UNIT-1.

CHP-1 = INTRODUCTION TO DATABASES AND TRANSACTIONS.

* What is database system.

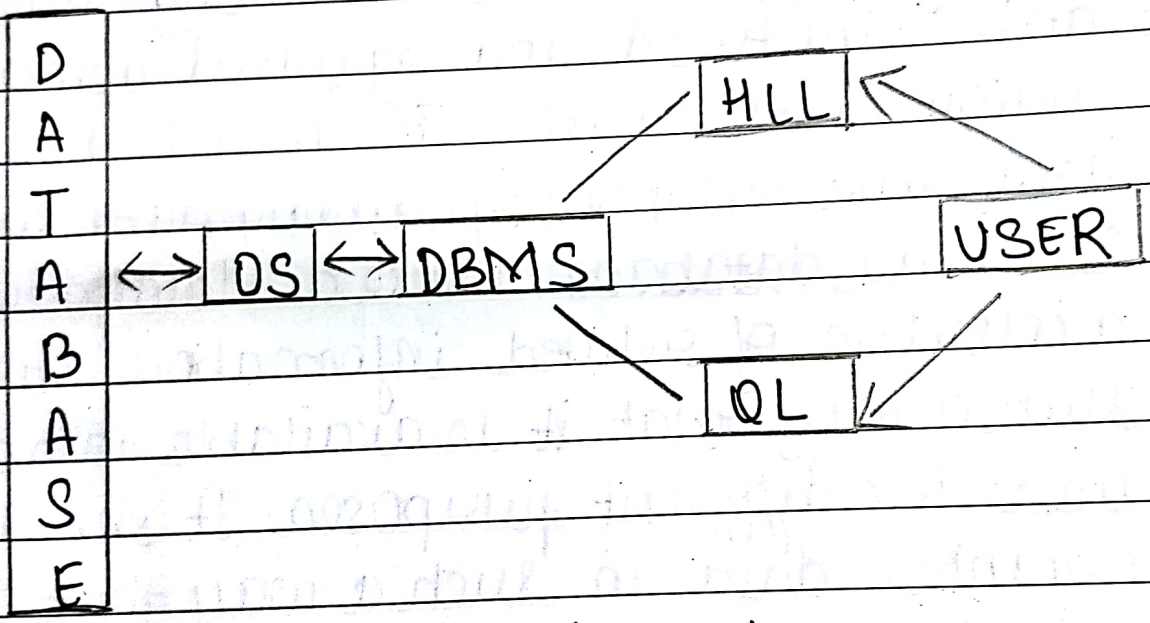
An organisation needs accurate and reliable data for effective decision making. This data can be structured and organised using various data models. This model shows the data and relationship among data and are called as database. Hence a database is a collection of related information stored in such a way that it is available to many users for different purposes. It should organise data in such a way that it eliminates or minizes redundant data.

A computer database gives us electronic filing system which allows the data to be referenced in multiple ways by multiple users. Hence database is a collection of related and relivant information which is organised in a way so that all data accessible. Some of the common data models are relational network etc.

Expt. No.:

A database management system is a combination of hardware, software (OS and software to handle database) that can monitor a database and manage updating and retrival of database which is stored in it.

★ Architecture of database.



HLL - high level language.
QL - query language.

* Purpose of database.

(1) Creating a file for addition, delition and modification of data.

(2) Retriving data collectively or selectively

(3) The data can be stored ~~as~~ or stored as indexed as per user requirements.

(4) Various reports can be produced from the system as per user requirement.

(5) It provides mathematical and other functions for manipulating data.

(6) It maintains data integrity

(7) The data redundancy must be eliminated or atleast reduced.

