20/Oct/2016

UNIT: 1 * "DATABASE MANAGEMENT SYSTEM"

* "INTRODUCTION TO DATABASE MANAGEMENT SYSTEM" *

* "Def" of DBMS" *

A DBMS is a collection of programe that enables users to create and maintain a Database. DBMS is a genral purpose application s/w system that facilities the proces of defining, constructing, manipulating and staring database among various users & application.

(i) Defining a database means specifying the Datatype, structure & constraints of the data to be stored.

(ii) Constructing the Database is the proces of storing the on some storage media which is controlled by DBMS

(iii) Manipulating a Database include function to query the Database retrerive specific data, update the database & so on.

(iv) Sharing a Database means allowing multiple users & programs to access the Database simultaneously

* Advantage & disadvantage of DBMS *

(a) Restricted unauthorized access *

when multiple users share large database then a DBMS should provide some security & authorization mechanism. It can be

login password or certain privileges to d/f categories of users.

(b) Providing storage structure for efficient query processing → ✓

DBMS must provide specialized data structure to speed-up search & retrieval of the desired records. Fields like Indexes are used for these purpose. A good DBMS must be capable of executing query efficiently.

(c) ~~Providing storage structure for efficient query processing~~ → ✗

(c) Providing persistent storage for data & records →

A DBMS is used to store data & records persistently (permanent). It means the data & records are kept in there original form & will not get corrupted modification.

(d) "Inforcing integrity constraintes" → ✓

A DBMS should provide capability for defining & inforcing certain integrity & constraintes. A simple type of integrity constraints involves specifying the data type for each data item.

(e) "Controlling Redundancy" → ✓

DBMS is use to control & eliminate redundancy. If it occurs in a table

* Note *

Redundancy means storage the some data and records multiple times.

(4) "providing multiple user interfaces" →

DBMS is used by many types of users with varying levels of technical knowledge. Therefore, a DBMS should provide a variety of user interface. Both forms style interfaces & menu driven interfaces are commonly used known as GUI.

→ "Limitations of DBMS" →

Inspite of many advantages DBMS has some limitations with respect to traditional file processing systems. This are as follow:

(a) High initial investment in H/w, s/w & training.
(b) Overhead for providing security, concurrency control, recovery & integrity constraints.
(c) A DBMS is not able to operate when two or more tables are required to be used as a single table.

* Dif Data models *

(1) * Relational data models →

"Relation: Student"

Attributes

Name	Class	Session	Regd. no	Roll no	City	Teleno.
Aadhi	BCA	2015-18	1017/15	01	Dumri	10101
Rohit	BCA	2015-18	2027/15	02	Dumri	20222
Sunil	BCA	2014-17	2567/11	49	Dumr	96211
Amit	BBA	2010-13	1058/10	39	Dumr	22911
Sanya	B.COM	2011-14	9099/09	26	Buxar	40556

Tuples

The Relational data models represents the database as a collection of relation. Each relation resembles as the tables of values. When a relation is thought of as a table of values, each row in the table represents a collection of relative data values. In Relational data model terminology, a row is called as tuple. The column header is known as attributes and the table as relation.

Attributes

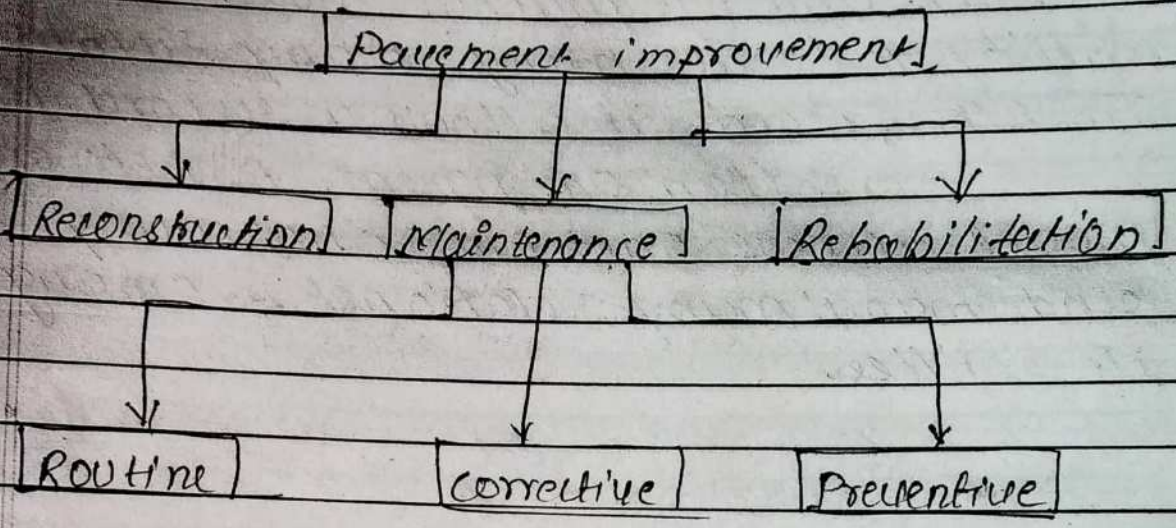
Name	Class	Session	Regd.no	Roll.no	City.	Teleno

Schema

* Schema *

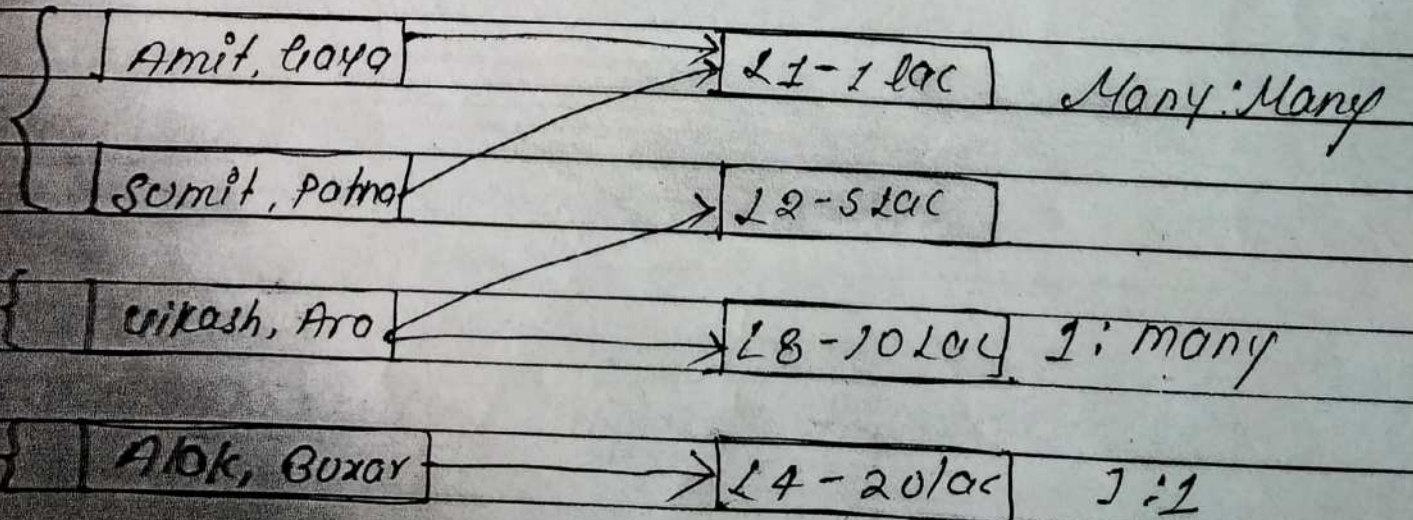
The attributes of the relation is known as schema.

