Data Model Vs. Schema

A data model is a higher-level, abstract representation of data and its relationships.. The schema, is a specific implementation that translates the concepts of the data model into a structure suitable for a particular database management system.

Data Model: Data models are collections of concepts for describing data.

A data model is a conceptual representation of the data and its relationships within an organization or system.

It defines the structure of data, the entities, their attributes, and the relationships between them.

Purpose of a data model is to provide a high-level, abstract view of the data to help stakeholders understand and communicate how data is organized and used within the organization.

Schema: A schema is a particular collection of data in tables using a given data model.

A schema, in the context of databases, is a blueprint that defines how data is organized and stored in a specific database management system (DBMS). It specifies the tables, columns, data types, constraints, and relationships that make up the physical or logical structure of a database.

The primary purpose of a schema is to provide a detailed, technical description of how data is stored and accessed within a particular database system.

Entity: Real world object distinguishable from other objects. An entity is described using a set of attributes

Relationship Types

Multiple Relationship: A multiple relationship, entity in one set is associated with multiple entities in another set. In other words, it represents a many-to-many relationship between two entity sets.

Binary Relationship: A binary relationship between two distinct entity sets. It's the most common type of relationship in database modeling, representing associations between two different types of entities.

Unary Relationship: A unary relationship where a single entity set is related to itself. Represents a connection between instances of the same entity type.

Ternary Relationship: A ternary relationship involves three entity sets and represents a more complex association between them. It often occurs when there are relationships or interactions between three different entities.

Index Files: Stored permanently on the hard disk

Indexing: Disadvantage to indexing: Increase in the cost of storage. The records to be included in an index should not be a decision variable when building indexes.

File Indexes: Good for searching Data

Used when data retrieved by queries is a small percentage of the available data in the table

Question to ask, Which record(s) should be part of an index

Hash Based Indexes: Good for Equality Statements

Indexes: Indexes are data structures used in databases and file systems to speed up data retrieval. They provide a way to quickly look up records or data based on specific key values.

Indexes contain a mapping of key values to the location of corresponding data. When you search for data, the system uses the index to find the location and retrieve the data efficiently.

Common types of indexes include B-tree, Hash, and Bitmap indexes, each suitable for different use cases.

Sorted Files: Sorted files are data storage structures where records or data are stored in a specific order, usually based on a key field.

The primary advantage of sorted files is that they enable efficient searching and range queries. Binary search can be used to quickly locate specific records within the sorted file.

Sorted files are commonly used in scenarios where data retrieval performance is critical, such as in database systems.

Heap Files:

Advantages

Efficient for bulking data, for relatively small relations as indexing overheads are avoided, when queries that need to fetch large proportion of stored records, best for random ordered data

Retrieving files in random order

Bulk loading data

Disadvantages

Not efficient for selective queries or for sorting, time-consuming as you need to load all the pages in the data

Heap files are data storage structures where records are stored in an arbitrary order without any specific sorting. Records are simply appended to the file as they arrive.

Heap files are easy to implement but may result in slower data retrieval since no inherent order or index exists for quick lookup.

Used for simple, write-intensive applications or as a temp. storage

Randomly Accessed Files: Randomly accessed files refer to data storage systems where you can directly access any record without traversing the entire data structure. This is achieved through direct addressing or calculated offsets.

Random access files are commonly used in low-level I/O operations, such as in file systems or storage devices, where you need to access specific blocks or sectors on a storage medium.

Buffer Manager: Buffer manager stages pages from external storage to main memory buff pool via buffer manager

File and index layers make calls to the buffer manager.

Buffer manager loads bit by bit into the page

Loads pages from hard disk to memory

B+ Trees: Leaf nodes in B+ trees store the actual data entries, while non-leaf nodes are used for routing and indexing. Non-leaf nodes guide the search process by indicating which child node to follow,y leading to the leaf node containing the desired data. B+ trees effective for efficient data retrieval and indexing in database system

DBMS

Advantages

Data independence: Application/Developer doesn't need to know data specifics

Efficient Data Access: Index the data, efficiently retrieve the data

Data integrity and Security: Securely store data

Data Administrator: One location that the data is stored, have a person who owns the database to manage the data

Concurrent Access and Crash Recovery: Allow for multiple data uses at one time

Reduce App Development Time: Application developer just needs to focus on the logic rather than the efficiency/security

Disadvantages

Complexity and Overhead, Cost, Learning Curve, Performance Bottlenecks, Scalability Challenges, Single Point of Failure

Designing a database

Phase 1: Requirement Analysis

What users (apps) expects from the database

Phase 2: Conceptual Database Design

Build Entity Relationship Diagram

Phase 3: Logical Database Design

Convert Entity Relationship diagram to logical database

Levels of Abstraction

Many views, single conceptual (logical) schema and physical schema

Views describe how users see the data

Conceptual schema defines logical structure

Physical schema describes the files and indexes used

Schemas are defined using DDL, data is modified/queried using DML

Views on top of conceptual schema, which goes down to the physical schema connected to the database.