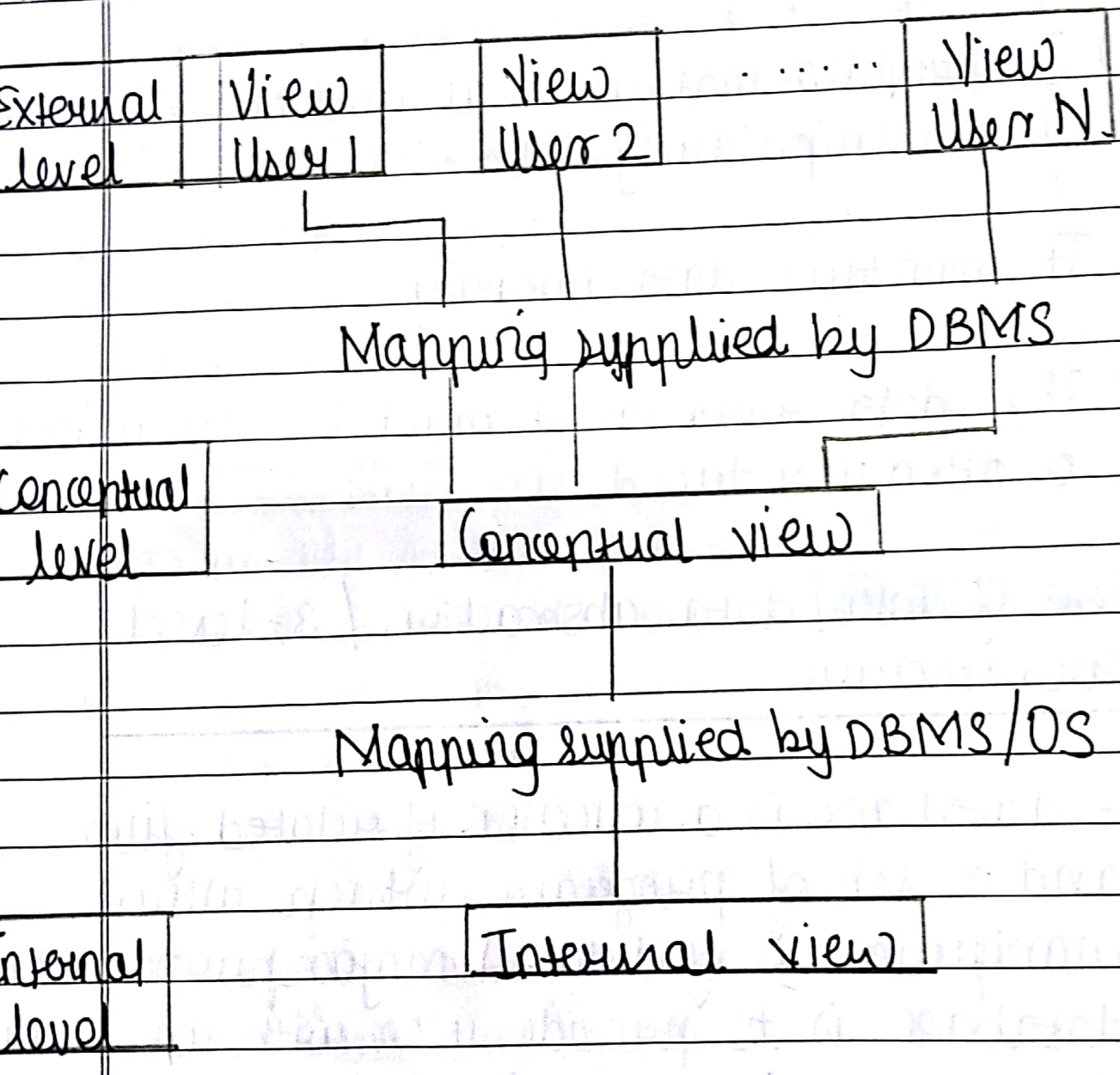
* View of data / data abstraction / 3-level architecture.

A database is a collection of related files and a set of program which allow manipulation of data. A major purpose of database is to provide user with an abstract view of data i.e certain details are hidden. This allows simplicity of data and prevents unnecessary complications. A database has 3-level or views they are :

Expt. No.:

- internal or physical view.
- Conceptual or logical view.
- External view.

★ 3-level architecture of DBMS.



- Internal view | Physical view :-

A this level the data is stored as a stream of bit and bytes or as raw bit.

Every database has to have only one internal view it is the lowest level of abstraction and is close to physical storage device. It indicates how the data will be stored and describes the data structure and excess methods. The internal view is expressed by the internal schema which contains the definition of records the method of representing data fields and excess aids used. A schema is an outline or plan that describes records and relationship existing in the view. The word scheme means a systematic plan for attaining a goal, is used interchangably in database with the word schema.

- Conceptual / logical / Global view :-

At this level abstraction of object concerned to the database are described. This view is a subset of a object model in the conceptual database that is used by an application. All the three views have to be mapped i.e there has to be correspondance among records at all three levels. i.e at this level all the

Expt. No.:

database entities and the relationship among them are defined. Every database has one conceptual schema. The description of data at this level is independant from physical representation it supports features such as data consistency and integrity

- External/User view :-

This is the highest level of data abstraction where selected and concerned data is accessed. Depending on number of users a database can generate one or more use. Each view is described by external schema. This level is defined by user or application where as conceptual or physical level are defined by DBA (database administrator)

Eg: (1) Logical view will think of data as follows:

Customer name datatype string
Customer address datatype string
Customer salary datatype integer.

(2) Physical view will think as follows:

Customer name string length 25
Customer address string length 51
Customer salary 5 digits.

(3) External view will think as follows:

Customer name
Customer address
Customer salary.

* Database architecture.

Describe the architecture of DBMS and 3 level architecture of database

* Relational database.

A relational database describes all it data in the form of tables. Tables are called relation in calculus. This database is based on a paper published by doctor Codd. In this paper he described 12 codd rules. For a database to be qualified as RDBMS atleast 6 codd rules must be.