(2) Hierarichial data model *



Hierarichal data model is a data model in which data is organized into a tree like structure. The data is stored as records which are connected to one another through links. A records is a collection of fields containing only one value. The child as only one parent but a child can have more than one childe. The root has no parent.

(3) "NETWORK MODEL" →



17/Nov/2016 Unit: 2 } Data model → ½c

→ ½c Concepts of Entities, Attributes, Association and Relationship → ½c

* Entities *

An Entity is a real world thing with an independent existence. It can be any object with a physical existence. e.g., Employee, college, currency, Flow etc.

* Attributes *

En entity has certain attributes. These are the particular properties of that Entity. Which is use to describe it.

Student

Name	course	subject	session	} Attributes

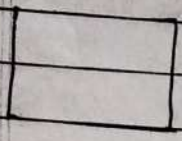
* Relationship *

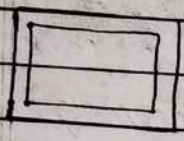
The logical association among Entities is known as Relationship.

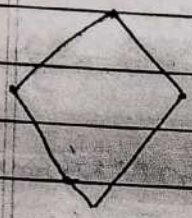
* Association *

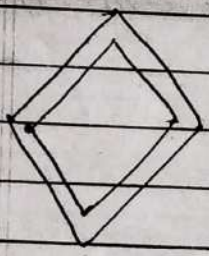
Every attributes is define by its set of values called association.

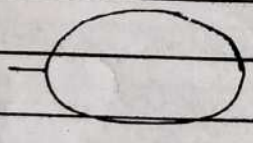
* Symbols used in ER-models and Diagram →

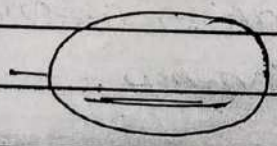
(1)  : Entity

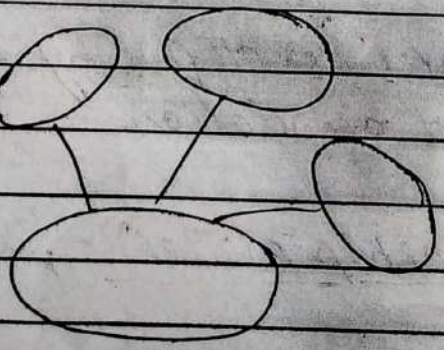
(2)  : Weak Entity

(3)  : Relationship

(4)  : Identifying Relationship

(5)  : Attribute

(6)  : Key Attributes

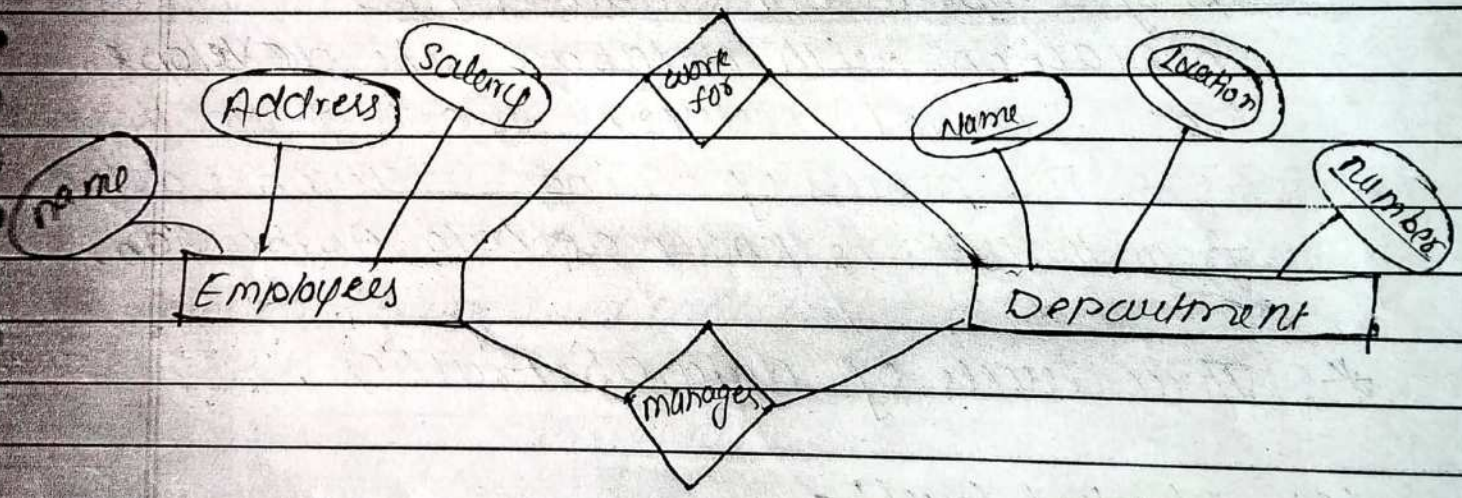
(7)  : Composite Attribute

(8) : Multivalued Attribute

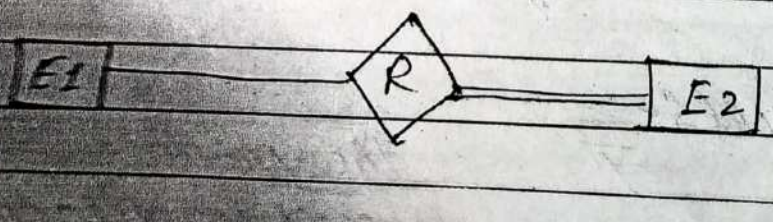
(9) : Derived Attribute.

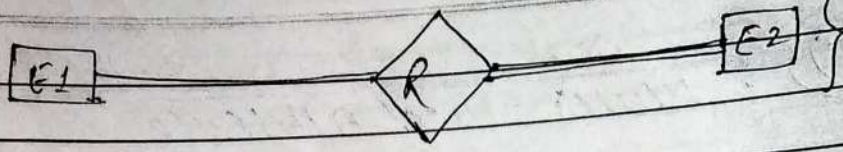
E-R model

ER-model is a widely used high level conceptual data model. The ER-model and its variations are frequently used for the conceptual design of database applications and many other database tools. The ER-model works around the real world entities & their associations among them.



{Fig: Entity Relationship Diagram}





ER-Diagram is the digramatic notation of ER-model. It is a visual representation of entities, attributes, relations, e.t.c., that describes how data is related to each other. It's notation and symbols are universal.

