CST-122

Data base creation & (designing) compelling

Data: collection of facts which provides information about something

Database: collection of data, specification, structure, constraint

Types of data:

→ Traditional
→ Recent

DBMS: set of applications that allows users to construct & manage databases.

* General purpose software that simplifies, creating. building, modifying & sharing databases across users & applications.

Metadata: Telling data about data.

Database catalogue: Tabular form.

Miniworld: Real-time world.

concurrency: Accessing one's data.

why DBMS:

* Traditional file systems were used before DBMS.

* DBMS got rid off many limitations faced by traditional file systems.

* Familiar with physical specifics. No concurrency.

Basis	File system	DBMS
Structure	Software that manages & organise the file in storage	software for managing database
Data Redundancy	Redundant data is present	No Redundant data
Backup & Recovery	No Backup & Recovery	provides backup & Recovery if Cost.
Query processing	No efficient query processing.	Efficient query processing
consistency	Less data consistency	
complexity		
security constraints		
cost		
Data Independence		
user Access		

Meaning

sharing

Data abstraction

Integrity constraints

Data base: collection of data

Data: known facts & have an implicit meaning.

Mini-world: Some part of real world stored in a database.

DBMS: Software that facilitate operat-ions on data base.

Features of DBMS:

Normalization

user-defined limitations/rules

Security

Data Backup

Data organisation

Goals of Database-design:

Minimize the redundancy, lossless join, Dependency preservation.

Draw-Backs of DBMS:

* Cost, complexity, through understanding of SQL is required.

* simple, well defined, Not expected to change are characteristics iff DBMS is not required.

* Access to data by multiple users is not required.

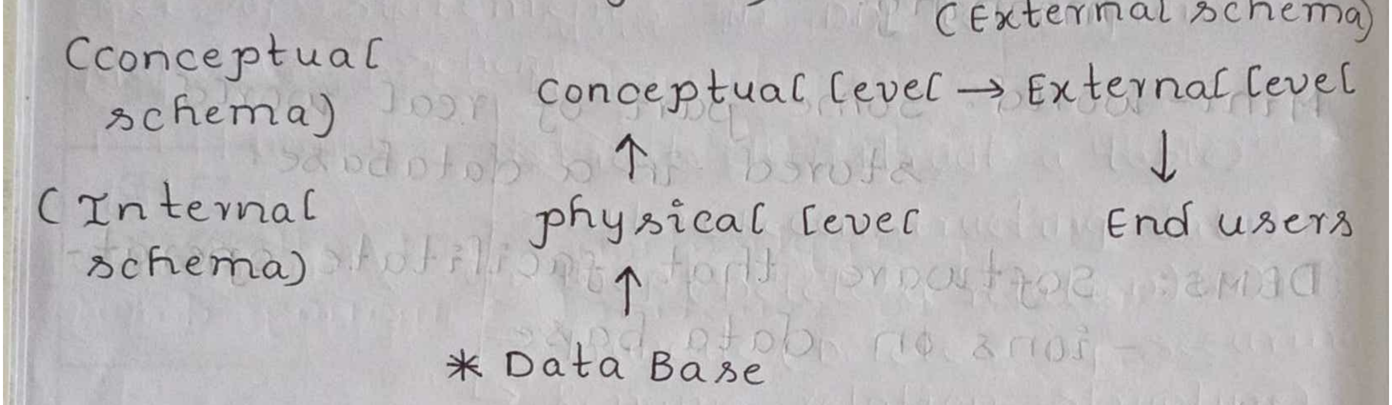
Database users:

