2. E.R. Model & Database

Page No:
Date: 25/9/19

* Entity Relationship Diagram / ER model

i) An entity relationship model describes the structure of database with the help of a diagram, which is known as a Entity Relationship Diagram (ER diagram).

ii) An ER model is a design or blueprint of a DB that can later be implemented as a DB. The main components of E-R model are entity set & relationship set.

iii) ER dia shows the relationship among entity sets. An entity set is a group of similar entities & these entities can have attribute, ER dia shows the complete logical structure of database

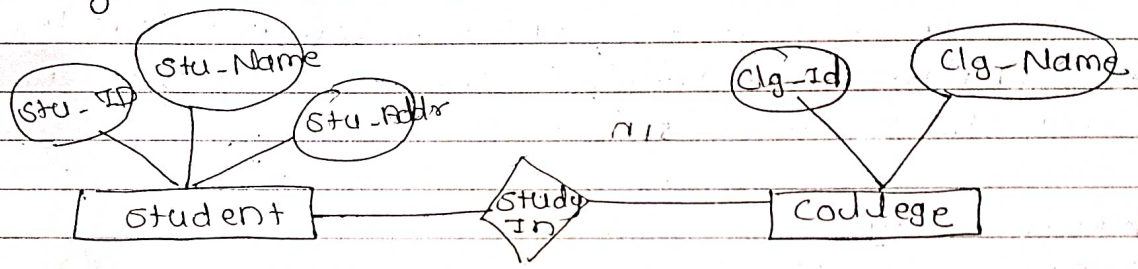
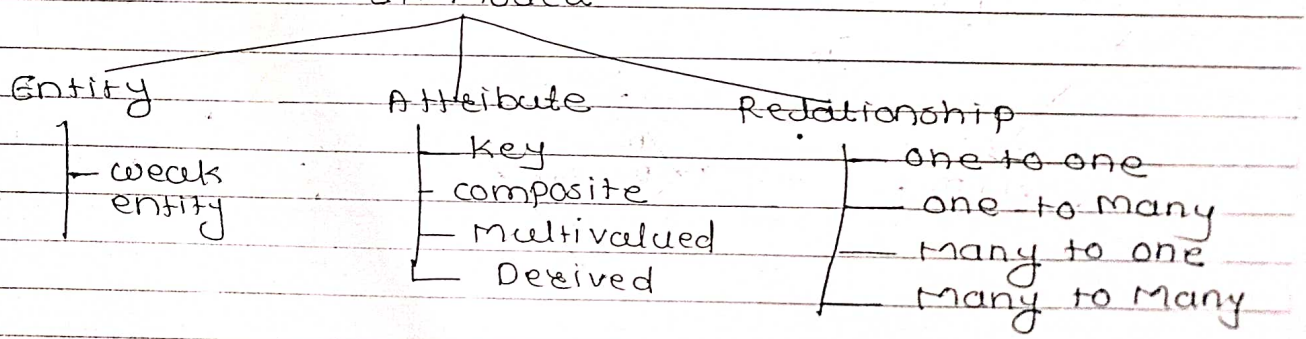


fig. Sample E-R diagram

iv) In above dia. we have two entities student & college & their relationship. Relationship between student & college is many to one as a college can have many students however student can't study in multiple colleges at the same time. Student entity has attribute std_id, stu-name & Stu-Addr & college entity has attribute such as Clg_id & Clg-Name.

Components of ER Diagram



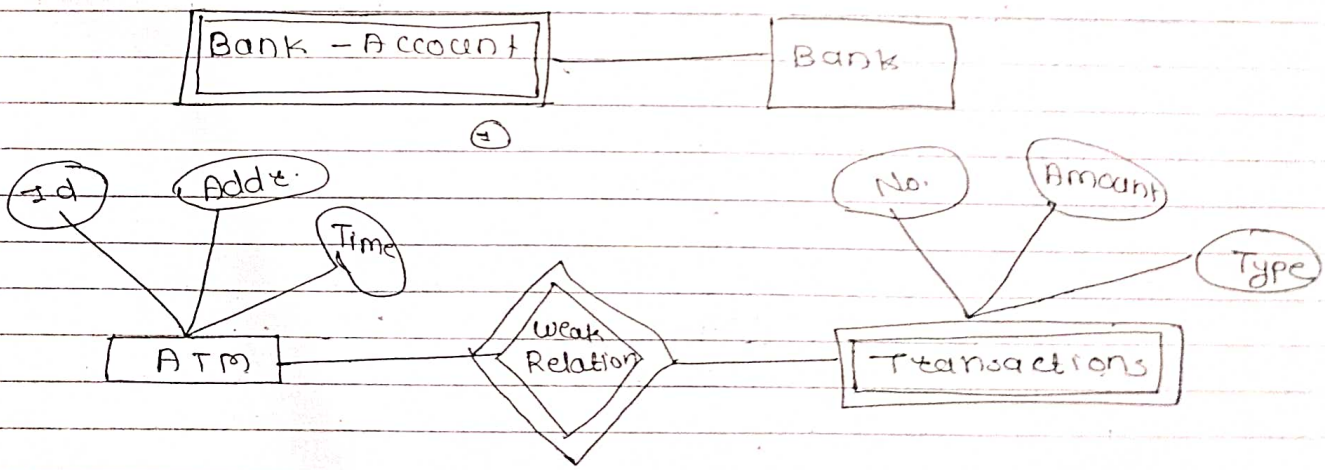
Sulekha

- ER diagram has three main components:
  1) Entity
  2) Attribute
  3) Relationship

1) Entity
i) An entity is an object or component of data.
ii) It is represented as rectangle in an ER diagram.

iii) In above fig. two entities student & college & the two entities have many to one relationship as many students study in single college.

Weak entity
i) An entity that cannot be uniquely identified by it's own relationship attributes & relies on the relationship with other entity is called an weak entity.
ii) The weak entity is represented by a double rectangle.
iii) e.g → a bank account cannot be uniquely identified without knowing the bank to which the account belongs so bank account is a weak entity.



Page No :
Date :

Strong Entity	Weak entity
1 Strong entity set always has a primary key	It does not have enough attributes to build a primary key.
2 It is represented by rectangle symbol	It is represented by double rectangle symbol.
3 It contains primary key represented by the underline symbol	It contains a partial key which is represented by dashed underline symbol.
4 The member of strong entity set is called as dominant entity set.	The member of weak entity set called as a subordinate entity set.
5 Primary key is one of its attributes which helps to identify it's member.	In a weak entity set, it is a combination of primary key & partial key of the strong entity set.
6 In the ER diagram the relationship between two strong entity set shown by using a diamond symbol.	The relationship between one strong & weak entity set shown by using double diamond symbol
7 The connecting line of the strong entity set with the relationship is single	The line connecting the weak entity set for identifying relationship is double.

Sulekha

2) Attribute

i) An attribute describes the property of an entity

ii) It is represented as oval in an E-R diagram

There are 4 types of attribute.

