Name: Ayush Jain
SAP ID: 60004200132
Div: B/B1

06/12/2021

DBMS - TT1

Q. 1) Define Physical Data Independence and logical Data Independence in three schema architecture of DBMS. list down the advantages.

→ The three-schema architecture can be used to further explain the concept of data-independence, which can be defined as the capacity to change the schema at one level of database system without having to change the schema at the next high level. We can define two types of data Independence:

1) Logical data Independence

It is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the conceptual schema to expand the database, to change constraints, or to reduce the database. Only the view defination and the mappings need to be changed in DBMS that supports logical data independence. After the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before. Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.

2) Physical Data Independence

It is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not to be changed as well. Changes to the internal schema may be needed because some physical files were reorganized • for example, by creating additional ocess structures - to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to changed the conceptual schema.

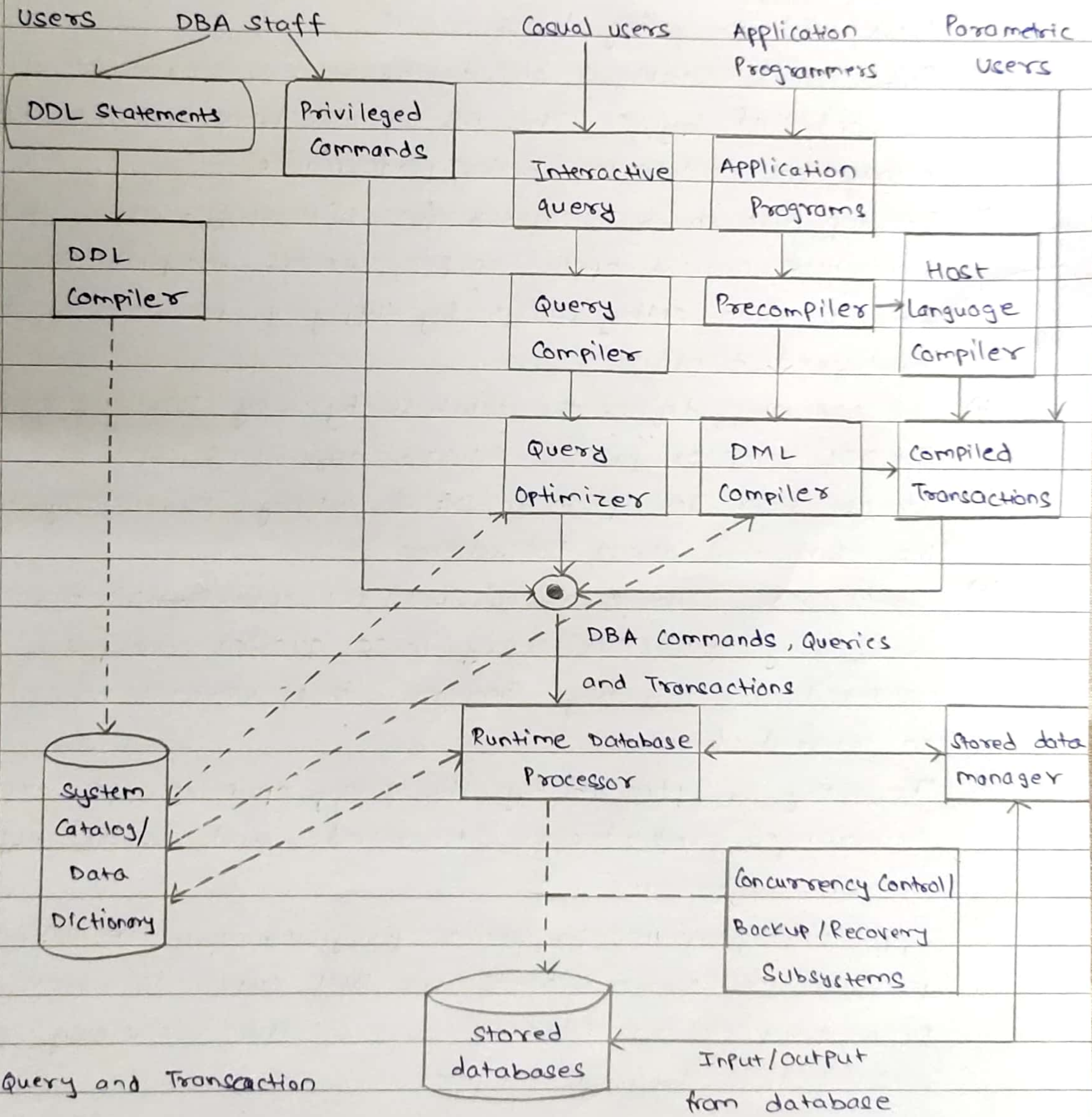
Advantages :

i) Alterations in data structures does not requires alterations in application programs.
2) Implementation details can be hidden from user.
3) Easily make modifications in the physical level is needed to improve the performance of the system.
4) It allows one to improve state which is undamaged or undivided.
5) Database incongruity is vastly reduced.