Expt. No.:

implemented including the data definition.

The tables should structure the data as follows.

(1) In a given column of table all items must be of same kind

(2) Each item should be a simple number or character.

(3) All rows of a table must be distinct.

(4) Ordering of rows is immaterial

(5) The columns of a table are assigned distinct names and the ordering is immaterial.

(6) A table having n columns is said to have n degree.

- Advantages of RDBMS.

(1) Ease of use.

(2) flexibility - different ~~levels~~ tables can be generated at view level.

(3) precision - data can be stored without ambiguity.

(4) Security - controls accessibility.

(5) data independence - it shows independence between levels.

(6) Data manipulation language - It allows the user to modify various data.

* Transaction management.

A transaction in a database is a set of instructions which are executed to update data items. Each data item is read and written once by the transaction to update data item. No transaction should violate any database constraints. Consider a following banking example.

Let 2 transaction occur as follows.

Expt. No.:

Balance = 600

constant (min bal = 500)

T_1	T_2
read bal	read bal.
write (withdraw). (bal - 50)	write (withdraw). (bal - 100)
commit	commit.

Above is the example of database inconsistancy. It can be overcomed by allocating the write lock to only one transaction to avoid inconsistancy. A transaction has following states:

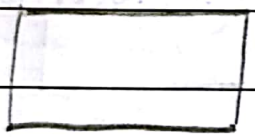
1) Active
2) Wait / Normal
3) Committed
4) Roll back / Canceled

NOTE A committed transaction cannot be rolled back.

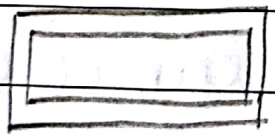
CHP-2 = DATABASE DESIGN, ER DIAGRAM AND UNIFIED MODELING LANGUAGE.

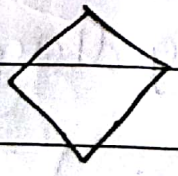
* Database design, ~~ER~~ and ER Model : overview.

(mobile)
ER diagram stands for 'entity relationship' diagram. It is a graphical representation showing relationship among data items / entities. The following symbols are used in ER diagram.

(1)  - Entity

→ It represents the entity set. Alternatively an entity set is a weak entity which depends on another entity.

 - weak entity.

(2)  - Relationship  -

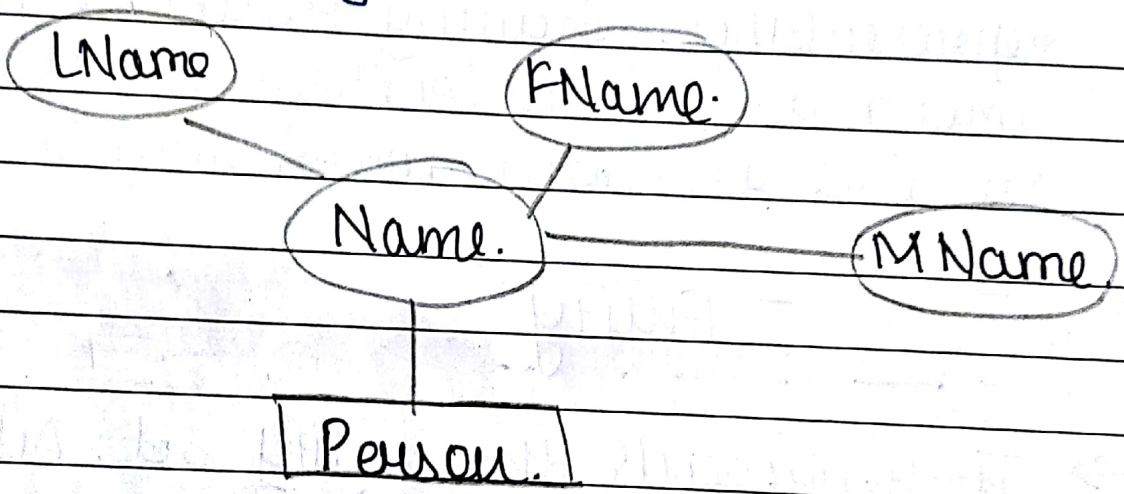
→ It shows relationship between 2 entities it can also be of type identifying

Expt. No.: Date :

relationship.

(3) —( )— - attribute.

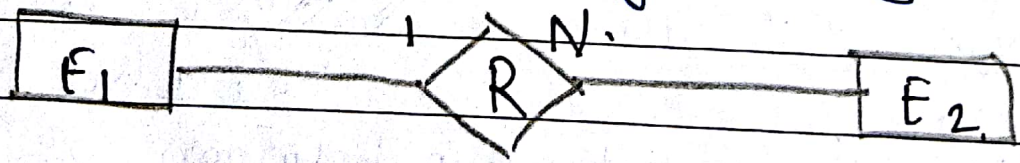
→ A key attribute is underlined which uniquely identifies an entity. It can be multivalued eg - Phone no. It can be a composite attribute. eg -



(4) —(- -)— - derived attribute.