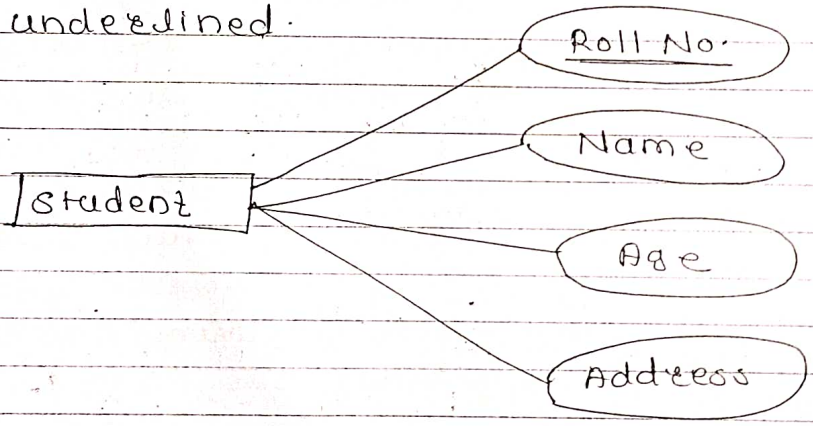
a) Key
b) composite
c) multivalued
d) Derived.

a) Key attribute

i) Key attribute can uniquely identify an entity from an entity set

ii) e.g, student roll no. can uniquely identify a student from a set of students.

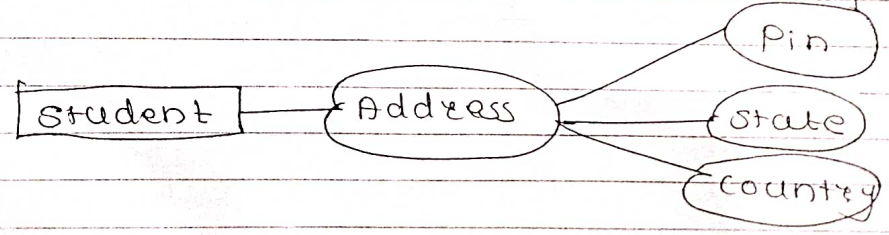
iii) Key attribute is represented by oval same as other attributes however the text of key attribute is underlined.



(2) Composite attributes

i) An attribute that is a combination of other attributes is known as composite attributes.

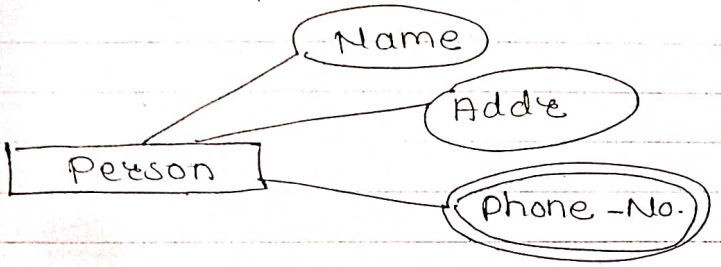
ii) e.g → In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country



Page No :
Date :

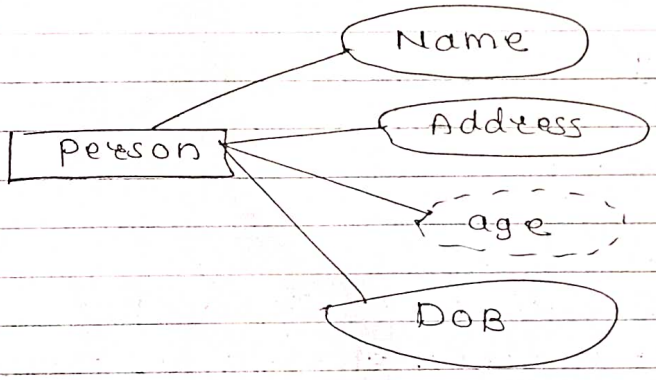
c) Multivalued attribute

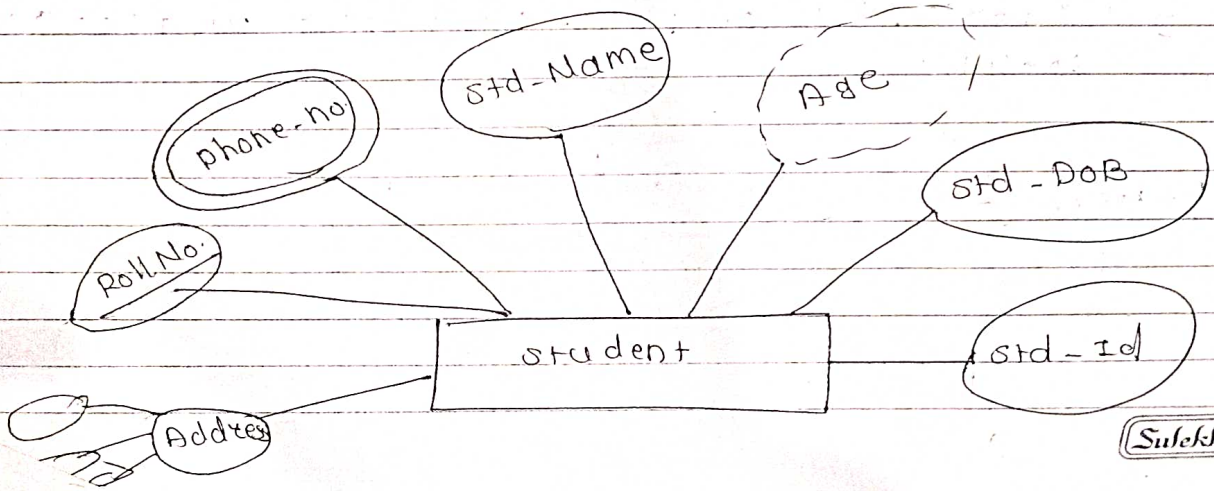
i) An attribute than can hold multiple values is known as multivalued attribute.
ii) It is represented with double ovals in ERD.
iii) e.g → A person can have more than one phone no. So the phone no. attribute is multivalued.