60004200132

Q. 2) Draw and Explain System Architecture of Database Management System.

→



Query and Transaction Execution:

(1) The above figure indicates DBMS components in a simplified form. The top part Indicates various users and their interfaces. The lower part shows internal modules responsible for storage of data and processing transcations.

(2) The DBMS catalogue and database are stored on a disk that is controlled by OS. The DB can also have its own buffer management module for better performance.

(3) The top part shows interfaces for DBA staff, casual users who formulate queries, application programmers and parametric users who do data entity work by supplying parameters to predefined transcations.

(4) The DBA staff defines the database by using DDL and other commands. The DDL compiler process schema definations, specified in DDL and stores their description in DBMS catalog. The catalog includes all the information about the schema.

(5) Casual users with occasional need for information from database do so using interactive query. These queries are passed and validated by a query compiler that converts them into an internal form.

(6) The query is optimized by the query optimizer, ie. expressions rearranged, redundacies eliminated and use of efficient search algorithms.

(7) Application programmers write programs that are submitted to the precomplier which extracts the DML commands which are sent to DML compiler for database access. The remaining program is sent to host language compiler.

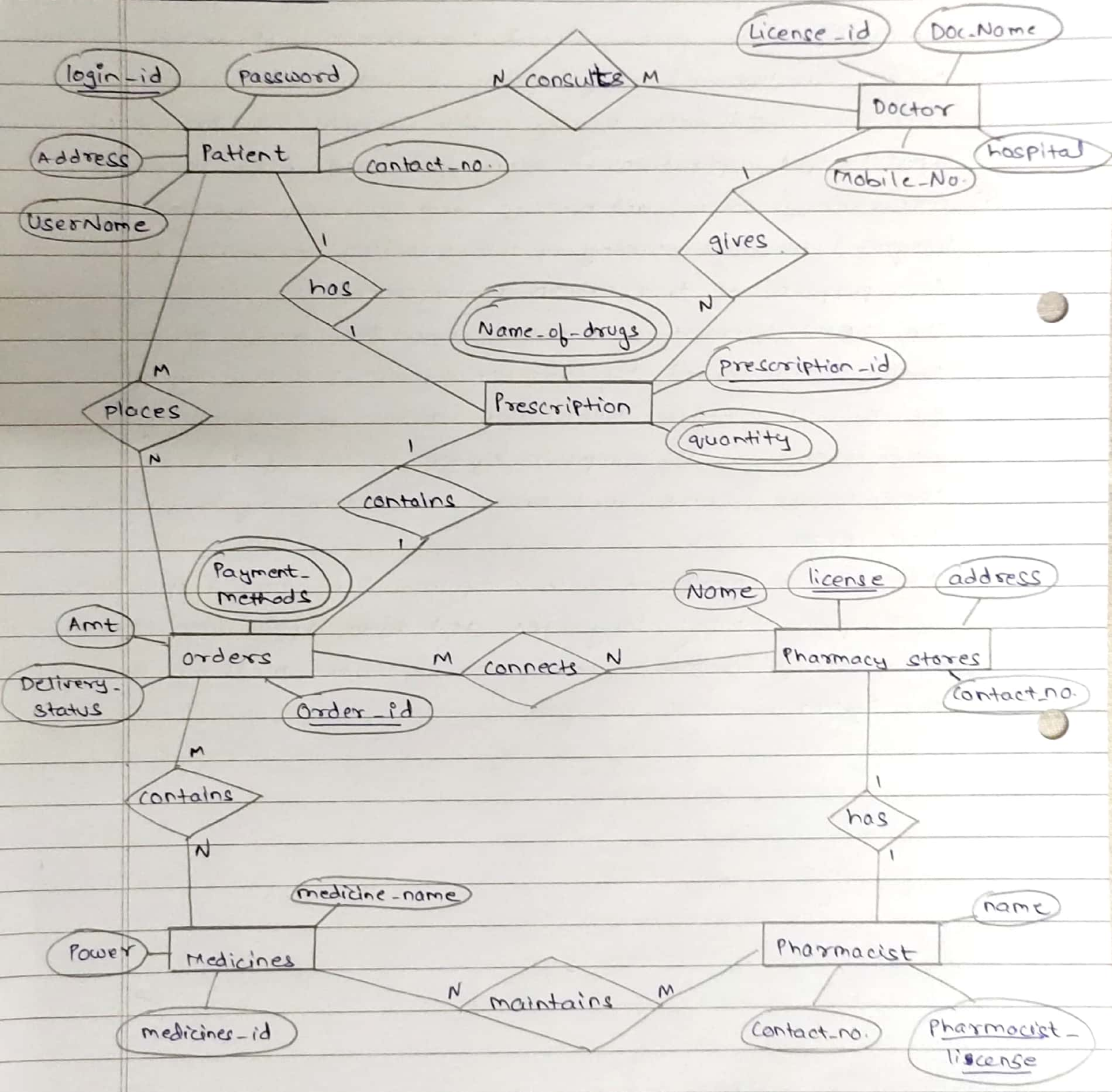
(8) In the lower part, the runtime database executes privileged commands, query plans, canned transactions. It works with system catalog and many update it with facts. It also works with the stored data manager for carrying out low level input/output operations between disk and main memory.