→ Alternatively an attribute can be a derived attribute.

ER diagram has following diagram

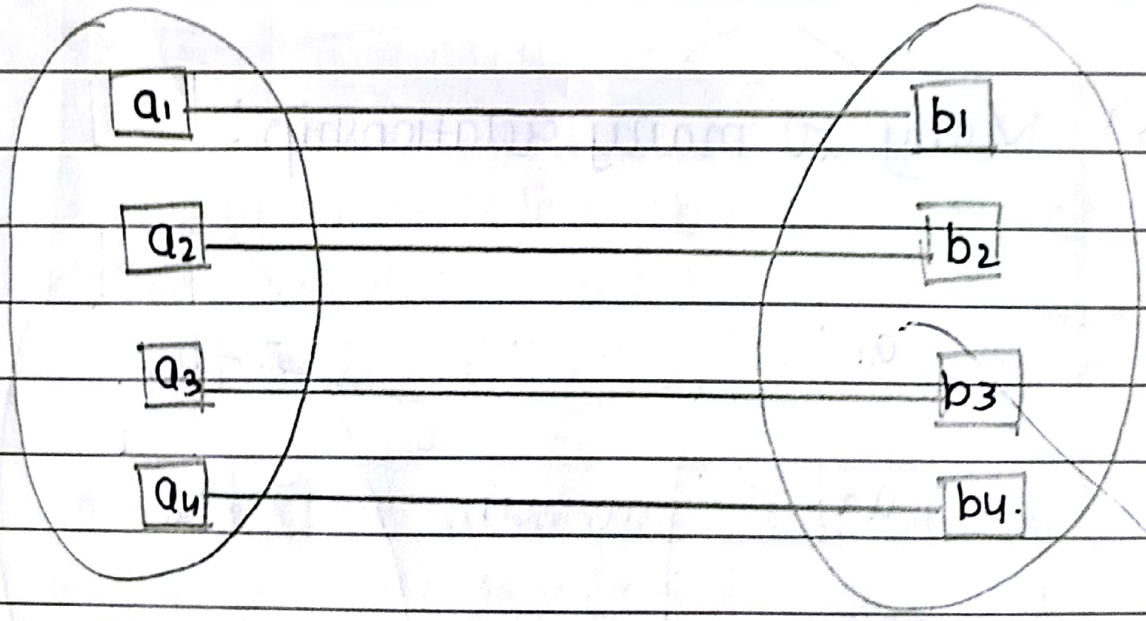


The links can be used to show constraints cardinality and participation. Total participation means common fields.
Cardinality of following types :.

(1) 1 to many
(2) many to 1
(3) many to many
(4) 1 to 1

The cardinality is also called mapping constraints

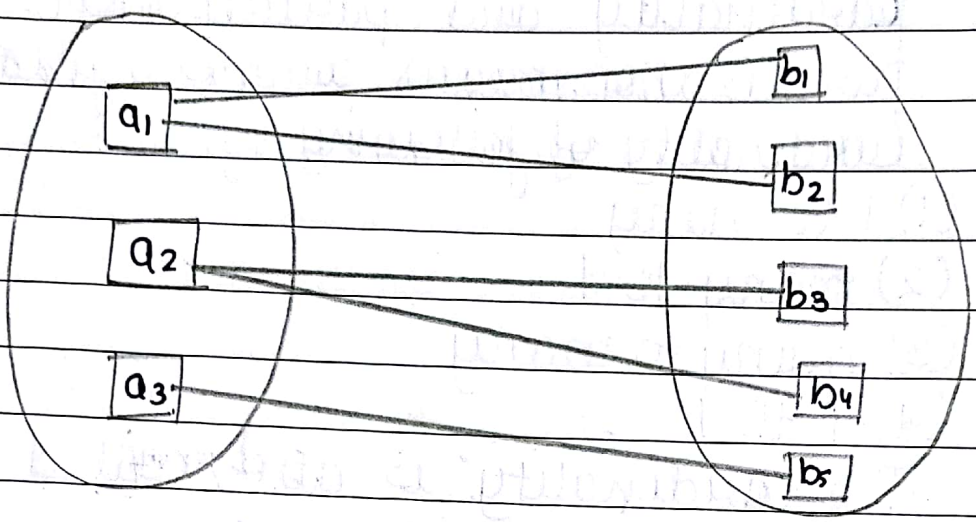
(1) One to one relationship :



Two attributes of entity has only one relationship. This is where one occurrence of an entity relates to only one occurrence in another entity. A one to one relationship rarely exist in practise.