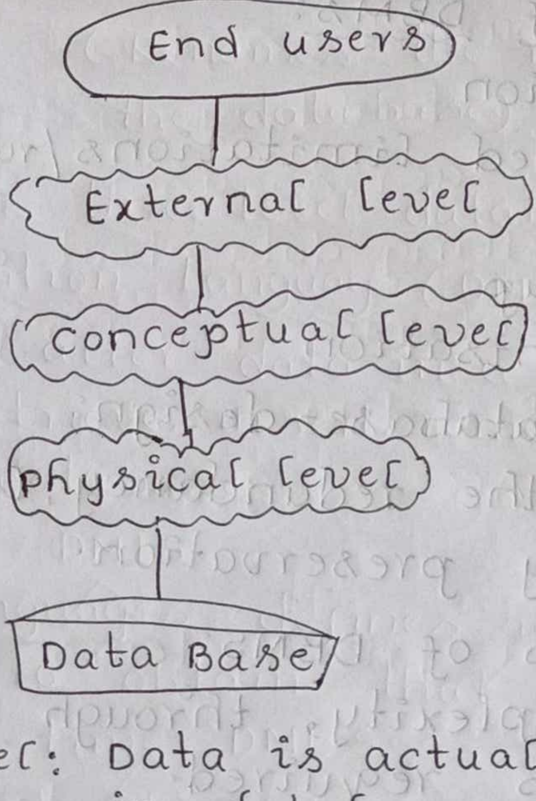
→ End user.
→ DB Administrator
→ DB designer

* use & control the database content, who design, develop & maintain (Actors)

* who design & develop

Three level Architecture of Data base:
(Blue print)

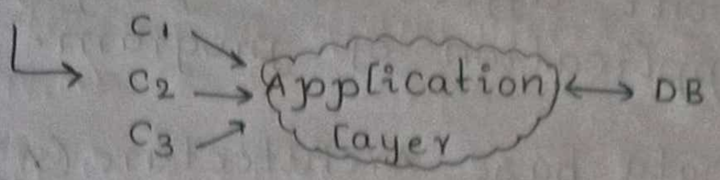




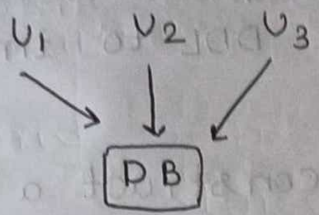
physical level: Data is actually stored in database.

Architecture in DBMS:

Tier-1: Tier-2: Tier-3

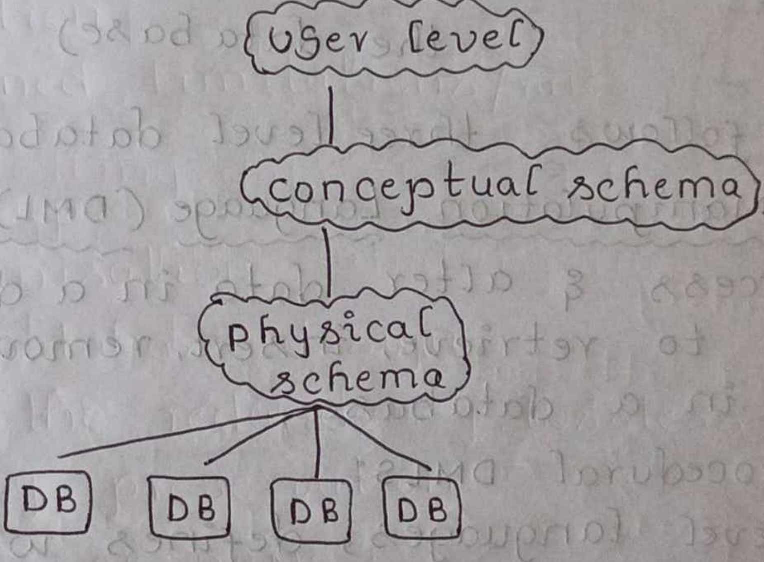


ODBC,
JDBC,
poor performance (more users),
query processing,
management of transactions.



concurrency: Discretions of using simultaneously.

Application layer: Partially processed data



view schema	conceptual	physical
user interaction	Tabular	Raw folders

Data Independence:
change in the data shouldn't be affected by the program. Data is seperated from the program. This would save time & cost requires as it doesn't affects the data at other levels of the data base.

Hence, change in the data at a level won't affect the program execution (or) application of program.

Data base Architecture (A)

* Differences between procedural & Non-procedura

Data definition language: (DDL)
[schema - structure] statements that will be used to implement the database schema. set of statements in DDL to implement database scheme:

→ create (construct a table)
→ Alter (Reconstructing data)
→ Drop (Delete a relation) (or) whol
→ Truncate (Deletes by securing
→ Rename relation structure)
(Renames the relation in the data base)

DDL follows three level database schema.

