Page No.
Date

Database Management System

- Database is a collection of inter related data which helps in efficient retrieval and manipulation of data and organizes data in the form of tables, views, schemas etc.

- DDL - Data Definition Language
Create, Alter, Drop, Truncate, Rename
- DML - Data Manipulation Language
Insert, Update, Delete

- DBMS over File System
Reduced redundancy of data — Easy of data access
Reduced inconsistency of data — Backup and Recovery of data

- 3 Tier Architecture
External level - User can view the data also
Conceptual level - User can view the data but only the structure
Physical level - User only also knows where the data is stored.

Data Independence - Change of data at one level should not affect other level
Physical Data Independence and Conceptual Data Independence

- Phases of Database Design
1) Data Collection 2) Conceptual Design - ER Model
3) Logical Design - Table Creation 4) Physical Design - Data Entry

- Data Abstraction - Hide irrelevant details
Physical - Lowest level of abstraction. It even tells how the data is stored in memory
Logical - Information stored in database can be viewed
View - Only a particular part of database can be viewed

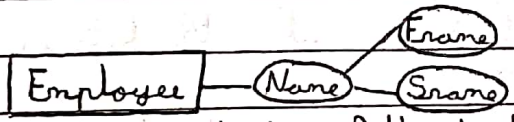
- Database Objects → Table, View, Index, Sequence, Synonym

- Disadvantages of DBMS
1) Cost of hardware and software 2) Training cost
3) Data conversion 4) Difficult to maintain integrity due to many users

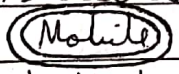
- ER Diagram → Provides a conceptual design of the database

1) Entity - Object with real world existence. Eg - Student
Each entity has its attributes (properties)
Strong Entity - Has its own distinct primary key. ▭
Weak Entity - Cannot form distinct primary key from its attributes. It depends on strong entity for primary key. ▭▭
Entity Set - Collection of all entities of same type

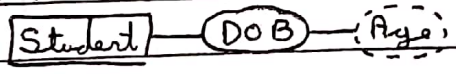
2) Attribute → Properties which describe entity. ⬭
Key Attribute - Primary Key (Roll No)
Composite Attribute - One attribute composed of many other attributes



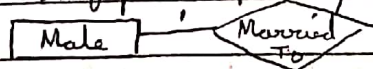
Multivalued Attribute - Attribute having more than 1 value



Derived Attribute - Attribute derived from other attributes

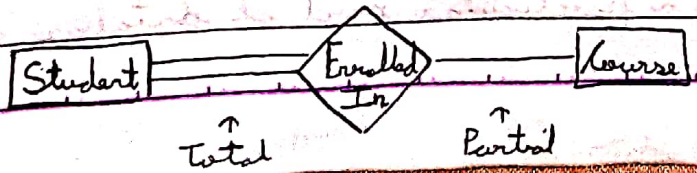


3) Relationship - Association among several entities
Degree of relationship - Number of participating entities. Eg - Unary
Cardinality - One to One



- One to Many (Teacher - Students)
- Many to Many (Student - Course)

Participation - Total - Each entity must participate
Partial - Entity may or may not participate



- Enhanced ER Diagram (EER)

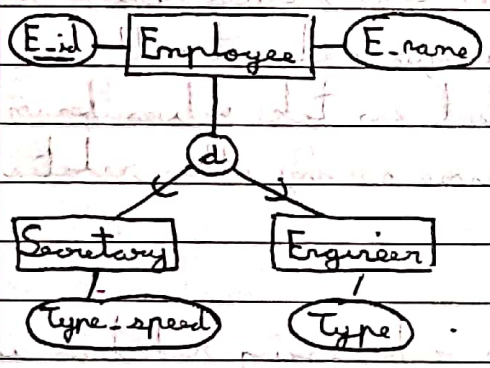
1) Super Class - An entity having its sub groupings is called as super class

2) Sub Class - An entity which is sub group of super class is called sub class

Sub class inherits all the attributes and relationships of super class

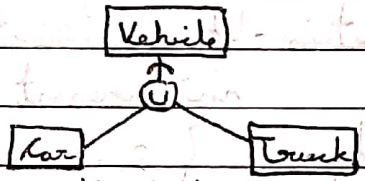
3) Specialization - Top Down approach of sub class / super class relationship

Generalization - Bottom Up approach of subclass / superclass relationship



Union - If an entity is a union of 2 other entities (both of them being independent) then its a Union. We denote it as ⓤ instead of ⓓ

If we consider car and truck as → independent entities (ie does not inherit attributes of vehicle) then Vehicle can be considered as a union of car and truck



- In One to Many Relationship, we need minimum of 2 tables
- In Many to Many Relationship, we need minimum of 3 tables
- In One to One Relationship with total participation, we require 1 table and in case of no total participation, we need 2 min tables