(9) Concurrency control and backup and recovery systems are integrated into the cooking of the runtime database processor for purpose of transcation management

(10) The DBMS interacts with the OS when disk access to DB or catalog are needed. If computer system is shared by many users, the OS will schedule DBMS disk access request along with the other processors. If computer system is mainly for handling DBMS server, DBMS will control main memory buffering of disk pages.

(11) DBMS also interfaces with compilers for general purpose host programming languages and with application servers and client programs running on seperate mediums through the system network interface.

Q.3) ER Diagram



Entities :

1) Patient - UserName, login-id, password, Address, contact-no.

2) Doctor - DOc.Name, license-id, hospital, contact-no.

3) Prescription - Prescription-id, quantity, name-of-drugs.

4) Orders - order-id, Amt, Delivery-status, Payment-methods.

5) Pharmacy stores - Name, license, address, contact-no.

6) Pharmacist - name, contact-no., Pharmacist-license.

7) Medicines - medicine-id, medicine-name, Power.

Pelationships :

1) One doctor is consulted by many patients. Similarly, 1 patient can consult many doctors. Hence cardinality is M:N

2) One patient has one prescription at a time. Similarly a prescription can belong to one patient only. Hence cardinality is 1:1

3) One doctor can give multiple prescriptions, but a prescription is only prescribed by one doctor. Hence cardinality is 1:N

4) One order can contain one prescription only. Similarly a prescription can be used to place one order only. Hence cardinality is 1:1

5) One patient can place multiple orders and similar orders can be placed by many patients. Hence cardinality is M:N

6) One orders connects multiple pharmacy stores and similarly one pharmarcy stores can provide multiple orders. Hence cardinality is M:N