d) Derived attribute

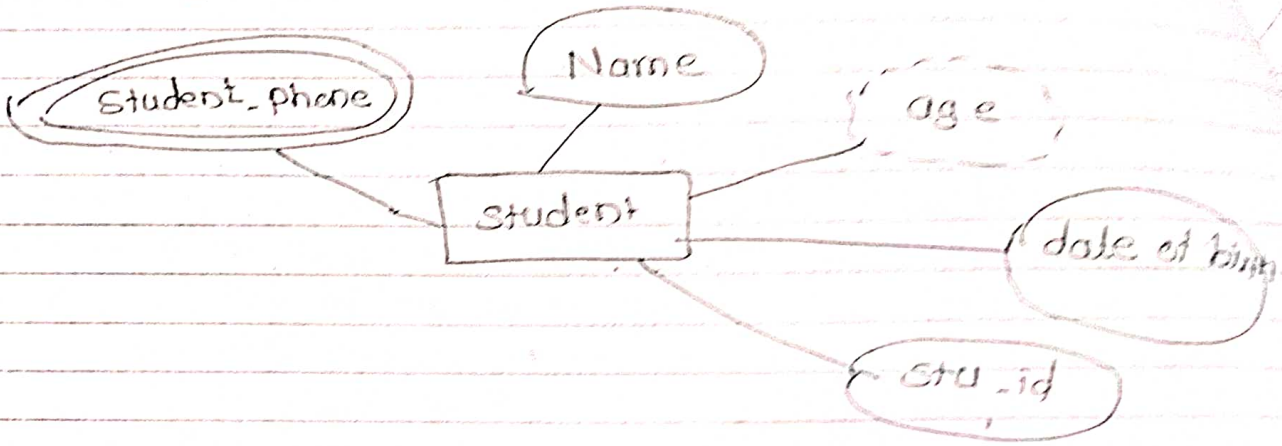
i) A derived attribute is one whose value is dynamic & derived from another attribute.
ii) It is represented by dashed oval in ERD.
iii) e.g → Person age is derived attribute as it changes over time & can be derived from DOB.





Sulekha

* E-R diagram with multivalued & derived attributes.



* Relationship

Cardinality → It defines the numerical attributes of the relationship between two entities or entity sets.

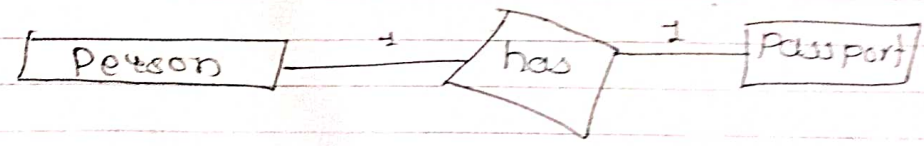
- A relationship is represented by diamond shape in ER diagram, it shows the relⁿ among entities. There are four types of cardinal relⁿ.

1) one to one
2) one to many
3) many to one
4) many to many

1) One to one relationship.

i) When single instance of entity is associated with the single instance of another entity then it is called one to one relationship.

ii) e.g → a person has only one passport & passport given to one person.

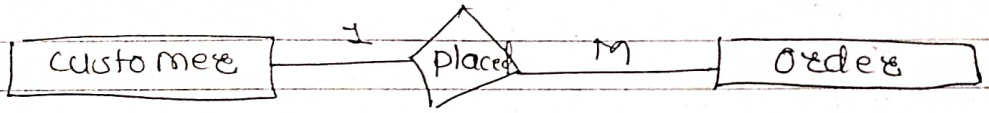


Page No :
Date :

2) One to Many

i) when a single instance of an entity is associated with more than one instance of another entity then it is called one to many relationship.

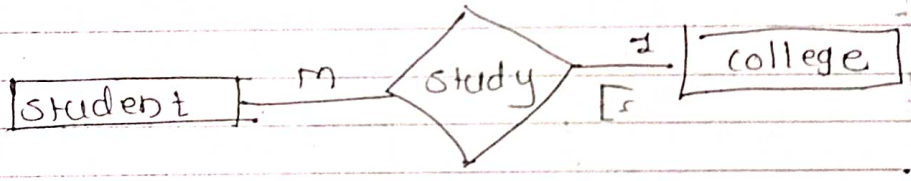
ii) e.g → A customer can place many orders but order cannot placed by many customer.



3) Many to one

i) when more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship.

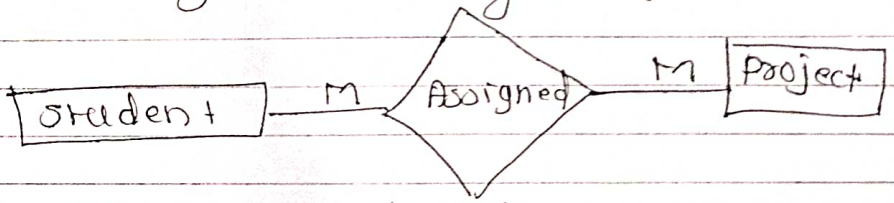
ii) e.g → Many students can study in a single college but student can't study in many colleges at the same time.



4) Many to Many

i) when more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationships

e.g → Assigned many projects & projects can be assigned to many students

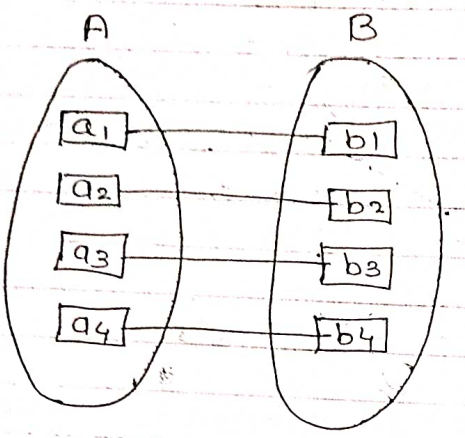


Participation constraints

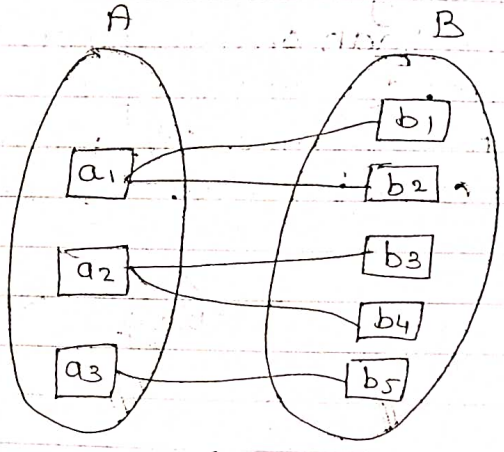
1)

Suleksha

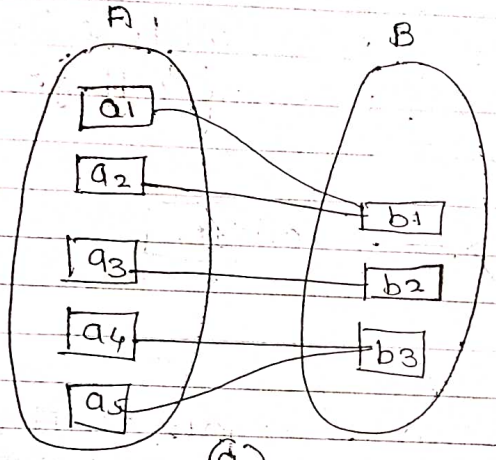
Mapping cardinalities



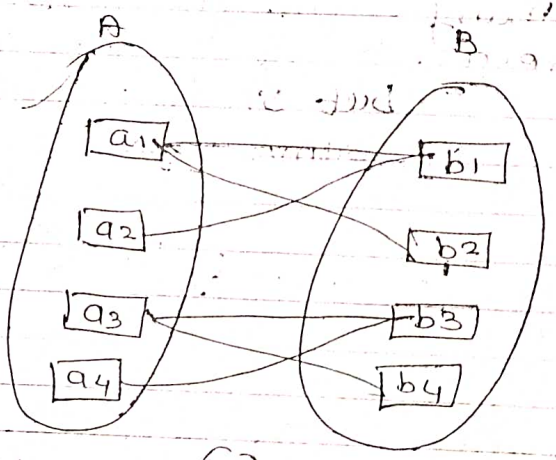
(a)
one to one



(b)
one to many



(a)
Many to one



(b)
Many to many

Page No :
Date :

