DBMS

Date: _____

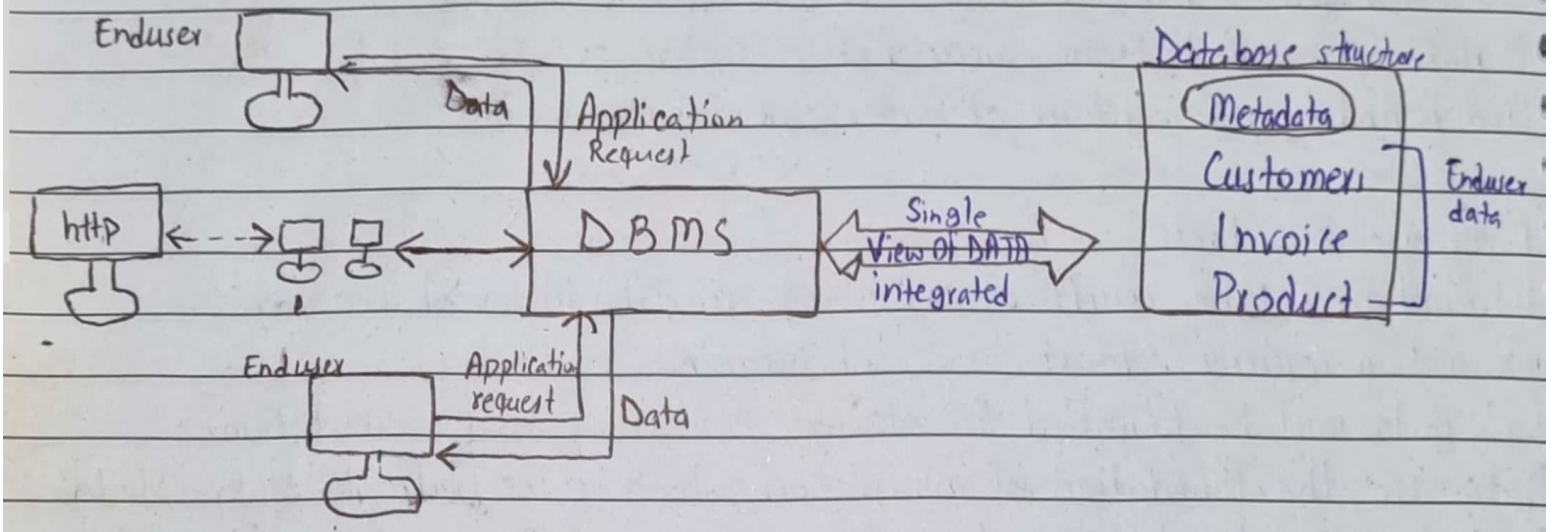
Intro: Good decisions require good information derived from raw facts.
Data managed most efficiently when stored in a database.
Databases evolved from computer file systems.
Understanding file system characteristics is important.

Data are raw facts.
Information is the result of processing raw data to reveal meaning.
information requires context to reveal meaning.
Raw data must be formatted for storage, processing, and presentation.
Data are the foundation of information, which is the bedrock of knowledge.
Data: building blocks of information
Information produced by processing data.
Information used to reveal meaning in data.
Accurate, relevant, timely information is the key to good decision making.
Good decision making is the key to organizational survival.

Database: shared, integrated computer structure that stores a collectio
of data.
End-user data: raw facts of interest to the end user.
Metadata: data about data.
Metadata provides description of data characteristics and relationships in data - Complements and expands value of data.
Database Management System (DBMS): collection of programs.
Manage structure and control access to data. DBMS is the intermediary betwee the user and database.
Database structure stored as file collection; - DBMS enables data to be shared
Access database through the DBMS.
DBMS integrates many users' views of the data.

Date: _____

DBMS manages the interaction between the end user and the database.



Advantages of DBMS: Improved Data Sharing, Control over redundancy, Improved Data Standards, Improved data Security, Improved data integrity, Better data integration, Minimized data inconsistency, Improved data access, Balancing of Coflicting Requirements, Faster Development of new application. Economy on scale, More control on Cocurrency, Better backup and Recovery Improved decision making, Increased end-user productivity.

Trade-offs/Limitations: High cost of DBMS, Hardware Cost, Conversion Cost, High Development Cost, Slower processing of some applications, Increased Vulnerability Difficult Recovery,

Databases can be classified according to:
Number of users.
Database location(s)
Expected type and extent of use

Date: _____

Single-user database: supports only one user at a time.
- Desktop database: single-user, runs on PC

Multiuser database supports multiple users at the same time. Eg: LMS whatsapp group DB
- Workgroup database supports a small number
- Enterprise database supports a large number.

Centralized database: data located at a single site. LMS

Example: Enterprise resource planning (ERP) systems, Goverment databases, Academics institutions, Inventory management systems.

Data warehouse: stores data used for tactical or strategic decisions.

Unstructured data exist in their original state.

Structured data result from formatting.
- Structure applied based on type of processing to be performed.

Semistructured data have been processed to some extent.

Extensible Markup Language (XML) represents data elements in textual form

XML database supports semistructured XML data.
↳ Extensible Markup Lang.

Distributed databases data distributed across several sites.

Eg: Social media platforms, E-commerce websites, cloud computing.

Operational Database (OLTP): support a company's day-to-day operations.

Transactional or production database

Eg: LMS, Banking System, Reservation System.

Field: A character of [illegible]
Record: A logically connected set of one or more fields that describes a person, place or thing
File: A collection of related records
↳ Row or tuple

Domain: Complete column ki values.

Degree (No. of columns)
Cardinality (No. of Rows).

Date: _____

Types of Databases

Structured	Unstructured/Big Data/No-SQL	Semi Structured
RDMS	Key-Value	XML-DB
OODBMS	Document Oriented	
ORDBMS	Network	
Mobile DBMS	Column Oriented	
In-Memory DBMS	Row Oriented	
Spatial DBMS	etc	
Temporal DBMS		
Multi-Dim DBMS		
Distributed DBMS		
Grid DBMS		

Database
↳ Table → columns → metadata
↓
rows ⇒ records.

Modification, Developing, Testing, Deployment, Maintainance.

Why Database Design is important.

Database design is ~~important~~ focuses on design of database structure used for end-user data. Designer must identify database's expected use.

Well-designed database: -) Facilitates data management
-) Generate accurate and valuable information.

Poorly designed database: -) Causes difficult-to-trace errors.

File & File System (lec 1, Pg 21-23)

File System: can't run adhoc queries in runtime

Data redundancy → data repetition, Data anomalies → issue or abnormalities that occure due to data inconsiltanly.

Procedures are like functions
↳ We defines, what functionlities the database will have

Constants are also used to define functionalities but they have limitations.

FMS: Data stored in file, which are typically organized in hierarchical structure

•) Organize in files •) Leads to redundancy •) FMS relies on manual efforts to main data consistency. •) Data consistency is low •) Query processing is not so efficient. •) Sharing is not easy	•) Organize in tables •) Minize redundancy through normalization techq •) Enforce data integrity through constraints, validations, while ensuring data consistency. •) Data consistency is high due to normalization •) Query processing is efficient. •) Sharing is easy due to centralized.

Date: _____

Problems with File System Data Management:

Simple file system retrieval task required extensive programming.

-) Ad hoc queries impossible. -) Changing existing structure is difficult.

Security features difficult to program.

-) Require extensive programming. -) System administration complex & difficult.

Structural dependence: The situation where the design or functioning of one component in a system relies on the specific structure or implementation details of another component. This tight coupling between components creates structural dependance and can make the system flexible & more challenging to maintain or evolve over time. Change in one module impact on another. LMS change in functionality will result in change in user interface

Structural independance: The ability of components within a system to operate independently of each other's internal structure or implementing details.

Data dependence: A change in any data characteristics requires change in all data access programs.

Data independence: data storage characteristics don't affect data access

↳ Logical data independence •) Physical data independence.

Practical significance of data dependence is difference between logical & physical forma

Data Format: How data from one table is depending upon the other table.

Clients requirement are met.

In database: Column name must be meaniful & must not be repetitive.

Logical data format: how human views the data.

Physical data format: how computer must work with data.

Each program must contain: Lines Specifying opening of specific file type, Record Specificatio

File defination. Storing customer name as single field: Better record defination break field into compound

Selecting proper field names important; field name are descriptive.

Each record should have unique identifier.

Date: _____

File system structure makes it difficult to combine data from multiple sources.
Data stored in different locations unlikely to be updated consistently.
Data redundancy: same data stored unnecessarily in different places.

Data inconsistency: different and conflicting versions of same data occur at different places.
Data anomalies: abnormalities when all changes in redundant data not made correctly: (-)Update anomalies (-)Insertion anomalies (-)Deletion anomalies

Insertion anomali:

	Sid	Cid	Cname	chri	Contact
5 row is insertion anomali	1	101	PF	3	
when 1 column is provided with s-id but	2	101	PF	3	
the rest of the field are left empty.	3	102	DB	3	
Deletion:	x 4	101, 102	PF, DB	3, 3	
Select from tbname where cid = 101	5	-	-	-	

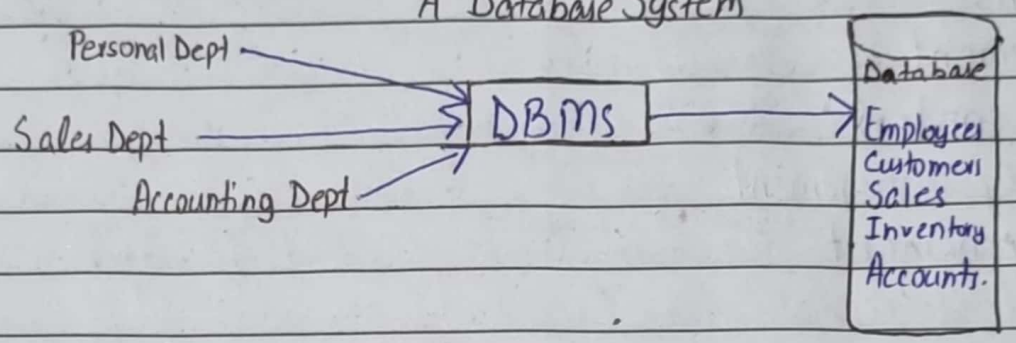
result: record 1 & 2 will be deleted but the 4th record will also be deleted which should not to be deleted.

