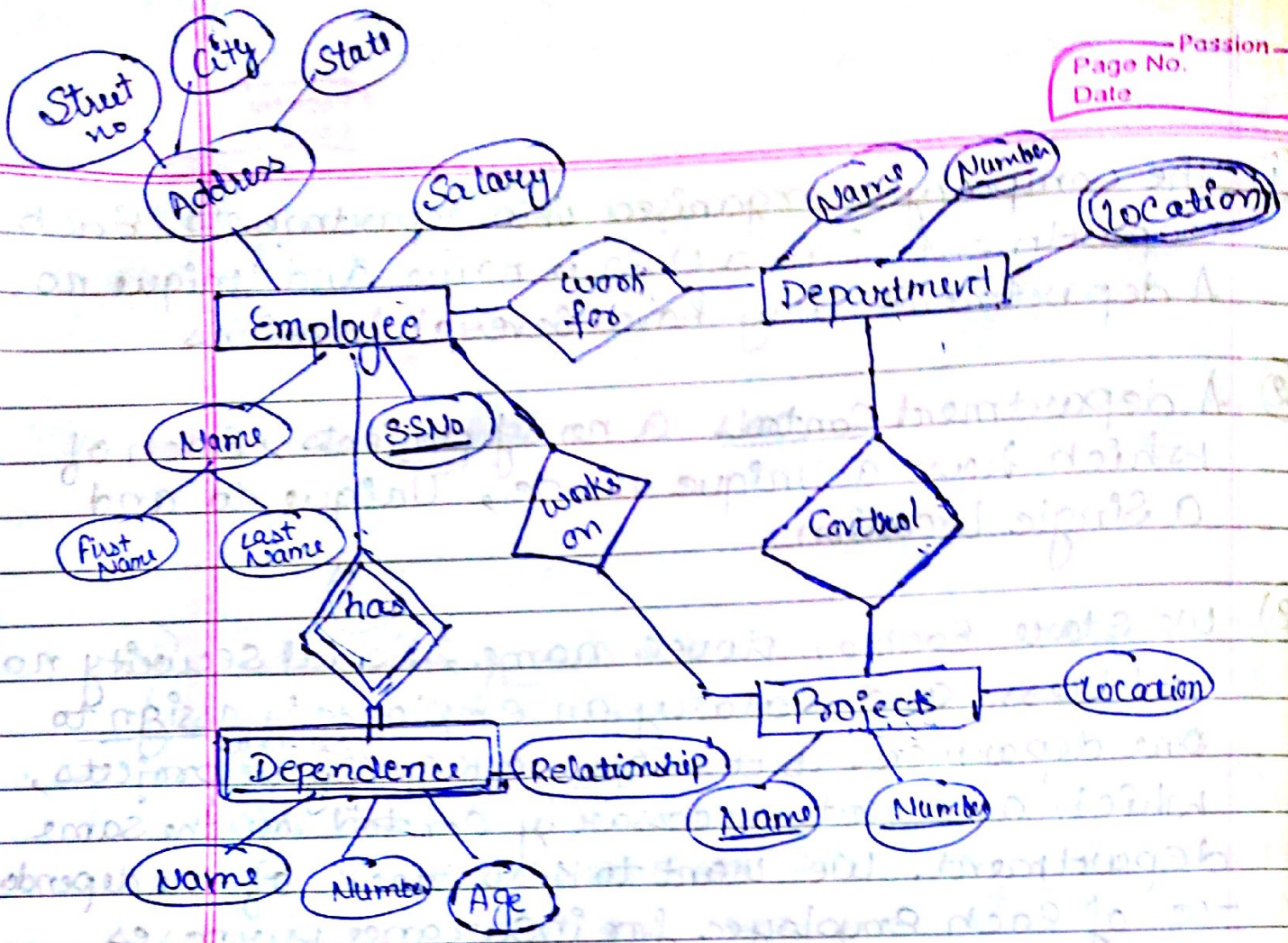
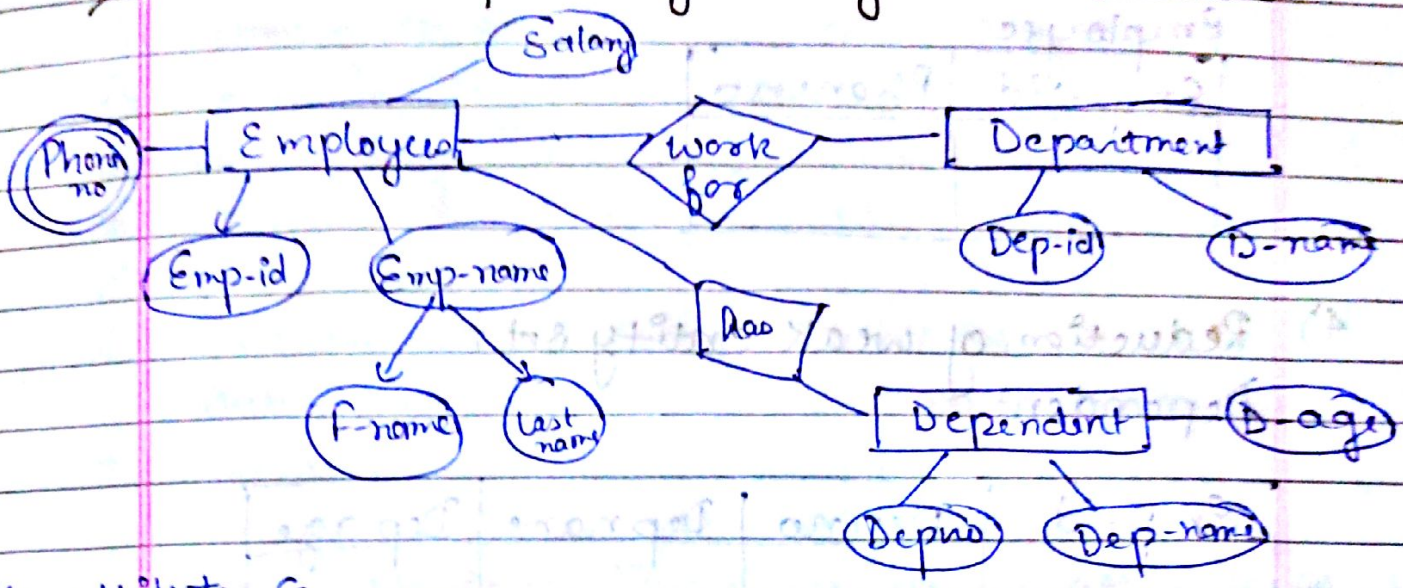