* Data Abstraction, Generalization, Specification

Data Abstraction

Database systems are made up of compler data structures to ease the user interaction with database, the developer hides internal irrelevant details from the user. The process of hiding irrelevants details from the users is known as Data Abstraction.

* Three levels of data abstraction:

(1) Physical level:-

These is the lowest level of data abstraction. It is use to the describe how data is actually stored in the database.

(2) Logical level:-

These is the middle level of data abstraction. It describes detail data structures.

③ View Levels -

This is the highest level of data abstraction. This level describes how the users interact with database

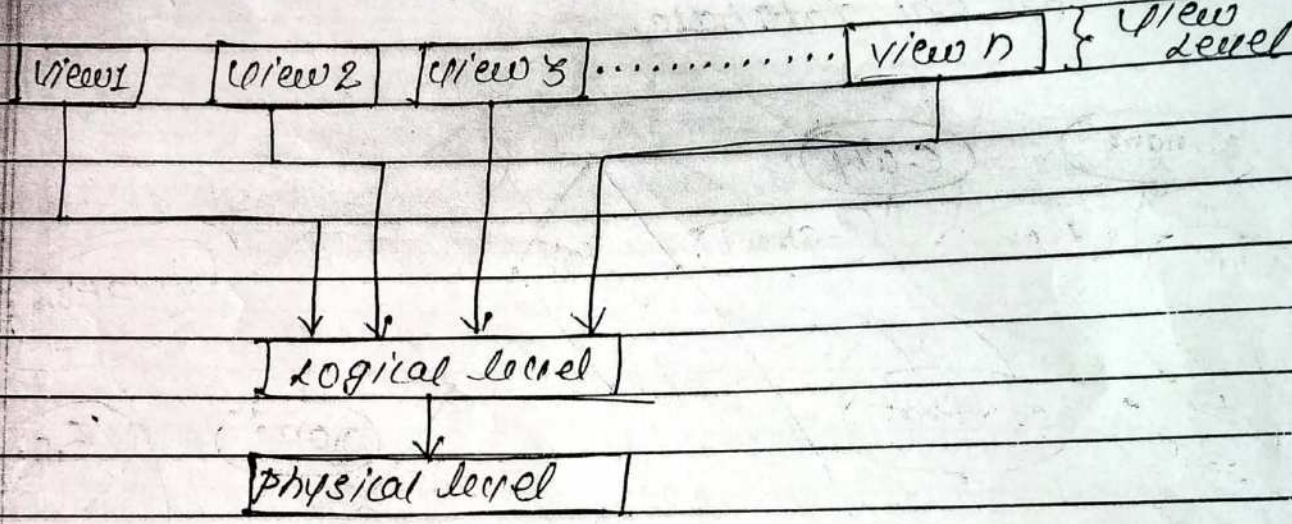
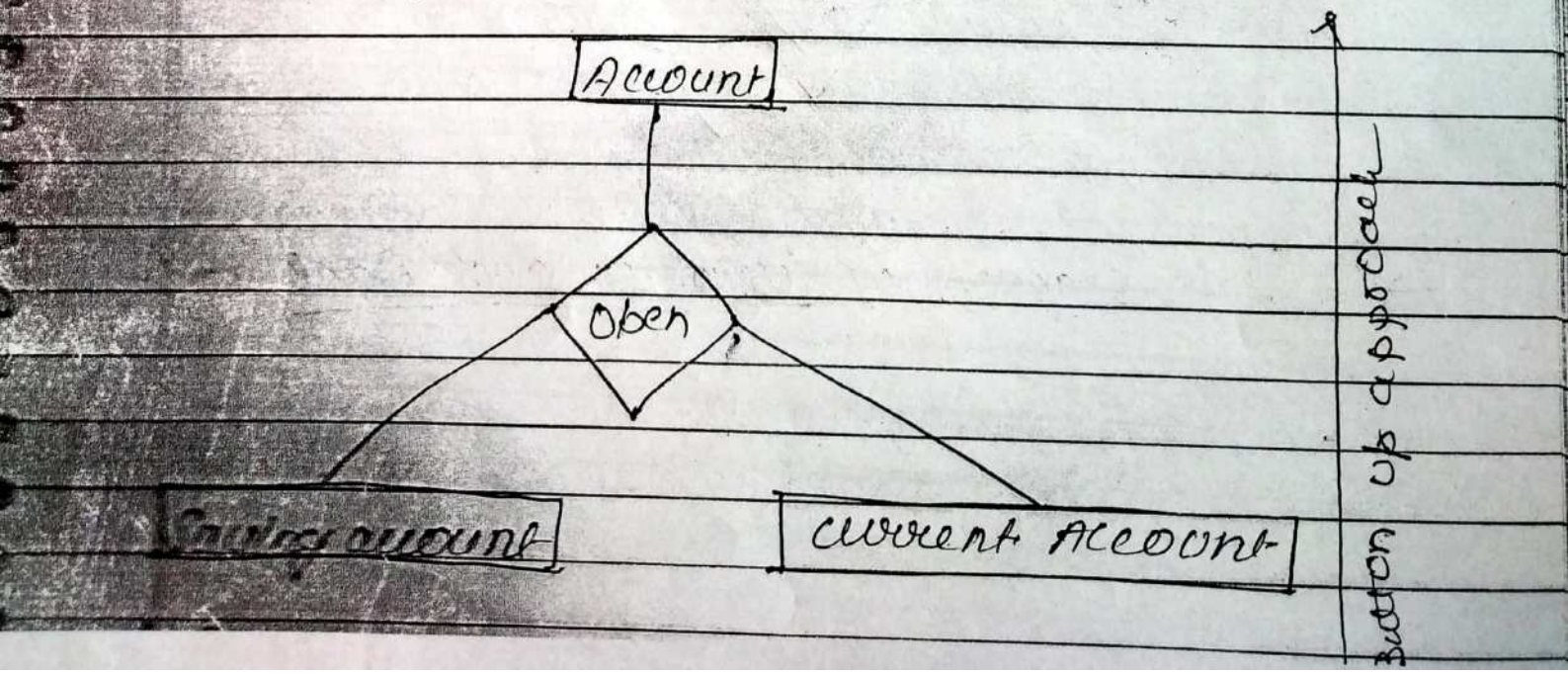


