Unit I

PAGE No.
DATE / /

Introduction To DBMS

1. Purpose of Database System
2. View of Data
3. Data Models
4. Data Definition Language
5. Transaction Management
6. Database Administrator
7. Database Users
8. Overall System Structure

* **Database**

Database is a collection of data. It contain information about one particular enterprise.

Example:

1. Manufacuring company: which store product data
2. Bank: which store customer banking data
3. Hospital: Which store patient data.
4. University: which store student data

* **Database Management System (DBMS):-** DBMS is the collection of interrelated data and the set of program to access the data.

Ex:- Oracle, Microsoft Access, Foxpro, etc.

* **Advantages of DBMS**

1. Centralized Management and Control Over Data: Database administrator is a person who has central control over the system
2. Reduction of Redundancies: Centralized control of data ~~under its control by any number of application~~ by DBA avoid unnecessary duplication of data.

3. Shared Data: DBMS allow the sharing of data under its control by any number of application programmer and users
4. Integrity: Centralized control also ensures that adequate checks are incorporated in the database to provide data integrity
5. Security: The DBA can entire that process access procedures are followed, including proper authentication schemes for access to the DBMS and additional check before permitting access to sensitive data
6. Conflict Resolution: DBA resolves the conflicting requirement of various users and application
7. Data Independence: DBA can modify the structure of data record. Modification does not affect other application

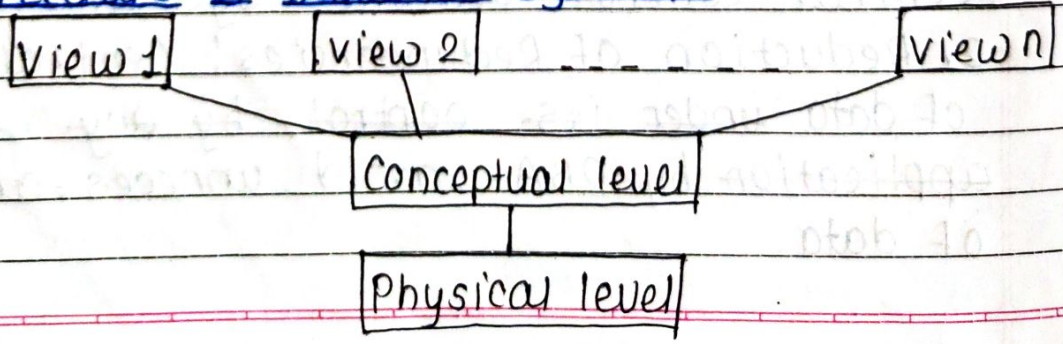
* Disadvantages Of DBMS

1. Number of Problems are associated with centralization of Data
2. Cost of Software and hardware
3. Complexity of back-up and recovery is increased

* Data Abstraction

Data Abstraction means to hide certain details of how the data is stored and maintained

- Architecture Of Database System



Interrelationship among the Three-level Of Abstraction

PAGE No.
DATE / /

* **Physical Level:-** It is the lowest level of abstraction It describes how the data is actually stored and defin describe the data structure and access method to be used by the database At the physical level, complex low-level data structure are described in detail

* **Conceptual level:-** It is also known as logical level. It is next higher level of abstraction. It describes what data is actually stored in the database and the relationship that exist among the data.

Ex:- type customer = record
name : string;
street : string;
city : integer;
end;

* **View level:-** It is highest level of abstraction. It describe only a part of the entire database. Views can also hide information for security purpose

- **Data:-** known facts that can be recorded and have an implicit meaning

- **Mini-World:-** Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university

- **Database System:-** The DBMS software together with the data itself. Sometimes, the application are also included.

PAGE No.
DATE / /

* Data Models

- Data Model is the collection of conceptual tool for describing:
  1. Data
  2. Data Schema
  3. Consistency Constraints.
- Data models are classified into following three categories:
  1. Object Based logical Data Model.
  2. Record Based logical Data Model.
  3. Physical Data Model.

- 1. Object Based Logical Data Model

- Object-based logical models are used in describing data at:
  i) logical level
  ii) View level
- Following are the object based logical models:
  1. Entity Relationship Model
  2. Object Oriented Model
  3. Semantic Data Model
  4. Function Data Model.

- Entity Relationship (E-R) model consist of a collection of basic objects, called entites and relationship among these object. An entity is an object that is distringuishable from other object by a specific set of attributes. A relationship is an associated among several time.

PAGE No.
DATE / /

Object Oriented Model is based on a collection of object. An object contain values stored in instance variables within the object and bodies of code that operate on the object. These bodies of code are called 'Method'

- Objects that contain the same type of value and the same method are grouped together into classes. A class may be viewed as a type defination for object

-2. Record-Based Logical Model: Record base logical ~~metho~~ models are used in describing data at:

i. logical level
ii. View level

In record-based models, database is structured in fixed format records of several types. Each type defines a fixed number of fields or attributes and each field is usually of fixed length

- Following are the record based data models:

1) Relational Model :- It uses a collection of table to represent both data and relationship among those data.

2) Network model: In network model, data is represe.nted as collection of record and relationship among data are represented by link, which can be viewed as pointed. The records in database are organized as collection of arbitary graphs.

3) Hierrarchical model :- It is similar to Network model, in the sense that data and relationship among data are represented by record and link respectively.

- Schema is sometimes called as intension of the database
- Instance is sometime called as extension or state of the database

PAGE NO. DATE / /

-3. Physical Data Model:-
- It is used to describe data at the lowest level. Following are the physical models.

1) Unifying model,
2) Frame memory model.

*Instances And Schemes
- Collection of information stored in the database at a particular (instance of time) moment is called an instance of the database.

The overall design of the database is called the database schemas.

- Database System supports three database schema

1) Physical Schema:- It is a lowest level i.e. at Physical level.
2) Logical Schema:- It is at the next or imediate level. i.e at logical level
3) Sub-Schema:- It is at the highest level, i.e. at the view level.

*Data Independence:- The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called data independence. There are two levels of data independence:

1. Physical Data Independences
2. Logical Data Independences.

- Physical Data Independences is the ability to modify the physical schema without causing application program to be rewritten.
- Logical Data Independences is the ability to modify the logical schema without causing

application programs to be rewritten.

* Database Languages

Data Sublanguage consist of two parts:
1) Data Defination language
2) Data Manipulation language

* Data Defination language :- 1) Database schema is specified by a set of definations which are expressed by a special language called Data Defination language (DDL)

- Data Dictionary :- The result of compilation of DDL statement is a set of tables, which stored in a special File called Data Dictionary or System Catalogues. This File contain meta data, i.e. data about data.
- Data Storage and Defination Language :- The storage structure and access method used by the database system are specified by a set of definition in a special type of DDL called as Data storage and Defination Language.

* Data Manipulation language :- DML is a language that enable users to access or manipulate data as organized by the appropriate data model

- Data manipulation means:-
  - ~~Data~~ To retrieve the information From database
  - To insert information into database
  - To delete information from database
  - To modify information from database.

- There are two type of DML:-

1) Procedural DML :- It requires a user to specify what data is needed and how to get that data

2) Non-procedural DML:- It requires a user to specify what data is needed without specifying how to get that data.

- Query is a statement requiring or requesting the retrieval of information
- Query language is the portion of DML that involves information retrieval

• Database Management System (DBMS)

- Collection of interrelated data
- Set of programs to access the data
- DBMS contains information about a particular enterprise
- DBMS provides an environment that is both convienent and ~~effective~~ efficient to use.
- Database Applications:
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customer, product, purchases
  - Human resources: employee record, salaries, tax deductions
- Databases touch all aspect of our lives.

• Purpose Of Database System.