Que Construct an E-R diagram of University system.

Reduction of E-R diagram into table

1) Reduction of Strong entity set into table



Key attribute Eg.

Employee

Emp id	E-name	Salary

Department

Dep-id	Dept name

Reduction of composite attribute into table

Employee

Emp-id	f-name	l-name	Salary

3) Reduction of multivalued attributes
Phoneno

Employee

Emp-id	Phoneno

4) Reduction of weak entity set :
Dependent

Emp-id	Depno	Depname	Dep age

5) Reduction of relationship sets
Work for

Emp-id	Department id

RDBMS. :
3 Components :
- Data Structure
- Data integrity
- Data Manipulation

Candidate key is a subset of Superkey.

Key : It is a set of one or more columns whose combined values are unique among all the

Page No.
Date

① ~~Candidate Key~~ :
occurances in a given table

Types of keys :-
1) Candidate key
2) Super key
3) Primary key
4) foreign key
5) Alternate key
6) Artificial key
7) Composite key

1) Candidate Key :- They are those attributes of the relational, which have the properties of uniqueness and irreducibility.

a) uniqueness :- No legal value of R ever contain two distinct tuples with same values for K.

b) Irreducibility :- No proper subset of K has the uniqueness property.

For example if the Combination of (Name, Class) is unique, then it can be identified as the candidate key if and only if Name and Class individually are not unique.

2) Super Key :- Super key follow the properdy of uniqueness, but not irreducibility.
A Super key has a uniqueness property but not necessarily the irreducibility property.
A candidate key is a special case of a Super key

Roll No. Class Name ...

Page No.
Date

eg:- If Roll-no is unique in relation STUDENT then, the set of attribute (Roll-no, Name, Class) is a Super key for a relation STUDENT, these set of attributes are also unique, but this combination of key is not having the property of irreducibility

→ Relation of patient in which Patient-number is unique. The patient-number is a candidate key and (Patient-number, Patient name) is a Superkey.
'A superset of a candidate key is a Superkey.

3) Primary Key:- The primary key is an attribute or a set of attributes that uniquely identify a specific instance of an entity. Primary key cannot contain any null value because we cannot uniquely identify multiple null values. ~~Ea Primo~~ primary key is a candidate which promote by database designer as primary key.

4) Alternate Key:- Exactly one of those candidate keys is chosen as the primary key and the remainder, if any, are then called Alternate Keys. An alternate key is a function of all candidate key minus primary key.