Fig: "Three levels of data abstraction

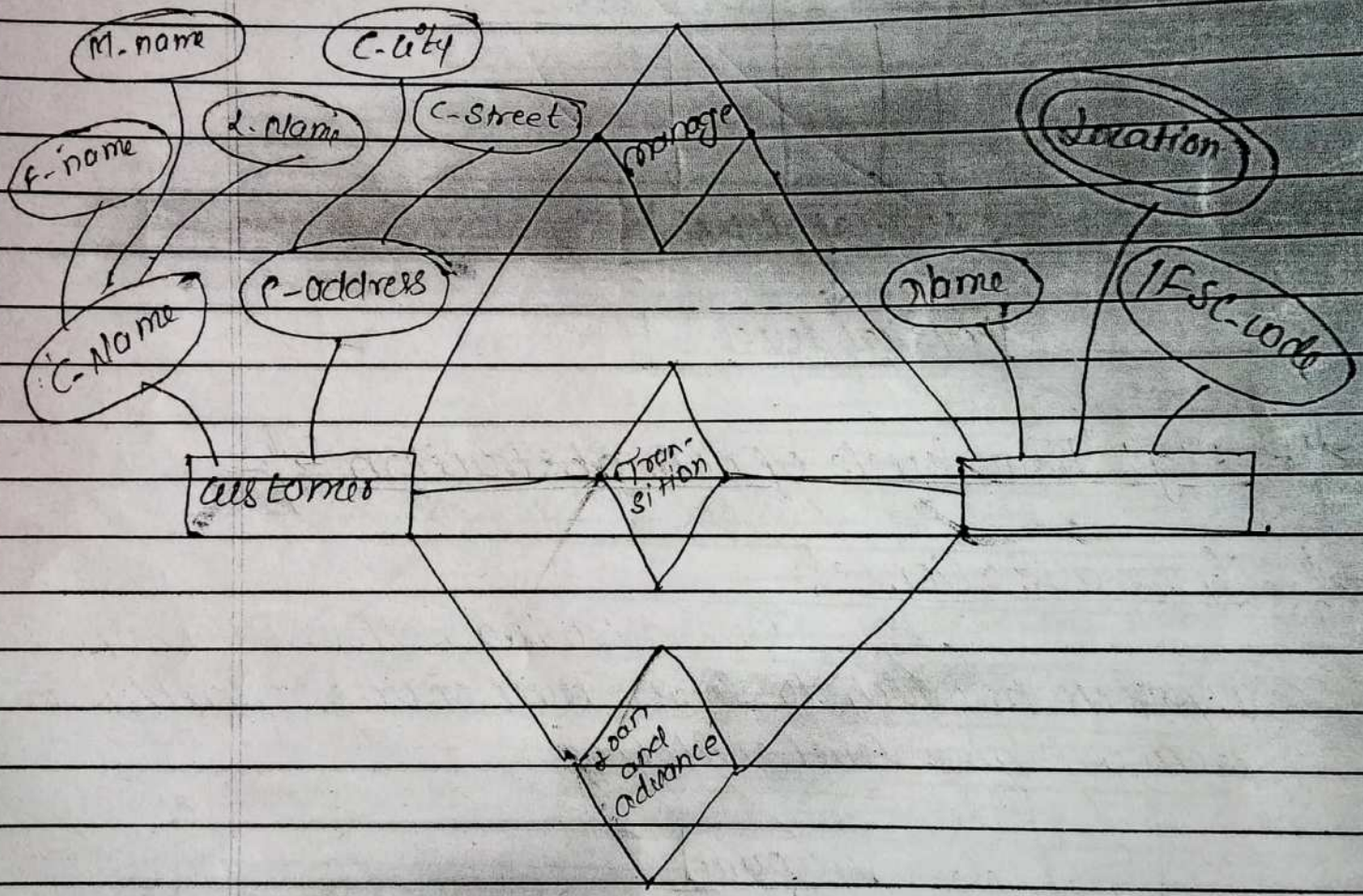
Generalization

Generalization is the bottom up approach in which to lower level entities combine to form a higher level entitities



Question → Draw a ER-Diagram of a Bank. The entities are customer & Bank.

Question → Draw an ER-Diagram of university and student database.



→ "E-R Diagram of Customer and Bank" ←

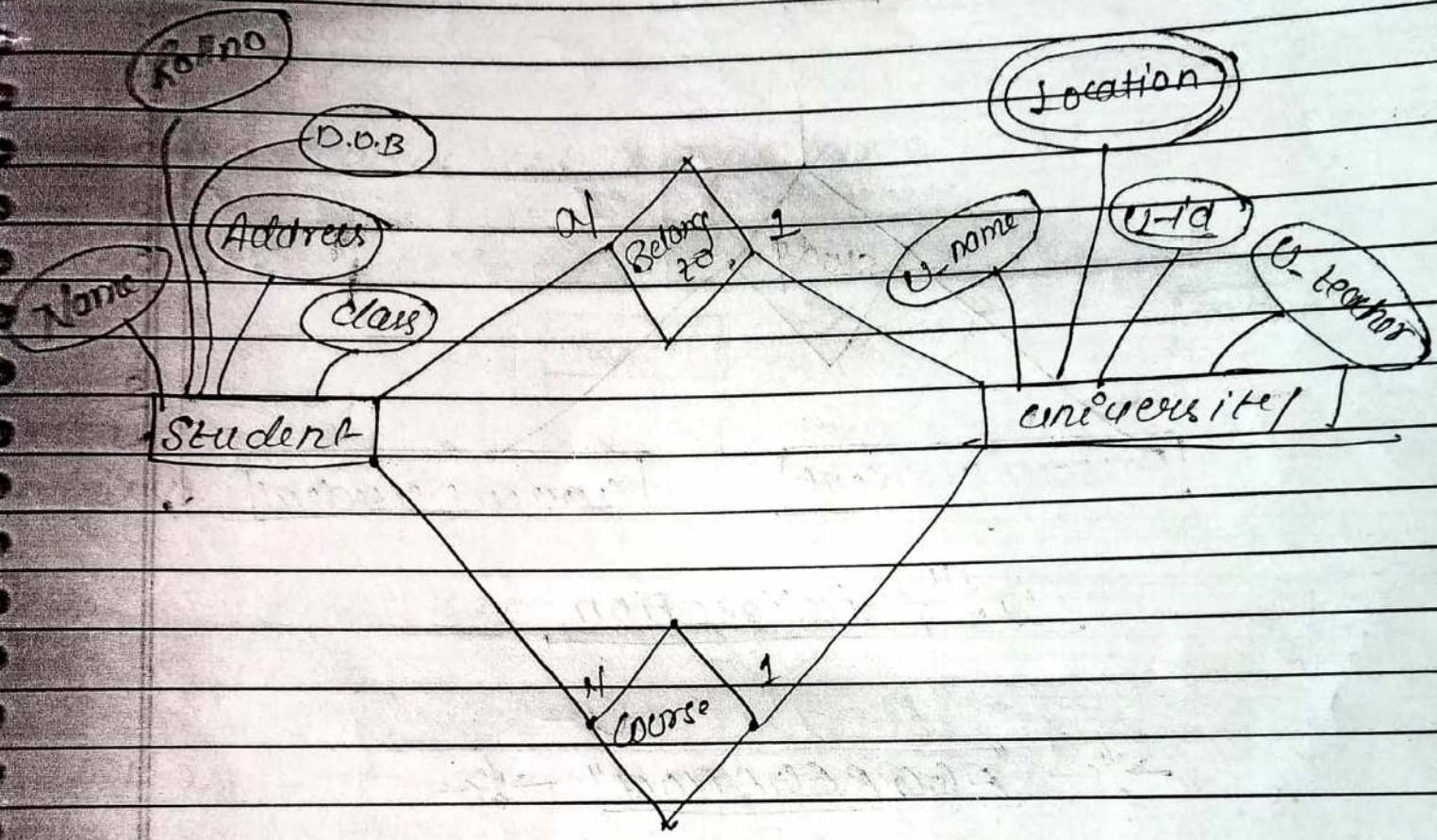


Figure: "E-R Diagram of university and student"

* Specialization *

Specialization is opposite of generalization It follows top down approach in which one higher level entity can be broken down into two lower level entity. In particular case of specialization some higher level entities may not have lower level entities set at all.