Update anomali:
If we have to update the chri of the table an error would occur at the data in record 4.

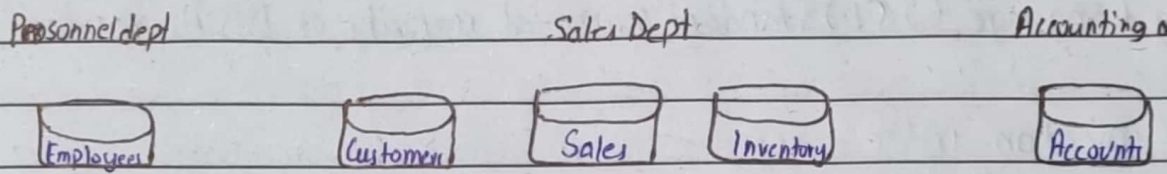
Database System consist of logically related data stored in a single logical data repository.
- Maybe physically distributed among multiple storage facilities.
DBMS eliminates most of the files System's problems
Current generation stores data structures, relationships between structures, acc Pat
- Taking care of defining, storing, managing all access paths & components.

Date: _____

A Database System



A File System



Database System: defines and regulates the collection, storage, management, use of data.

Five major parts of a database system.

Hardware, Software, People, Procedures, Data.

Hardware: all systems physical devices.

Software: three types of software required: Operating System Software, DBMS software
Application programs & utility software.

People: all users of the database system: System & database administrators
Database designer. End user
System analysts & programmers.

Procedures: instruction and rules that govern the design and use of DB system.

Data: the collection of facts stored in the database.

DB System created & managed at different level complexity, solutions must be cost effective as well as tactically & strategically effective. DB technology already in use affects selection of a database system.

Date: _____

DBMS Functions:

Security Management: user security / data privacy — which user access which items.

Multiuser access control: Cocurrent access donot affect integrity.

Backup & recovery management: Backup & data relovery ensure safety & integrity.

Data integrity management: Minimize redundancy, Maximize consistency. Data relationship stored in data dictionary

DBMS provides access through query language.

Query Language: is a nonprocedural language.

Structured Query Language (SQL) Standard supported majority of DBMS vendors.

Database Communication interfaces:

Communication accomplished in several ways:

End users generates answer to queris by filling in screen forms through web browser

DBMS automatically publishes predefined reports on a Web site.

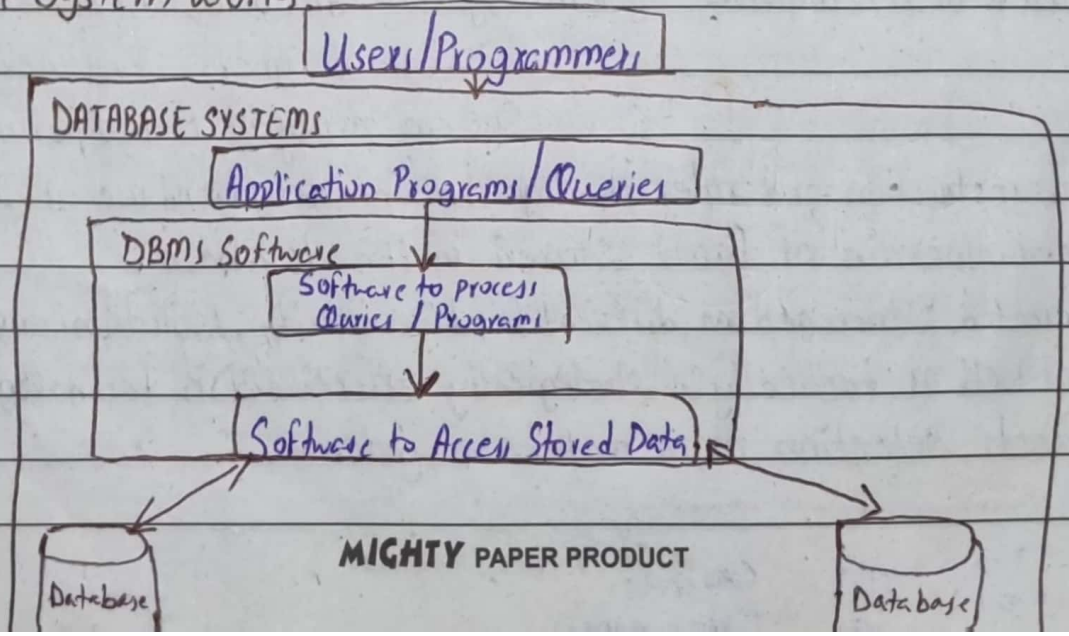
DBMS connects to third-party systems to distributes info via e-mail.

Disadvantages of database System:

-) Increased cost -) Management Complexity -) Vendor dependence

-) Frequent upgrade/replacement cycles.

How Database System Works



...database and it varies on database.
->They are document. ->They modify on customer requirement
->Sources of business rule ->Company managers ->Policy Makers
->Department Managers.

Column -> attribute
Row -> Tuple

Date:

Data Modeling: -> Repetative
-> Iterative.
*) Change in database model.

Entity: -> student / real world objects
Entity-set -> multiple students, collection of same type of ent

Relationship:
*) One-to-One: Person & Cnic *) One-to-Many: Doctor & Patient *) Many-to-One: Courses & Studen

Attributes: charaterictics of entity.

Constraints: restriction on attributes / specific condition applied on attribute.

Schema: view for end-user, display for end-user

↳ if u divide it in different modules then it will be subschema. Eg: LMS -> faculty module, student module

↳ Complex structure development, Manging Problem. Tree Structure.

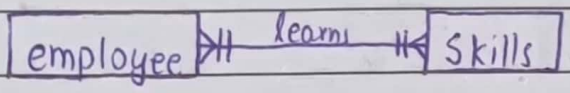
Hierarchical -> One to many not possible, only one parent for children.

Network -> 1 to many possible, one children can have multiple parents.

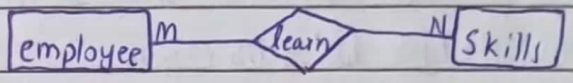
↳ Created to represent complex data.

Entity Relationship Model:

*) Crow's Foot:



*) Chen Notation



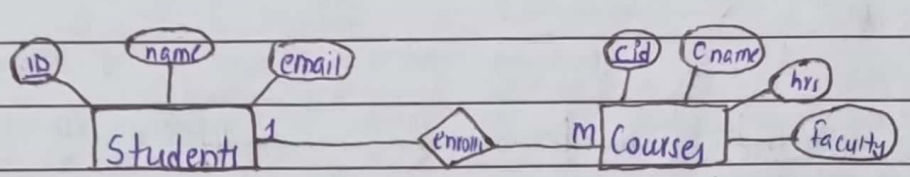
: entity : Relationship : Attributes.

Unique: unique value, null value possible

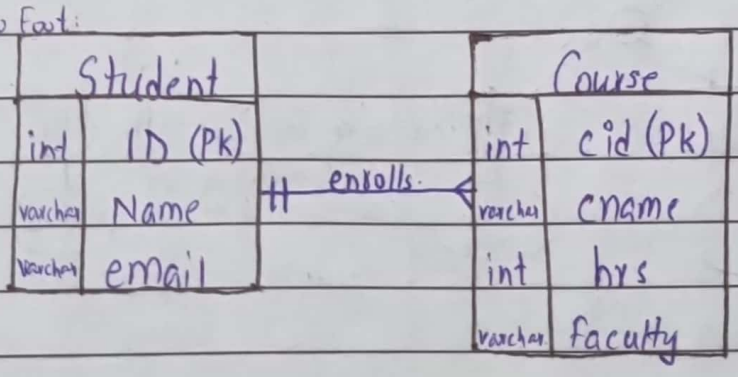
Primarykey: unique & not null.

Not null: repeatation possible & Not null

When we combine 2 Primary key it will become a composite key.



Crow Foot:



The Relation Data Model:
Data define in tabular form.
Each row called tuple.
User Interface
Set of tables
One to many relation
Sql.

—— links (one to one) ||—— one & only one.
+—< one to many
>—+ many to one
o—— Zero or one.
>o—— Zero or many

Date: _____

Supertype: → Parent (Super entity)
Subtype: → Child (Sub Entity)

→ Inheritance: Subtype entities inherit values of all attributes of the supertype.
↳ An instance of subtype is also an instance of the supertype.
hierarchy start from parent

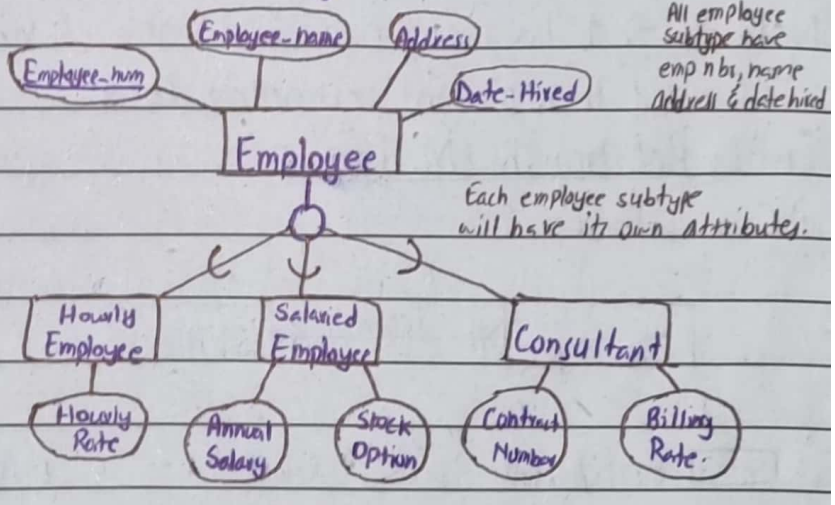
Composite Attribute: Attributes which can be divided further.
Example: Name can be divided in first name & last name.