In the early days, database applications were built on top of file system

Drawbacks of using file systems to store data:

- Data redundancy and inconsistency - duplication of information in different files.
  - Uncontrolled duplication of data is undesira

-ble for following reasons:
- Duplication costs time and money to enter data more than once.
- It tasks additional storage space thus again increasing associated costs.
- It may lead to data inconsistency

- Difficulty in accessing data
  - Need to write a new program to carry out each new task

- Drawbacks of File System (cont.)
- Data isolation - multiple files and formats
  - When data is isolated in separate files, it is more difficult to access data ~~is isol~~ and to ensure that data is correct.
  - Also, the structure of file depends on application programming language. Thus the direct incompatibility of such files makes it difficult to process jointly.
- Integrity problem - Integrity constraints (e.g. account balance > 0) becomes part of program code
  - Hard to add new constraints or change existing ones.
- Atomicity of updates - Failure may leave database in an inconsistent state with partial update carried out.
- Concurrent access by multiple users
  - Concurrent accessed needed for performance
- Security Problem

Database systems offer solution to all the problems.

* Advantages OF DBMS

- Control of data reduadancy
- Data consistency
- More information from the same amount of data
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale
- Balance of conflicting requirements
- Improved data accessibility and responsiveness
- Increased productivity
- Improved maintenance through data independence

* Disadvantages OF DBMS

- Complexity - provision of the functionality we expect from DBMS makes it extremely complex
- Size - complexity and breadth of functionality makes DBMS extremely large piece of software
- Cost of DBMSs - it varies significantly depending on the environment & functionality provided
- Additional hardware costs:- to achieve required performance, it is necessary to procure large memory.
- Performance:- DBMS is written to be more general, to cater for many application rather

than just one.

• Components OF DBMS

1. Hardware - DBMS and the applications required hardware to run
- It can range from PC to mainframe or network of computer
- It depends on the organization's requirement and the DBMS used.

2. Software - It comprises of following;
- DBMS software itself
- Application program
- Operating system including network software

3. Data - Most important component from end-user's point of views
- It acts as a bridge between the machine components and the human component
- The database contains both operational data and the metadata

4. Procedures - It refers to the instructions and rules that govern the design and use of the database.
- user of the system require documented procedure on how to use/run the system
- It may consists of instructions like
- log on to the DBMS
- use a particular DBMS facility or application program
- start and stop DBMS
- Make backup copies of the database
- Handle H/W and S/W failure
- Change structure of table to improve performance

5. Peoples - i.e USERS

• Database Administrator.

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs!
- Database administrator's duties includes!
  - Schema definition
  - Storage structure and access method defination
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

* Database Users

Users are differentiated by the way they expect to interact with the system

1. Application programmers:- They are the developer who interact with the database by mean of DML queries. The DML queries are written in the application program like C, C++, Java, Pascal, etc. These queries and converted into Object code to communicate with the database.

2. Sophisticated users:- They are database developer who write SQL queries to select/insert/delete/update data. They do not use any application or program to request the database. They directly intereact with the database

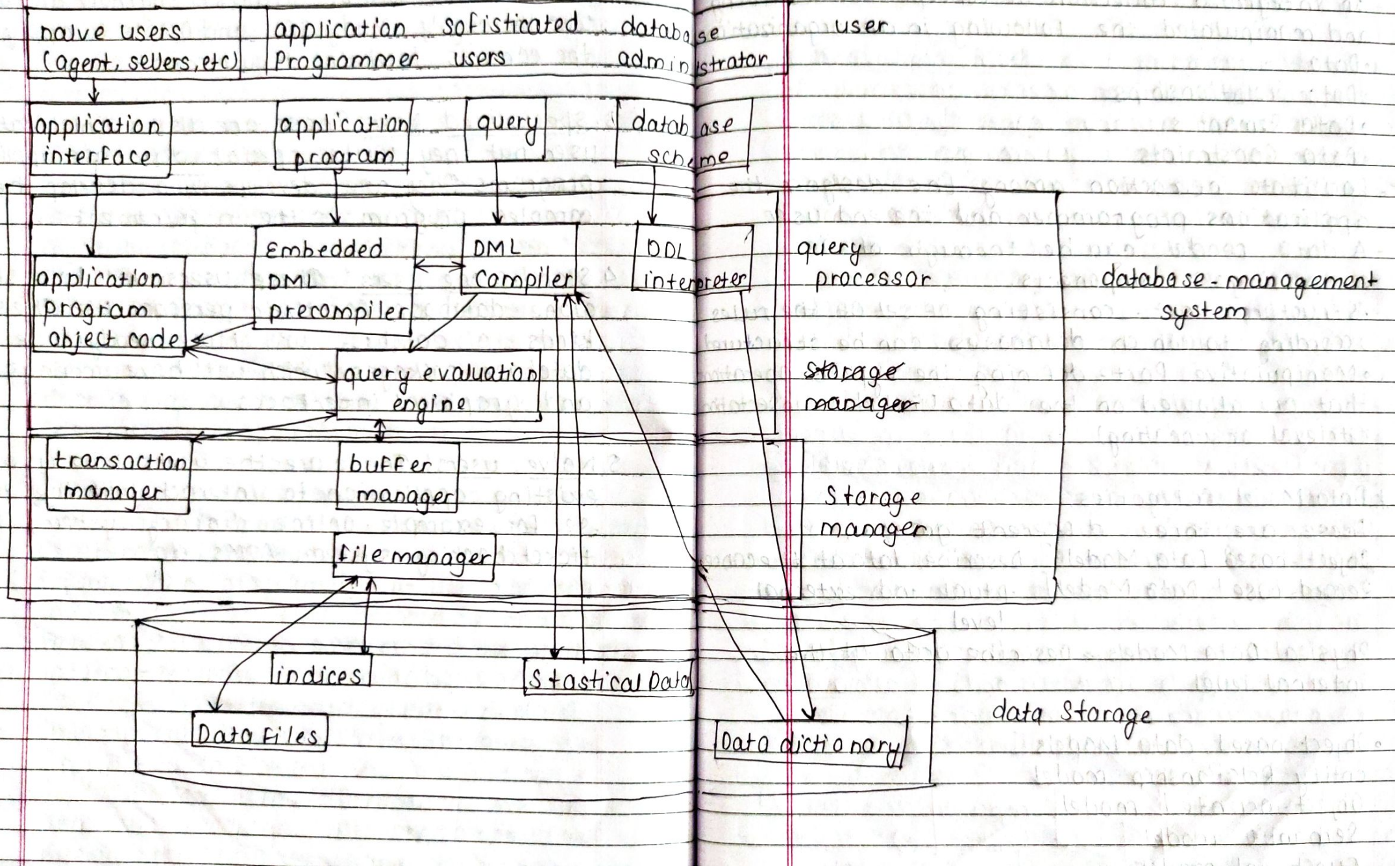
by mean of query language like SQL. These users will be scientist, engineer and analyst who thoroughly study SQL and DBMS to apply the concepts in their requirement

3. Specialized Users: These are also sophisticated user, but they write special database application programs. They are develops who develop the complex program to the requirement

4. Stand-alone users :- These users will have stand alone database for their personal use. These kinds of database will have readymode database packages which will have menus and graphical interfaces

5. Naive users :- These are the users who use the existing application to interact with the database. for example, online library system, ticket booking system, ATMs, etc.

* Overall System Structure



PAGE No.
DATE / /

* Data Model

- An integrated collection of concept for describing and manipulated the following in an organization
  - Data
  - Data relationships
  - Data Semantics
  - Data Constraints
- Facilitate interaction among the designer, the applications programmer, and the end user
- A data model can be through of as comprising 3 components.
  - Structural Part - consisting of set of the rules according to which databases can be structured
  - Manipulative Part - defining the type of operation that are allowed on the data (includes insertation retrieval or updating)

* Data Model Categories.