1. External Schema (View)

The view level is the highest level of abstraction and measures how data is presented and accessed by end users and applications.

It defines various "views" or subsets of the data that are tailored to the specific needs of different user groups.

2. Conceptual Schema

The logical level is an intermediate level of abstraction that focuses on the logical structure of the data without concern for the physical implementation.It defines the overall structure of the database, including tables, columns, relationships, and constraints.

Data models, such as Entity-Relationship Diagrams (ERDs) and schema diagrams, are used to represent the logical level of the database.

3. Physical Schema

Lowest level of abstraction in a DBMS.It deals with the actual storage and retrieval of data on the physical storage devices (e.g., hard disks).

TDBMS manages the details of data storage, such as file structures, indexing methods, and data access paths.

ACID

Atomic: All or Nothing (All read/write actions are executed or aborted)

Consistency: No violation of integrity constraints

Isolation: Concurrent Changes invisible to end-users

Isolation implies serializability

Durability: Once committed, it persists (written to DB), never lost

Decreasing order of speed for processing data:

CPU Registers, L1 Cache, L2 Cache, L3 Cache, Main Memory (RAM), Hard Disk

Commands

Query Language

Procedural/Imperative Language: Instructs system to perform a sequence of operators to get the end result. Relational Algebra

Non-Procedural/Declarative Language: Tells what data is to be retrieved but does not tell the system how to retrieve it. Relational Calculus/SQL

Relational Algebra

Selection: p is predicate (Precondition)

Projection: a is attributes, projection will filter out the other attributes

Union: List the student information for all comp sci or EE majors students

Difference: CSE Majors - EE Majors

Intersection:Lists the student information for those students who are double majoring in computer science and electrical engineering. Returns common tuples between r and s

Cartesian Product: List all possible combinations of computer science professors teaching computer science courses. Combines everything, join has a predicate (condition) for filtering.

Selection: List the student information for seniors majoring in computer science, filters certain tuples in a relation based on a given criteria.

Join: List the professor information along with the information on the courses that the professor is currently teaching

Cartesian = A*B rows= Total, X but not Y = Difference, X and Y = Intersection, X or Y = Union

Select

The SELECT keyword is used to retrieve data from a database. It specifies the columns you want to retrieve and the table from which you want to retrieve the data. A basic SQL statement often starts with SELECT, followed by a list of column names or expressions.

From

The FROM keyword is used to specify the table or tables from which you want to retrieve data. The source of the data you are querying. In a basic SQL statement, you typically specify the table after the FROM keyword.

Where

The WHERE keyword is used to filter the data you retrieve based on a specific condition or criteria. It allows you to narrow down the results to only those rows that meet the specified condition. In a basic SQL statement, the WHERE clause follows the FROM clause and includes the filtering condition.

Distinct

Ensures that the result set of a query contains only unique values in a specified column or a combination of columns

Relational DB to ER Diagram

Entities in an Entity-Relationship (ER) diagram are typically converted to tables when creating a relational database. Each entity in the ER diagram corresponds to a table in the relational database, and the attributes of the entity become the columns of the table. This process involves mapping the conceptual representation in the ER diagram to the logical structure of tables in a relational database.

Here's how the conversion typically works:

Entities to Tables:

Each entity in the ER diagram becomes a table in the relational database. The name of the entity becomes the name of the table.

Attributes to Columns:

The attributes of the entity are mapped to columns within the table. Each attribute is converted into a column with the same name in the table.

Primary Key:

The primary key of the table is determined based on the primary key attribute(s) of the entity in the ER diagram.

Foreign Keys:

If there are relationships between entities in the ER diagram, foreign keys are introduced in the corresponding tables to establish referential integrity and link related data.

Cardinality and Multiplicity:

The cardinality and multiplicity constraints of relationships in the ER diagram are translated into foreign key constraints and indexing strategies in the database schema

Clustered Vs. Unclustered Index

Understanding the workload for clustered vs. unclustered

- For each query in the workload: What relations does it access and which attributes are retrieved during selections/joins, and how selective are they conditions likely to be

A database may only have one clustered index

Data Storage and Indexing

Cluster vs. Unclustered: In order of data is the same as or close to, order of data entries, then this called a clustered index. A file can be clustered on at most one search key

Clustered Index

Examples of clustered indexes

B+ tree index, Group By query, Equality queries, Structure and Organization:

Clustered Index:

Ideal for tables that are frequently queried for range-based operations or when you need to retrieve the entire table efficiently.