Data Manipulation Language (DML): Allows use to access & alter data in a data base. Let users to retrieve, insert, remove, (or) edit data in a database.

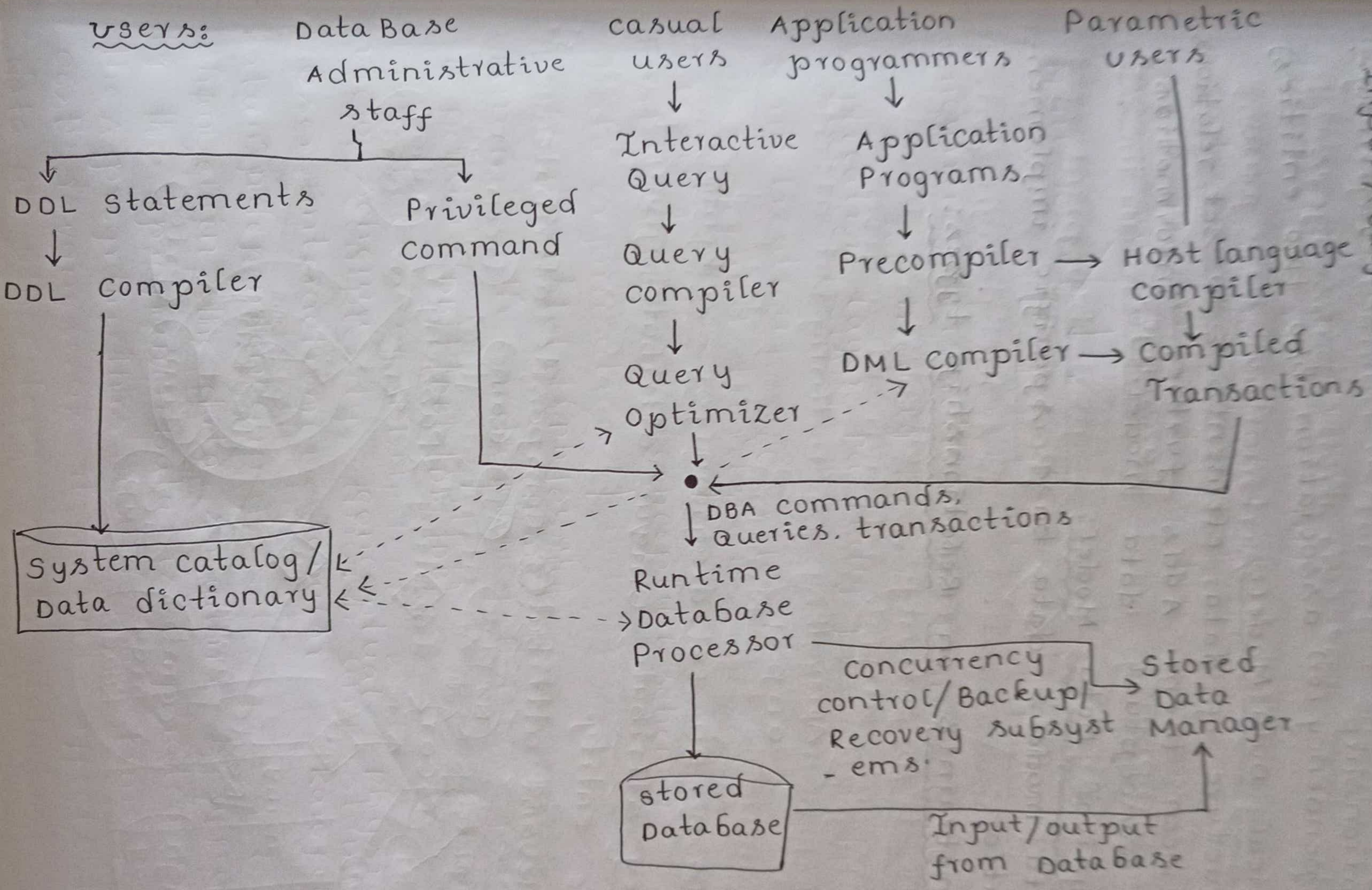
(i) Procedural DMLS:
Low level languages, defines what data is needed, how to obtain that data. Called as one-at-a-time DML's.