Derived Attribute ↳ (dashed)

Multi-Valued: attributes having multiple values. Std 1 | OOP, PF, DB → course is multivalued.
↳ Define by (double oval)

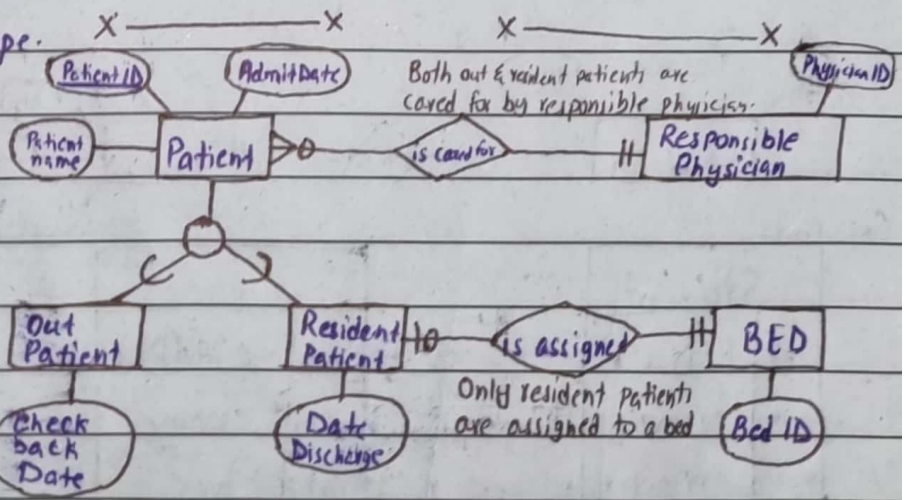
•) Supertypes will have all the common attributes of subtypes
Subtype attributes will be different for each subentity.

•) Data redundancy will be minimized if we make supertype & subtype.



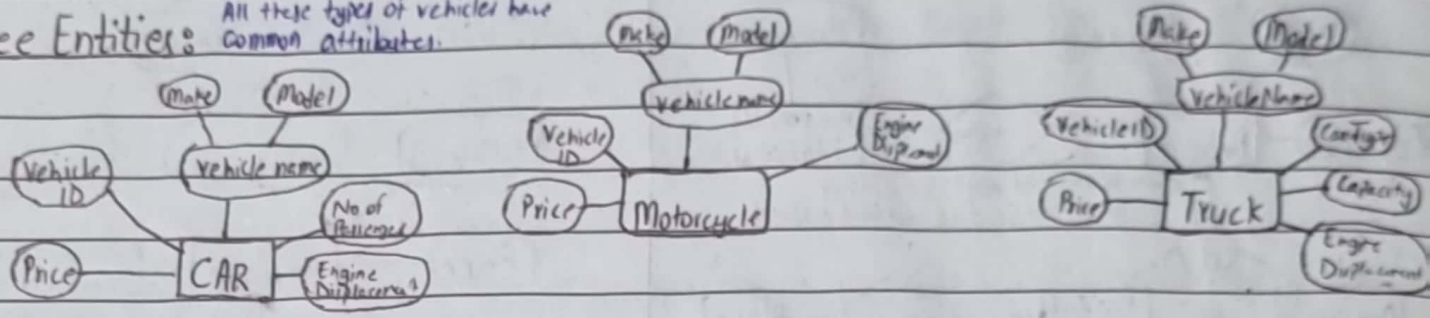
Generalization: Bottom-Up Approach.
From subtype to supertype.

Specialization: Top-Down Approach.
From supertype to make further Subtype.

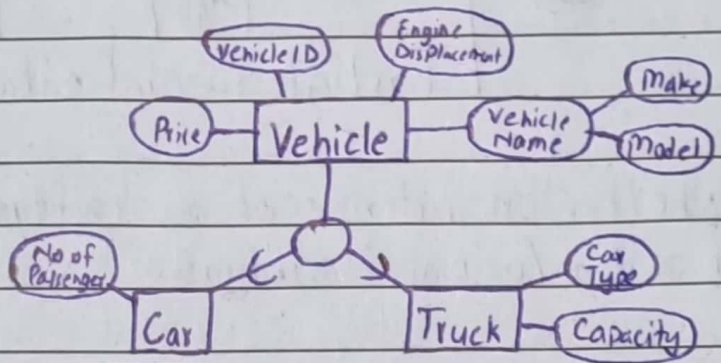


Date: _____

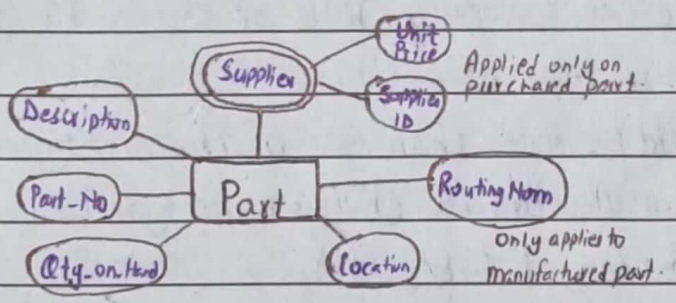
Three Entities: All these types of vehicles have common attributes.



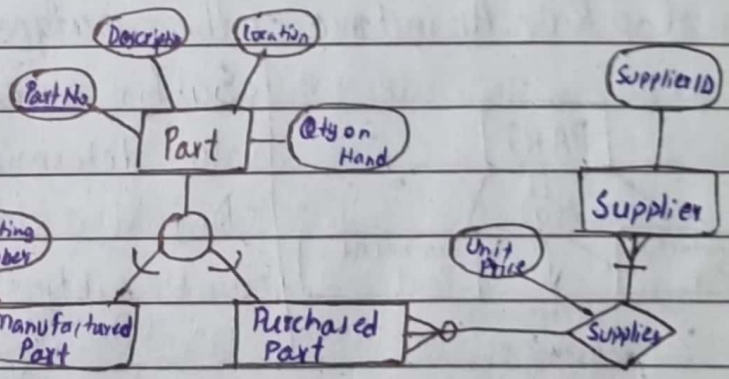
Generalization to vehicle supertype: So we put the shared attribute in super



Note: no subtype for motorcycle, since it has no unique attributes.



Specialization to Manufactured & Purchase



(created 2 subtypes

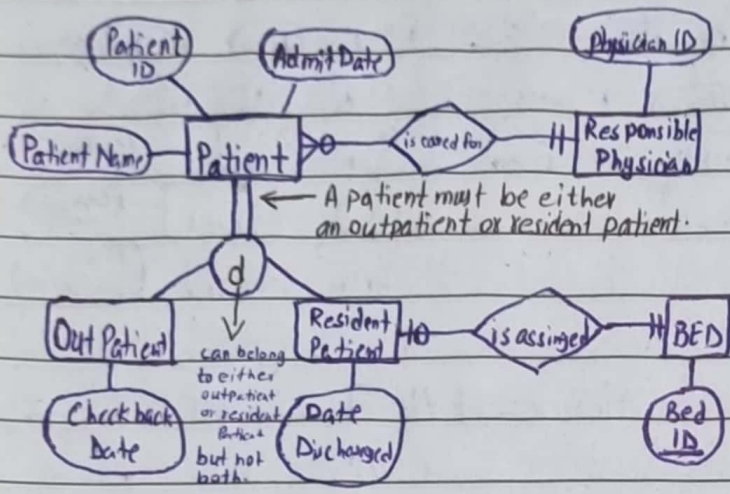
Completeness Constraints: Whether an instance of a supertype must also be a member of atleast one subtype.

Yes Total Specialization Rule: II (doubleline) Patient data must belong to either outpatient or resident pati

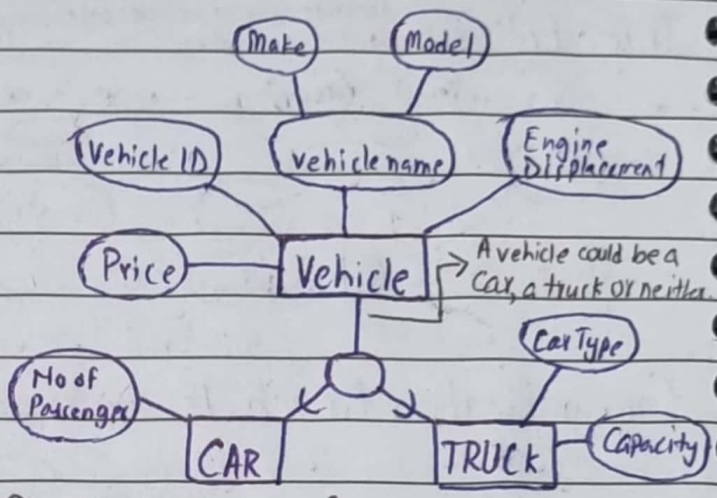
No Partial Specialization Rule: I (singleline) its belong to one category or does not belong to any other categ

↳ Overlap not possible.