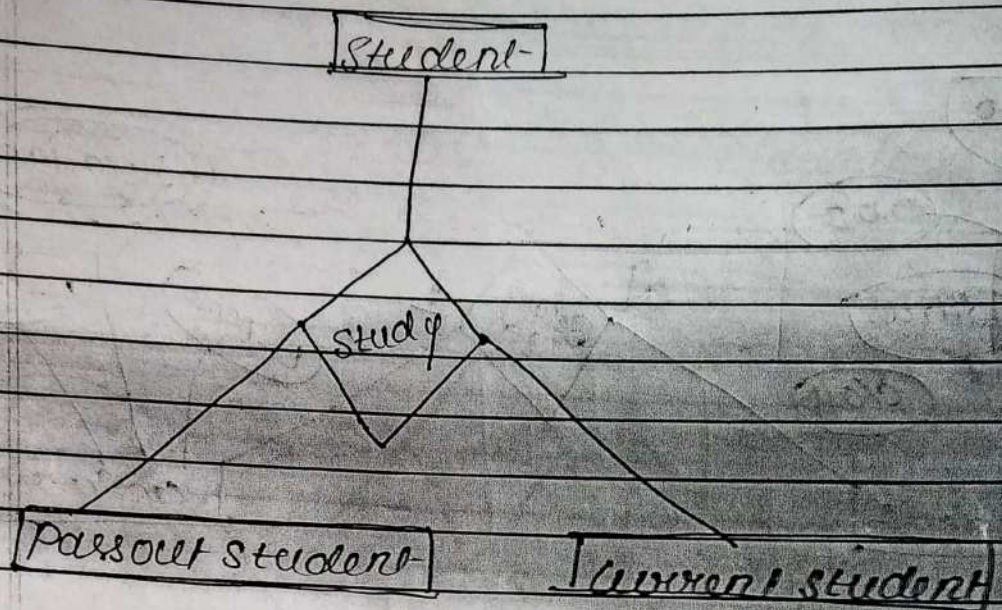


Fig: "specialization"

"AGGREGATION"

Aggregation is a process in which relationship b/w two entities is treated as a single entity

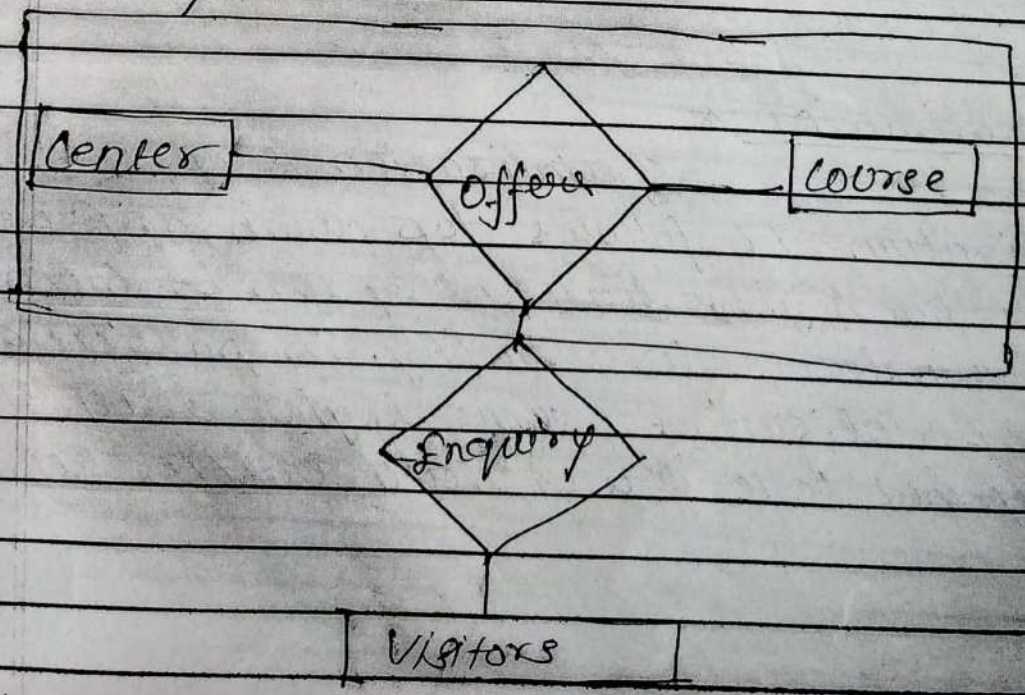


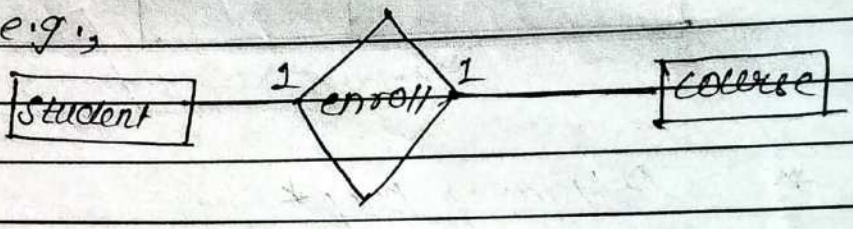
Fig: Aggregation

Relationships b/w entities

Binary relationships :-

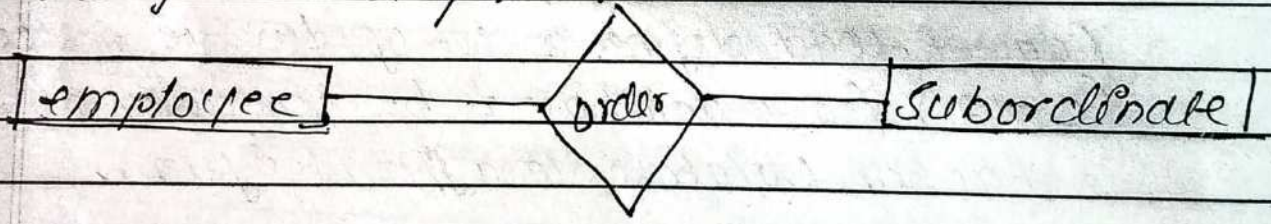
Binary Relationship exist b/w two entities. It is further divided into three types.