(ii) Non-procedural DMLS:
Precisely define the data requirement without specifying the way to access it. Set-a-time DMLS.

statements of DML:

Select, Insert, update, Delete

DBMS Interfaces: Menu, form, GUI, Natural, voice Input / output, ... etc



Database Environment system.

Conceptual modeling using the entity Relationship model

thing)
(A real world (properties

The 3 basic components are Entity, Attribute, Relationship.

(Dependency (or) association b/w two entities)

Logical Data Model:

structure of data elements & to set relations between them. Adds further information to conceptual data model.

Physical Data Model:

Data model's data base specific implementation.

Constraints on Relationship types:

Entity Relationship Diagram symbols & Notations: Rectangle, oval, diamond to represent relationships between elements. entities & attributes. There are some sub-elements

□ - Entity , ◇ - Relationship ⧈ - weak entity,

◈ - weak Relationship, ⬭ - Attribute,

◎ - Multivalued Attribute

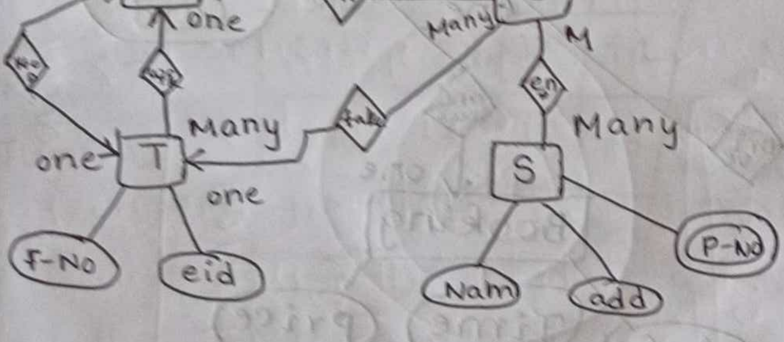
case-study (Example-1): ER diagram

step-1: Identify the nouns & the given requirements,

Entities: university. (only one instance) → DataBase
Department,
Teachers,
Courses,
Students,
HOD (Indirectly teachers)

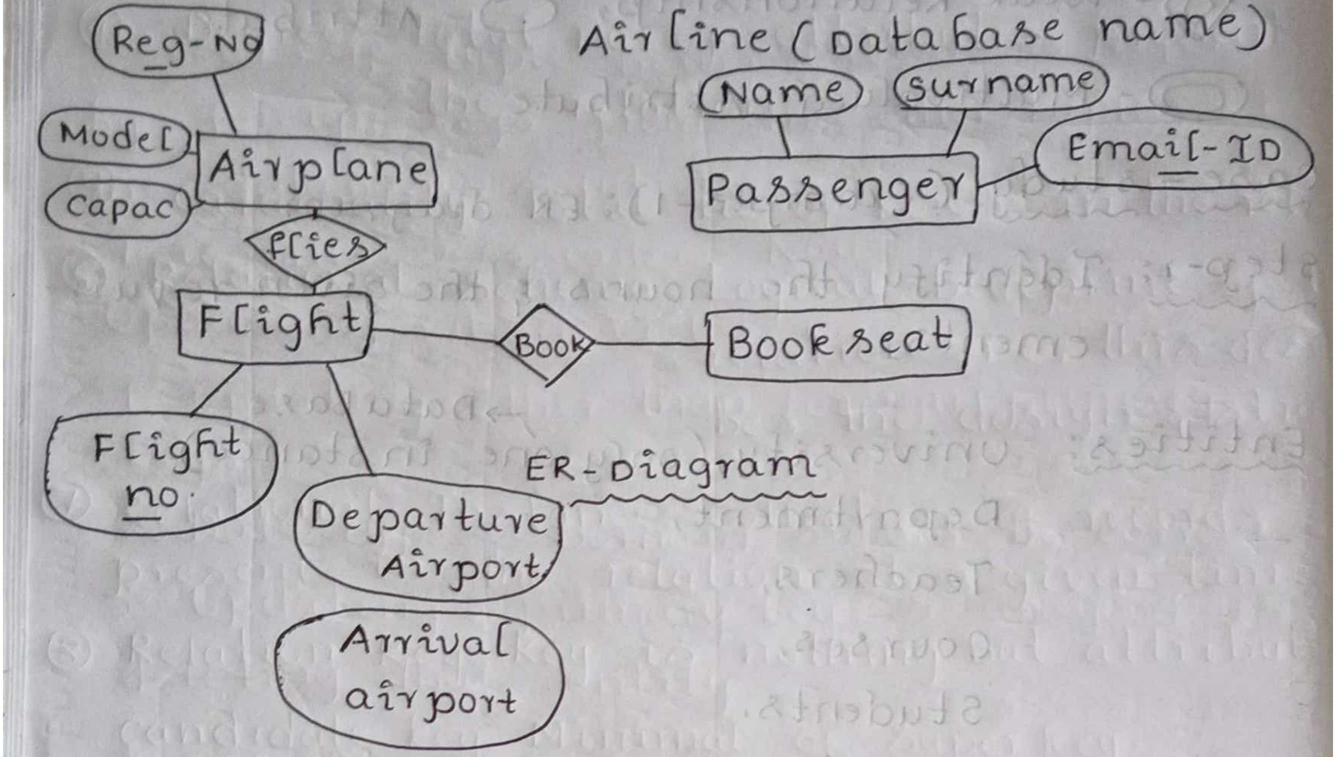
Final entities after excluding HOD, university.