These are three different groups:

Object-based Data Model } Describe data at the conceptual and external level

Record-based Data Model }

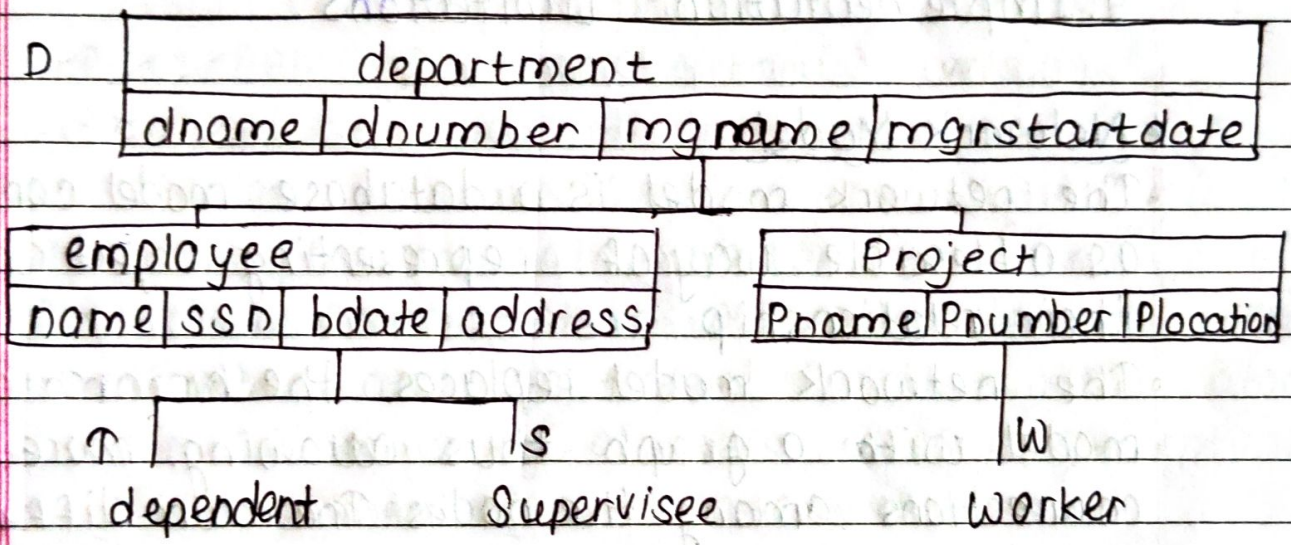
Physical Data Models → Describe data at the internal level

- Object-based data Models
  - Entity Relationship model
  - Object-oriented model
  - Semantic model
  - Functional model
- Record-based logical models
  - Relational model, Network model, Hierarchical model

- Physical Data Models
Unifying model. Frome memory

* Hierarchical Model

- A hierarchical data model is a data model in which the data is organized into a tree-like structure.
- The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent.
- All attribute of a specific record are listed under an entity type



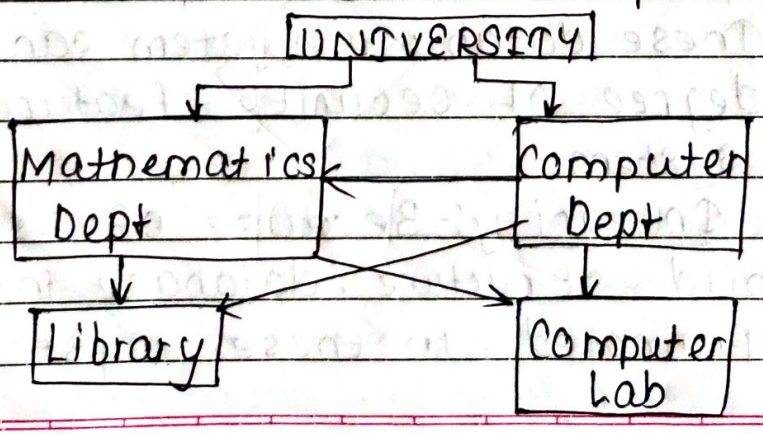
* Advantages

1) Simplicity:- Data naturally have hierarchical relationship in most of the practical situation
2) Security:- These database system can enforce varying degree of security features unlike flat-file system
3) Database Integrity:- Because of its inherit parent-child structure, database integrity is highly promoted in these system.

4. Efficiency: The hierarchical database model is a very efficiency

* Disadvantages :-
1. Complexity of Implementation
2. Difficulty in Management
3. Complexity of Programming
4. Poor Portability
5. Database Management Problems
6. Lack of Structural Independence
7. Programs Complexity
8. Operational Anomalies
9. Implementation Limitations.

* Network Model
- The network model is a database model conceived as a flexible way of representing object and their relationship
- The network model replaces the hierarchical model with a graph thus allowing more general connections among the nodes. The main difference of the network model from the hierarchical model is the ability to handle many to many relationship. It other words it allow a record to have more than one parent



PAGE NO.
DATE / /

* Advantages:-
1) Conceptual Simplicity
2) Capability to handle more relationship types
3) Ease of data access
4) Data integrity
5) Data Independence
6) Database standards.

* Disadvantages:-
1) System Complexity
2) Operational Anomalies
3) Absence of structural independence

* Three-schema Architecture
- Defines DBMS schemas at three levels.
- Internal schema at the internal to describe physical storage structures and access paths. Typically uses a physical data model.
- Conceptual schema at the conceptual level to describe the structure and constraints for the whole database for a community of users. Uses a conceptual or an implementation data model
- External Schemas at the external level of describe the various user views. Usually uses the same data model as the conceptual level

- Mapping among schema level are needed to transform request and data. Program refer to an external schema, and are mapped by the DBMS to the internal schema for execution
- Mapping between internal and conceptual schemas is known I/C mapping
- Mapping betn external and conceptual schemas is

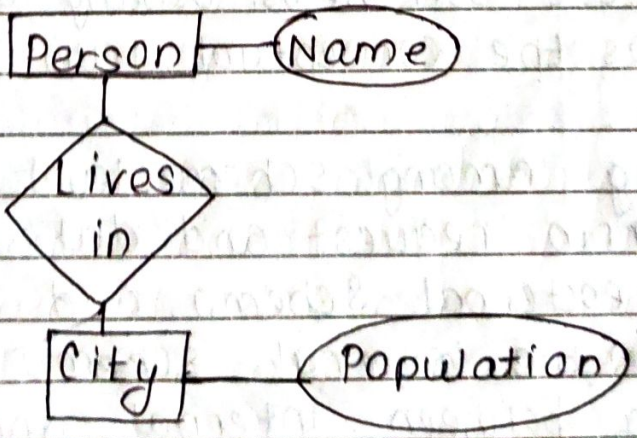
known E/C mapping

* Data Independence

- The disjointing of data discriptions from the application program (or user-interfaces) that user the data is called data independence
- Data independence is one of the main advantages of DBMS.
- The inteface bet^n the various levels and componen -ts should be well defined so that changes in some parts do not seriously influence others
- These are two kind of data independence
  - Physical Data independence
  - Logical Data independence.

* Entity-Relationship Model

- An entity Relationship (ER) diagram is a specialize graphic that illustrate the interrelationship bet^n entities in a database
- ER diagrams often use symbol to represent three different type of information. namely, entities, attributes and relationship



- How do we start ERD

• Define Entities: These are usually nouns used in descriptions of the system, in the discussion of business rules, or in documentation; identified in the narrative.

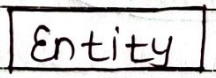
• Define Relationship: These are usually verbs used in descriptions of the system or in discussion of the business of the business rules; (entity.... entity); identified in the narrative

• Add attributes to the relations: These are determined by the queries, and may suggest new entities

* ER Diagram Symbols

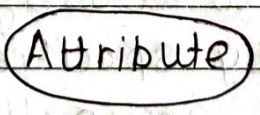
1) Entity

- An entity is an object or concept about which you want to store information



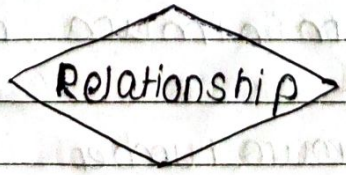
2) key attributes

A key attributes is the unique, distinguishing characteristic of the entity.