5) Composite keys:- A primary key that is made up of more than one attribute is known as Composite key.

Ex:- WORK

Employee ID	Project ID	Hours-Worked
01	01	200
01	02	120
02	01	50
02	03	120
03	03	100
03	04	200

It is subset of superkey.

6) Artificial keys:- An artificial key is one that has no meaning to the business or organization and perform function of a primary key in relation. Artificial key are permitted when

(1) no attribute has all the primary key properties
(2) the primary key is large and complex

eg:-

Enrollment		
Student	Class	row-id
AK		PK

7) Foreign key:- Foreign key are the attributes of the table which refers to primary key of some another table. Foreign key are used to link together two or more different tables which have some form of relationship with each other. These foreign keys is a reference to the tuple of a table from which it was taken. This tuple being called as Referenced or target tuple.

Target Attribute

Foreign key

eg:- Employee

Department → Target table

Emp-id	Name	Salary	Dep-id
1	Abc	30000	10
2	Xyz	40000	20
3	Pqr	50000	30

Dep-id	Dep-Name
10	Sales
20	Market
30	Product

eg:-

Student

Rno	Name	Class-Code
1	A	2
2	B	1
3	C	-

Class

ClassCode	Name
1	B.TECH
2	B.TECH
3	BBA

(2) Data Integrity :- Basically it consist of two rules
1) Entity Integrity Rule
2) Referential Integrity Rule

(1) Entity Integrity Rule :- This rules states that in a relation, the value of the attribute of a primary key can not be null.

(2) Referential Integrity :- It states that if a foreign key exists in a relation, either the foreign key value must match the primary key value of some tuple in its home relation or the foreign key value must be completely null.

Passion
Page No.
Date

Employee

Emp-id	Name	Salary	Dep-id
1	A	3000	10
2	B	4000	20
3	C	5000	30
4	D	6000	40

Department

Dep-id	Dept name
10	Sales
20	Market
30	Product

either should match or null.

Enterprise Constraints :- These are the additional rules which are given by the user & DBA of a a database. eg <, >.

Passion
Page No.
Date

Codd's Rule:- Founder of RDBMS is Dr. E.F Codd
There are 12 rule are as following:-

(1) Information rule:- All information represented in form of tables
(2) Guaranteed Access rule:- To Access any info we use key.
(3) Comprehensive data Sublanguage rule:- There should be a particular language to support RDBMS rule.
(4) View updating rules:- If we change any data in Database then it should be automatically updated in all records or tables
(5) High level Insert, update & Delete:- The language we used it should contain these rule insert, update & Delete and perform all function.
(6) Physical data independency:- Change in lower level not effect higher lvel and this rule support.
(7) Logical Data idependency:- Change in conceptual level and doesnot effect external level.
(8) Integrity Independency:- RDBMS support all Integrity rule.
(9) Non Subversion rule:- Any language we use to access the database and, that language will support our Integrity independency.
(10) Systematic treatment of null Value:- There should be special treatment of null value.
(11) Data Description rule:- Data we describe it should be in form of table

(12) Distribution Independence :- It should be platform independent

Data Manipulation :-
(1) Relational Algebra
(2) Relational calculus.

Q Difference b/w Relational Algebra & Relational Calculus.

	Relational Algebra	Relational Calculus.
(1)	It is procedural language In this we follow the procedure & step	(1) It is a non procedural language In this we automatically get output
(2)	We can combine 2 or more table to get an another table Both are non user friendly.	

Relation Algebra :- A (+) B

Relational operator :- Type of Relational operator
(1) Traditional Set Operator
(2) Special operator

(1) Traditional Set operators :- → Union
→ Intersection
→ Difference
→ Cartesian Product

Union :- Combine two table

Employee

Emp-id	Name	Dept-id

Dep-id

Dep-id	Name

→ Intersection: Common element

→ Difference: The Difference between two set S1 & S2 produces a set, which contain all the member of one set, which are not in the other

R

Cust.name	Cust-Status
Ram	Good
Shyam	Excellent

S

Cust-name	Cust Status
Karan	Average
Ram	Good

(1) R ∪ S

Cust.name	Cust.Status
Ram	Good
Shyam	Excellent
Karan	Average

(2) R ∩ S

Cust.name	Cust.Status
Ram	Good

(3) R − S ⇒ Shyam.