step-2: Relationship between the entities check the verbs (or) role.



Case study- example-2:

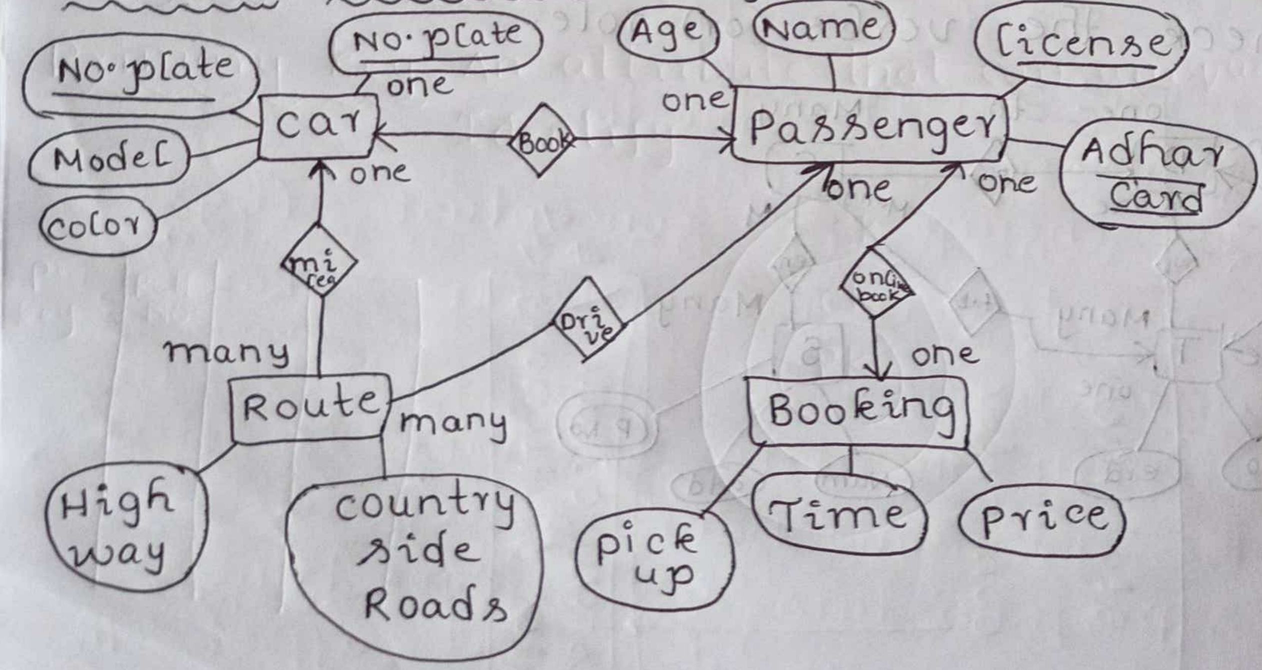
- * Airline has one (or) more airplanes.
- * An aeroplane has a model number, a unique Registration Number, capacity to take one (or) more passengers.