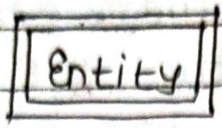
3) Relationships

Relationships illustrate how two entities share information in the database structure



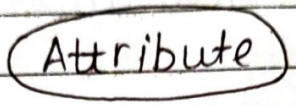
4) Weak Entity

- A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone



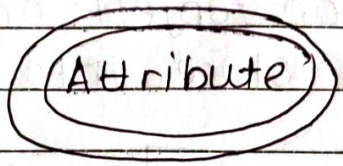
5) Derived attribute

- A derived attribute is based on another attribute



6) Multivalued attribute

- A multivalued attribute can have more than one value



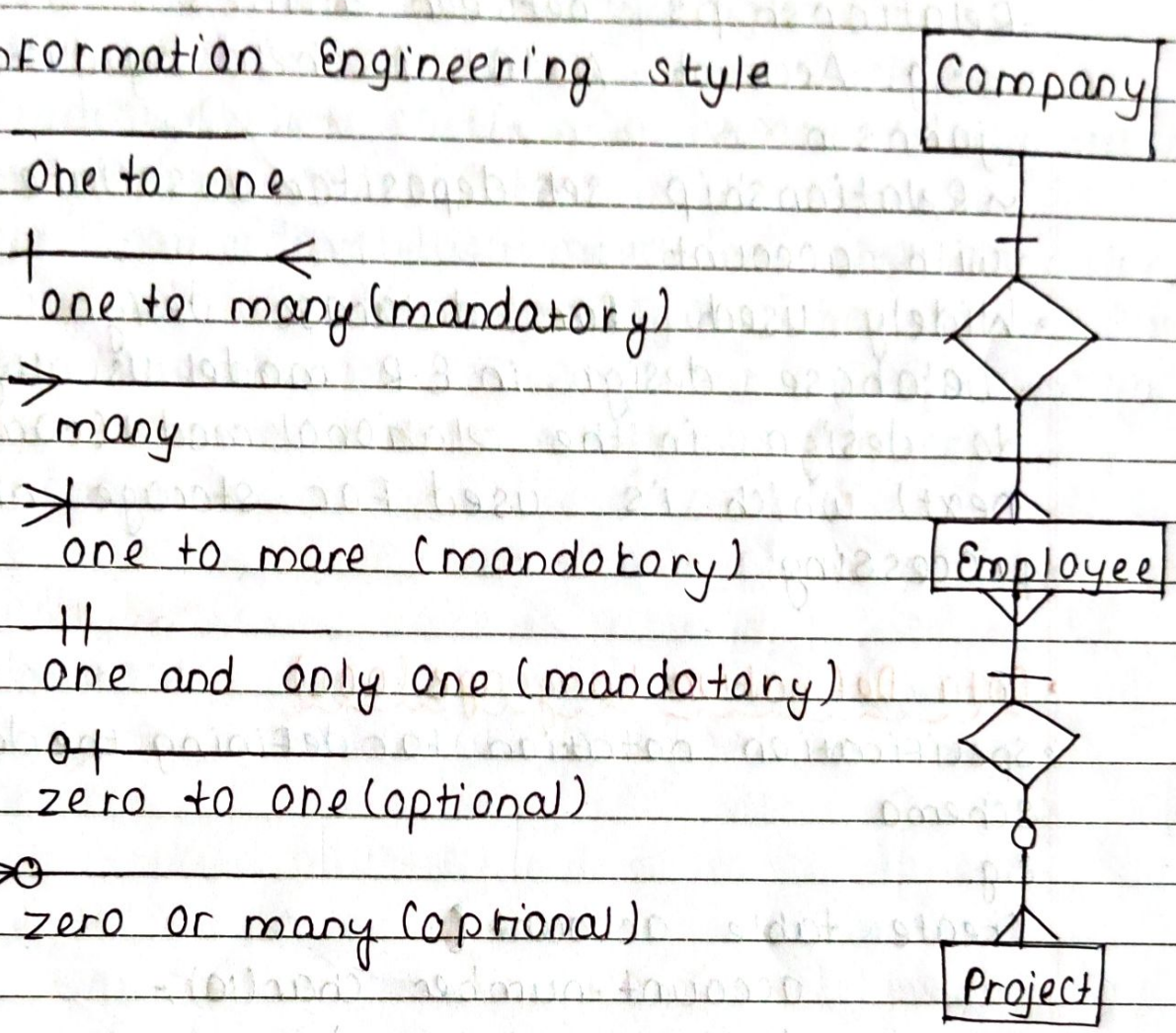
* How to express Relationship

- Cardinality specifies how many instances of an entity relates to one instance of another entity
- Ordinality is also closely linked to cardinality
- While cardinality specifies the occurrence of a relationship, ordinality describles the relationship as either mandatory or optional
- In other ward, cardinality specific the maximum number of relationship and ordinality the absolute minimum number of relationships.
- When the minimum number is zero, the relationship is usually called optional and when the ~~num~~

minimum number is one or more the relationship as usually called **mandatory**

- **Relationship Symbols**

Information Engineering style



one to one

one to many (mandatory)

many

one to more (mandatory)

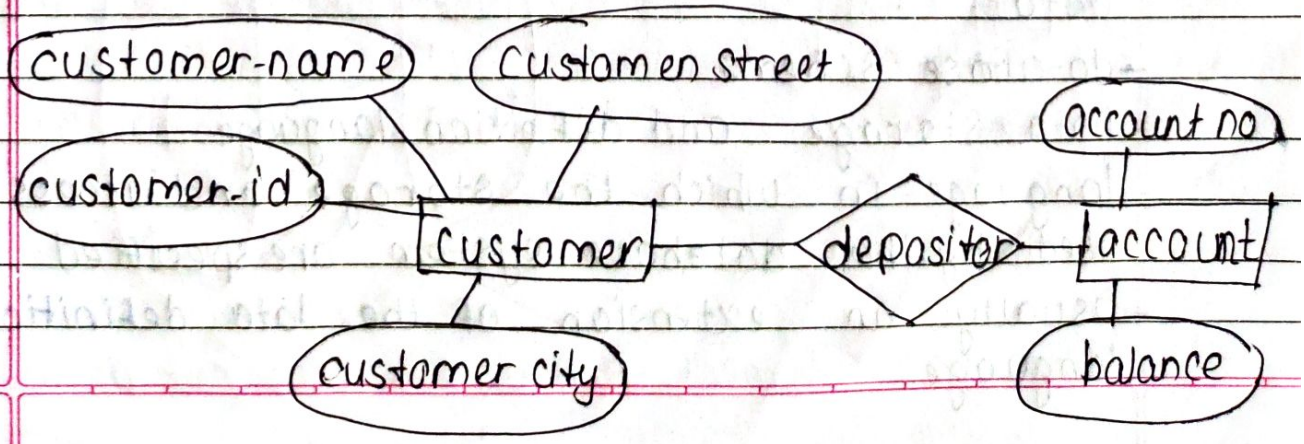
one and only one (mandatory)

zero to one (optional)

zero or many (optional)

* **Entity Relationship Model**

Example of schema in the entity-relationship model



PAGE No.
DATE / /

• Entity Relationship Model (cont.)

- E-R model of real world
  - Entities (objects)
    - E.g. customers, accounts, bank ~~balance~~ branch
  - Relationships between entities
    - E.g. Account A-101 is held by customer johnson
    - Relationship set depositor associate customer with account
- Widely used for database design
  - Database design in E-R model usually converted to design in the relational model (coming up next) which is used for storage and processing

• Data Definition Language (DDL)

- Specification notation for defining the database schema

E.g.

```
Create table account (
        account-number char(10),
        balance integer)
```

- DDL compiler generates a set of tables stored in a data dictionary
- Data dictionary contain metadata (i.e. data about data)
  - database schema
  - Data Storage and definition language
  - language in which the storage and access method used by the database system are specified
  - Usually an extension of the data definition language.