(4) R × S = { (Ram, Karan) (R R), (S K), (S R) }

R

Name	Roll no
Ane	10
Mim	8

S

	Roll
Ajay	11
Arun	12

R × S = { Ane 10 ...

ii) Special operators:
- → Selection
- → Projection
- → Joins
- → Division

(1) Selection operation: It yields a horizontal subset of a given relation that is the subset of row should be selected with in the given relation for which a particular

Condition is satisfied. Sign of selection is (σ)

Employee :

Emp-id	Name	Salary

For row: $\sigma_{salary > 10,000}(Employee)$

② Projection: The projection operation on a table, simply form another table by copying specified columns from the original table. Symbol of projection is (Π)

To select name of employee :

$\Pi_{name}(Employee)$

For salary :

$\Pi_{salary}(Employee)$

We want the name of all the employee having salary less than 70,000

Or say: $\sigma_{salary > 70000}[\Pi_{name}(Employee)]$

Roll 11 12

Passion
Page No.
Date

Codd's Rules :-

(1) Information Rule :- All information in a relational database including tables names, column names is represented in the form of tables. This simple view of data speedsup design and learning process) user productivity is improved since knowledge of only one language is necessary to access all data such as description of the table and attribute definitions, integrity constraints [Action are taken when constraint are voliated. Access of

data is restricted
All info also stored in table

(2) Guaranteed Access Rule :- Every piece of data in relational database, can ibe accessed by using a primary key value that identifies the row name and column name. User productivity is improved since there is no need to resort to using physical pointers or address. It provide data independence

or make it possible to retrive each piece of data in a relational database

(3) Comprehensive Data Sub-language Rule :- The RDBMS may support several languages. But atleast one of them should allow the user to do all following define table, view, query & update data, set integrity constraints, set authorizations & define transaction. User productivity is improved since there is just one approach that can be used for all database operations.

(4) View Updating Rule :- Any view that can be updated theoretically can be updated using the RDBMS. Data consistency is ensured since the changes made in the views are

transmitted to base table and vice-versa.

(5) High level Insert, update & Delete :- The RDBMS support insertion, updating and deletion at a table level. The performance is improved since the commands acts on a set of records rather than one record at a time.

(6) Physical Data independence :- → Application program is not effected by changes in the physical access or storage method. The Database administrators can make changes to the physical access and storage method, which improve performance and donot require changes in the application programs or request.

(7) Logical Data independence :- Logical changes in tables and views such as adding/deleting columns or changing field lengths need not necessitate modifications in the programs ~~or in~~.

(8) Integrity Independence :- Integrity Constraints (like table & view definition,) are stored in the one line catalog or data dictionary and can therefore be changed without (integrity independence) necessitating changes in the application program.

(9) Non Subversion Rule :- If the RDBMS has a language that accesses the information of records at a time, this language should not be used bypass the integrity constraints. This is necessary for data integrity.

(10) Systematic Treatment of Null values:- In RDBMS null values should be supported for the representation of missing ~~information~~ and inapplicable information. The database management must have a consistent method for representing null values.

(11) Database Description Rule:- The description of database is stored and maintained in the form of tables. It allows the user with appropriate authority to Query information in a similar way or language.

(12) Distribution Independence:- The RDBMS package must have distribution independence Thus RDBMS must make it possible for the database to be distributed across multiple packet

Comparision:-

Relation Algebra	Relational Calculus
(1) It is a procedural method for solving Queries	(1) It is non-procedural method of solving the Queries
(2) The Solution to the data base access problem using a relational algebra is obtained by stating what is required and what are the steps to obtain that information	The Solution to the database access problem using a relational Calculus is obtained simply by stating what is required and letting the system find the answer
It is used as vehicle for implementation of Relational Calculus	Relation Calculus Queries are converted into equivalent relational algebra format by using Codd's reduction Algorithm and then it is implemen

	with the help of relational algebra operators
(4) Relational algebra operators are used as a yardstick for measuring the expressive power of any given language.	A lang. is said to be complete if it is at least powerful as the Calculus that is, if any relation definable by some expression of the calculus is also definable by some expression of the language in question

Difference :-

(1)
- Relational Algebra is a procedural language that can be used to tell the DBMS how to build a new relation from one or more relation in the database.

compulsory even they are having heterogeneous platform for both the operating system & hardware

- Relational Calculus is non-procedural language that can be used to formulate the definition of relation in terms of one or more database relation.

(2) Relation Algebra :- User has to specify what is required and what are the procedure or steps to obtain output.