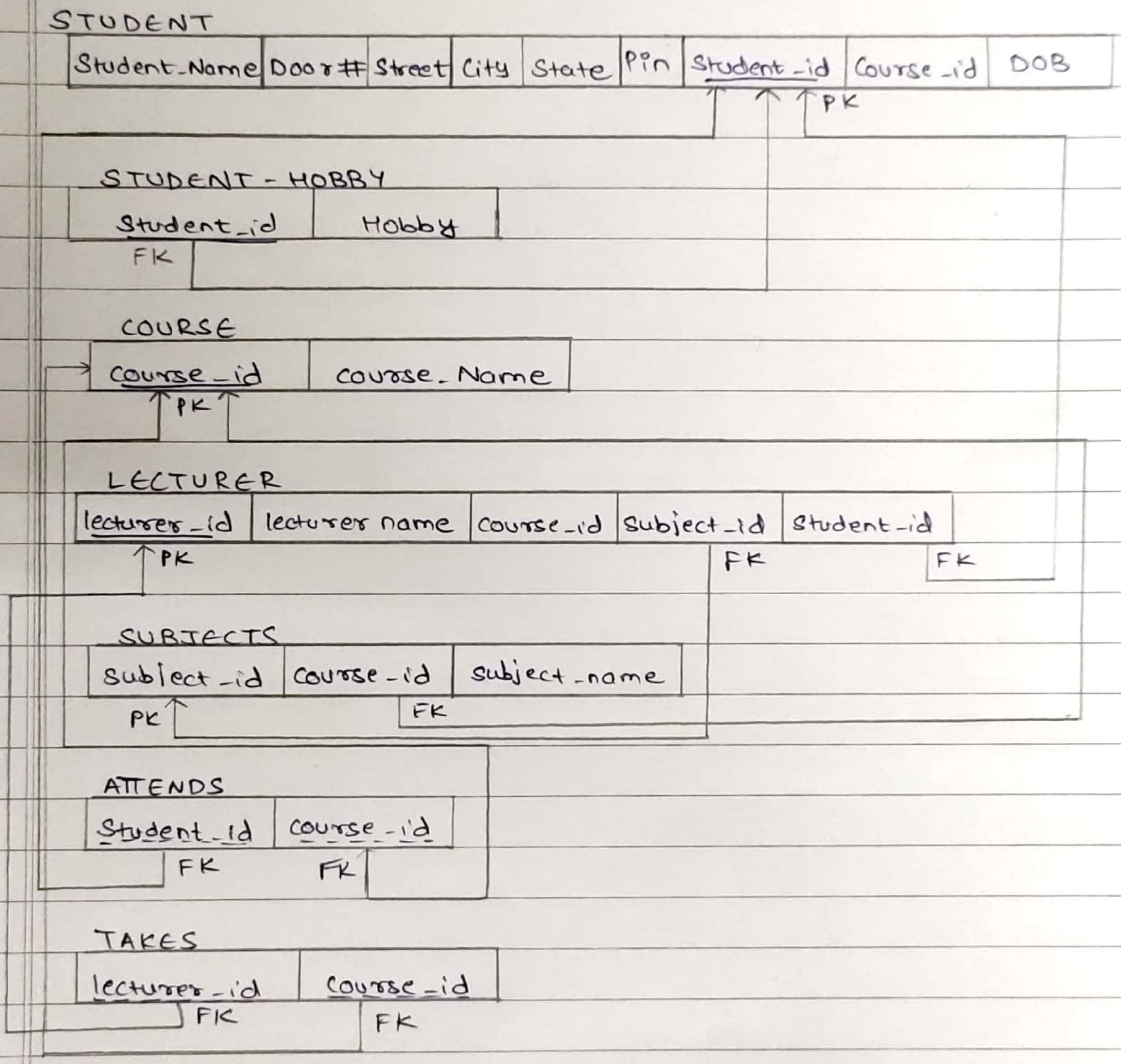
7) One pharmacy store has one pharmacist working there, similarly one pharmacy work at one pharmacy store only. Hence cardinality is 1:1

8) One pharmacist maintains many medicines and similarly one medicine is maintained by many pharmacist. Hence the cardinality is M:N

9) One order contains many medicines and similarly one medicine can be in many orders. Hence the cardinality is M:N