- Aggregation - Converting a part of relation into an entity

- Impedance Mismatch - Problems which occur due to difference between database model and programming language model
Eg - Datatype mismatch

- Relational Model - Represents data in the form of table
Attributes form the columns
Domain of Attribute - Possible values an attribute can take
Each row represents a tuple
Values of some attributes in a tuple may be unknown and is represented by NULL
Relational Model should stisfy 13 Codd Rules
Degree - Number of attributes
Cardinality - Number of tuples

Constraints - Set of rules for performing insertion, deletion, manipulation
1) Domain Constraint - An attribute can take values inside the domain
2) Key Constraint - A key should be unique and not null
3) Referential Integrity - When one attribute can take values from other attributes of same or any other relation

Anomalies
1) Insertion - We can't insert a row in referencing relation if it is not present in referenced relation
2) Deletion - } Same as insertion
3) Updation - }

Delete/Update Cascade - It will delete the tuples from referencing relation if the value used by referencing attribute is deleted/updated from referenced relation (update)

- Keys
1) Candidate Key - Minimal set of attributes which uniquely identifies a tuple
More than one possible
2) Super Key - Set of attributes which uniquely identifies a tuple
Adding one or more attributes to CK gives SK. CK is a SK
3) Primary Key - One CK chosen among all is called primary key
4) Alternate Key - CK other than PK, is called alternate key
5) Foreign Key - Used to reference one attribute to another

- Schema Design Strategies
1) Top Down Approach
2) Bottom Up Approach
3) Inside Out Approach
4) Mixed Approach

- Schema Integration
1) Identifying Conflicts - Name Conflicts, Type Conflicts, Domain Conflicts, Constraint Conflict
2) Modifying Views to conform one another
3) Merging of views and restructuring

- Relational Algebra - Foundation for databases and SQL
1) Projection (π) - Used to select column data. By default, it removes duplicates
2) Selection (σ) - Used to select required tuples
3) Rename (ρ) - Rename an attribute
4) Set Operations - Union ($\cup$), Intersection ($\cap$), Set Difference ($-$)
5) Cartesian Product ($\times$)
6) Join ($\bowtie$) - $\bowtie_{condition}$, Left Outer Join ($\bowtie$), Right Outer Join ($\bowtie$), Full Outer Join ($\bowtie$)

- Normalization - Process of organising data to reduce or eliminate data redundancy.
Problems - Size of database increases
Functional Dependency - $A \rightarrow B$. For all instances of particular value of A, there is same value of B. If $A \rightarrow B$, $B \not\rightarrow A$
1) Trivial - $ABC \rightarrow AB$, $ABC \rightarrow A$, $ABC \rightarrow ABC$
2) Non Trivial - Id $\rightarrow$ Name
3) Semi Trivial - $AB \rightarrow BC$

- First Normal Form - No multi valued attributes in the table
- Second Normal Form - No partial dependency and is 1NF
Partial Dependency - Proper Subset of any candidate key gives non prime attribute

- Third Normal Form - No transitive dependency and is 2NF
Transitive Dependency - Non prime attribute derives another non prime attribute
- Boyce Codd Normal Form (BCNF) - $X \Rightarrow Y$, X should be a super key
Prime attribute cannot be derived

BCNF is free from redundancy.
A relation with only 2 attributes is always in BCNF
If all attributes are prime attributes, relation is in 3NF
A relation with singleton candidate keys is always in 2NF

$\alpha \rightarrow \beta$
If all α is Super Key, BCNF
If α is Super Key or β is a prime attribute, 3NF
If α is a proper subset of CK and β is non prime, 2NF fails

- Lossless Decomposition - If a relation is decomposed, it should give the original relation back when combined
$attr(R_1) \cup attr(R_2) \Rightarrow attr(R)$
$attr(R_1) \cap attr(R_2) \not\Rightarrow null$
$attr(R_1) \cap attr(R_2) \Rightarrow attr(R_1) / attr(R_2)$

- Dependency Preserving - If a relation is decomposed, it should give the original functional dependencies back when combined

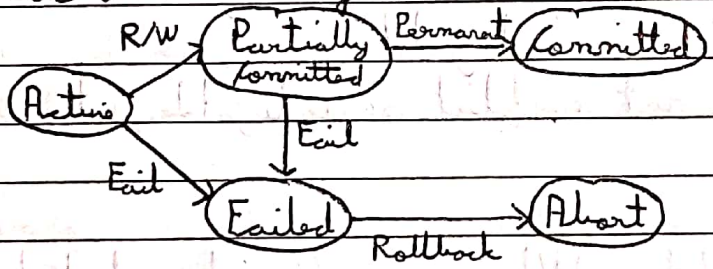
- Fourth Normal Form - No multivalued dependency and in BCNF
- Fifth Normal Form - Lossless Join