Relational Calculus :- User just specifies what is required and need not to specify how to obtain it

Strong Entity Set	Weak Entity Set
(1) It has its own primary key	It does not have sufficient attribute to form a primary key on its own.
(2) It is represented by rectang	It is represented by double rectangle
(3) It contain primary key represented by an Underline	It contain Partial key represented by dashed underline
(4) The member of Strong entity set is called dominant entity set	The member of weak entity set is called Subordinate entity set
(5) The relationship between two strong entity set is represented by diamond System	The relationship b/w one Strong and a weak entity set is represented by a double diamond sign. i.e Identifying relationship
(6) The line connecting Strong entity set with the relationship is single	The line connecting weak entity set with identifying relationship is double
(7) Total participation in the relationship may or may not exist	Total participation in the identifying relationship always exist

Page No. Passion
Date

Relational Algebra :-

Joins :- General form
↳ Theta Join.

Theta Joins :- It is a cartesian product operation on the two tables followed by a restriction operation on the resultant tables

eg :-

Employee Product

Name	Product
Raja	Pen
Spash	Pen
Raja	Pencil
Spash	Rubber

Product Customer

C-Product	Customer
Pen	Karan
Pen	Suneet
Pencil	Suneet

Eq: Raja Pens

Name	Product	Customer
Raja	Pen	Karan

Operation
↳ Comparision operator
→ equal =
→ Not equal to a
→ Greater than >

	Name	Product	C-Product	Customer
Equal= ✓	Raja	Pen	Pen	Karan
✓	"	"	Pen	Suneet
Not equal →	"	"	Pencil	Suneet
✓	Spash	Pen	Pen	Karn
✓	"	Pen	Pen	Suneet
	"	"	Pencil	Suneet

Equi Joins :

Natural Joins : The projection operation which does eliminate one of the duplicated column resulting from the equi join, the natural join is obtained.

Name	Product	Customer
Raja	Pen	Karan
"	"	Suneet

→ Inner Joins :

→ Outer Joins : If it requires stack the value exist in only one table must appear in the output then the solution is outer joins.

Name ~~Product~~

These are of two type

1) Left outer joins
2) Right outer joins

Relational Calculus :

Difference b/w Algebra and Relational Calculus

Relational Calculus
→ Tuple
→ Domain

Page No. Passion
Date

Normalisation :-

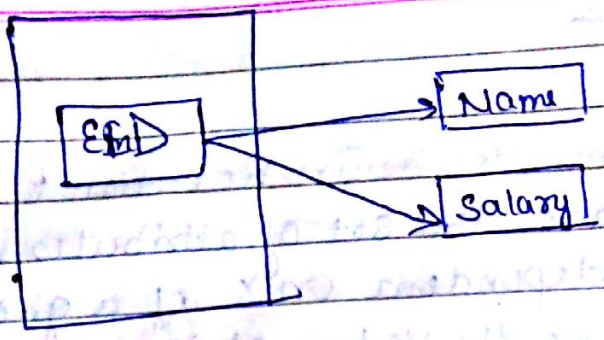
1) Fuctional Dependency :- In Relation R, X & Y are the two subset are the set of attributes, Y is said to be function dependent on X if a given value of X uniquely determine the value of Y
It is denoted by $X \rightarrow Y$ to
Y is Called Determined
where X is Called determinant

Employee

EID	Name	Salary
1	XYZ	30,000
2	Abc	20,000
3	Pqr	10,000

$X_{(EID)} \rightarrow Y_{(Name, Salary)}$

Functional dependency chart :- It is the graphical representation of functional dependencies among the attributes in any relation
Steps :-
1. find out the primary key attribute
2. Make a rectangle write all the primary key attributes inside it
3. write all the non primary key attribute outside the rectangle
4. use the arrows to show the functional dependencies among the attributes