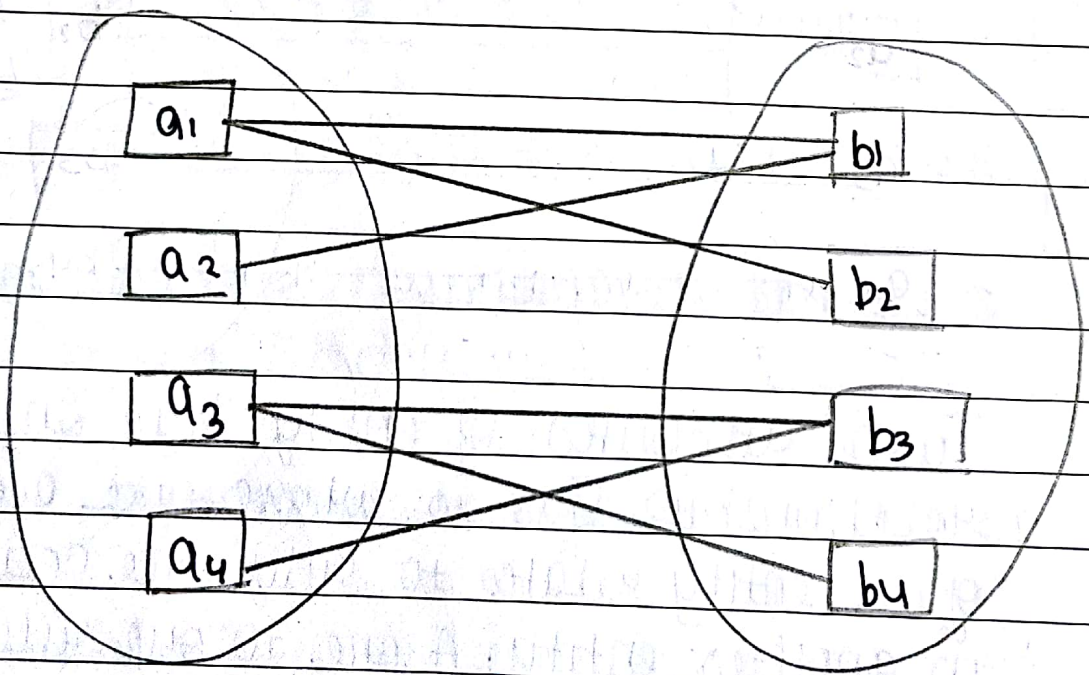
Expt. No.: Date :

(2) One to many relationship :



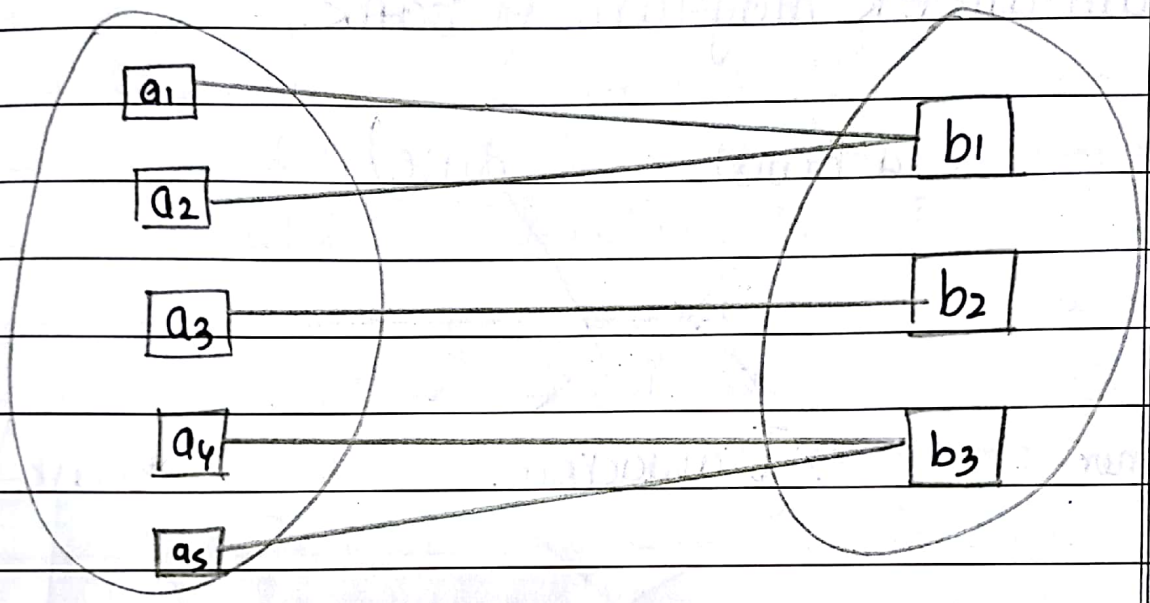
It is where one occurrence in an entity relates to many occurrences in another entity.