(i) One-to-one :- e.g.,



(ii) One-to-many :- e.g.,

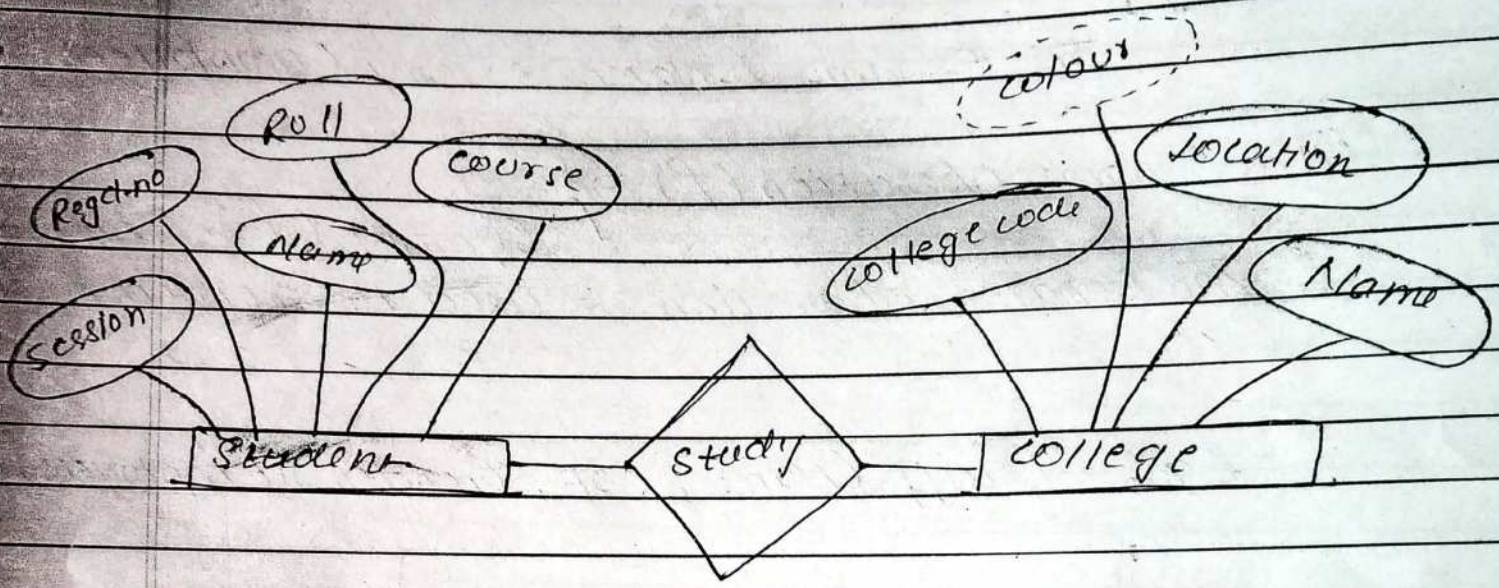
(iii) Many-to-Many :- e.g.,



Recursive Relationship

When an entity is related with itself. It is known as Recursive relationship.

Question → Draw an E-R diagram of student and college.



→ "Relational Algebra Operation" →

(i) Select operation, (σ), (ii) Greek letter small Sigma, (iii) symbol (σ) Greek letter Sigma.

(i) Select operation :-

The select operations selects tuples that satisfy a given predicate.

eg., σ amount > 1200 (Loan)

(ii) Projection (Π) →

Projection is presented by Greek letter (Π). The projection operation selects certain columns from the table and discards other columns. If we require only certain attributes of a relation we use

projection operations.

* e.g.,

Π F_name, L_Name, salary (Employee)

(iii) Rename operation (ρ) →

ρ Symbol (ρ). Rho.

The Rename operation is used to rename the relation name or attribute name.

e.g., ρStaff (Name, DOB, Gender) (Employees)

$\rho_S(B1, B2, B3 \ldots Bn)^{(R)}$

(iv) Union operation. (∪) →

In Union operation all the attributes of R1 and relation R2 are merged together. Its symbol is "∪".

e.g., R1 ∪ R2

(v) Intersection operation (∩) →

In Intersection operations those attributes which are common to both relation R1 and R2 are taken into a/c & form a new relation. Its symbol is "∩"

e.g., R1∩R2.

(vi) Difference operation ("−") → It is denoted by the symbol "−". A Relation R−S is such a relation which includes all tupules (Record) which are present in relation R but not in relation S.

(vii) Cartesion product (X) → A Relation R1XR2 and produces such a relation that has all the attributes of R1 and R2 and includes all possible combinations of tupules R1 and R2.

e.g.,

R1 (1, 3, 5)
R2 (2, 4, 6)
R1XR2 (12, 14, 16
32, 34, 36
52, 54, 56)

26/11/20

Unit-3 Relational Data Model

Student :-

Attributes or domain → ↓ ← Primary key

Student name	Reg. no	Roll. no.	Session	course	← Schema
Rohit	20155002	02	2015-18	BCA	↓
					Tuple / Row / Record

* Primary key *

Primary key is the most important attribute which helps to identify all the tuples & records uniquelly.

→ ↓ ← "codd's Rule" → ↓ ←

Edgar F. codd defines a set of twelve rules (0-12) for a database to be considered as relational Database Management System.

* Rule: 1) Information Rule → ↓ ←

Everything in database must be stored in table formate.

* Rule: 2) Guaranteed Access Rule → ↓ ←

Every single data elements is guaranteed to be accessable logically with a combination of table name & attributes. No other means can be use to access data.

* **Rule: 3} Systematic use of NULL values →** The NULL value in a database must be used systematically & in uniform manner. NULL can be interpreted as missing data, data not known or data is not applicable.

* **Rule: 4} Active Online Catlog →** The structure description of the entired database must be stored in an online catlog known as data dictionary.

* **Rule: 5} Comprehensive data sub-language →**

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation and many other forms of data operations.

* **Rule: 6} View updating Rule. →** All the views of database which can theoritically be updated must also be updated by the system.

* **Rule: 7} High-level insert, update & delete rule →**

A database must support high level insertion, updation & deletion of all records. This must not be limited to a single record (a row).

* Rule: 8) Physical data independence →
The data stored in a database must be able to access by the same manner even if the physical structure gets changed.

* Rule: 9) Logical data independence →
Logical data independence must be independent of its users (view/applications). This is one of the most difficult rule to apply & follow.

* Rule: 10) Integrity Independence →
Database must be independent of the applications that it uses. This rule makes the database independent of the front-end-application & its interface.

* Rule: 11) Distributed Independence →
This rule is regarded as the foundation stone of distributed database system. In this rule, the end user must not be able to see that data is distributed over various location user should always get the impression that the data is connected at one site only.

* Rule: 12) Non sub version Rule →
If a system has an interface that provides access to low

lower records than the interface must be able to subvert the system & by pass security & integrity constraints.

* Primary key *

A primary key is used to identify individual tuples in a relation. A primary key constraints cannot have NULL values. Having NULL values for the primary keys means that we cannot identify some tuples.

* Foreign key *

When two relations are associated in such a manner that at least one of the attribute is common among them than the primary key of first relation becomes the foreign key in second relation.