Q.4 Convert ER diagram into Relational model

→



STUDENT

Student_Name	Door#	Street	City	State	Pin	Student_id	Course_id	DOB

STUDENT - HOBBY

Student_id	Hobby

COURSE

course_id	course_Name

LECTURER

lecturer_id	lecturer name	course_id	subject_id	student_id

SUBJECTS

subject_id	course_id	subject_name

ATTENDS

Student_id	course_id

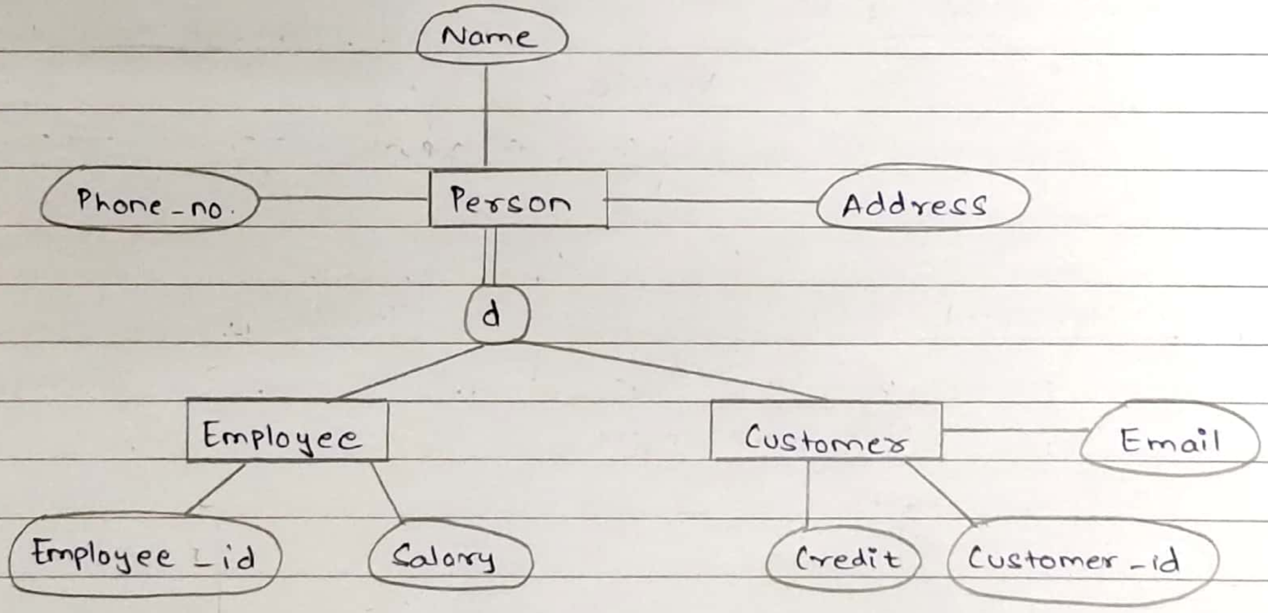
TAKES

lecturer_id	course_id

Q. 5) Explain specialization and generalization with example ?

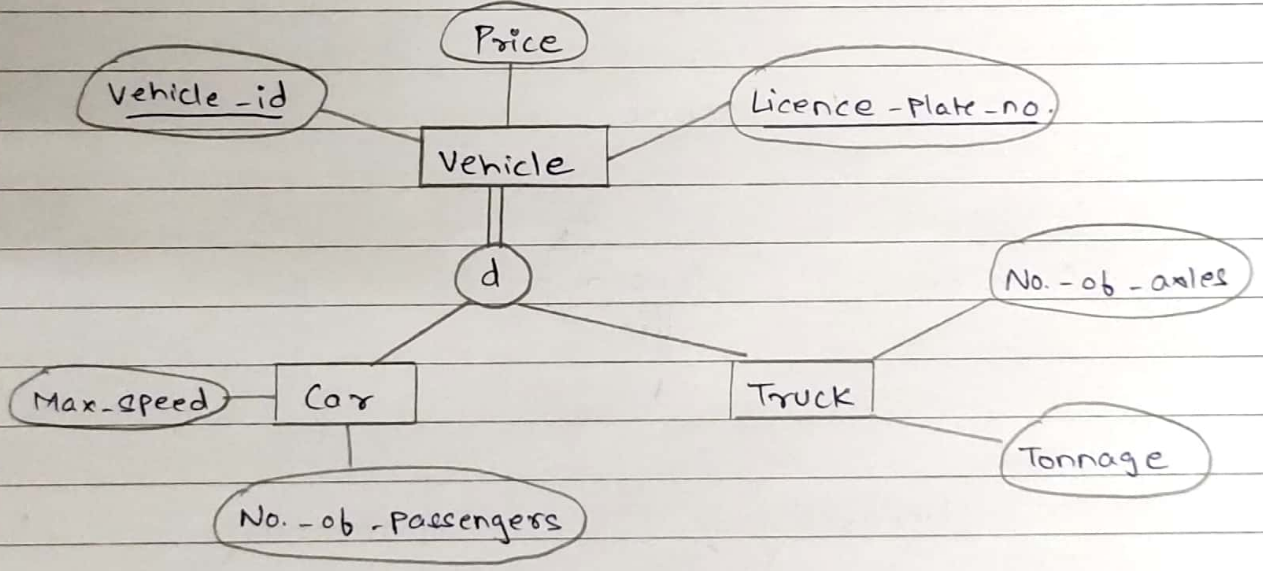
→ Speacialization :

It is a process of defining a set of subclasses of an entity type ; this entity type is called the superclass of speacialization. The set of subclasses that forms a speacialization is defined on the basis of some distinguishing characteristic of the entities of the superclass. We may have several speacilizations of the same entity type based on different distinquiching characteristics.



Generalization :

We can think of a reverse process of abstraction in which we supress the differences among several entity types, identify their common features, and generalize them into a single superclass of which the orginal entity types are special subclasses. For example, consider entities car and truck. Since they have several common attributes, they can be generalized into entity type vehicle.


Price
Vehicle_id
Licence-Plate-no.
Vehicle
d
No.-of-axles
Max-speed
Car
Truck
Tonnage
No.-of-Passengers