(P) Draw an ER diagram for car Rental

Key attribute: Unique attribute given to the entity.

Relevant attribute: composite attribute.

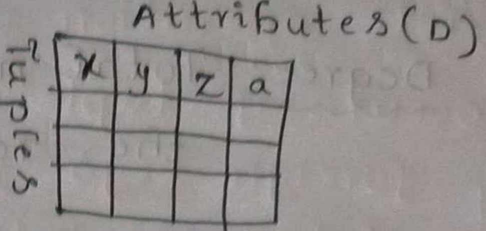


Ⓟ Draw an ER diagram for BMW dealership

car, price

Relational model:

Logical representation & managing of data in tabular format (2D Table).



x	y	z	a

Rows: tuples

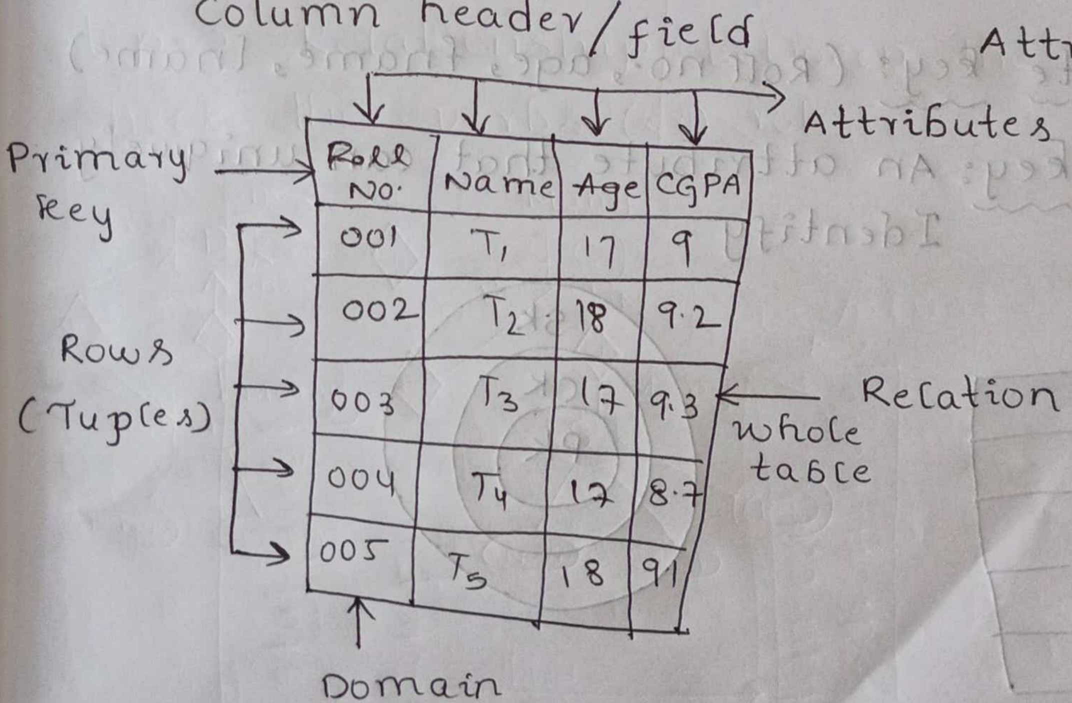
columns: Attributes,

Information:

use of Data: (tables) representation:

* To store Data,
* which will provide efficient & flexible way.
* It is based on the Mathematical concepts
* Informal representation . Formal terms

Informal representation	Formal terms
column	Attributes
Rows	Tuples
All possible column values	Domain Attributes
Tables definition	Schema of Relation
Column header/field	Attribute.



Roll No.	Name	Age	CGPA
001	T_1	17	9
002	T_2	18	9.2
003	T_3	17	9.3
004	T_4	17	8.7
005	T_5	18	9.1

① Relation: 2D Table used to store Data.

② Tuples: Row of the relation which depicts Real world entities.

③ Attributes: columns in relation which tells about properties of entity.

④ Degree: Total no. of attributes present in the relation

⑤ Cardinality: Total no. of Rows in the relatio

⑥ Relational schema: Logical blueprint of the relation, i.e; this descri-bes the design & structu

⑦ Relational Instance: Collection of records present in the relation at a given time.