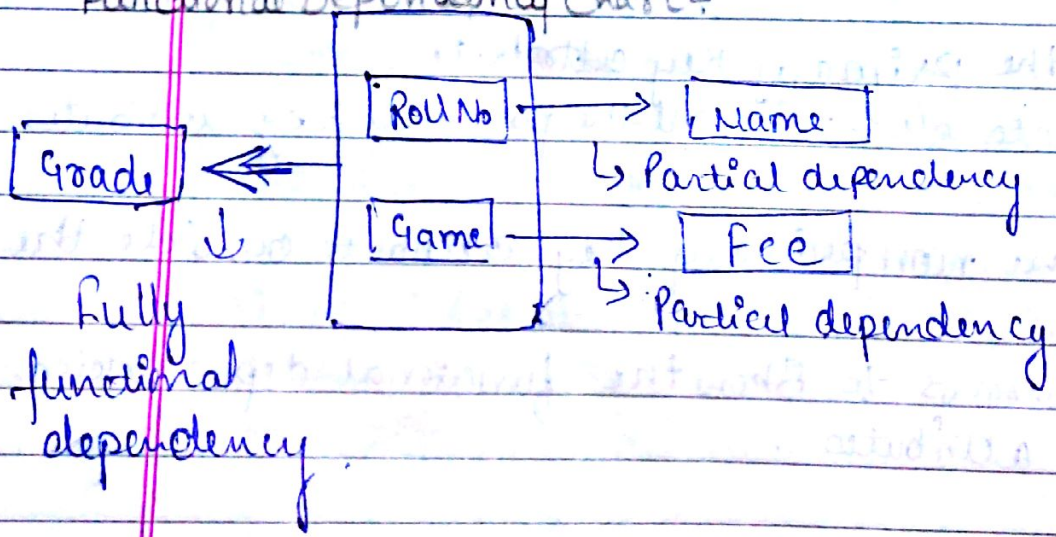
→ Types of functional Dependency :-

① Partial & Fully Dependency.

- Partial Dependency :- Suppose we have more than one attribute in primary key. Let A be the non primary key attribute. If A is not dependent upon all the primary key attribute than partial dependency exists.

- Fully functional Dependency :- Let A be the non primary key attribute & value of A is dependent upon all the primary key attributes then A is said to be fully functional dependency.

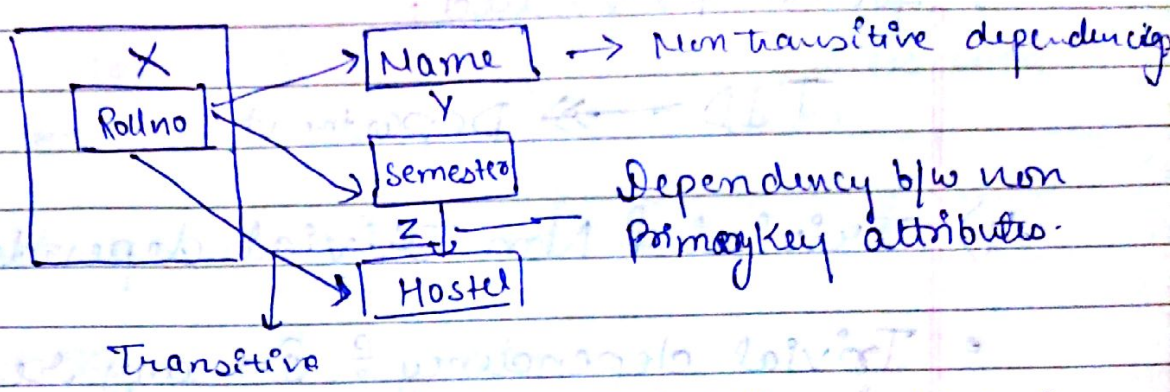
Functional Dependency Chart :-



Page No. Passion
Date

(2) Transitive & Non-Transitive dependency :-

- Transitive dependency is due to dependency between the ~~Known~~ non-primary key attributes.
Suppose in a relation R, $X \to Y$, $Y \to Z$ then $X \to Z$



- Non Transitive :- Any functional dependency which is not transitive is known as Non Transitive dependency.

NOTE :- Non transitive dependency exist if there is no dependency b/w the ~~Known~~ non primary key attributes

(3) Single valued & multiple valued dependency :-

- Single Valued :- In any relation R, if for a particular value of X, Y has single value then it is Known as single valued dependency.

Teacher ID	Name	Department
1	Abc	CSE
1	Abc	ME
2	XYZ	Civil
2	XYZ	ECE
3	Pqr	E.C.E
3	Pqr	CSE

→ for single value

TeacherID → T Name

→ for multiple valued :

TID →→ Department

4) Trivial & Non Trivial dependency :-

- Trivial dependency :- In any relation R, $X \to Y$ is trivial if $Y \subseteq X$
- Non Trivial depency :- In any relation R, if $X \to Y$ if $Y \not\subseteq X$

Anomalies :- undesired result

→ Insertion
→ Deletion
→ Update

EID	Name	Salary	DeptNo	Department
4	Pq r	39,000	34	Marketing
5	ABC	20,000	3	Marketing

→ Normalization: It is a process by which we can decompose or divide any relation into more than one relation to remove the anomalies in the relational database

Normal forms: It is a step by step process and each ~~pro~~ step is known as Normal form.