Date: _____



Total Specialization rule:

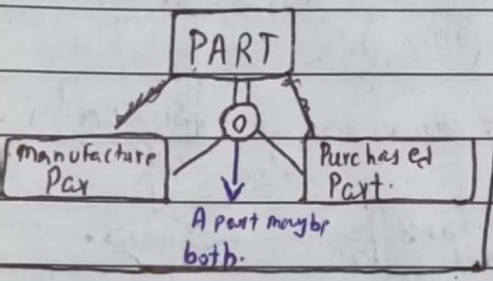


Partial Specialization:

Disjointness Constraints: Whether an instance of a supertype may simultaneously be a member of two (or more) subtypes.

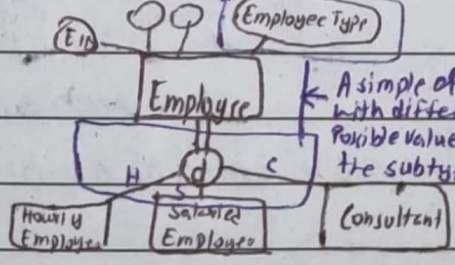
Disjoint Rule: An instance of the supertype can be only ONE of the subtypes.
↳ (d) Eg: Patient can only belong to one type at a time.

Overlap Rule: An instance of the supertype could be more than one of the subtypes.



Subtype Discriminator: An attribute of the supertype whose values determine the target subtype(s).

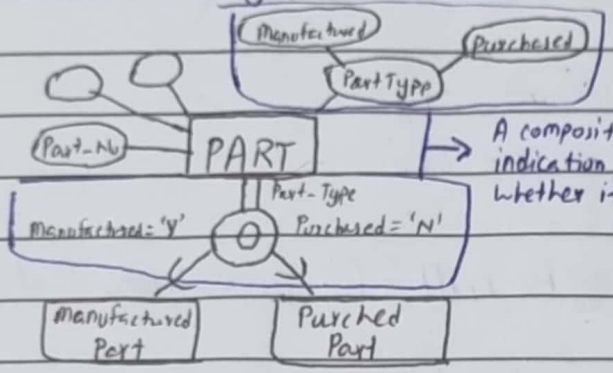
- Disjoint: a simple attribute with alternative values to indicate the possible subtype. If specific value given.



E_ID	E_Type
1	Salaried
2	Consultant
3	Hourly

Subtype Discriminator: Date: _____

Overlapping: a composite attribute whose subparts pertain to different subtypes. Each sub parts contains a boolean value to indicate whether or not the instance belongs to the associated subtype. If boolean value given:



→ A composite attribute with subattributes indication "Yes" or "no" to determine whether it is of each subtype.

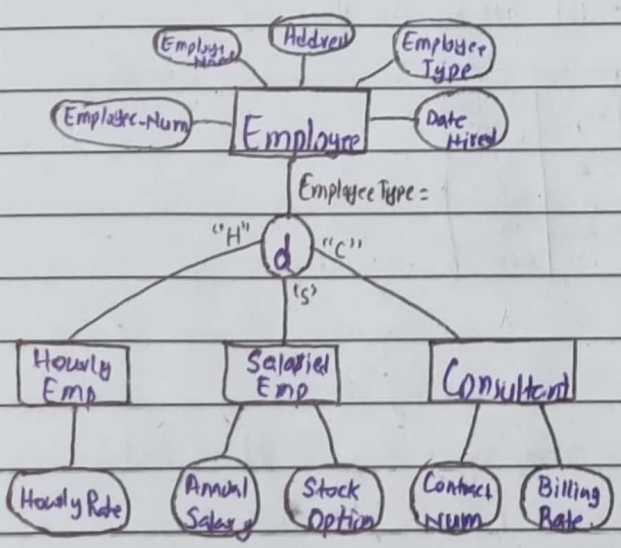
Part

Part_No	Manufacture	Purched
1	Y	Y
2	N	Y
3	N	N
4	Y	N

Transforming EER Diagrams into Relations:

Mapping Supertype/Subtype Relationships:

-) One relation for supertypes & for each subtype.
-) Supertype attributes (including identifier and subtype discriminator) go into ↑ supertype relation
-) Subtype attributes go into each subtype; primary key of supertype also become the primary key of subtype relation.
-) 1:1 relationship established between supertype & each subtype, with supertype as primary table.



Mapping Supertype/subtype relationships to relations:

Employee

Employee_Num	Employee_Name	Address	Emp_Type	Date_Hired

Hourly_Employee

H_Employee_Number	Hourly_Rate

Salaried_Employee

S_Employee_Number	Annual Salary	Stock Opti

Consultant

C_Employee_Number	Contract Num	Billing Ra

Referential Integrity: [illegible]
Must match a primary key value in the relation of one side (or foreign key can be null)
↳ constraints are drawn via arrows from dependent to parent table.

Date: _____

Relation: Table is called relation.
↳ every relation have unique name. And also every column have unique name.
↳ each row is unique. Value are atomic.

Integrity Constraints:

Domain Constraints: Primary key, Foreign key, Default, check;
↳ eg: column kae andr konsi value allowed hogi, we use check.

Entity Integrity: No primary key, attribute may be null - All primary key fields must have data.

Action Assertions: Business Rule. or Defined Rules
Read, Delete, Insert, Update.

Differential Integrity: Applied when we link two tables together.

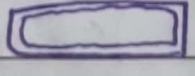
Delete Rules:
Restrict: We can't delete data if data is dependent or used by child.
Cascade: If we delete data from parent, so data from child also deleted if dependent.
Set to Null: If we delete data from parent, so in child null will be display if //

St ID	Courses
1	OOP, PF
2	DD, PF, OOP
3	PF, CAL, OOP
4	DB, PF, OOP
5	
6	

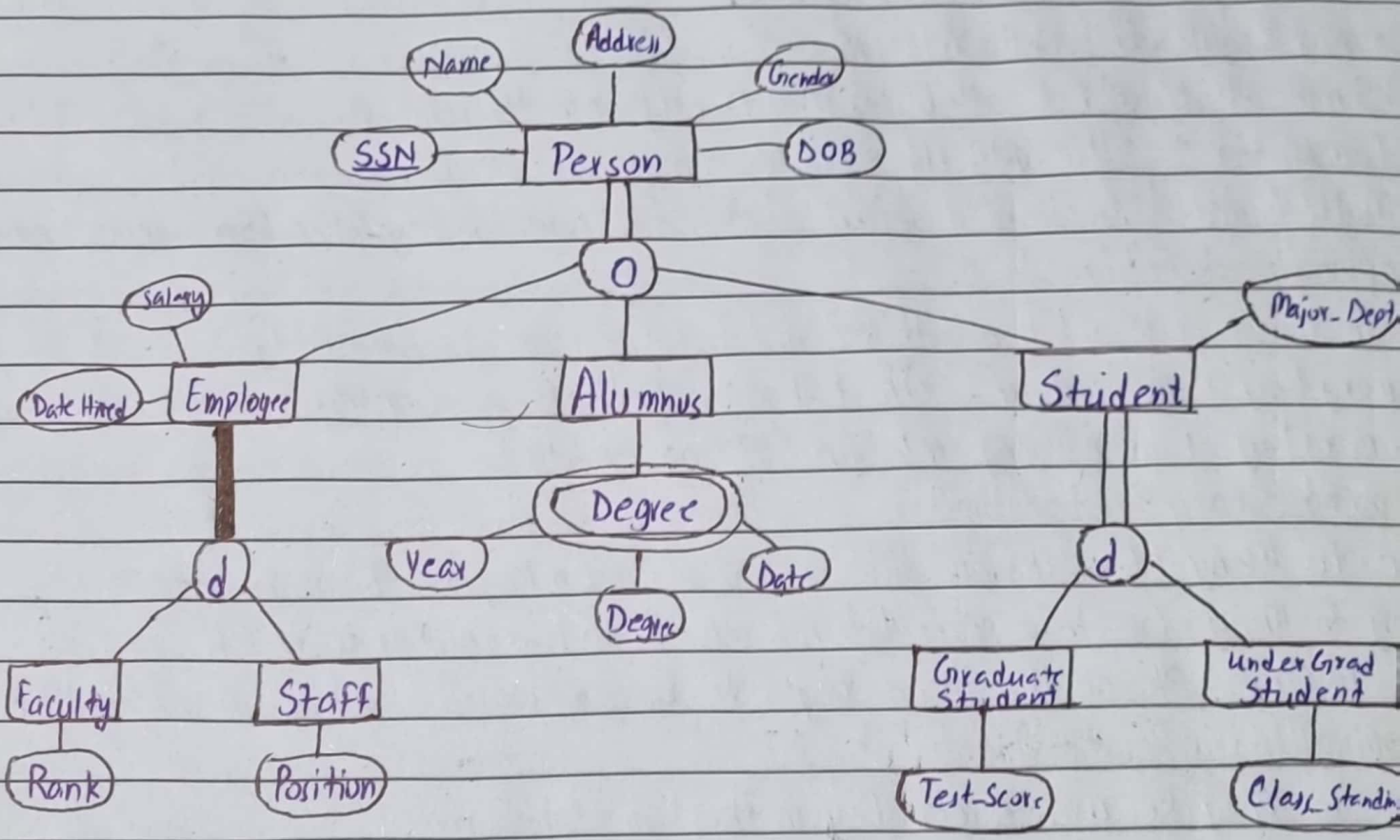
∴ Courses is multivalued. We Create seperate table for course that is multivalued And symbol is double oval

Composite key ↳ by combining 2 or more attributes.

St ID	Course
1	OOP
1	PF
2	DB
2	PF
2	OOP

Strong Entity: Consist of primary key.
Weak Entity: Don't have primary key, so we make composite key in that table.
↳ Symbol: 

Example of supertype/subtype Hierarchys Date: _____



Supertype: || Put Not Null: means must have subtype.
| Don't put Not Null: means subtype optional.

SubTypes: ⓞ Employee (Y/N), Alumni (Y/N), Student (Y/N): Multiple Fie

ⓓ Student_type ('U', 'G'): Single Field.