Keys

i) A super key of an entity set is a set of one or more attributes whose values uniquely determine each entity.

ii) A candidate key of an entity set is minimal super key.
- social security is candidate key of customer
- account-no. is candidate key of accnt.

iii) Although several candidate keys of th may exist, one of the candidate keys is selected to be the primary key.

iv) The combination of primary key of the participating entity sets forms a candidate key of relationship set.
- must consider the mapping cardinality & the semantics of the relationship set when selecting the primary key.
- social security, accnt no. is the primary key of depositor.

Reduction of an ER schema to Tables

i) Primary keys allow entity sets & relationship sets to be expressed uniformly as tables which represent the contents of the DB

ii) A DB which conforms to an E-R diagram can be represented by a collection of tables

iii) For each entity set & rel^n set there is a unique table which is assigned the name of the corresponding entity set or rel^n set.

iv) Each table has a no. of columns which have unique names.

v) Converting an E-R diagram to a table format is the basis for deriving relational DB design from an ER diagram.

1) Representing entity sets as tables

i) A strong entity set reduces to table with same attri

customer-Name	social security	c_street	C_city
A	019-12-1112	Main	Harrison
B	067-89-9011	North	Rye
C	821-12-1331	Main	Harrison

customer table

ii) A weak entity set becomes a table that include a column for primary key of the identifying strong entity set.

loan-no	payment-no	Pay-date	pay-amount
L-17	5	10 May 2021	50
L-23	11	21 Jan 2022	70
L-15	22	10 Feb 2022	300

Payment table

2) Representing Relationship sets as tables.

i) A many to many relⁿ set is represented as a table with columns for the primary keys of the two participating entity sets & any descriptive attributes of the relⁿ set.

social security	accnt-no.	access_date

depositor table

ii) The table corresponding to a relationship set linking a weak entity set to its identifying strong entity set is redundant. The payment table already contains the information that would appear in loan payment table i.e columns loan-no & payment-no.

Page No :
Date :

3) Representing Generalization as Table

1) Form a table for generalized entity accont: form a table for each entity set i.e generlized include primary key of generalized entity set.

table	table attributes
accnt	accnt-no., balance, accnt-type
saving-accnt	accnt-no., interest-rate
checking_accnt	accnt-no., overdraft-amnt

2) Form a table for each entity set i.e generlized

table	
savings-accnt	accnt-no, balance, interest rate
checking-accnt	acent-no, balance, overdraft-amnt

4) Relations corresponding to Aggregation.

customer

cust-name	cust social security	street	city

loan

loan-no.	amount

borrower

cust-social-security	loan-no.

employee

emp-social security	name	phone-no

loan officer

emp-social security	cust social security	loan-no

Sulekha

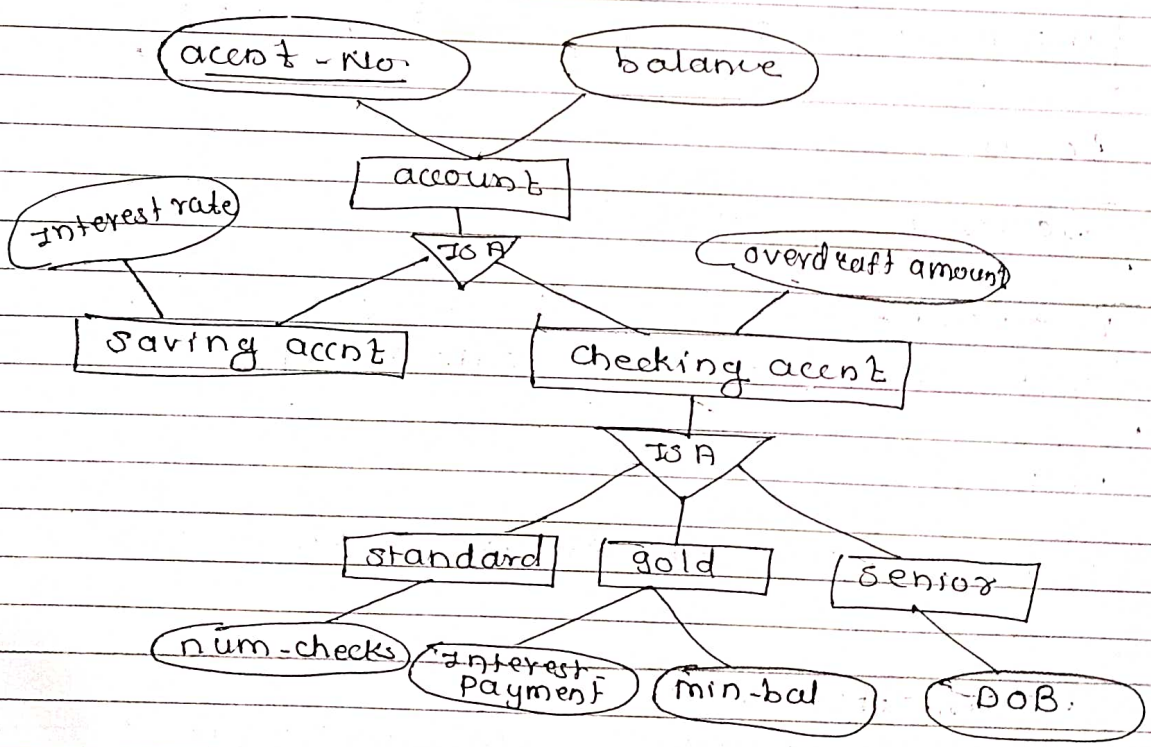
Extended E-R Features

1) Specialization
2) Generalization
3) Aggregation

1) Specialization

i) Top down design process
ii) we designate subgrouping within an entity set that are distinctive from other entities in the set.
iii) These subgrouping becomes lower level entity sets that have attribute or participate in relationship that do not apply to the higher level entity set.
iv) Depicted by a triangle component labeled ISA i.e saving acct is an account.

Example



Page No :
Date :

Generalization :

i) A bottom up design process combine a no. of entity sets that share the same features into a higher level entity set.

ii) specialization & generalization are simple inversions of each other, they are represented in an ER diagram in the same way.

iii) Attribute inheritance
a lower level entity set inherits all the attributes & relationship participation of the higher level entity set to which it is linked.