* Candidate key *

In absence of primary key, the next most suitable attribute which can help to identify the tuples in a relationship uniquely is known as candidate key.

Relational Algebra and calculus

Relational Algebra is the basic set of operations for relational models. This operations enable users to specify basic retrieval request. The algebra operations thus product new relations. A sequence of

relational algebra operations forms a relational algebra expressions whose result will also be a relation.

* The relational algebra is very important for various reasons:

(i) It provides a formal foundation for relational model operations.

(ii) It is use as a basic for implementing & optimizing queries.

(iii) Sum of its concepts are encorporated into SQL.

The relational calculus provides a high level declarative notations for specifying relational calculus is important because it has firm bases in mathematical logic.

Relational algebra operations

(1) Select operations

The select operations is use to select a subset of tuples from a relation with satisfies select condition. It is denoted by symbol "Sigma (σ)".

e.g.,

σ dno = 4 (employees)

σ sal > 30,000 (Employs)

(ii) Project operation (Projection Operation) :-

The project operation selects certain columns from the table & discards other columns. If we required on the certain attributes of a relation, we use project operation, it is denoted by "π".

(iii) Rename Operation:-

The Rename Operation can rename either the relation name or the attributes name. It is denoted by the "Rho (ρ)".

* e.g.,

$\rho S(B_1, B_2, B_3 \ldots B_n)^{(R)}$

$\rho_{Staff(name, DOB, Gender)}^{(employ)}$

$\rho_{(name, DOB, Gender)}^{(employ)}$

* e.g., of project operation *

~~Πname, to~~

$\Pi_{Fname, Lname, Sal}(employ)$

(iv) Union operation :-

In union operation all the attributes of relation 1 (R1) & attributes of relation 2 (R2) are merged together. It is denoted by the "U"

* e.g., R1 U R2.

(v) Intersection Operation :-

An Intersection operation those attributes which are common two both relations R1 & R2 are taken into a/c & form a new relation. Intersection operation is denoted by symbol "Ω".

* eg.,

$R1 \Omega R2 = \{2, 4\}$

(vi) Difference or minus operation → Ⅎ is denoted by the symbol "−". R-S is a relation which includes all tuples which are present in R but not relation S.

(vii) cartesian product → Symbol (X)

$R1 = \{a1, a2, a3\}$

$R2 = \{b1, b2, b3\}$

$R1 \times R2 = (a1, b1), (a1, b2), (a1, b3)$
$(a2\ b1), (a2\ b2), (a2\ b3)$
$(a3\ b1), (a3\ b2), (a3\ b3)$

R1 X R2 produces a relation that has the attributes of R1 & R2 & includes all possible combination of tuples R1 & R2.

SQL (Structured Query Language)

SQL is used to access database records &

information. It is independent of any database application s/w. All the database support SQL with little with modification in it syntax.

* The SQL has three parts:-

(1) DDL (Data Definition Language) → It is use to define the database structure.

(2) DML (Data manipulation language) → It is use to modify & alter the given data.

(3) DCL (Data control language) → It is use to control the database. DCL commands is are use by DBA (Database administrator).

→ SQL (Structured query language, Sequel) →

IBM develop the original version of SQL, as part of the system R project in the early 1970's. Many of the products now support SQL language. SQL has established itself as the standard relational database language.

(1) DDL (Data Definition Language) → DDL provides commands for defining relational database

Schemas, deleting relations & modifying relations.

(i) Create commands:-

* Syntax:

Create table <table name> (A1, A2, A3... An, Integrity constraints (A i));

* where, A1, A2 ... are attributes of the table.

* example →

create table customer
customer_name varchar (20),
customer_street char (30),
customer_city char (30),
Primary key (customer_name),

customer name	customer stree	customer city

→ Schema

Create commands create a Database schema. It means to create the structure of the table.

(ii) Drop command:-

* Syntax *

Drop table <table_name>

* e.g., Drop table customer.

* Drop command is use to remove the relation from database.

(iii) Alter command :-

* Syntax :-

Alter table <table_name> add A
or Alter table <table_name> Drop D

* e.g.,

Alter table customer add customer i'd.
Alter table customer drop customer i'd.

(2) DML (Data manipulation language) →

(a) Select command :- Syntax :-

select <attributes>
from <relation / Table_name>
where condition

* e.g., select customer_name, customer_street
from customer
where customer_city = patna

(b) Insert command :-

* Syntax :-

Insert into <Table_name>

* e.g., Insert into customer values ("Amit", "Boring road", "Patna").

(C) Delete command :-

* Syntax : Delete from <Table name>
where <condition>

* e.g., Delete from customer
where customer-name = Amit.

(d) Update command :-

In certain situation, we want to change a value in a tuple without changing all the values in the tuple. For this purpose, we use update command.

* Syntax :- Update <table-name>
where (condition)

* e.g., update a/c
set balance = balance * 1.05
where balance > 1000

(3) DCL (Data control language) →↓←

DCL commands are use by database administrators to keep the database safe & secure any unauthorised user. Grant & Revoke are common command use by DBA.

Unit: 4 "Normalisation"

Normalisation is a process of decomposing tables to eliminate data redundancy (Repeatation) & undesirable characteristic like Insertion, anomaly, Updation anamoly & Deletion anamoly. Normalisation is a multistep process that puts table into tabular form by removing duplicate data from the table.

* There are four normal forms:

(a) First normal form (1NF).
(b) Second normal form (2NF).
(c) Third normal form (3NF).
(d) fourth Normal form (BCNF).

(a) First normal form (1NF) →↓←

Rule: "Each attribute in the table must have atomic (single) values."

emp.id	emp-name	emp add	emp-mob
101	Rohit	Dumri	012345
102	Kunwar	Dumri	0162850

Employee (Table 1NF)

(b) Second Normal form (2NF) →↓←

Rule: A table is said to be in 2NF. If it holds —

(i) The table is in 1NF. No non prime attribute is

* Note:
Non-prime attributes :- An attribute that is not part of any candidate key is known as non-prime attribute.

* e.g.,

Teacher id	subject	Age
101	Math	16
102	Physics	35
103	Chemistry	60

Candidate key : Teacher_id, subject
non-prime attributes - Age.

* This table is in 1NF because each attribute has atomic values. However, it is not in 2NF because non-prime attribute teacher "age" is dependent on Teacher-Id. These violates the rule that says "No non-prime rule that says". No non prime attribute is dependent on the proper subset of any candidate key.

* To make the table follow 2NF we break it into two tables.

Table 1 "2NF"

Teacher-id	Teacher age
112	54
114	48

Teacher-id	sub
112	math
114	eng.

(c) 3NF :-

Rule: (i) Table must be in 2NF.
(ii) Transitive functional dependency of non-prime attributes on any super key should be removed.

* Super key *

Super key is a set of one or more attributes to uniquely identify a tuple in a relation.

Transitive functional Dependency

Transitive functional dependency can only occurs in a relation of three or more attributes. These dependency helps us to normalise the database in the 3NF. A functional dependency is set to be transitive if it is formed by two functional dependencies.