Date: _____

Transforming EER Diagram into Relations:

Mapping Regular Entities to relations:

1) Simple attributes: E-R attributes map directly onto the relation.
2) Composite ″ : Use only their simple, component attributes.
3) Multi-Valued ″ : Becomes a separate relation with foreign key taken from superior entity.

Mapping Weak Entity:

Becomes a separate relation with foreign key taken from superior entity.
Primary key composed of: ◦) Partial identifier of weak entity.
◦) Primary key of identifying relation (strong entity).

Mapping Binary Relationships:

One-to-Many: Pk on the one side become a foreign key on the many side.
Many-to-Many: Create a new relation with Pk of two entities as its Pk.
One-to One: Pk on the mandatory side becomes as foreign key on optional side.

Mapping Unary Relationship:

-) One-to-Many: Recursive foreign key in the same relation.
-) Many-to-Many: Two Relations: ◦) One for entity type.
◦) One for associative relation in which the Pk has two attributes, both taken from the primary key of the entity.

Mapping Supertype/Subtype Relationships:

-) One relation for supertype & for each subtype.
-) Supertype attributes (including identifier & subtype discriminator) go into supertype relati
-) Subtype attributes go into each subtypes; primary key of supertype relation also become Pk of subtype relation.
-) 1:1 relationship established between supertype & each subtype, with supert as primary table.

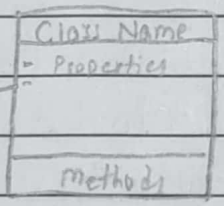
Date: _____

The Object-Oriented (OO) Model:

Data & Relationships contained in single structure known as an object
OODM is the basis for OODBMS - Semantic data model.
Objects contains operations.
Object is an abstraction of a real-world entity.
Attributes describe the properties of an object.
Objects that share similar characteristics are grouped in classes
Classes are organized in class hierarchy
Inheritance: object inherits methods & attributes of parent class.
UML based on OO concepts that describe diagrams & symbols.
-) Used to graphically model a system.

-) real world containers in which we define attributes & properties related to the class.
have same properties/fun/behaviours. -) Instance of class -) Class = Property/fun.
Method = Behaviour

UML:



-) Datatype include in all except chen notation, hierarchical, network model.

Inheritance: One class use the features of another class, when both classes have similar features.

Extended relational data model (ERDM)

-) Semantic data model developed in response to increasing complexity of the application.
-) Includes many OO model's best features.
-) Often described as an object/relational database (O/RDBMS)
-) Primarily geared to buisness applications.

Extention of relational model
Define complex datatype
Use advance/user defined datatype
Recursive relation.
Turnary relationship (3 tables)
To transport data from one system to another we use XML.
Ad-Hoc queries

Semantic
- Extended relation model
- Object Oriented

Object Orient...
Relational
ER Model
] imp.

Check Pg# 35/37 Lec-3 Date: _____

Database Model & the Internet:

-) Internet drastically changed role & scope of database market.
-) Dominance of web has resulted in growing need to manage unstructured information.
-) Current database support XML
  -) XML: the standard protocol for data exchange among systems & Internet sour

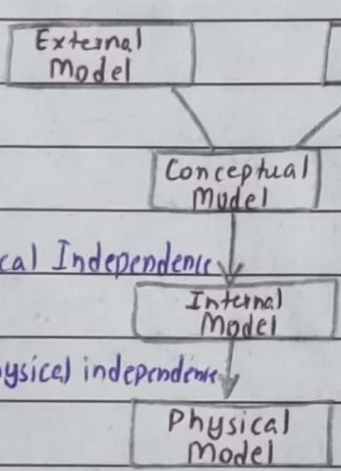
Data Model?

Business rule sae info aik conceptual model bana sake, uski design par kaam krske - It will be easier for us otherwise we will have restructure DB again & again.

⇒ Provide important detail & hide unimportant details.

Degrees of Data Abstraction: Database designer starts with abstracted view, then add detail
)External •)Conceptual •)Internal.

External Model: End users view of the data enviroment.
ER diagrams represents external views
External Schema: specific representation of the external view.
)Entities. -)Relationships -)Processes. -)Constraints.



Degree of Abstraction		Characteristics
High	ER Object Oriented	Hardware-independent Software-independent
Medium	Relational	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent. Software-dependent.

nsure security constrainst in database design.
implifies application program development.

Conceptual View interact with logical view which interact with physical view.

 Date: _____

The Conceptual Model: Represents global view of the entire database.
All external views integrated into single global view: conceptual schema.
ER model most widely used. ERD graphically represents the conceptual schema.
Doesn't depend on the DBMS software & Hardware used in implementation of the model.
Changes in hardware/sofware don't affect database design at conceptual level.

The Internal Model: Representation of the DB as 'seen' by the DBMS
-) Maps the conceptual model to the DBMS.
•) Internal schema depicts a specific representation of an internal model.
•) Depends on specific database software. Change in DBMS software requires internal model be changed.
Logical Model (Table structure on paper)
Logical Independence: change internal model without affecting conceptual model.

The Physical Model: Operates at lowest level of abstraction.
-) Describes the way data are saved on storage media such as disks or tapes.
Require the defination of physical storage & data access methods.
Relational model aimed at logical level
↳ Doesn't require physical-level details.
Physical independence changes in physical model don't affect internal model.

Levels of abstraction:

Models	Degree of Abstraction	Focus	Independent of
External	High	End user views	Hardware & Software.
Conceptual	↑	Global view of data (DB model independent)	" "
Internal	↓	Specific database model	Hardware
Physical	low	Storage & access model	Neither hardware nor software.

Conceptual → relation blw data.
Internal → datatype.
Physical → Storage requirement

Date: _____

External Model: What details enduser will have, we will define checks

Conceptual view: Logics of database, → Entities, connection blw entities, How one entity is dependent on another entity.
↳ Most used model is ER-Model.

Internal Model: Internal Schema. Table & the datatype of attributes. Constriant applied on it.

Physical independence: If we change or apply any constriant it will not make any impact on storage. Internal model isn't dependent on physical model.

If modify structure or any design etc in conceptual it will not affect internal model →
↳ Logical Independence.

Lecture: 04

Weak Entity → No Primary key.

Strong entity's Primary key & composite attribute of weak entity together can be composite for weak entity.

Mapping Binary Relationship: 1 to 1, M to M, 1 to M.

M to M

CID	Name
1	DB
2	DB lab
3	PF

Courses

SID	Name	email
1	A	
2	B	
3	C	

Student

SID	CID
1	1
1	2
2	1
2	3

enrollment

→ composite key = Primary key of the both table.

Weak entity composite key = Strong entity Primary key + attribute in weak entity.

If relationship have attributes then we will make its table.

Associative entity: Has its own attributes & link with 2 tables.

The Pk of those 2 tables will be foreign key in associative entity.

In one to one we donot create table for relationship its attributes cuz its doesn't have multiple values. Attributes will be added into secondary table in case to one to one.

We will create table for associative entity in one to many & many to many.

Associative entity → ◇ become its relationship & entity as well.

Date: _____

Unary relationship = relationship blw single entity.

Split entity in such a way that is considered 2 entities but actually it is one entity.

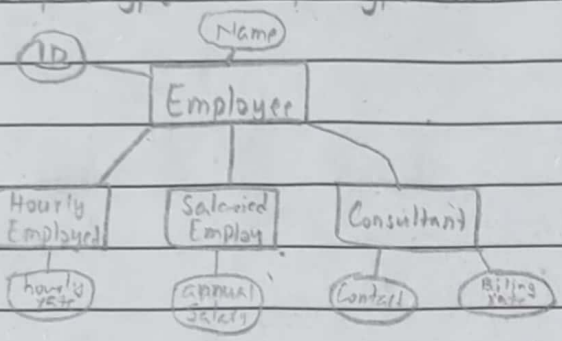
ID	Name	Design	Manager-ID

Conceptually ManagerID is another entity.
→ self join is use to connect them.

•) Unary is not possible in one to one.

ER → Relation Schema (tuenary)
ER → Table.

Supertype & Supertype both entities will have their table.



Employee Table
Hourly Employee → attributes → ID, hourly rate
Salaried " → " → ID, annual salary.
Consultant → " → ID, Billing rate, contact.

Normalization:

→ Remove anomalies.
→ Avoid Duplicate data facing problem while adding.

Goal to avoid anomalies.
- Insertion Anomali: ↑ data, data shouldn't be incomplete & duplicate.
- Update " : Update at one place but not at other location - lead to inconsistency.
- Delete " : Wanted to delete std(1) but we delete (1,0).

ID	Name	Project
1	A1	P1
2		

When we are deleting project the whole data of employee will be deleted

Insertion A: adding new rows forces user to create duplicate data.
Deletion A: deleting rows may cause a loss of data that would needed for other future row
Modification A: changing data in a row forces changes to other rows cuz of duplication.

•) A table shouldn't pertain to more than one entity type.

A → B → C

both attribute are dependent

Date: _____

ID	Course Name	Credit Hour

*) When we enter course we have to add CH they are dependent on each other this is called functional dependency.

Candidate key: Attributes in a table through which we can uniquely identify.
One of the candidate key will become Pk.
Eligible column which can be declared Pk are candidate key.
Consider one column, it is type of composite key.
Difference is that in composite we consider atleast 2 column but in candidate we consider 1 column.

Steps in Normalization:

1st NF: We need to remove multivalued attributes. If the table doesn't have multivalued attributes means we are at 1st NF.
2nd NF: Remove Functional & Partial dependency - Ensure that all the values of the table depends on Pk.
3rd NF: Remove transitive dependencies. A → B → C
4th BCNF: Remove multivalued dependencies - Multivalued dependency means data is dependent on multiple components.

eid	ename	Pid	Pname	Grade	Salary
1	Ali	1,2	P1 P2	17	1k
2	Zain	1	P1	16	8k
3	Ahmed	3,2	P2	16	10k

Atomic Value: Single row single value.