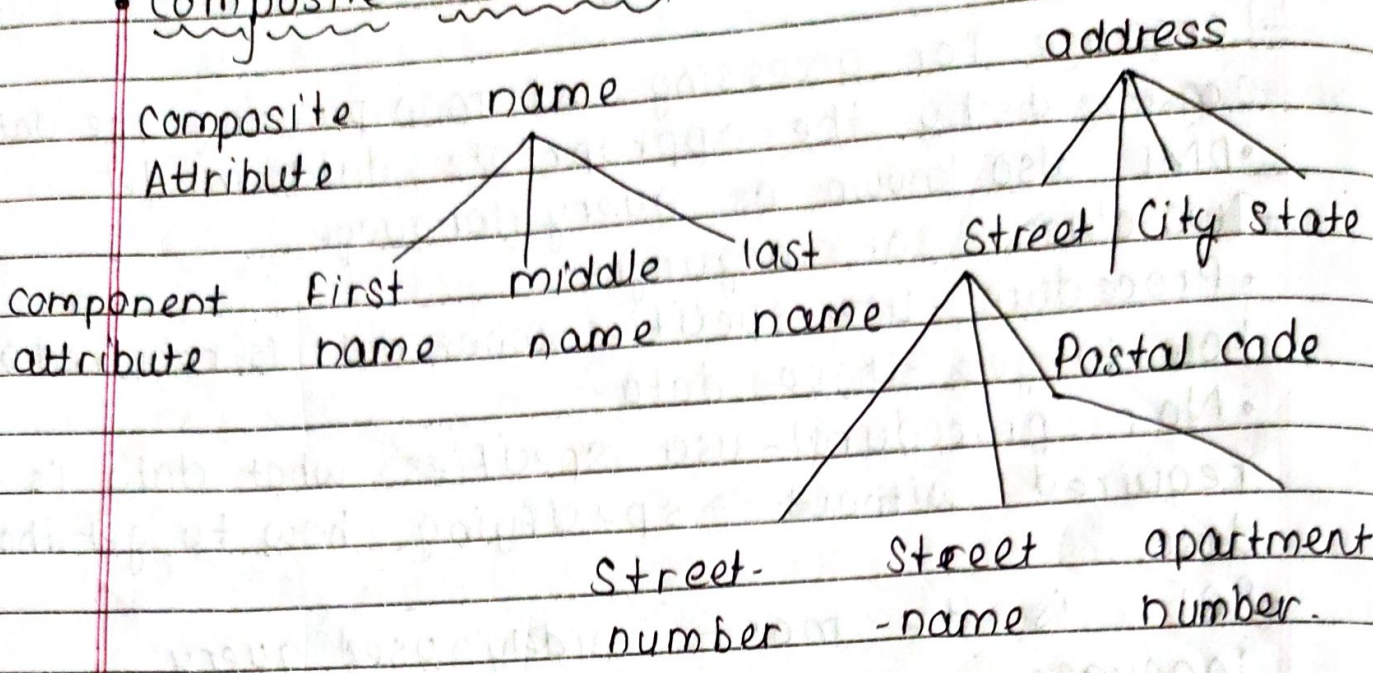
PAGE No.
DATE / /

* Data Manipulation Language (DML)

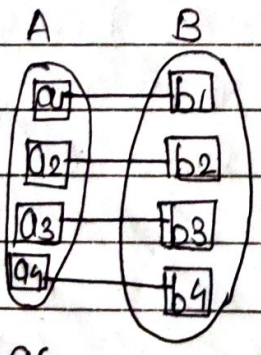
- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - Procedural - user specifies what data is required and how to get those data
  - Non procedural - user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

DATE

- Composite Attributes

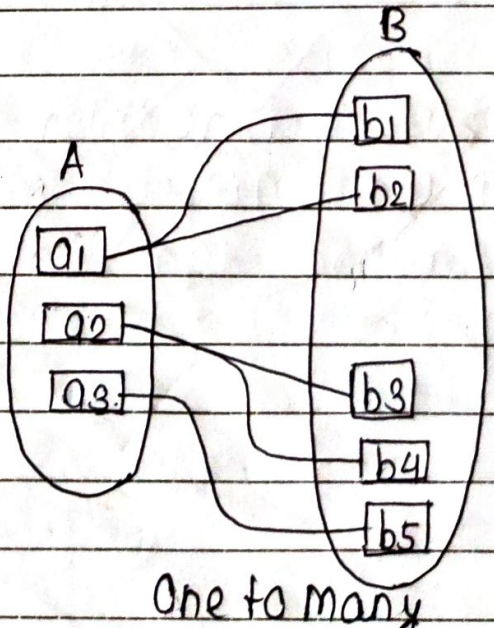


* Mapping Cardinalities.



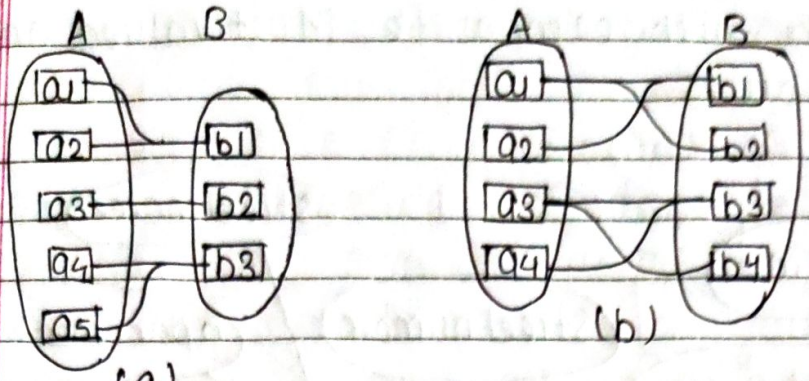
Mar

One to One



One to many

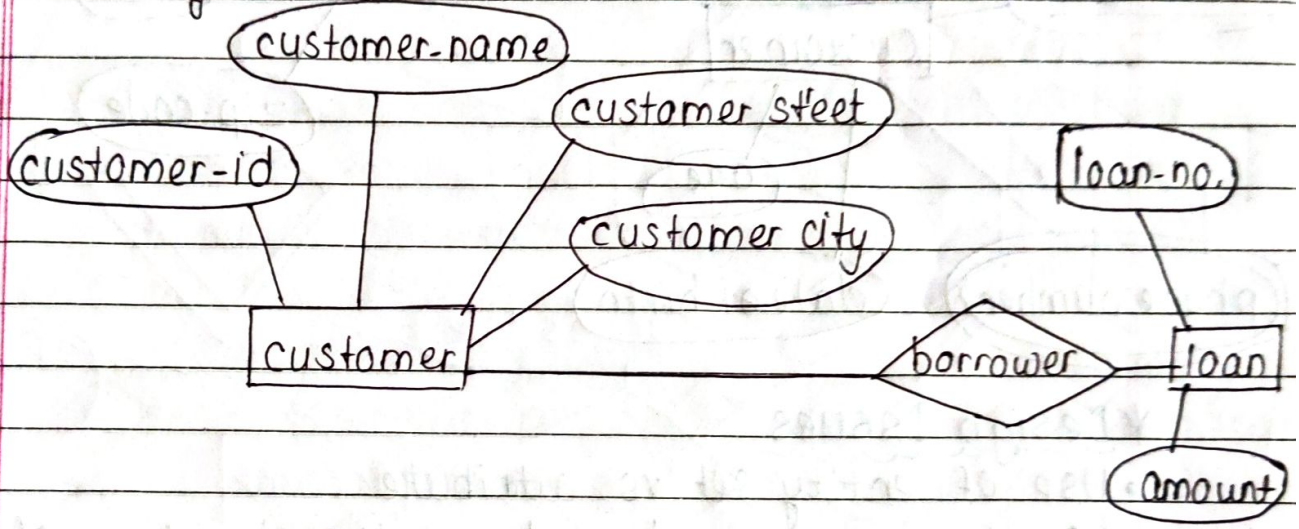
Note:- Some element in A and B may not be mapped to any elements in the other set