Good for columns that have a high degree of uniqueness (e.g., primary keys).

In a relational database, a table can have only one clustered index.

The data rows in the table are physically reordered or "clustered" based on the order of the clustered index key.

The leaf level of the clustered index structure contains the actual data rows themselves.

When you create a clustered index, you are essentially reorganizing the data on disk to match the order of the index key.

Non-Clustered Index (Unclustered Index):

Suitable for columns that are frequently used in WHERE clauses for filtering or when you need to look up individual rows quickly.

Effective for columns with low cardinality (few distinct values) where the data is not well-suited for clustering.

In summary, the main difference between clustered and non-clustered indexes is in how they organize and store data. Clustered indexes determine the physical order of the data in the table, whereas non-clustered indexes provide an additional data structure for efficient data retrieval without affecting the physical storage order. The choice of which index to use depends on the specific query patterns and performance requirements for a given database table.

A table can have multiple non-clustered indexes.

The data rows in the table are not physically reordered by the non-clustered index.

A non-clustered index structure has a separate set of pointers to the actual data rows.

The leaf level of a non-clustered index contains a copy of the indexed columns along with pointers to the corresponding data rows.

Recovery Process

Recovery Algorithm: Undo, Redo, Analysis

The log contains information used by the recovery process to restore the consistency of a system.

- Transaction ID
- Type of operation
- Items accessed by the transaction to perform the action
- Old state value
- New state value

Concurrency

Interleaves transactions

Shared Lock - Read

Exclusive Lock - Write

Lock Manager manages lock-based concurrency control for transaction

Deadlock

Cycle of transaction waiting for locks to be released by each other.

Deadlock detection

Create a waits-for graph:

- Nodes are transactions
- There is an edge from Ti to Tj if Ti is waiting for Tj to release a lock

Periodically check for cycles in the waits-for graph

Strict Two-Phase Locking is from the two phases, lock acquisition and lock release

Databases enter a recovery state when there is a failure of any kind

What is a characteristic of big data? **Data comes from different sources in different formats.**
What does data independence mean for a DBMS? **Not required to know how the data in the app is stored.**
What is the relationship for the number of physical schemas to logical schemas in a DBMS? **1:1**
What is the most widely used data model today? **Relational Model**
Which of the following is true about data schemas? **May be multiple external schemas for a given database.**
Given 2 tuple sets (A w/ 30 and B w/ 5) what is the maximum number of tuples in the union? **15**
Mathematical operator is used to display information about students majoring in CS but not EE? **Difference**
Which statement about the selection operator is true? **It filters certain tuples in a relation based on given criteria**
Consider a relation, Student, with the following attributes: Name, ID, number, Major, Grad Year, Consider a query list the names and id numbers of students majoring in CS which operator is used: **Projection**
Consider 2 sets, A with 10 and B with 5. Whats the max number of intersection of tuples? **5**
What is the join operation for? **Join 2 relations from the db**
Which statement about the selection operator is correct? **It displays certain tuples based on given criteria, The selection operator can work on a relation even with a single attribute.**
Which of the following is a correctly formatted db query? **What is the name of the student with ID number 2357?**
What is the role of a relationship in a db? **It connects 2 entities**
What is required in order to use the cartesian product operator on 2 relations?
What operation must be completed before a join? **Cartesian**
What connects two entities in a db?
What must be listed after a select query? Attributes
What must be listed after the FROM command in a SQL query? Tables
Whjat must be listed after the WHERE command in a SQL query? Conditions
Which SQL statement displays name, ID, number, and major of business students?
SELECT Name ID_Number, Major FROM Student WHERE School='Business'
Where is the database stored in a computer? Hard Disk
What is the correct order of processing speed of major units in a computer from fastest to slowest? CPU, Cache, memory, hard disk
Why is the processing speed of a traditional computer hard disk lower than a modern SSD? Because hard disk is a mechanical device.
What is the software component in a computer that loads pages from hard disk into memory? Buffer Manager
Which of the following database operations can be achieved by using hash-based indexes? To retrieve student data where age to equal to 20
If a database table has m rows and n columns, the maximum number of clustered indexes for this table? 1
What is the difference between a data model and schema?
Think of the three levels of abstraction: Which description of Conceptual Schema is accurate?Conceptual Schema defines the logical structure that is used for the data.
What are advantages of using a Database Management Systems (DBMS)? *Select all that apply.Data integrity and security,Data independence,Concurrent Access and crash recovery*
Consider these three phrases A, B, and C:
A. Conceptual Database Design (Build ER diagram)
B. Requirement Analysis
C. Logical Database Design (Convert ER diagram to R-DB schema)
B-A-C
What are advantages of using Heap Files for storage?
For fetching large proportion of stored records for a query
For handling relatively small relations
For bulk loading of data
Which option is the *best* to use while trying to create a database that is based on retrieving a range of records that are ordered by some condition?
Sorted files
When creating a Relational DB from an ER diagram, which conversions take place?
Entities become tables
Which of the following is created by the FROM clause, when multiple tables are listed after the FROM clause
Cartesian Product
Can you list two types of query that are significantly benefited by using a B+ tree index in DB system?
Range query and Group-By query
Which type of query would be benefited by hash-based index?
Equality query
In the course we have discussed the "ACID" properties of Transactions. What does the "D" in "ACID" refer to?
Durability: Updates made by committed transactions should persist in DB system.
In which scenario is the Isolation property followed?
A currently executing transaction that does not permit other transactions' access until its commitment (locking mechanism)
Logging is one of the most popular system recovering techniques. It contains several types of information to support the recovery process. Which set contains four of these types?
Transaction identifier, operation type, old value, data item
What is the difference between a data model and schema?
A data model is a collection of concepts for describing data while a schema is a description of a particular collection of data.