Pid	Pname
1,2	P1, P2

1	P1
1	P2

eid | ename | grade | salary

Pid | Pname] 1st NF

Insertion Anomali: When we are entering the data of an employee we have to assign value to project name attribute. We cannot leave it empty. This is insertion anomali.
Attribute is completely dependent on Pk not on the part of Pk is partial dependence.
To remove partial dependences we will split the table into two tables - One for employee & one for course & then we create a relationship b/w them.
relation will be linked

eid | Pid] 2nd NF

Functional dependency: One attribute value determines value of another attribute value. Eg: % dependent on grades.

Date: _____

Third Normal Form:

-) Eliminate transitive dependencies
-) dependency on non-key component in transitive dependency.
Eg: Pay scale is dependent on designation.

| eid | ename | Grade |

| Grade | Salary |

These two tables are linked. If we want to update in grade & salary table it will automatically updated in eid, ename, grade table.

Total 4 tables:

| eid | ename | grade | salary |

Split into

| eid | ename | grade |

| grade | Salary |

| Pid | Pname |

| eid | Pid |

So we have total 4 tables after 3rd NF.

Normalization: Process of organizing the tables in a way to reduce data redundancy
Normalization divides tables & there is a relation blw them.
-) Flexiable database design -) Database Security -) Reduce data Redundancy -) Organized DB.

OrderNo	OrderDate	Customer ID	Fname	Lname	Street Add	City	State	Zip	Email	Sale Person ID	Sale Pers Name	Sale Pers Phone	Prd No	Prod Des	Unit Price	Qt

Already in 1NF as no Multivalued attributes, attributes are atomic but we need to define PK

Full Dependency

2NF: Remove Partial Dependency

SALES(OrderNo, ProductNo, Qty)
Product(ProductNo, ProductDes, UnitPrice)
Order(OrderID, OrderDate, CustomerID, Fname, Lname, StreetAdd, City, State, ZIP, Email, SalepersonID, SalepersonName, SalepersonPhone).

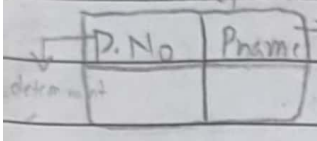
3NF: Remove Transitive Dependency.

SALE(OrderNo, ProductNo, Qty)
Product(PrdNo, PrdDesp, UnitPrice)
Order(OrderID, OrderDate, CustomerID, SalepersonID)
Customer(CustomerID, FName, LName, StreetAdd, City, State, ZIP, Email)
SalePerson(SalepersonID, SalePersonName, SalePersonPhone)

Functional Dependency: The value of one attribute (the determinant) determines value of another attribute.

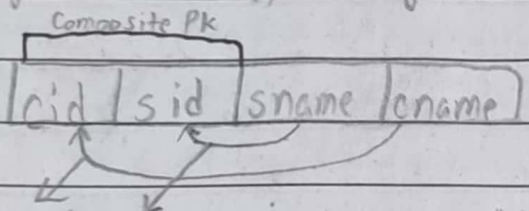
Superkey: Any key that uniquely identifies each row.
Candidate: A superkey without unnecessary attributes.
Referential Integrity: Fk contain a value that refers to an existing valid tuple (row) in another relation.
Secondary key: key used strictly for data retrieval purpose. Date: _____

Attribute of normalized table: -) Atomic Value -) Unique Name / data of table / column / records.
Funtional Dependency: When 1 column determines the value of another value.



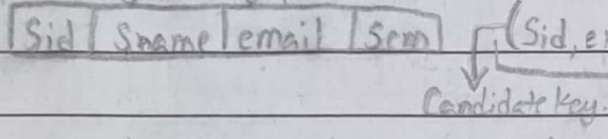
P-name → depends upon P.No

Determinant → determines the value of dependent.
Dependent → its value depends upon the determinent.
Fully Functional Dependency: When all the attributes completely depend upon primary key.



As both sname & cname don't depend upon composite key but part of composite key

Functional Dependency exist but not fully functional dependency.
key Attribute: Any attribute that is a part of of the composite Primary key.
Super key: Every possible combination that can uniquely identify the data.



In super key it creates combinations, sets of potential Pk.
Candidate keys are extracted from the Super key with minimal combinational value
The candidate key consider multiple colums without creating combinations.
Than the primary key is decided or chosen out of the candidate key.
Sid & email are candidate key from Superkey then Sid chosen as Pk due to fact email can be empty.
Null value in a column can create problem for the aggregate function.

The link created blw two tables (relation) must match the values in the Primary table with the secondary table.
Referential Integrity = ?
the values in the primary key column should match the values in foreign key column.

Relational Algebra: Selection, Projection, Union, Set Difference & Cross Product Date:

Different operators used for operations applied on data.

1) Union 2) Intersect 3) Grouping 4) Joins.

Union → used in place of full join in SQL - To apply union some conditions be met.

-> Same No. of columns -> Same datatype (in order) -> Order should be same.

Col2 → [int | varchar] → Col2 Col2 ← [int | varchar] → Col2.

Intersection: common data applied through inner join in SQL.

Difference: Common element are excluded & the uncommon value of 1st table are included only. (2nd column's values are not included even if they are uncommon)

Product: Implement in the form of cross-join the values of the first table are multiplied by each value of 2nd table (sets are created).

A (1, 3) B (a, b, c)

(1,a) (1,b) (1,c) 3 x 3 = 9 pairs

6 x 3 = 18 pairs.

Selection: Specific row is selected through which claim Horizontal fitting.

Select column-name from Table where condition.

Projection: Specific column is selected vertical fitting, no duplicate data in out

(π) Select column-name from Table

Joins: to merge different tables in order to find an output/result.

[D-id | Dname] Department [Cid | Cname | Did] Course Merging individual table to contain results.

Join → theta (also called natural Join)

↳ Equi → Inner, Outer → Left, Right, Full (Union)

Inner Join (Natural Join) We only define the column names of the first table only just the name of 2nd table it identifies 2nd column based on de

Left, Right] both use (=) sign.

Union, Intersection, Difference, Product] All these applied through joins.

Natural Join: Links table by selecting row with common values in common attribute(s)

EquiJoin: Links table on the basis of an equality condition that compares specified columns.

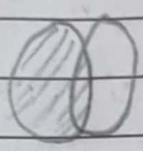
Theta Join: Any other comparision operator used.

Outer Join: Matched pair retained & unmatched values in other table left null.

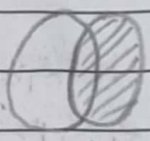
Date: _____

In equi Join we need to specify the names of the columns of the 2nd table as well as the 1st table.
Operators are not specified in natural join but specified in equi Join.
Natural Join merge the columns of 1st & 2nd table.

Left Join:
All data in left table.



Right Join



All data in right table.

Union, Intersect has their own keywords.

Divide Operator: the values of 1st table is checked w.r.t 2nd table and if the value match than that value is the result.

Code	Loc
A	5
A	6
A	9
B	5
B	9

Divide

Code
A
B

A = 5
B = 5
result → 5

Order No	Prod No	Cust No	Name	Add	City	St	Zip	Order date	Promised date	Desc

2NF

Cust No	Name	Add	City	St	Zip

Order No	Order Date	Promised Date	Qty Order

Prod No	Desc	Unit Price

Order No	Cust No	Prod No

3NF

Cust No	Name	Add	City

City	St	Zip

System Catalog: Contains Metadata, detailed system data dictionary that describe all object within DB.

Date: _____

Scalar → Single row single output Func.
Upper(), Lower() length() concat()
Agg func → multiple rows single output.
↳ count(), max(), min(), avg(), sum().

data dictionary: Structure of the table, with attributes, checks and datatype of attributes. ↖ Contains Metadata: data about data.

database Object: Joins, stored Procedures.

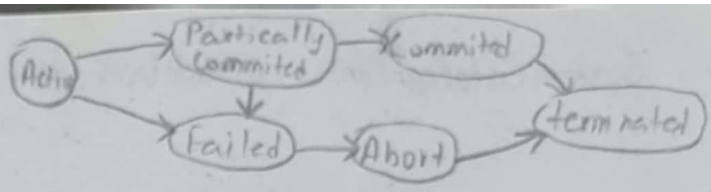
Indexes are used in DB to reduce the searching time. Index | Type | Page | 1

Cluster Index: Address of data is saved in the form of pointers.
↳ is created by default on PK column where a pointer is pointing to data.

Indexing → Mapping of key & values.
Create Index ____ on table(col)

Unique Index: has to implied (is not default) Each index is associated with only o
→ Can be applied only column with unique constrait (has null values as well)
One table can have multiple non-clustered indexes
↳ has to apply doesn't sort data.

Cluster Index: Sort in Ascending order (alphabet or numeric) defau
1 table can only have 1 cluster index.



Transaction generally represent change in DB.

Logical unit of work that must be either entirely completed or aborted. Date: _____

Transactions: Group of statements written in a block.

↳ May or may not be completed means it will execute completely or terminates in between.

From one consistant state to another consistent state. (Changes the DB)

Properties of transaction:

Atomicity: Process should be complete means line of code execute without any error, execute even with interuption. / Either all or none

Consistency: Befor & after transaction data should be valid. Before start & After completion Sum of money should be same.

Isolation: If any resource is using data than no other can use that data.

-) Two transaction can't access same data at a time.

Durability: If transaction is completed then we can't reverse it. / Permanent Changes.

Serializability: If multiple transaction perform at a time using same data then we will create mechanism like scheduling.

-) Concurrent transaction can be perform on multiuser database.

-) What is defined in transaction table? TransactionID, Transaction State, Timestamp, List of Lock, Undo info, LSN, Status Flag.

-) Scenario given, identify problem?

Transaction log: Start → Previous Null. End → Next Null.