→ Properties of Normalisation:

1. Remove different anamolies
2. Decomposition must be ~~for~~ lossless
3. Preserves the necessary dependency
4. Reduce Redundancy

In Normalisation there are 5 Normal form

1. First Normal form:
   - ↳ Composite attributes
   - ↳ Flattening the table
   - ↳ Decomposition of table

Teached ID	Name	Subject
1	XYZ	ADA DBMS C, C++
2	PQR	Java, Automata ADCE

(1) Composite attribute

Teacher ID	First Name	Last Name	Subject

(2) Flattering the table

Teacher ID	Name	Subject
1	XYZ	ADA
1	XYZ	DBMS
1	XYZ	C
1	XYZ	C++
2	Pqr	Java
2	Pqr	Automata
2	Pqr	ADCP

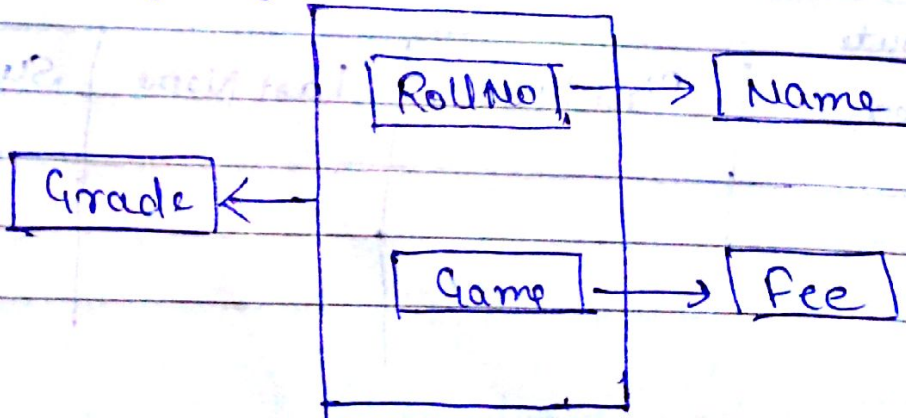
(3) Decomposition of table :

Teacher ID	Name

Teacher ID	Subject

A relation is in the first normal form if the domain of each attribute contains only atomic values it means atomicity must be present in the relation

(4) Second Normal form : A relation is in the second normal form if it is in the first normal from and all the non primary key attribute must be fully functionally dependent upon the primary key attribute



Roll No	Name	Game	Fee	Grade

Partial dependency:

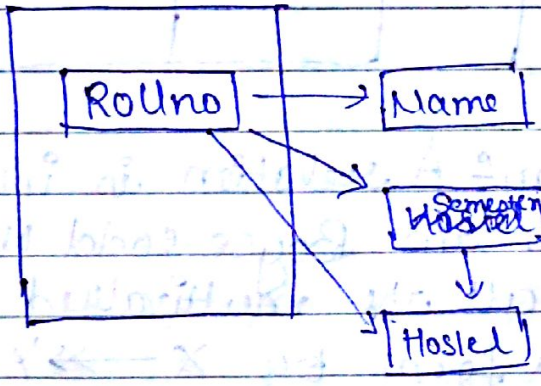
Roll no	Name

Game	Fee

Full dependency:

Roll No	Game	Grade

III) Third Normal form: A relation is in the third normal form if it is in the second normal form and Non primary key attributes must be non transitively dependent upon primary key attributes.



Transitive:

Hostel	Semester

Non transitive:

Roll no	Name	Semester

BCNF (Boyce Codd Normal form) :-
A relation is in BCNF if and only if all the determinants are the candidate key

$X \rightarrow Y$ → determined

↓
determinant

Roll no.	Teacher	Subject

↳ Roll no.
→ (Roll no, Teacher ID)
↳ (Roll no, Subject)

Roll no	Teacher ID

Teacher ID	Subject

IV:- Fourth Normal form:- A relation is in fourth form if it is in the Boyce Codd Normal form (BCNF) and for all the multivalued functional dependencies of the form by $X \twoheadrightarrow Y$