- Denormalization - Process in which we add redundant data to one or more table to avoid costly joins in relational database. It is ~~of~~ applied after the process of normalization.
However updates and inserts are expensive and difficult

- Transaction - Set of logically related operations. It includes
1) Read - Reads value from database and stores it in buffer in main memory
2) Write - Writes value back to database from buffer
3) Commit - Changes are made permanent
4) Rollback - If a transaction is not able to execute all operations successfully, all changes done by transaction are undone

- ACID Properties
1) Atomicity - Either all of them should be executed or none
2) Consistency - Transactions should not leave the database in inconsistent state
3) Isolation - Results should not be visible to others before commit
4) Durability - Once committed, changes made should be permanent

- Transaction State Diagram



- Locking protocols are used as a means of concurrency control
Data Structure - Hash Table

Atomicity is managed by Transaction Management Component
Isolation is managed by Concurrency Control Component
Durability is managed by Recovery Management Component
A transaction should be performed only if ACID properties are satisfied

- Concurrency - Multiple transaction at one time
1) Waiting Time Decreases
2) Response Time Decreases
3) Resource Utilization Increases
4) Efficiency Increases

- Dirty Read - If a transaction reads a value of another uncommitted transaction, it is called dirty read

- Schedule - Order of performing a transaction
Serial Schedule - Non contact switching. Always consistent
Non Serial Schedule - Contact switching between transaction

- Conflicting Transaction - If 2 transaction operate on same data and one of the operation is write operation, then the transaction is said to be conflicting

- Conflict Serializability - When a non serial schedule can be proved equivalent to ~~one~~ serial schedule, by swapping non conflicting instructions. A conflict serializable schedule is consistent

If a schedule is conflict serializable, then it is consistent.
Also it is view serializable
However if its not conflict serializable, it is not necessary inconsistent

- View Serializability - When a non serial schedule can be proved view equivalent to any all possible serial schedule.

First check for conflict serializability, if it is conflict serializable, it is view serializable.
If it is not conflict serializable, check for blank writes, if no blind writes, then it is not view serializable.
If blind writes exist, we need to prove view equivalent by checking if the initial reads, final writes and intermediate reads are performed by same transaction

Create Index index-name On Persons (Last Name)
If we use Unique Index, duplicates are not allowed

Page No.
Date

- Concurrency Control Techniques - Using and creating protocols such that conflict serializability, view serializability and consistency is maintained

1) Time Stamping Protocol - Each transaction can be assigned a time stamp for entry into the system to avoid conflicts
2) Lock Based Protocol - Transaction should acquire lock on the data, so that no other transaction can use the same data.

- Deadlock - A system is in a deadlock state when every transaction in the set is waiting for another transaction in the set.

Starvation - If a particular transaction is not able to be executed or does not get a chance to be executed

Deadlock Prevention

1) To ensure no hold and wait, each transaction locks all the data before it begins execution.
2) To ensure no cycle wait, impose an order of data acquire

- Indexing - Method to optimize the performance of the database by providing faster retrieval of data
It is a data structure technique to quickly locate and access data
It consists of search key and reference pointer
The size of index file is much smaller than the database, hence faster access

(Unsorted) Dense Index - There exists an index record for every data file (value)
(Sorted) Sparse Index - Index record exists for only few items in data file (Not every record has an index)
Multi Level Indexing - Create index table for an index table. Index file is always sorted. Implemented with B/B+ tree
B - Tree - Search, Insert, Delete - $\log n$
B+ Tree can store more entries as compared to B Tree
In B+ tree, all keys are present in the leaf nodes, hence insertion and deletion is easier and searching is faster

1) Primary Indexing - Main File should be sorted. Primary key is used as anchor attribute. Sparse Indexing.
Number of index = Number of block
Number of access = $\log_2 N + 1$

2) Clustered Indexing - Main file should be sorted. Secondary key (Non key) is used as anchor attribute. Sparse and Dense Indexing. There will be an index to each value (unique)

3) Secondary Indexing - Main file unsorted. Dense Indexing. Each value has to be indexed.
Number of index = Number of entries

- SQL - Structured Query Language
Query Language to create, maintain and retrieve data from relational databases

- View is a virtual table based on result set of SQL statement. It prevents exposure of entire data, just the data required is displayed
Create View view_name As Select Customer_name from Customers where Country = "Brazil"

- Trigger is a stored procedure which is automatically invoked when special event in database occurs

- Stored procedure - It is type of function so the code can be reused again and again
Create Procedure SelectAllCustomers As Select * from Customers Go
Exec SelectAllCustomers

- Cursor is used to retrieve data one row at a time unlike SQL commands which operate on all rows at one time. Used when records needs to be updated in a row by row manner.