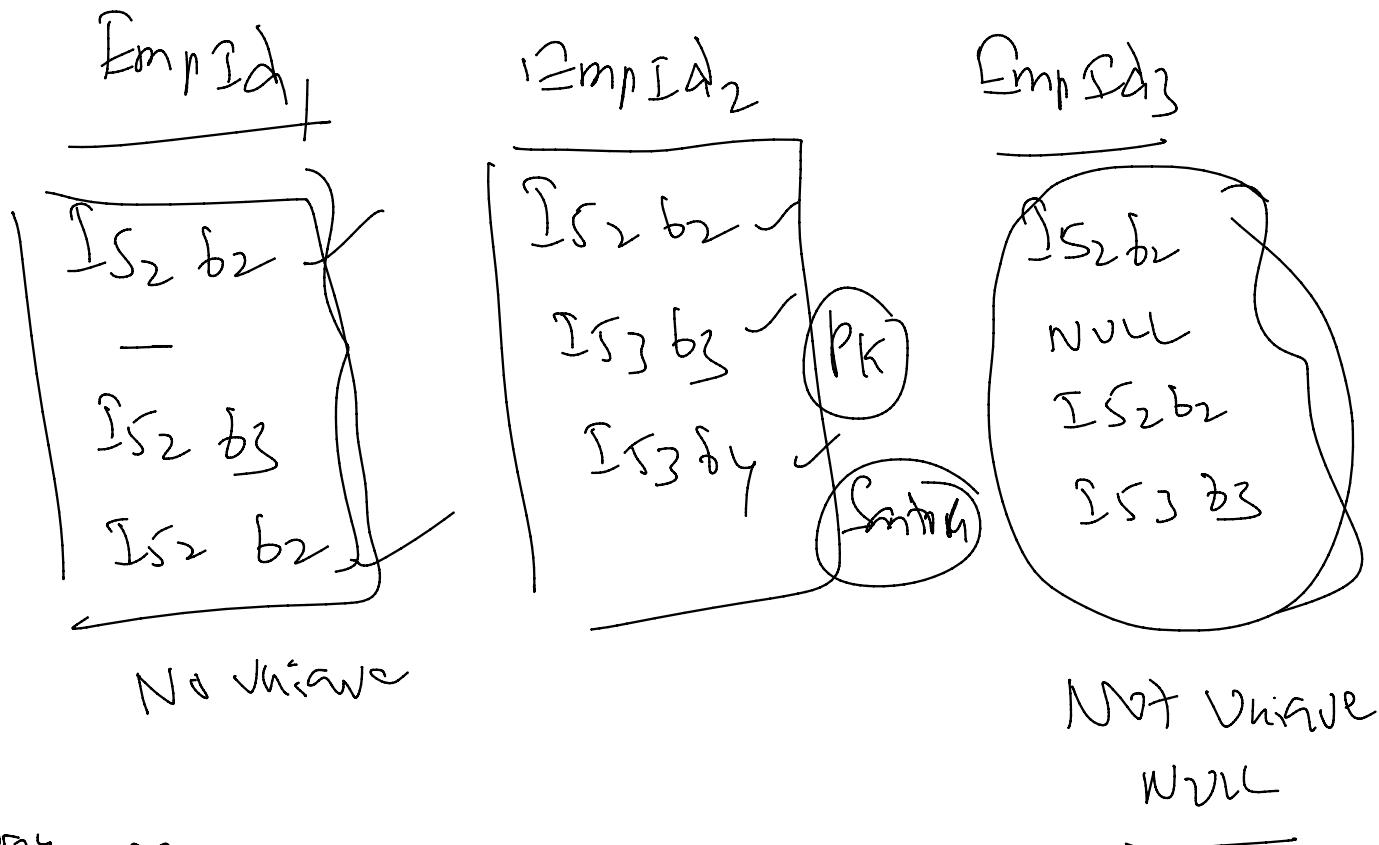
Value  
Not null  
unique

→ Not null  
unique

Not null + Unique = primary key

1 2 3

1 , 2 , X



Day - 02

Key Take aways

- ① RDBMS
- ② Relations
- ③ functional dependency
- ④ object Entity

⑤ DB objects

⑥ Models { Hierarchical  
    ↳  
    Relational }

⑦ Create DB

Use DB

Show tables

⑧ Create Table

- Datatypes [ int, varchar ]
- Constraints [ NOT NULL, unique, primary key ]
- AUTO\_INCREMENT