Design constraints on a Generalization :

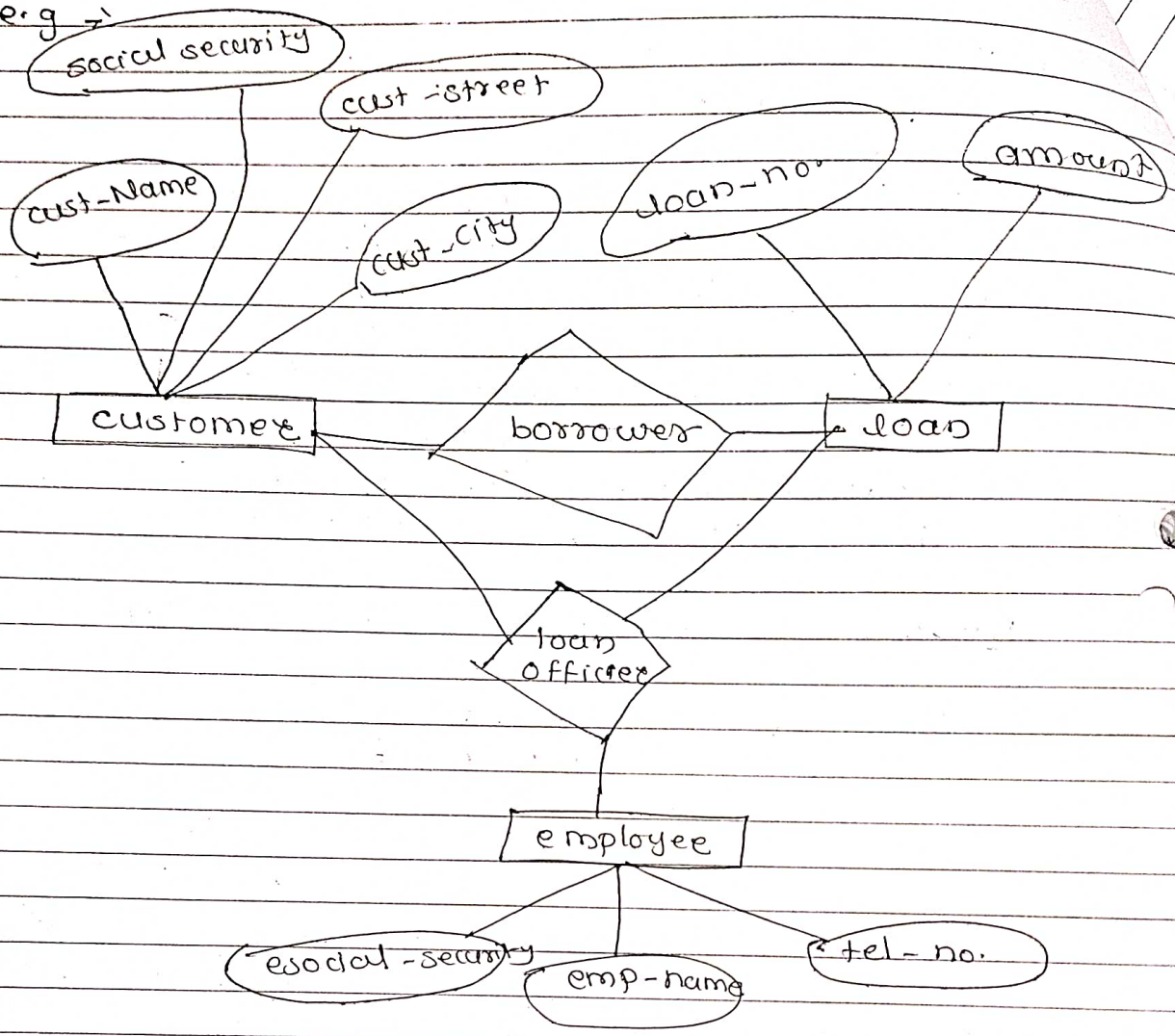
i) constraint on which entities can be members of a given lower level entity set.
- condition defined
- user defined

ii) constraint on whether or not entities may belong to more than one lower level entity set within a single generalization
- disjoint
- overlapping

iii) Completeness constraint
Specifies whether or not an entity in the higher level entity set must belong to atleast one of the lower level entity sets within a generalization
- total
- Partial

Sulekha

e.g :-



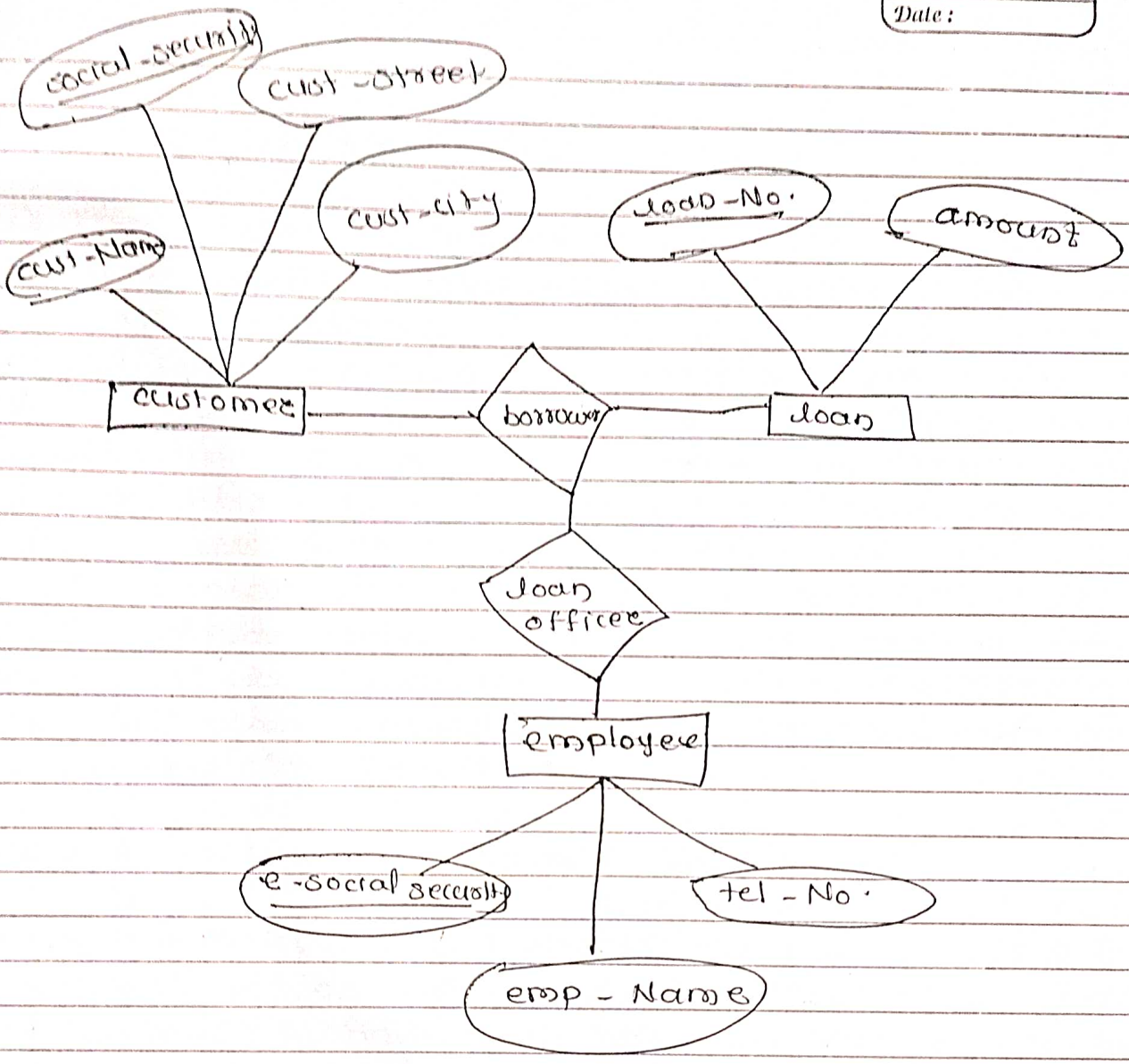
Loan customer may be advised by a loan officer