$T1: A = A + 100, B = A - 100$

$T2: A = A * 1.06, B = B * 1.06$

Assuming T1 arrives first, what will be the final values of A and B if the two transactions are processed using a serial schedule?

◉ A=636, B=530

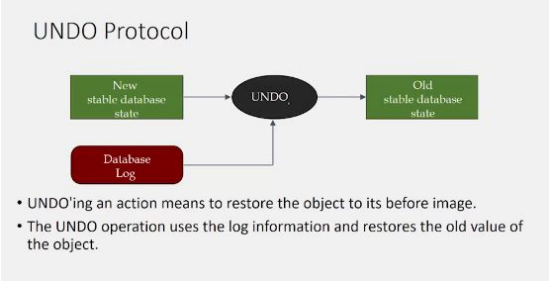
$T1: A = A + 100, B = A - 100$

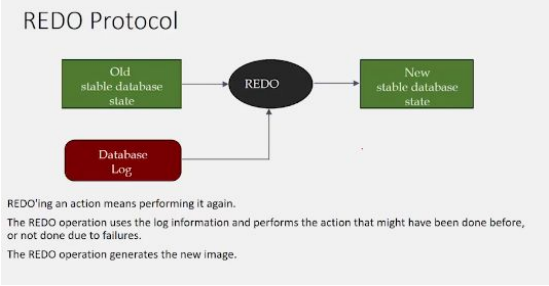
$T2: A = A * 1.06, B = B * 1.06$

Assuming T1 arrives first, what will be the final values of A and B if the database management system interleaves transactions such that the transactions are interleaved after every part of the transaction?

For example, first A in T1 is executed, then A in T2 is executed, then B in T1 is executed, and finally B in T2 is executed.

◉ A=636, B=568.16





Redo: Committed before the crash
Undo: Already Perform

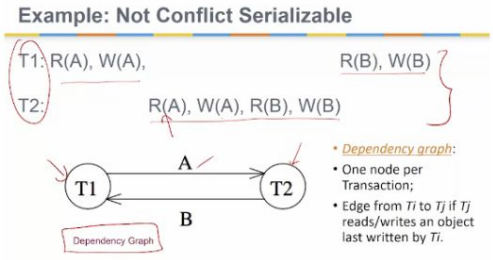
Write-ahead Log Protocol.
- If a system crashes before a transaction is committed, then all the operations must be undone. Only need the before images (Undo)
- Once a transaction is committed, some of its actions might have to be redone. Need the after images (Redo)

WAL protocol:
- Before a stable database is updated, the **undo** portion of the log should be written to the stable log.
When a transaction commits, the **redo** portion of the log must be written to stable log prior to the updating of the stable db

Database needs to figure out what transaction failed using the Aries recovery algorithm
Analysis: Scan the log forward to identify all Acts that were active, and all dirty pages in the buffer pool at the time of the crash.

Is this schedule conflict serializable? Explain why.

T1	R(A) W(A)			R(C) W(C)
T2			R(B) W(B)	
T3		R(C) W(C) R(B) W(B)		



If you build the graph and there's no cycle then the schedule is conflict serializable which is good, else it is not conflict serializable which is bad and may lead to inconsistencies.

Is there a deadlock in this schedule? Explain why.

T1	S(A), R(A)	S(B)			
T2		X(B), W(B)		X(C)	
T3			S(C), R(C)		X(A)
T4				X(B)	

Yes, a cycle is produced, produces a waits-for graph