Concurrency Control: When 2 different transaction use same data at a time There will be some problem

Both transaction read the data. Then one use updated value but other transaction use old read value.

Lost update problem: When concurent data Two concurrent transaction update same data elem One of the update is lost; Overwritten by other transac

Jiski transaction pehle commit hogi uski update loss hojae gi.

Uncommited Data: 2 transaction but due to any reason one transaction is rollback - So transaction will see uncommited data.

Order matters when read or write on same transaction.

Inconsistent Retrievals:
a) First transaction access data. b) Second transaction alter the data. c) First transaction access the data again
Transaction might read some data before they are changed & other data after changed.
Yield inconsistant results.

Date: _____

Schedulers: Allow delay transaction - Schedual the transactions.

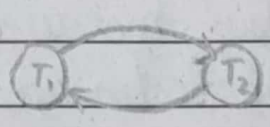
Lock: When a transaction isn't complete, till that time locked that transactio
so that resource can't be accessed
When 2 transaction use same resource Table, DB, row, Column, Pages, Feeds
can be lock.
Binary lock: Locks can be check by Flag: 0 means no lock & 1 → lock
Exclusive lock → Stop from both reading & writing.
Shared Lock → Lock on reading.

	S	X
S	T	F
X	F	F

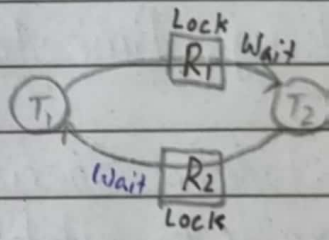
a) Agr aik block par shared lga ho a hai tou exclusive nhi lag skta

Deadlock: A condition where a resource wait for other resource, and other resource
wait for the First resource. Dependent on eachother.

Eg:



T1 dependent on T2
T2 dependent on T1



Transaction support is provided by two SQL statement: COMMIT & ROLLBACK.
Transaction sequence must continue until:
a) COMMIT: statement is reached. b) ROLLBACK statement is reached.
c) End of program is reached d) Program is abnormally terminated.

if no loop/cycle in precedence graph
then it a conflict Serializable

R-W
W-R
W-W.

Date: _____

Serializability:

Transaction are performed one after another without interleaving of operations.
No serial execution would allow these three problems to arise.
Regardless of which serial schedule is chosen
Serial execution will never leave the database in an inconsistant state & considered to be correct, even though different results maybe produced.
Goal: Allow transaction to execute concurrently & make sure no interference occur & produce database state as of truly serial execution
In Serializability following factors are imp:
If two trans are only reading a variable, they don't conflict & order is not important.
If two trans. operate on (read/write) completely separate variable they don't conflict and order is not important.
If one trans. writes to a variable & another either reads or write the same variable, the order of execution is important.
→ Serial → Consistant, but waiting. •) Parallel → Better Performance.
Scheduler: is used to allow operations to executed immediatly, delayed or rejecte
Serializability canbe achieved in many ways but most used techniques are: locking, time stamping and optimistic technique.

The Scheduler: Special DBMS program
Purpose is to establish ~~program~~ order of operations within which concurrent transaction are executed
Interleaves execution of database operation: •) Ensure serializability •) Ensure Isola
Serializable schedule:
Interleaved execution of transaction yields same result as serial execution.

Date: _____

Concurrency Control with Locking Method:

Lock: Guarantees exclusive use of a data item to a current transaction.
Required to prevent another transaction from reading inconsistant data.
Lock Manager: Responsible for assigning & policing the locks used by transaction.
Lock Granularity: Indicates level of lock use:

-) Database-level lock: Entire database is locked.
-) Table-level lock: Entire table is locked.
-) Page level lock: Entire diskpage is locked.
-) Row-level lock: Allows concurrent trans to access diff rows of same table
  -) Even if rows are located on same page.
-) Field-level lock: Allows concurrent transaction to access same row.
  -) Requires use of diff fields (attributes) within the row.

Lock Types:

Binary Lock: Two states: locked (1) or unlocked (0)
A flag is used in field, record, page or file to indicate portion of DB is loc

(X)
Exclusive lock: AND SHARED LOCK. (S) → Allow to read only.
↳ R & W both.

	S	X
S	✓	×
X	×	×

Deadlocks: Three techniques to control.) Prevention) Detection) Avoidance.

-) Choices of deadlock control method depends on database enviroment
  -) Low probablity of deadlock, detection, recommended.
  -) High probablity, prevention recommended.

ormalization: Date: _____

ID	Name	Address	Contact No	Prog ID	Prog - Name
01	Anas	Jauhar, Karachi	9002, 8007	P1	BBA
02	Ali	DHA, Lahore	2253, 4482	P2	BBM
03	Blil	Bhaxia, ISL	6789	P1	BBA
04	Usama	DHA, Karachi	3245	P3	CS
05	Kami	North, Karachi	2432, 5132	P2	BBM

1NF : [SID | Contact No] [SID | Name | Address | Prog ID | Prog Name]

2NF : // [SID | Name | Address | Prog ID] [Prog ID | Prg Name]

3NF : As there is no transitive depency so its in 3NF.

Date: _____

Relationships within the Relational Database:

1:M relationship: Relational modeling ideal.
Should be the norm in any relational database design.

1:1 relationship: Should be rare in any relational database design

M:N relationship: Cannot be implemented as such in the relational model.
↳ can be changed into two 1:M relationships.

CS221004 Assignment#02 M.Anas Date: 26-06-24

No SQL databases like MongoDB or Redis and traditional databases like MySQL, PostgreSQL or SQL Server serve different purposes & have distinct characteristic. No SQL databases are great for handling lots of data that doesn't fit nearly into tables, like social media post or sensor data. They can scale up easily as more user join in, making them perfect for apps that need to grow fast like real-time gaming or streaming services - In contrast, tradistional databases are like reliable organizers of the digital world - They ensure data integrity and support complex transactions, making them ideal for applications where consistency and accuracy matter most like baking system or online stores.

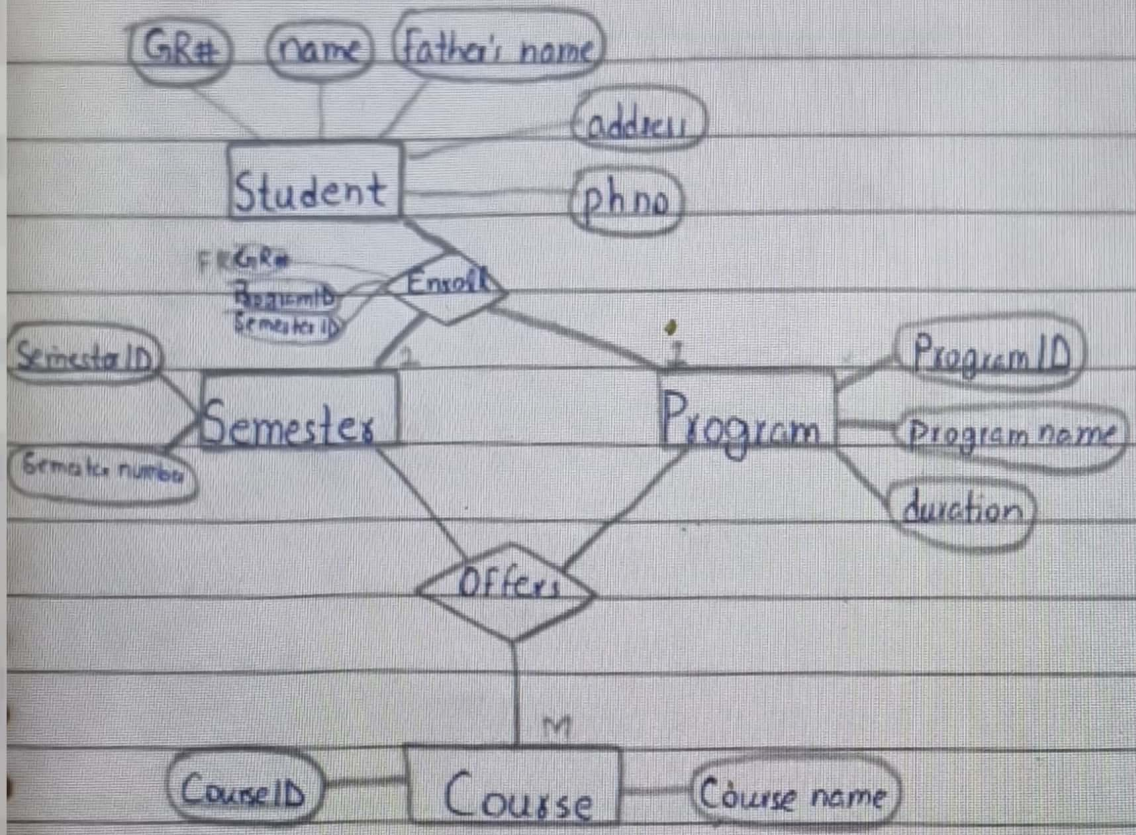
When you picking between them, consider the nature of your data & the specific needs of your application - If your data is varied and migh chang alot or frequent schema changes so NoSQL could be the way to g But if your application requires to stay in order, super reliable and ACID complaince a relational database maybe more suitable - It

It ultimately comes down to choosing the databases that aligns with your project's requirments & long term goals.