i) Relationship sets borrower & loan officer represent the same information

ii) eliminates this redundancy via aggregation
- treat relationship as an abstract entity
- Allows relationships between relationships
- Abstraction of relationship into new entity

iii) without introducing redundancy the following diagram represents that
- A customer takes out a loan
- An employee may be loan officer for a customer loan pair

e.g →


social-security
cust-street
cust-city
cust-Name
loan-No.
amount
customer
borrow
loan
loan officer
employee
e-social security
tel-No.
emp-Name

Page No :
Date :

Purpose of Normalization

i) The purpose of normalization is to identify a suitable set of relations that support the data requirements of an enterprise.

ii) Characteristics of a suitable relations are
   a) The minimal no. of attributes necessary to support the data requirements of enterprise.
   b) Attribute with a close logical relationship are found in the same relations.
   c) Minimal redundancy with each attribute represented only once with important exception of attributes that formall or part of foreign keys which are essential for the joining of related relations.

How Normalization support DB Design.

i)- Fig. shows. Approach 1 shows how normalization can be used as bottom up standalone DB design tech! & approach 2 shows how normalization can be used as validation technique to check the structure of relations, which may have been created using a top-down approach such as ER modelling.

ii) The opportunity to use normalization as a bottom-up standalone technique (approach 1) is often limited by the level of detail that the database designer is reasonably expected to manage. However this limitation is not applicable when normalization is used as validation technique (approach 2) as DB designer focuses on only part of DB such as single relation at any one time therefore no matter what the size ~~of example~~ or complexity of DB, normalization can be usefully applied.

Sulekha

Data Redundancy & Update Anomalies

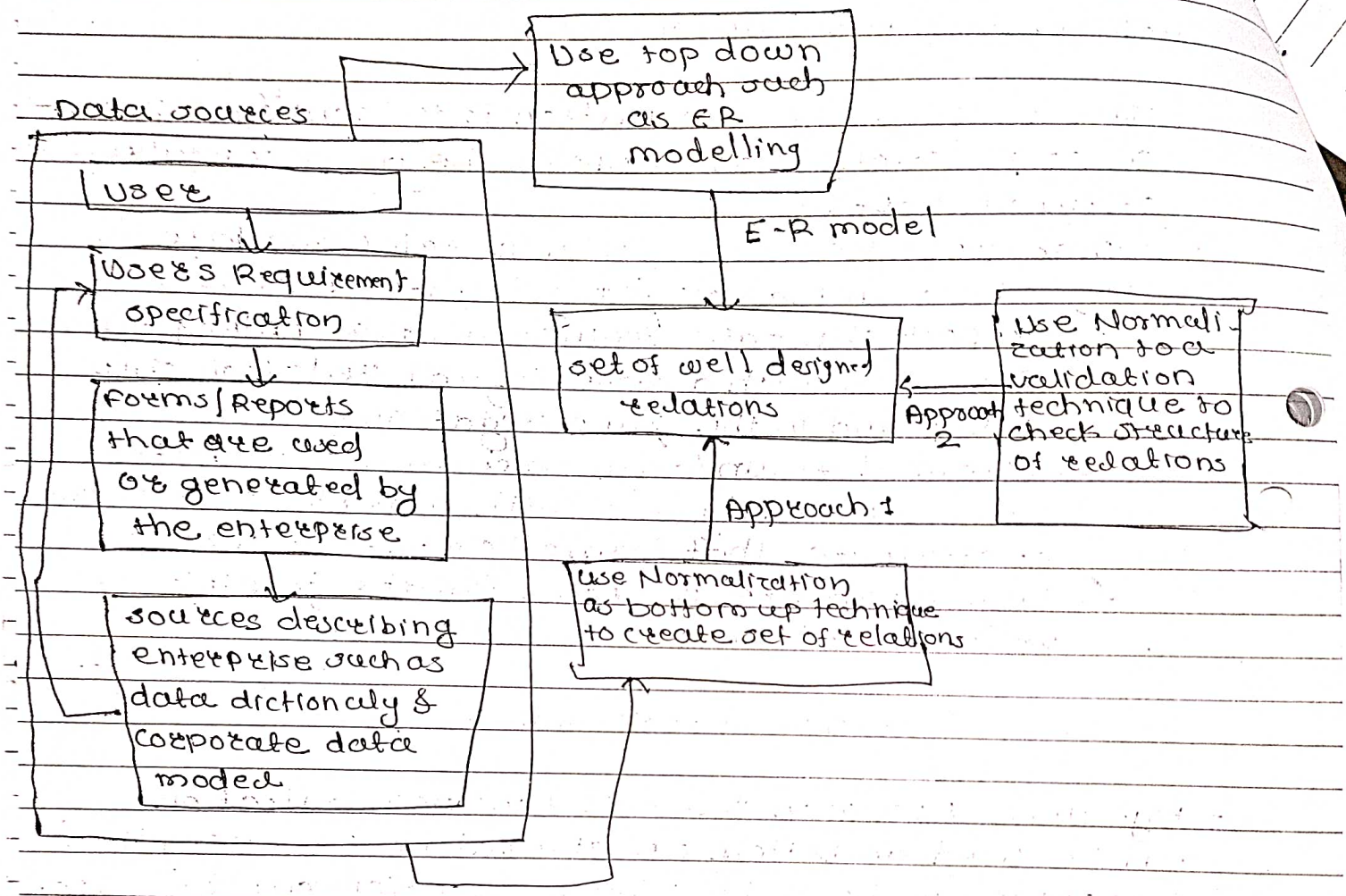


fig. How Normalization can be used to support database design.