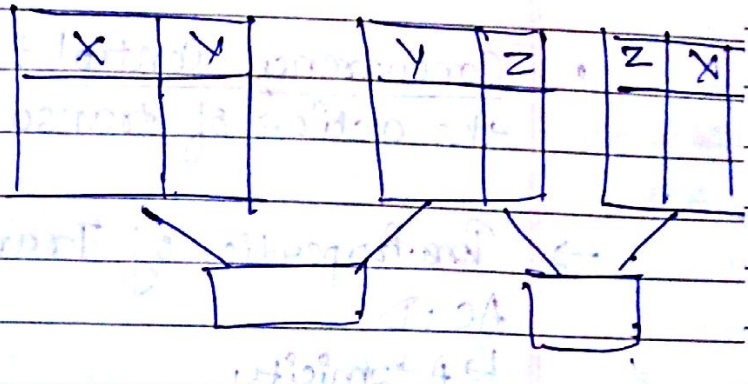
Programmer	Project	Module

Programmer	Project

Project	Module

V Fifth Normal form (Join dependency):-
Let R be a given relation upto fourth Normal form or (4NF). and it decompose into (R_1, R_2, R_3 --- R_n) The Relation R satisfy the join dependencies if and only if joining R_1 to R_n = R

X	Y	Z
X_1	Y_1	Z_1
X_2	Y_2	Z_2
X_3	Y_3	Z_3



Transaction & concurrency Control

- **Transaction**: It can be defined as a unit or any part of any program at the time of its execution. During transaction the data items can be read or updated or both

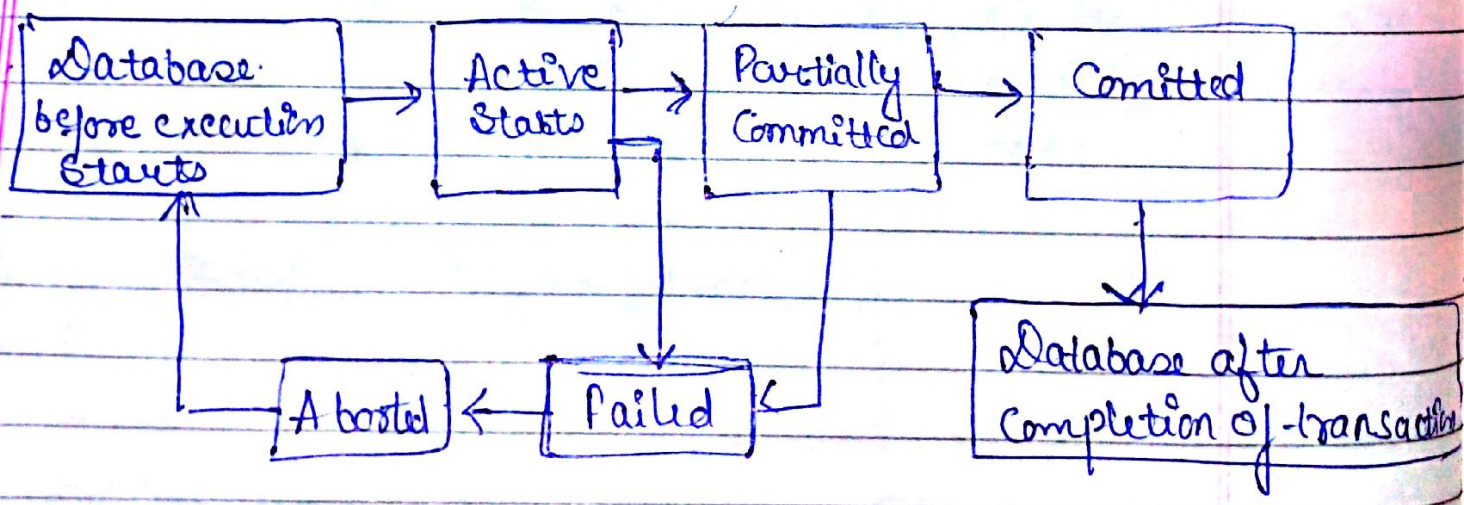
- **Concurrency Control** : It is the activity of co-ordinating the action of transactions, that operate simuntaneously

→ Tha Properties of Transactions.

ACID
- ↳ Atomicity
- ↳ Consistency
- ↳ Isolation
- ↳ Durability.

- **Durability**: Changes are made permanent to the database after Succesfull. Completion of the transaction

→ **Transaction States**:



Passion
Page No.
Date

- Serializability: Considered the set of transactions $T_1, T_2, T_3 \dots T_i$. S_1 is the state of database after they are concurrently executed and successfully completed and S_2 is the state of database after they are executed in any serial manner and successfully completed. If S_1 and S_2 are same then the database maintain the serializability.

- Concurrent Execution: If more than one transaction are executed at the same time then they are said to be executed concurrently

- Recoverability: To maintain the Atomicity of the database, undo effects of any transaction has to be performed in case of failure of that transaction. If undo effects successfuls then that database maintains the recoverability. This process is also known as role back.

→ Construct an ER diagram for Banking System

Construct an ER diagram for Hospital Management System

Hospital management :-