(3) Many to many relationship :



This is where many occurences in an entity relate to many occurences in another entity. Normally they occur because an entity has been missed.

(4) Many to one relationship:



A many to one relationship is where one entity contains value that refer to another entity that has unique values.

An entity is an object that exist and is distinguishable from other objects.

Entity set - they are entities of same type.

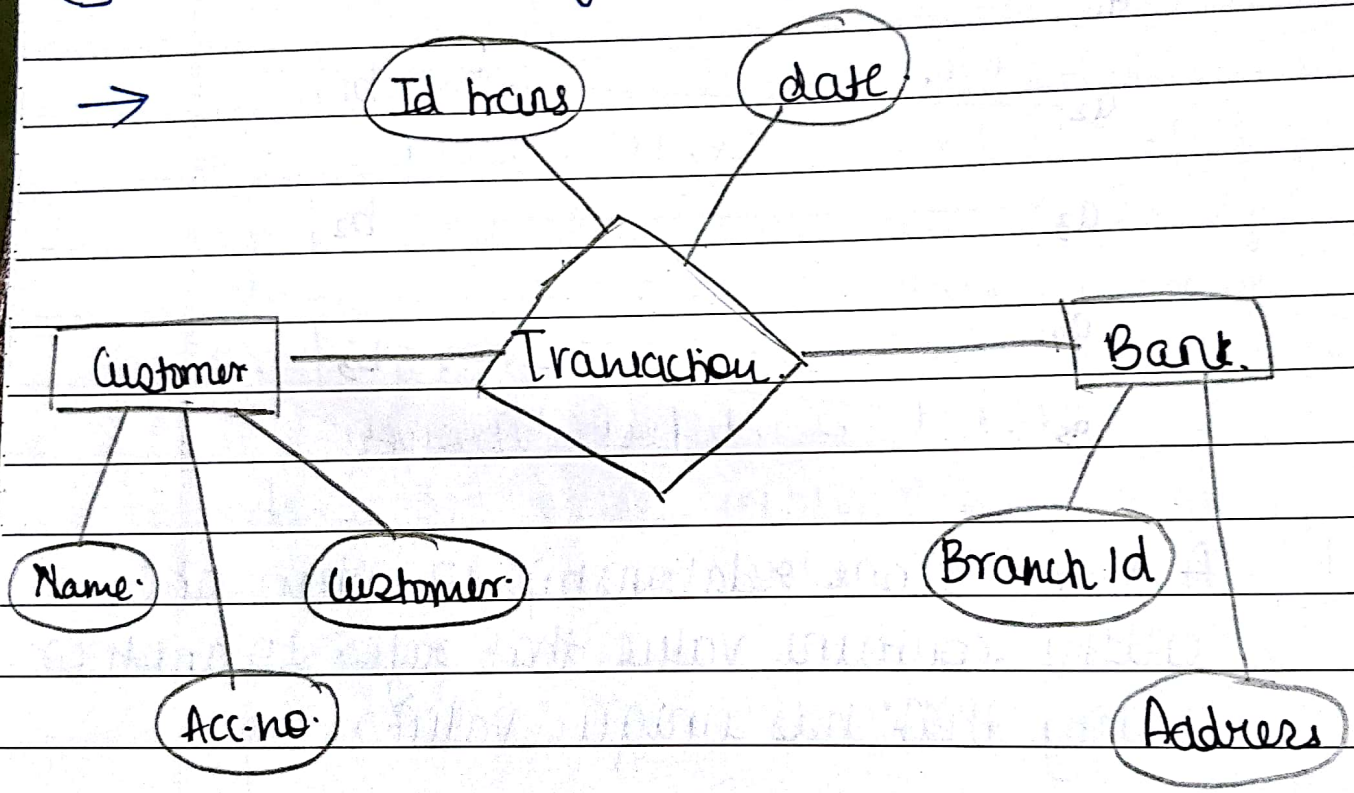
Attribute - They are set of properties that an entity has eg - name, etc

Domain - These are set of permitted values for an attribute.

A relationship is association among several entities. A relationship set is a set of relationships of the same type.

[Q] Draw an ER diagram of bank.

→



* ER Model / ER diagrams / ERD issues

ER Model is based on ER diagram. It is used for database design. It shows the relationship among entities. It offers following advantages:-

(1) It simplifies representation of data entities and relationship

(2) It shows aggregation, generalisation and specialisation.

It offers following disadvantages :-

(1) Details are difficult to show and ER diagrams can be very large.

(2) It is difficult to draw ER diagrams using computers i.e automation of ER diagram is difficult to achieve.