Page No :
Date :

Data Redundancy & Update Anomalies

i) Updates to the data stored in the DB are achieved with minimal no. of operations thus reducing the opportunity for data inconsistencies occuring in DB.

ii) Reduction in the file storage space required by the base relation thus minimizing cost.

So Relational DB also rely on the existence of a certain amount of data redundancy. This redundancy is in the form of copies of primary keys acting as foreign keys in related relations to enable the modeling of relationships between data.

Example →

Staff → (staffNo., sName, position, salary, branchNo.)
Branch → (branchNo., BAddr)
Staffbranch → (staffNo., sName, position, salary, branchNo, BAddr).

Staff

staffNo.	sName	position	salary	branch No.
21	John	Manager	30000	B005
37	Ann	Assistant	12000	B003
14	David	Supervisor	18000	B003
9	Mary	Assistant	9000	B007
5	Susan	Manager	24000	B003
41	Julie	Assistant	9000	B005

Branch

branchNo.	BAddress
B005	22 Deer Rd. London
B007	16 Argyll St. Aberdeen
B003	163 Main St. Glasgow

fig. 1

Sulekha

staffBranch

staffNo	SName	Position	salary	branchNo	BAddr
21	John	Manager	30000	B005	22 Deer Rd. London
37	Ann	Assistant	12000	B003	163 Main st Glasgow
14	David	Supervisor	18000	B003	163 Main st Glasgow
9	Mary	Assistant	9000	B007	16 Argyll St. Aberdeen
5	Susan	Manager	24000	B003	163 Main st. Glasgow
41	Julie	Assistant	9000	B005	22 Deer Rd. London

fig. 2

In the staffBranch relation there is redundant data the details of branch are repeated for every member of staff located at branch, the branch details appear only once for each branch in the staff relation to ~~represent where each member of staff is located~~ in the branch relation & only the branch number (branchNo) is repeated in the staff relation to represent where each members of staff is located. Relations that have redundant data may have problem called update anomalies which are classified as insertion, deletion or modification anomalies.

Insertion Anomalies.

There are two main types of insertion anomaly

1) To insert the details of new members of staff into staffbranch relation, we must include the details of the branch at which the staff are to be located.
e.g → To insert details of new staff located at branch no. B007 we must enter the correct details of branch no. B007 so that branch details are consistent with values for branch B007 in other tuples of the staffBranch relation.
- The relation shown in fig. 2 dont suffer from this potential inconsistency becoz we enter only branch no. for each staff member in the staff relation. Instead details of branch no. B007 are recorded in DB as single tuple in the branch relation.

Page No :
Date :

The process of Normalization

i) Normalization is a formal technique for analyzing relations based on their primary key & functional dependencies.

ii) It is a method of organizing data in the DB which helps you to avoid data redundancy, insertion, update & deletion anomaly.

Types of Normal Form

1) 1NF
2) 2NF
3) 3NF
4) BCNF
5) 4 NF
6) 5NF

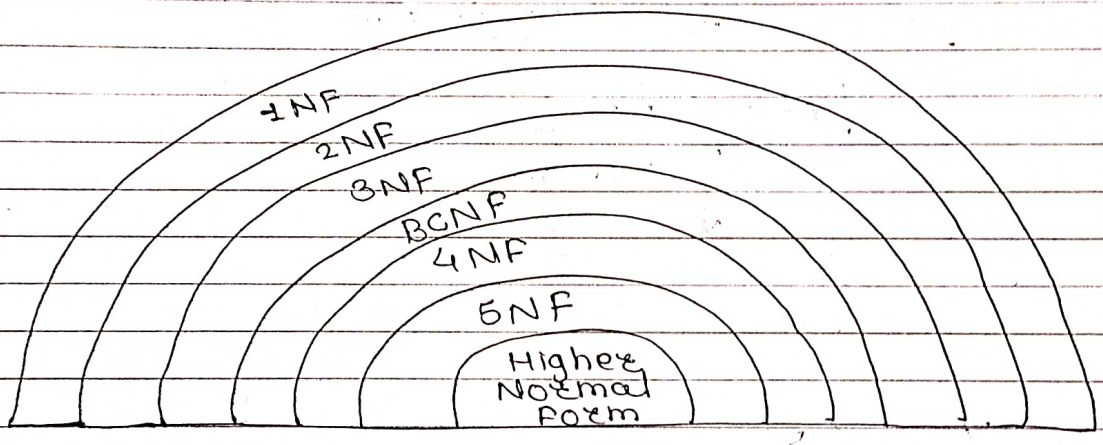
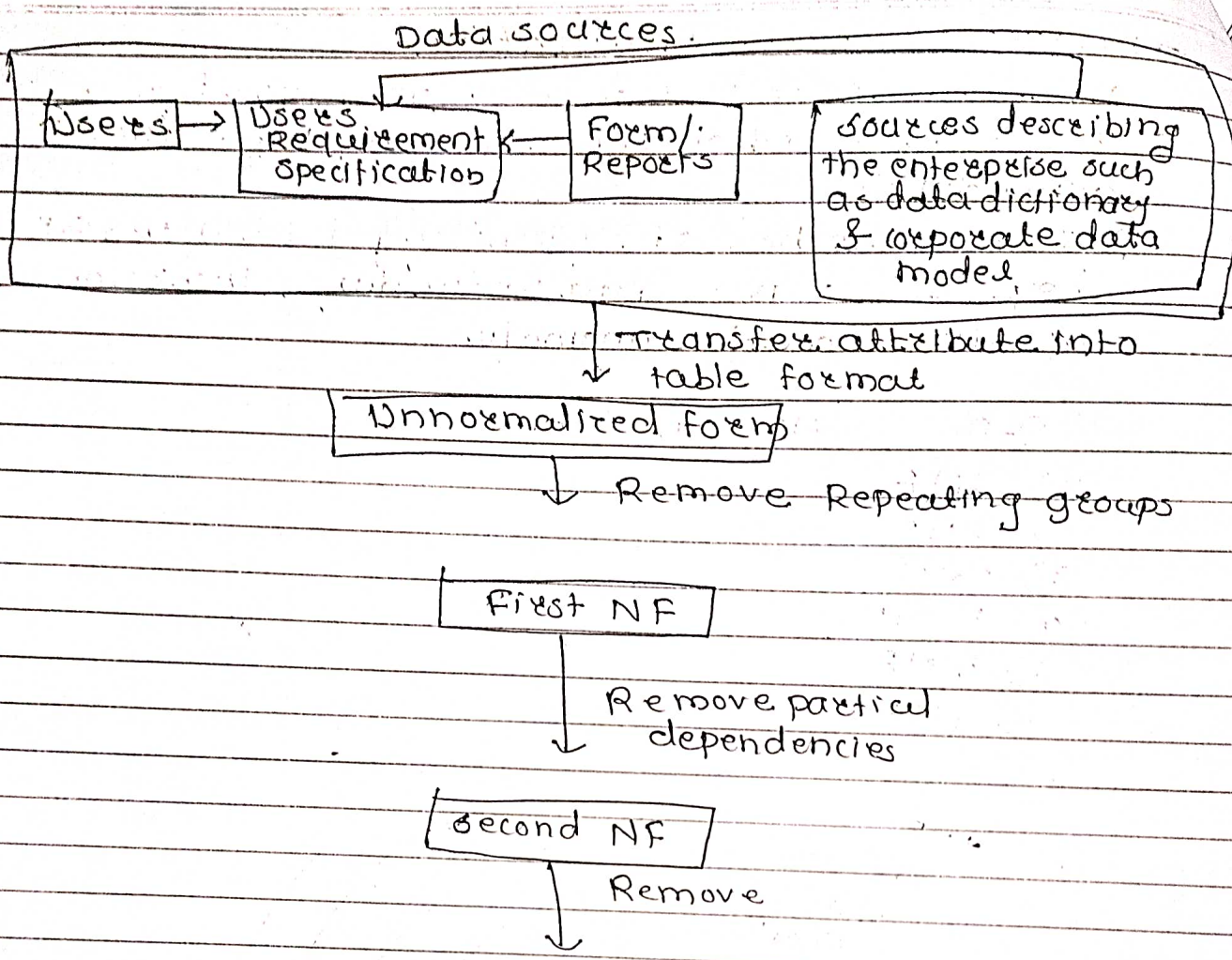