Assignment # 01

Date _____

Student Information System:

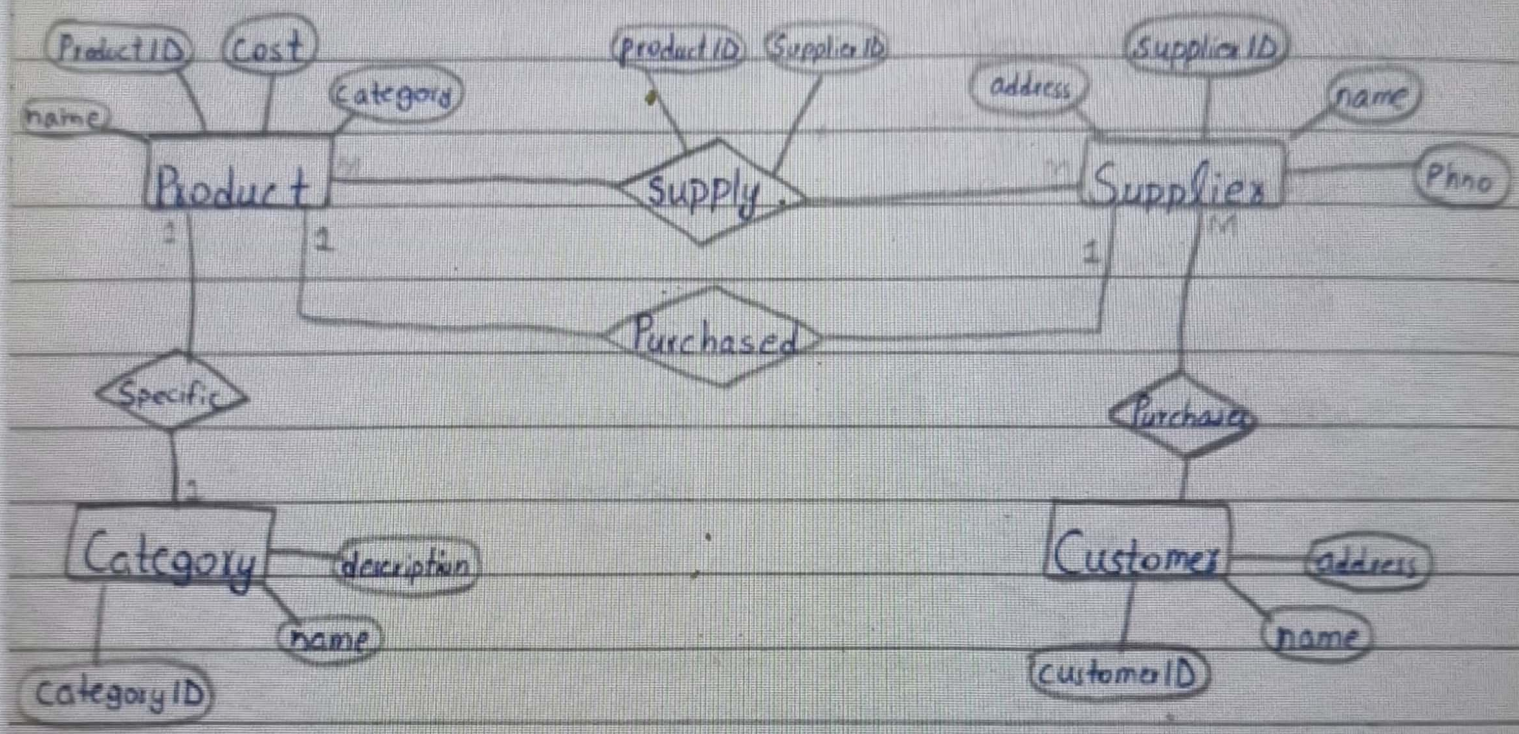


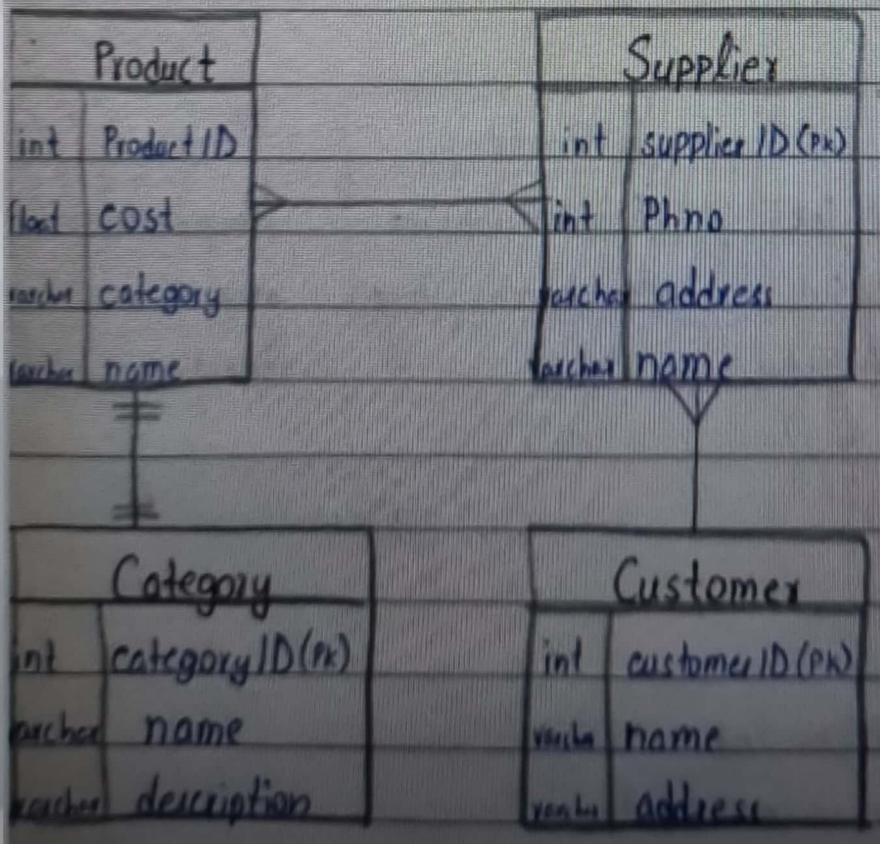
Student		Enroll (Rel)		Semester		Program	
GR# (PK)	int	GR# (FK)	int	SemesterID (PK)	int	ProgramID (PK)	int
name	varchar(50)	ProgramID (FK)	int	Semester name	varchar	Program name	varchar
Father's name	varchar(50)	SemesterID (FK)	int			duration	int
address	varchar(100)						
phno	int						

Offers (Rel)

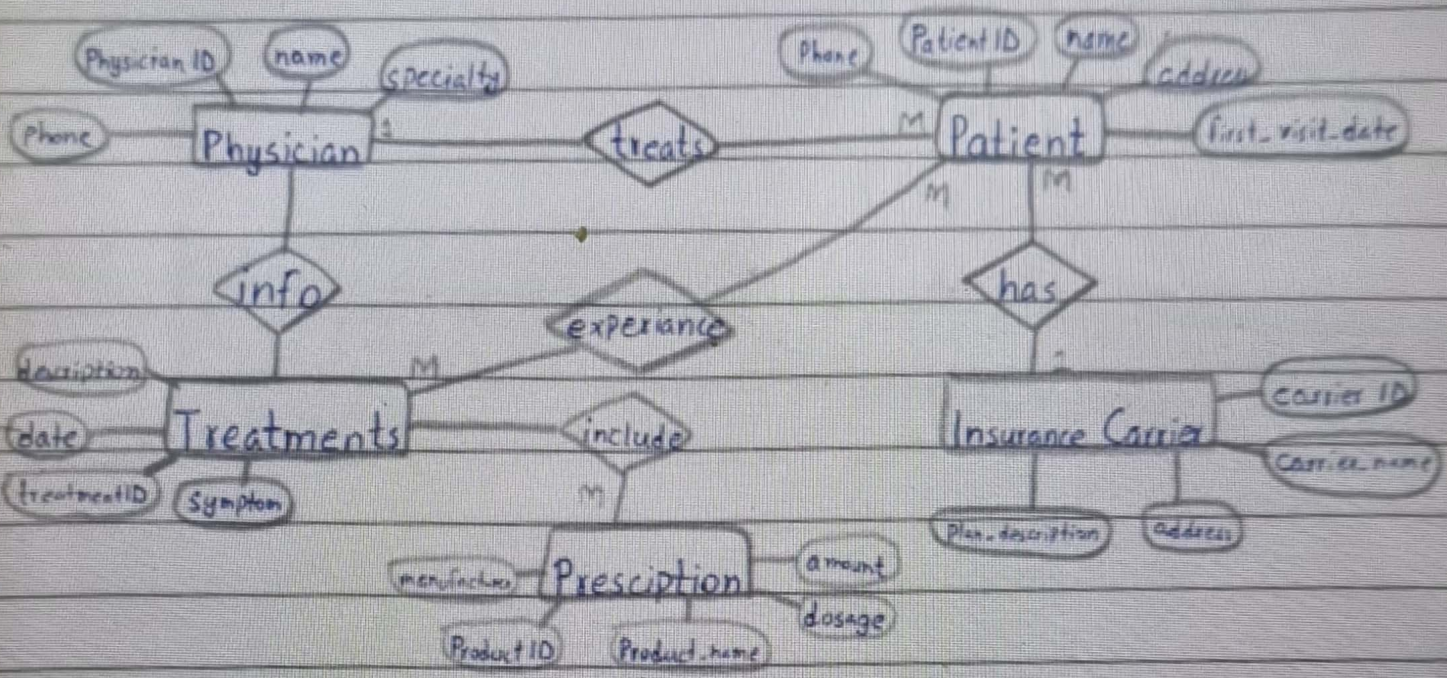
Course	
CourseID (PK)	int
Course name	varchar(50)

Super Store Inventory Control System:





Hospital Management System:



Physician	
int	physicianID
varchar	name
int	phone
varchar	speciality

Patient	
int	patientID
varchar	name
int	phone
varchar	address
data	first_visit_date

Insurance Carrier	
int	carrierID
varchar	carrier_name
varchar	address
varchar	plan_description

Treatement	
int	treatmentID
date	date
varchar	description
varchar	symptom

Prescription	
int	Product_ID
varchar	Product_name
Float	dosage
Float	amount
varchar	manufacturer