* e.g., $x \to z$ is a transitive FD. If the following three FD holds true —

$x \to y$

y does not $\to x$

$y \to z$

Book (x	Author (y)	Author Age
Let us C	Yashwant Kanitkar	35
N/w	Frozen	60
Database	Korth	62

(i) $x \rightarrow y$

Book → Author

(If we know the Book, we know the author

(ii) y does not → x

(If we know the author, we doesn't know the book.

{Author} does not {BOOK}.

(iii) $y \rightarrow z$

{Author} → {Author Age}

* Let us consider student detail table.

(i)

Stu-i'd	Stu-Name	Street	City	State	Pin

* In these table, Student-i'd is primary key but street, city & state depend on PIN number. The dependency b/w PIN number & other fields is called as Transitive dependency. Hence, for 3NF we need to move the street, city & state to new table with PIN number as primary key.

(ii)

Stu-i'd	Stu-Name	PIN
101	Rohit	802120
102	Nitin	802120

"New Student Detail"

(iii)

PIN	STREET	CITY	STATE
802120	Dumri	Buxar	Bihar
802120	Dumri	Buxar	Bihar
802120	Dumri	Buxar	Bihar

→ Address detail →

* Advantage of Transitive functional dependency

(i) Amount of data Publically is removed
(ii) Data integrity is achieved.

(iv) BCNF (Boyce codd normal form) →

It is also known as 3.5NF. It is slightly stronger version of 3NF. BCNF was developed in 1974 by Raymond F. Boyce and Edgar F. Codd. If a relational schema is in BCNF than all the redundancy base on functional dependency is remove. A relational schema arc is in BCNF if & only if $X \to Y$ & X is a superkey.

UNIT-5

09/12

Database security →

Database security is an important issue to keep the database informations correct & prevent it from any unwanted changes.

→ security policies :-

(a) Identification
(b) Authorization. →
(c) Authentication. → Means to make sure that the person accessing the database is who he claims to be. Many authentication system such as fingerprint, Retina scanner or biometrics are used to make sure unauthorised person don't use pass

(a) Identification →

Identification is the process of identifying the user. If the user is a legitimate person than he is allowed as a genuin person. In database to identify the user, user given login and password.

(b) Authorization →

Authorization is the process. which is manage by the database manager or database administrators (DBA). The DBA obtains information about the current authenticated. The DBA then authorizes the authenticated user that which database operations the user can perform or access.

"Encryption and Decryption"

* Encryption *

Database Encryption is the process of converting data within a database in plane text formate into a sipher text formate by means of is suitable algorithm. Encryption of a database is a costly affair and its requires more storage space than the original plan text.

To Encrypt a database following aspects are taken into consideration. such as:

(i) Determine the need for Encryption.
(ii) Determine which data needs to be Encrypted.
(iii) Determine which algorithm will be best-suited for Encryption
(iv) There are numerous Encryption algorithm & it is to be selected intelligent.
(v) Database Encryption can be provided at the file level.

DESCRIPTION

Database Decryption is the process of converting meaning less Sipher text into the original text. Using keys, generated by the Encryption Algorithm. Modern database are Encrypted through very complex algorithm & without

knowing the description keys. It is impossible to obtain original plane text.

Integrity rules and constraints

Data integrity is insured in database system by a series of Integrity constraints.

* Some of the Integrity constraints are:

(a) Entity Integrity:-

It gives the concept of primary key. Entity Integrity is an Integrity rule. which states that every table must have a primary key & the column (s) chosen from the primary key must be able to identify tuples uniquelly in a relation.

(b) Referential Integrity:-

It gives the concept of foreign key. The Referential Integrity rule states that - Any foreign key values can only be in one of two states. The usual state is that the foreign key refers to a primary key of some table in the database. A foreign key value can be NULL.

Auditing

Auditing :

Database Auditing is the process of monitoring & recording of selected user-database actions. It can be based on individual actions. Auditing is used for the purpose of -

* Enable future accountibility for the correct actions taken in a particular schema, table affecting specific contents.
* Investigate doubtful activate.
* Monitor & gather data about specific database activating
* Detect problems within authorization or access control implementing.
* Auditing is imp. to keep the database safe & secure over a long period of time & serve the user.

Database Recovery mechanism

It is one of the most important part of database system. A database must have a robust, safe & secure database recovery mechanism with the help of recovery mechanism that lost data is able to retrieved.

* There are many techniques to recover the lost

commonly used technique are:-
Roll back and check points

⇒ Roll back:-

It is also known as transaction roll back. If a transaction fails for whatever be the reason. It may be necessary to Roll back a transaction. If any data values had been changed by the transaction & written to the database. They must be restored to the previous value. The log entries are used to restore the values of data item that must be roll back.

⇒ Check Points:-

A check pointing is a technique in which record is written into the log file periodically. when the system writes out to the database on the disk. As a consequence of this all transactions that have taken place have their entries in the log file. All the updates will be record in the database during this check pointing technique. when a transaction failure occurs then the immediate log file is searched. If the lost data is found then the system recover otherwise the next log file is searched to found the lost data. The process goes on till the lost data is recovered.

17/Dec/2016 Unit : 6

Feature Trends in DBMS

- Concept of Distributed database system.
- knowledge based database system.
- Expert database system.
- object database system.

"Distributed Database System"

A Distributed Database management system is a centeralized application that manages a distributed database as if it where all stored on the same computer. The DBMS synchronises all the data periodically & in cases where multiple users must access the same data, ensures that updates & deletes, perform on the data at one location will be automatically reflected in the data stored on the servies such that all the users will be able to using in the real time. All the operations taken place will be updated in the real time.

knowledge based database system

It is used to store complex structured &

constructed information use by computer system. A knowledge based system consist of a knowledge base that represents facts about the real world & interference engine that can reason that about those facts & uses rules an other forms of logic to deduces new facts. A knowledge based system provides knowledge in the form of document & media that could be leveragride by humans. In knowledge based system, knowledge is the primary driving force which is used by computers & other automatic machines to generates & share new facts & knowledges.

Expert Database system

An expert system is a computer system that emulates the decision making ability of human experts. Expert system are design to solve complex problems by reasoning about knowledge, represented mainly as if then rules rather than convential procedural code. The first expert system was created in the 1970's & than in the 1980's. The expert system is mainly made up of knowledge base & "interface engine". Expert system is used to inforecasting & predictions, medical evience that diagnouses planning and designing e.t.c.

object database system

(OODBMS object oriented database management system)

An OODBMS is the result of combining object oriented programming principals. Object Oriented programming concept. such as - Encapsulation, polymorphism & Inheritance are enforced as well as database management concepts ACID properties (Automicity, consistency, Isolation and Durability). which leads to system integrity, support for a adhoc Query language & secondary storage management system. which allows for managing very large amount of data. An OODBMS is a full stell object oriented development environment as well as database management system. An OODBMS is better suited to Handle complex & enterrelated data than RDBMS.