(a) (b)

Many to one

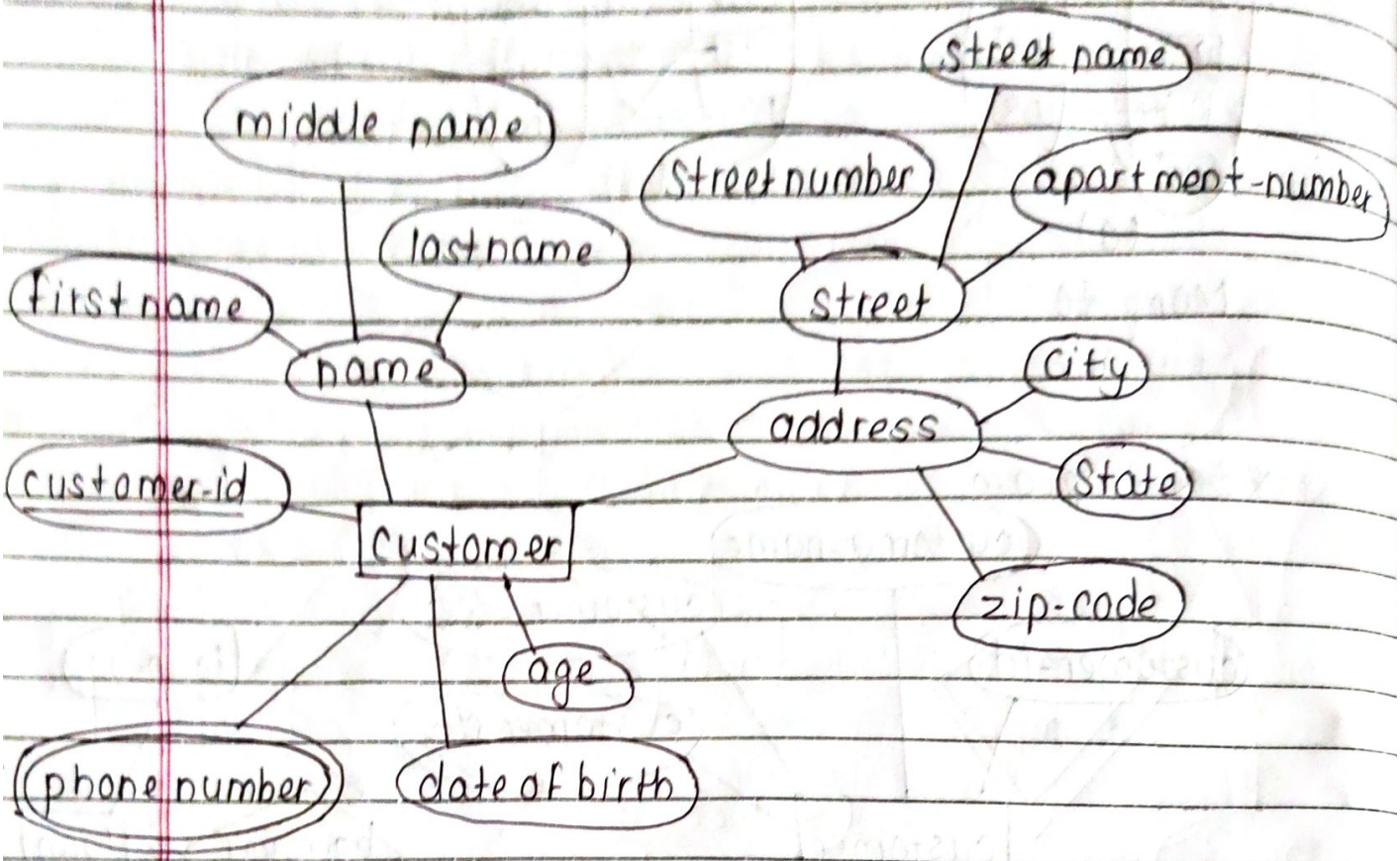
* E-R Diagram



- Rectangles represent entity sets
- Diamonds represent relationship sets
- lines link attributes to entity sets and entity set to relationship sets
- Ellipses represent attributes
  - Double ellipses represent multivalued attribute
  - Dashed ellipses donate attributes
- Underline indicate primary key attributes (will study later)

PAGE NO.
DATE / /

* E-R Diagram, with composite, Multivalued and Derived Attributes.



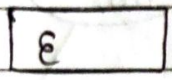
* Design Issues

- Use of entity set vs. attribute
  - Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question

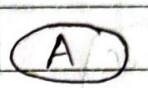
- Use of entity set vs. relationship sets
  - Possible guideline is to designate a relationship set to describe an action that occurs between entities.

- Binary versus n-ary relationship sets
- Although it is possible to replace any nonbinary (n-ary, for n>2) relationship set by a no. of distinct binary relationship sets, a n-ary relationship set shows more clearly that several entities participate in a single relationship
- Placement of relationship attributes.

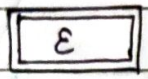
* Summary of Symbol used in E-R Notation



Entity Set



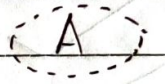
Attribute



Weak Entity Set

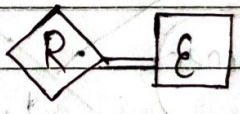
Multivalued Attribute

Relationship Set

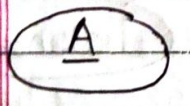


Derived attribute

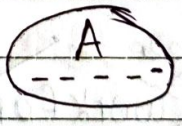
Identifying Relationship Set for Weak Entity Set