* **Weak entity sets**

let A be the weak entity set

$\{a_1, a_2, a_3 \cdots a_r\}$

let B be the strong entity set on which A is dependent, where the primary key of B consisting of attributes

$\{b_1, b_2, b_3 \ldots b_s\}$

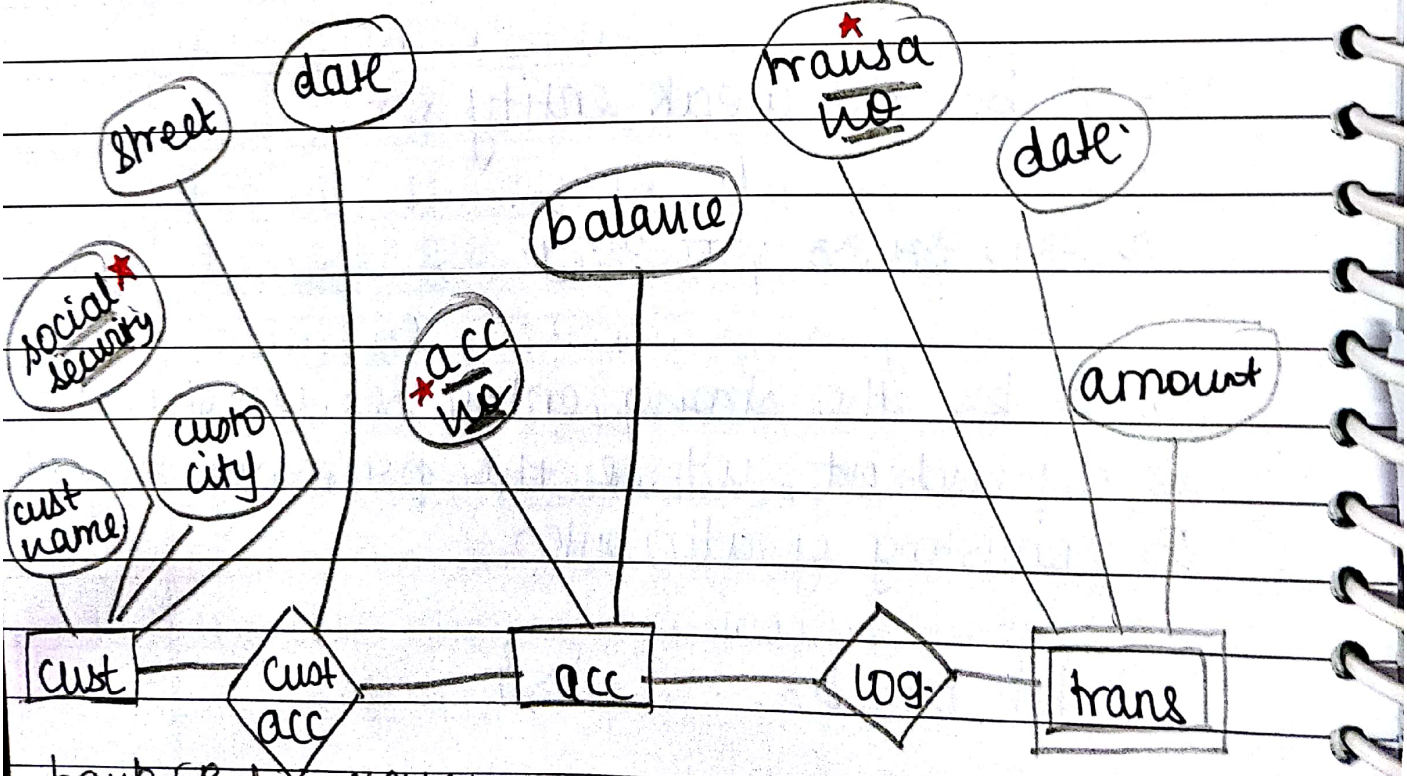
The entity set A is represented by a table called A with one column for each attribute set.

$\{a_1, a_2 \ldots a_r\} \cup \{b_1, b_2 \ldots b_3\}$

A weak entity set can be shown by the ER diagram of a bank. The entity set transaction has 3 attributes :-

- transaction number
- date
- amount

The primary key of the entity account is account number. This makes transaction a weak entity set.



bank ER diagram.

NOTE: A primary key is the attribute which uniquely identifies a record such attributes are given red star in ER diagram. In the table such primary key will have unique value i.e no two records can have common primary key.

* constraints

Database is governed by Codd's rules. The constraint are limitations or conditions imposed on database. For example a constraint can be :-

(1) 2 entities may have mapping cardinality of one to many, many to one, one to one and many to many

(2) A primary key may be composite i.e it may be made up of one or more fields/attributes. Such a primary key should not have attributes which depends on attributes that does not belong to primary key.

(3) Two other constraints are integrity rules ~~(a) enti~~

(a) entity rule - If an attribute A is a prime key then it cannot be null

(b) referencial integrity - A foreign key cannot be deleted if it is a primary key of another table.

* Codd's Rule:

The Relational database management system or RDBMS is based on a paper published by Dr. Codd. Atleast 3 or 6 rules must be obeyed for a DBMS system to be called a RDBMS. (According to some standards 6 rules must be observed). The summary of all 12 rules is :

A truly relational database must be manageable entirely through its own relational capabilities.