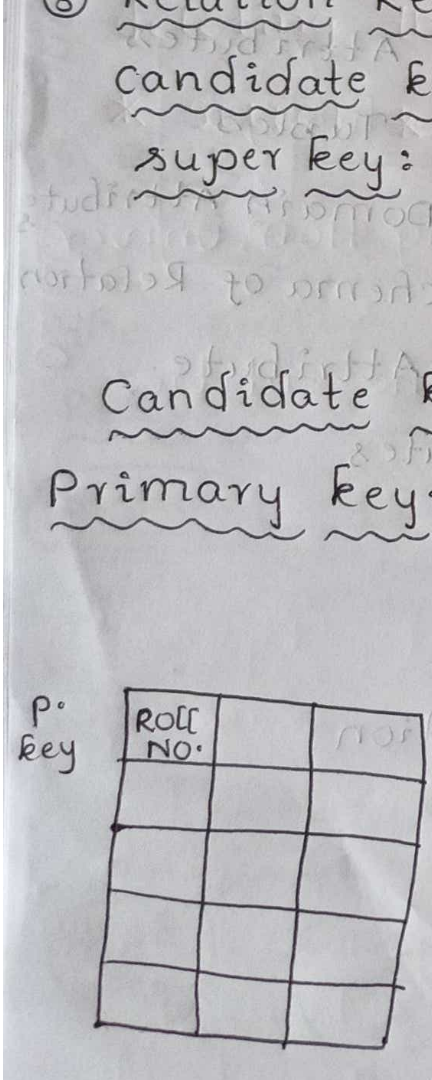
⑧ Relation key: key is nothing but attribute.

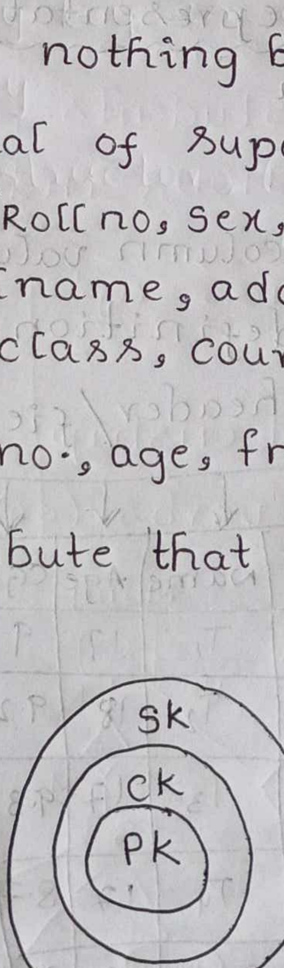
Candidate key: Minimal of super key.

super key: student (Roll no, sex, age, fname, lname, addr, email-ID, class, course, section).

Candidate key: (Roll no., age, fname, lname).

Primary key: An attribute that can uniquely Identify





2. **Key constraint:** (uniqueness)

RN	F-name	Age
001	x	17
002	x	18

3. **Entity Integrity: constraint**

4. **Referential Integrity constraint:**

RN	FN	Age

RN	add	Eid

```
CREATE DATABASE Svudent_DB;
DROP DATABASE student_DB;
```

Insert operation: The insert operation provides a list of attribute values for new tuple t that is to be inserted into a relation R.

tuple: Row

```
INSERT
```

Delete operation:

Deletes specific row that fits the criteria specified in Delete query.

Syntax:

```
DELETE FROM (table-name) WHERE (condition:)
```

Update (or) Modify operations:

update (or) modify to change the values of one (or) more attributes.

```
UPDATE table-name
SET column1 = value1, column 2 = value 2, ...
WHERE condition;
```

Insert operation violations:

1. Domain constraint
2. key constraint
3. Entity integrity constraint (Null values)
4. Referential integrity constraint

Delete operation violation:

occurs if tuple being deleted is referenced by foreign keys from the other tuples in the database. Solutions

1. restrict, 2. Cascade. 3. Null

Update operation violations:

1. updating other than foreign, primary keys.
2. Data type & Domain should be correct.
3. Updating primary key is as good as deleting that entity.

* SELECT * FROM Instructor; (Prints) all values & tuples)