Total Participat-ion of Entity Set in Relations-hip



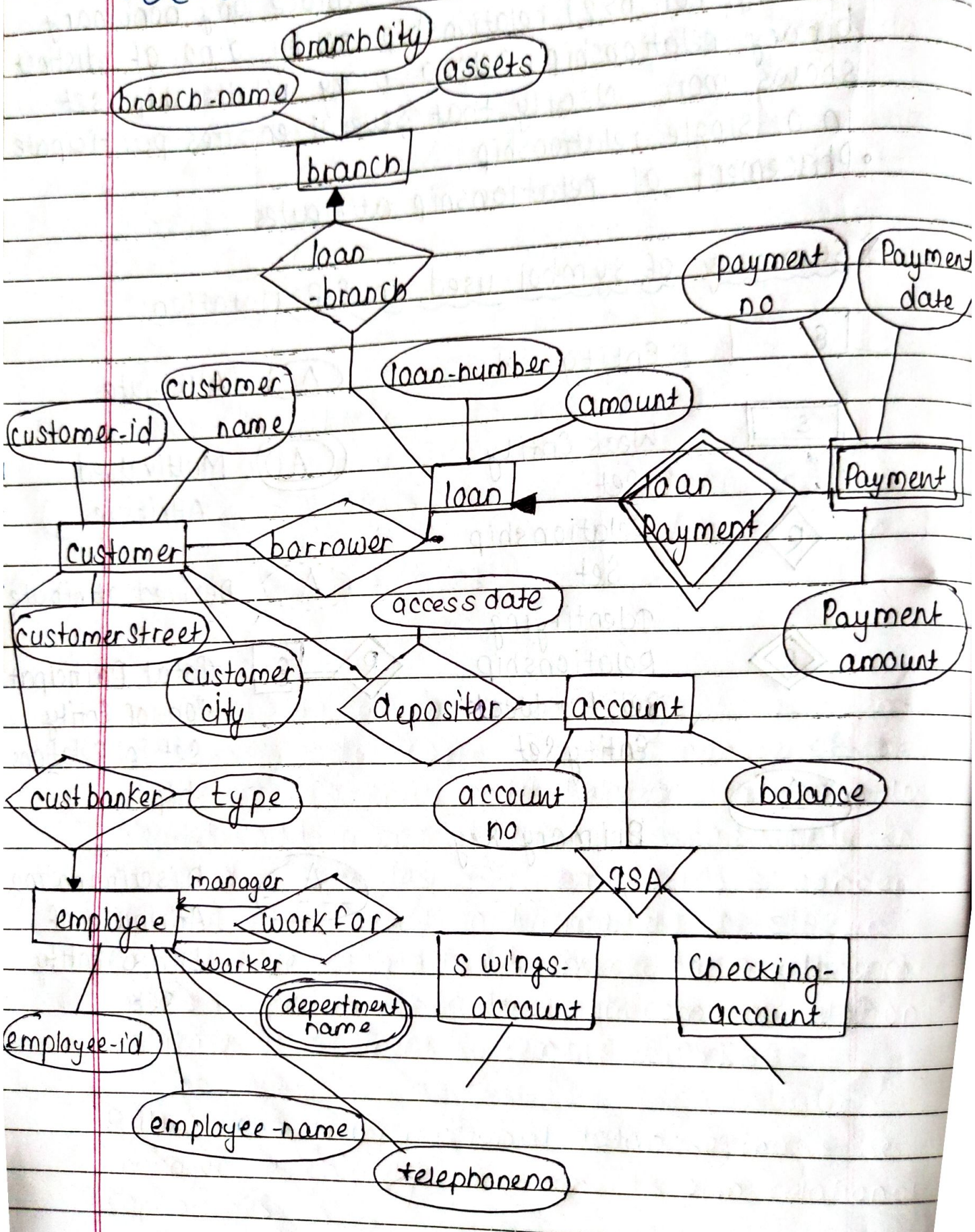
Primary Key

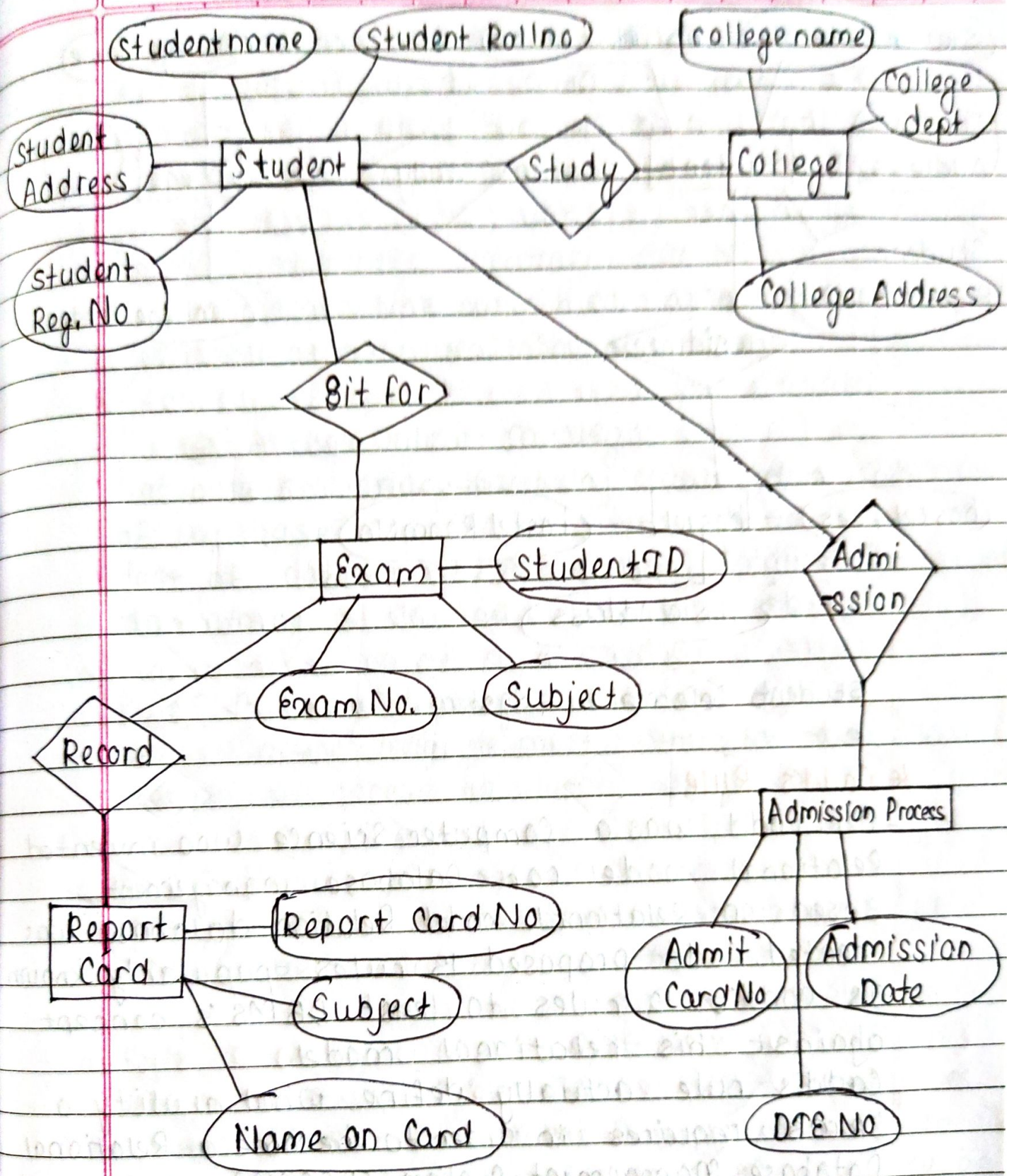


Discriminating Attribute of Weak Entity Set

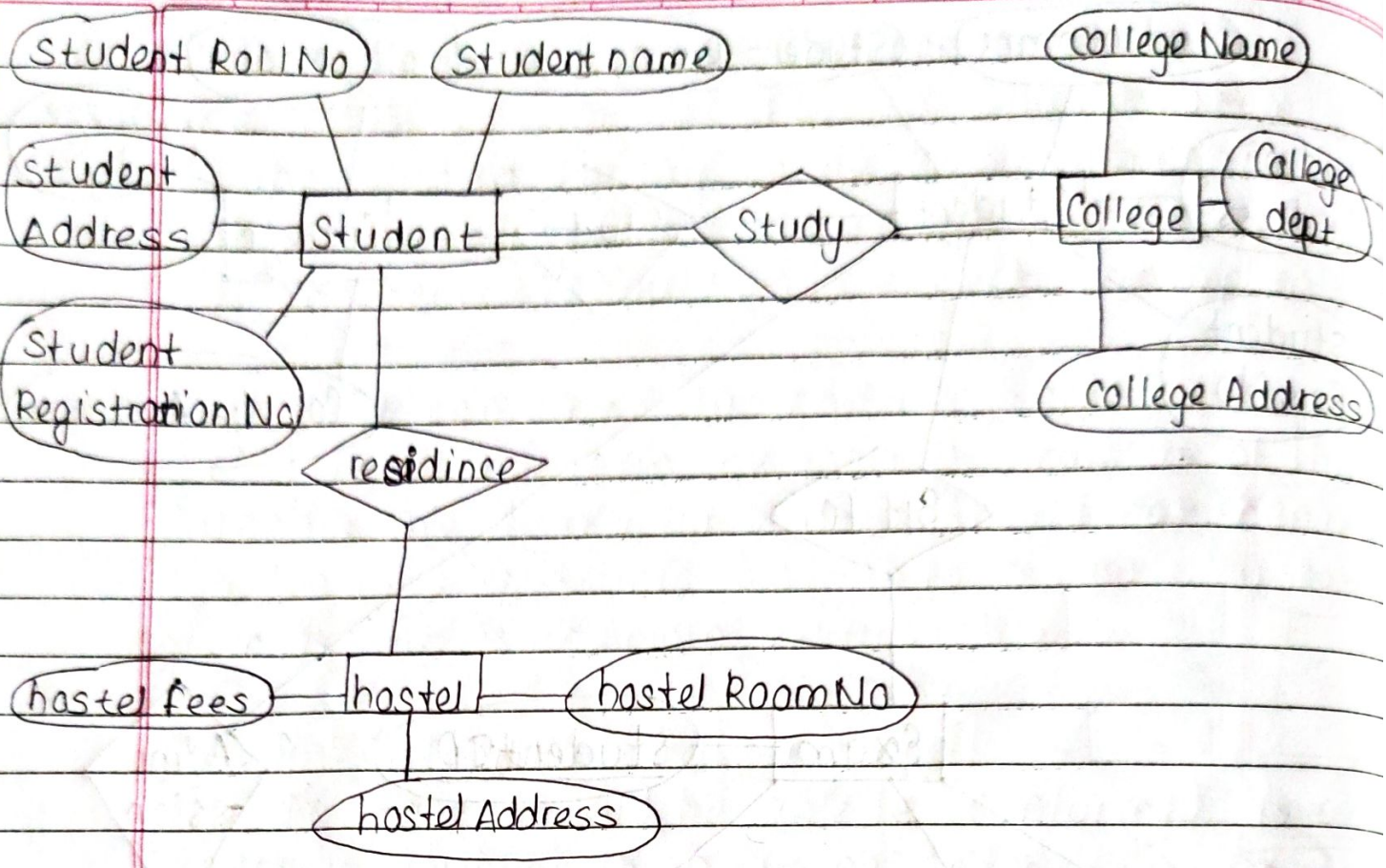
PAGE No.
DATE / /

* E-R Diagram for a Banking Enterprise





E-R Diagram For Student Information



Student Information System

* Codd's Rule

- E.F. Codd was a Computer Science who invented Relational model for Database management. Based on Relational model, Relation database was created. Codd proposed 13 rules popularly known as Codd's 12 rules to least DBMS's concept against his relational model
Codd's rule actually define what quality a DBMS requires in order to become a Relational Database Management System (RDBMS)

Rule 0:- Any truly relational database must be manageable entirely through its own relational capabilities

Rule 1: The Information Rule
All information in a relational database is represented explicitly at the logical level and, in exactly one way - by values in table.

Rule 2: Guaranteed Access
Each unique pair of data (automatic value) should be assessible by: Table Name + primary key (ROW) + Attribute (column)

Rule 3: Systematic treatment of NULL
NULL has several meaning, it can mean missing data, not applicable or no value. It should be handled consistently

Rule 4: Active Online Catalog
Database dictionary (catalog) must have description of database. Catlog to be governed by same rule as rest of database. The same query language to be used on catalog as an application database

Rule 5: Powerful Language
One well defined language must be there to provide all manner of access to data

Rule 6: View Updating rule
All view that are theoriatically updatable should be updatable by the system

Rule 7: Relational Level Operation
There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

PAGE No.
DATE / /

Rule 8: Physical Data Independence
The physical storage of data should not matter to the system. It say some file supporting table were ranamed or moved from one disk to another. it should not effect the application

Rule 9: Logical Data Independence
If there is change in the logical structure of the database the user view of data should not change. Say, if a table split into two tables, a new view should given result as the join of two table.

Rule 10: Integrity Independence
The database should be able to conformed its own integrity rather than using order program. This is also make RDBMS independent of front-end

Rule 11:- Distribution Independence
A database should work properly regardless of its distribution accross a network

Rule 12: No subversion rule
If low level access is avowed to a system it should not be able to subvert or bypass integrity rule to change data. This can be achieved by some sort of looking or encryption

- Difference Betn DBMS and RDBMS

DBMS	RDBMS
- DBMS data is stored in form of rows and column	- RDBMS data stored in form of table
- Data stored in DBMS is temporarily.	- where as in RDBMS is permanently
- DBMS is Database manage -ment System	- RDBMS is relation datab -se management System
- In DBMS keys are not used	- In RDBMS keys are used
- In DBMS duplication of rows and column	- But in RDBMS there is no duplication of rows and columns
- In DBMS rows and colu -mn are independent	- In RDBMS row and column are dependent
- DBMS is for single user only	- RDBMS is for multi user