The 12 rules are as follows :

(1) Rule 1 = The information rule.
→ All information is explicitly and

logically represented in exactly one way as data values in table. It means data as well as metadata / data dictionary must also be represented as table

(2) Rule 2 = The rule of guarantee access.
→ Every item of data must be logically addressable by resorting to a combination of table name primary key and column name. This can be achieved by application program or specialised languages such as SQL (structured query language). This rule means that intersection of row and column will be a data item having a value or it may be null

(3) Rule 3 = The systematic treatment of null values.
→ It means that if a value does not exist for a data item it must be null It means that a character field or integer value not having a value must have same null values.

NOTE: Primary key cannot be null.
eg: Gender
This can have values M or F.
Alternatively 'X' can be used for data not available and 'Y' indicating data not applicable. The value outside these four values is null value.

(4) Rule 4 = Database description rule.
→ It means that every database must have a data dictionary or metadata. with the RDBMS that is constructed of tables AND/OR views. It means that data dictionary is mandatory.

(5) Rule 5 = The comprehensive sub language rule:
→ It means that a database must have atleast one language which supports the following features:
(a) data definition
(b) view definition
(c) data manipulation
(d) integrity constraints
(e) authorization.
(f) transaction boundaries

(6) Rule 6 = The view updating rule.
→ All view that can be updated in theory can also be updated by the system.

(7) Rule 7 = The insert and update rule.
→ A RDBMS must allow a new data to be inserted or updated without affecting or deleting data.

(8) Rule 8 = The physical independence rule.
→ It means that whenever changes are made to storage, the data must still be accessible without changing the software
⇔ i.e applications must be limited to interfacing with logical layer without affecting the ~~upper layers~~ lower layers.

(9) Rule 9 = The logical data independence rule
→ Application program must be independent of changes made to the tables. It also means that the base tables can be combined as non loss. join

(10) Rule 10 = Integrity rule.
→ There are 2 integrity rules : Integrity rule 1 (entity integrity) - If the attribute A of relation R is a prime attribute of R, then A cannot accept null values. Prime key cannot be null. Integrity rule 2 (Referencial integrity) - It is concerned with foreign keys i.e Attributes of a relation having domain that are primary key of another relation.

(11) Rule 11 = Distribution Rule.
→ A RDBMS must have distribution independence This is one of the more attractive aspects of RDBMS Database system built on the relational framework are well suited to today's client server database design.

(12) Rule 12 = No Subversion Rule.
→ If an RDBMS supports a lower level language that permits for example row-at-a-time processing, then this language must not be able to bypass and integrity rule or constraints.

* Relational Schemas.

A schema is singular for scheme. It is used to define a database table along with the fields. It represents structure of relation i.e a relation comprising of all it attributes. Upper case letters are used to denote schema and lower case is used to denote the attributes

eg:- • Structure of schema.

Table name (field names)

eg:- * Student (roll no, name, address)

Fees (class, amount)

Result (course, class, roll no, marks)

In the above table, primary keys are underlined. The following is the example of the relation and its schema.

branch-name.	acc-no	cust-name	balance.
Downtown.	101	Johnson.	500
Mianus.	215	Smith.	700
Perryridge.	102	Hayes.	400
Round Hill	305	Turner.	350
Perryridge.	201	Williams	900
Redwood.	222	Lindsay.	700
Brighton.	217	Green.	750
Downtown.	105	Green	850

Deposit-scheme = (branch-name, account-no, customer-name, balance)

* Introduction to UML

UML - unified modeling language.

It is a standard language for specifying, visualising ~~and~~, constructing and documenting the artifacts of software system. It provides a conceptual model for designing a database. UML diagrams have 3 important features or building blocks they are:

- Things
- Relationship
- Diagrams.

- Things : Things are the most ~~building~~ important building blocks of UML. Things can be - Structural
  - Behavioral
  - Grouping
  - Annotational

(1) Structural things -

The structural things define the static part of the model. Following are the structural things :
* class, interface, collaboration, use case, component and node.

(2) Behavioral things -

A behavioral thing consist of the dynamic parts of UML. Following are the behavioral things : Interaction and state machine.

(3) Grouping things –

It can be defined as a mechanism to group elements of a UML model together. There is only one grouping thing available known as package.

(4) Annotational things –

This can be defined as a mechanism to capture remarks, description and comments of the UML model. Note is the only one annotational things available.

- Relationship : Relationship is another most important building block of UML. It shows how elements are associated with each other and this association describes the functionality of an application. There are 4 kinds of relationship available – Dependency
  – Association
  – Generalization
  – Realization.

- Diagrams : UML diagrams are the ultimate output of the entire discussion. All the elements, relationship are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process.