fig → relationship between the Normal form


Data sources.
Users
Users Requirement Specification
Form/ Reports
Sources describing the enterprise such as data dictionary & corporate data model
Transfer attribute into table format
Unnormalized form
Remove Repeating groups
First NF
Remove partical dependencies
Second NF
Remove

Page No.:
Date: / /

Functional Dependency

i) Functional dependency is a type of constraints exists between multiple attributes of a relation.

ii) we can say column X is functionally dependent on other column Y. If data value in column X change when data value in another column Y is modified.

iii) FD determines the set of values or attribute based on another attribute.

e.g → student table

Id	Name
01	AAA
02	XXX

when name of student is changed it's ID also need to be changed so we can say column is depend on ID column.

iv) FD is used to avoid data redundancy or minimize means same data should not be repeated at multiple location in same DB.

v) It is denoted by →, $X \rightarrow Y$ & can be written as Y is functionally dependent on X or X determines Y

Types of Functional Dependencies

1) Full Functional Dependency

i) A functional dependency $A \rightarrow B$ is a full functional dependency if removal of any attribute from A means that the dependency does not hold any more

ii) Example → (EmpNo, Project No) → Hours

i.e EmpNo → Hours

& ProjectNo → Hours.

Hours are fully FD on both empno & project No.

2) Partial Functional Dependency

i) Partial functional dependency means that a non key column is depend on some column in composite primary key of a table

ii) Example → student table

Id	Name	DOB
1	AAA	1/1/1986
2	XXX	2/2/1987

In above table Id & Name of student together making its composite key, so Id column is partially dependent on name because when name of employee changes ID also need to be changed.

iii) An FD $A \rightarrow B$ is a partial dependency if there is some attribute $x \in A$ (x subset of A) that can be removed from A & dependency will still hold.

e.g → (EmpNo, Project No) → Name

i.e EmpNo → EName

3) Transitive dependency (Trivial)

i) when one non key attribute is functionally dependent on another non key attribute then such a dependency is called as transitive dependency.

ii) Non key attribute → Non key attribute

iii) An FD $X \rightarrow Y$ in a relation R is a transitive dependency if there is a set of attributes Z i.e not subset of any key of R & both $X \rightarrow Z$ & $Z \rightarrow Y$ holds true

iv) e.g → Empdept (ENo, EName, DNumber, DMgrNo)

ENo → DmgrNo is transitive

Dependency of DmgNo on Key attribute Eno is transitive as DmgrNo is depends on Dnumber which is depend on Eno.

Eno → DNumber & DNumber → DmgrNo.

Eno	Ename	DNumber	DmgrNo
1	AAA	10	1
2	XXX	10	1

4) Multivalued dependency

i) Multivalued dependency is a relationship which accepts the cross product pattern.

ii) $X \twoheadrightarrow Y$ is said to hold for relation R(X, Y, Z) if for given set of values of X, there is set of associated values of attribute Y & X values depend on X values & have no dependence on set of attrib-Z

iii) e.g → Employee (EName, Addr, Car) which can be given as

EName $\twoheadrightarrow$ Address

EName $\twoheadrightarrow$ Car

$\alpha \twoheadrightarrow \beta$ says relationship between α & β independent of rel^n betwee α & $R - \beta$

That means EName $\twoheadrightarrow$ Address relationship is independent of EName & (EName, Car) rel^n.

i.e EName $\twoheadrightarrow$ Addr is independent of EName $\twoheadrightarrow$ Car

EName	Addr	Car
AAA	Kodoli	Maruti 800
BBB	Vathar	Omni

we will take one more relation.

Product purchased	custName	Street	City
Bread	AAA	AB	Mumbai
Egg	AAA	H.C Road	Goa

- Relation between custName & Product purchased is independent of rel^n between customer & his addr.
- If AAA has purchased bread, we want all AAA address associated with that purchase

custName → Street, City

2) Establish new branch ~~staff~~ with no members of staff B008, 57 princes st Edinburgh, so No staff members so staff No. must be NULL, but staffNo. is the primary key of the staffBranch table, so cannot be NULL.

3) Deletion Anomalies

i) If only one staff is working at a branch & if that staffs information is deleted, then branch details are also lost from dB.

ii) e.g → if staff 21 is record is deleted, the branch information of branch_no B005 is also lost. but the design of fig. 2 branch table avoid this problem because branch tuples stored separately from staff tuple & attribute & attribute branchNo relates two relations.

4) Modification

i) If we change the value of one attributes of a particular branch in the staff branch relation.

ii) ~~e.g~~ → This anomaly occures when dupdicate data is updated only one place & not in all instances, Hence it makes our data or table inconsistent state.

Page No.:
Date: / /

Armstrong's Axioms - closures of FD

i) Axioms are nothing but rules of inference which provides a simple technique for reasoning about FD.

1) Primary rules

a) subset property (axiom of reflexivity)
If Y is a subset of X then $X \to Y$ & also $X \to X$

b) Augmentation
If $X \to Y$ then $XZ \to YZ$

c) Transitivity
If $X \to Y$ & $Y \to Z$ then $X \to Z$

2) secondary rules.

a) Union → If $X \to Y$ & $X \to Z$ then $X \to YZ$
b) Decomposition → If $X \to YZ$ & $X \to Y$ then $X \to Z$
c) Pseudo Transitivity → If $X \to Y$ & $YZ \to W$ then $XZ \to W$

3) Canonical cover

A FD set X is canonical or minimal if the set has following properties

1) Each right set of FD of X contains only one attribute.
2) Each left set of FD of X is minimal. It means there should not be any extraneous attribute is present in dependency.
3) Reducing any FD will change the content of X