- DBMS does not satisfies Codd's rule	- RDBMS satisfies codd's rule
- E.g. DBMS: Dbase, Sysbase, Foxpro	- E.g. RDBMS, Oracle, MSSQL, server, MySQL

PAGE No.
DATE / /

* What is a Record?

A single entry in a table is called a Record or Row. A Record in a table represents set of related data. For Example, the above Employee table has 4 record Following is an example of single record

ID	Name	Age	salary
1	Anurag	34	13000
2	Aditi	28	15000
3	Sumit	20	18000
4	Shravani	42	20000

1	Anurag	34	13000

* What is Table?

In Relational database, a table is a collection of data elements organized in terms of rows and colums. A table is also considered as convenient representation of relations. But a table can have duplicate tuples while a true relation cannot have duplicate tuples. Table is the most simplest form of data storage. Below i's an example of Employee table

ID	Name	Age	Salary
1	Anurag	34	13000
2	Aditi	28	15000
3	Sumit	20	18000
4	Shravani	42	20000

* What is Field?

A table consist of several record (row), each record can be broken into several smaller entities known as field The above employee table consist of four field ID, Name, Age and Salary

ID	Name	Age	Salary
1	Anurag	24	10,000
2	Aditi	34	20,000
3	Sumit	21	21,000
4	Shravani	30	25,000

ID	Name	age	Salary

* Database keys

keys are very important part of Relational database They are used to establish and identify relation betn table. They also ensure that each record within a table can be uniquely identified by combination of one or more fields within a table.

Super key
Composite key
Non-key attribute
Foreign key
Compound key
Artificial key
Candidate key
Primary key
Secondary or Alternative
Non-prime attribute

Simple key

PAGE No.
DATE / /

* Super key

Super key is defined as a set of attributes within a table that uniquely identifies each record within a table. key is a superset of Candidate key

Employee	
Employee-name	→ Primary key
Employee-DOB	→ Candidate key
EMP-Address	
DEPARTMENT NAME	

Super key

* Candidate key

Candiate keys are defined as the set of field from which primary key can be selected. It is an attribute or set of attribute that can act as a primary key for a table to uniquely identify each record in that table

candidate primary key

Student Id	Fname	Lname	Course id
109	Anurag	kule	PC01
21	Shiv	Kale	PC02
32	Shrivani	Patil	AC02
44	Mrunal	Mali	PC21

* Primary key

Primary key

Student Id	Fname	Lname	Courseld
109	Anurag	kwe	PC01
21	Aditi	Patil	M210
33	Apurva	mali	MC21
35	Amit	Kale	PC21
100	Sumit	kapse	PC21

primary key is a candidate key that is most appropriate to become main key of the table. It is a key that uniquely identify each record in a table. The primary key are

* Candidate key which is not selected to be Primary key is called as an alternate key.

compulsory in every table.

- The properties of a primary key are:
- Model Stability
- Occurrence of minimum fields
- Defining value for every record i.e. being definitive
- Feature of accessibility

* Foreign key

Student Id	First name of Student	Last name of Student	Course Id
12342	Anurag	kule	102
12343	Aditi	Mali	202
12350	Priti	kale	312
12362	Anand	Patil	521

Relationship ↑

Course Id	Course Name
102	Chemistry
202	Physics
312	Math
521	Biology

→ Primary Key

In the context of relational database, a Foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. In simpler words, the foreign is defined in a second table, but it refers to the primary key in the first table.