CSCI 585: Database Systems

Midterm Study Guide

Test Details:

- Friday October 19th, 2018
- 17:00 - 18:00
- SGM 124

Major Topics:

- ER Diagrams
- SQL Queries
- Normalization
- Transaction Management
- Optimization
- Distributed Databases
- Business Intelligence

Sources of Information

- CSCI 585 Lecture Notes, Olivera Grujic [Lecture]
- Database Systems, Coronel, Morris [DBS]

ER Diagrams

Sources

Definitions

1. Weak Entity
    - An entity that cannot be uniquely identified by its own attributes
    - Must use a foreign key in conjunction with its own attributes to create a primary key

2. Weak (Non-Identifying) Relationship
    - A relationship between two entities where neither entity relies on the other's primary key
    - Entity is existence independent of other identities
    - PK of child does not contain PK component of parent
3. Entities
4. Relationships
5. Attributes
6. Keys
7. Hierarchy
8. Mandatory vs. Optional
9. One vs Many
10. Strong vs. Weak

Concepts

Relationships

- Double crossed single line - One
- 01 crossed single line - Optional One
- Specify max and min if we are given them
- Many to Many relationships should be converted to One-to-Many-to-One, where the many represents the A-B Connection Table

---

SQL Queries

Sources

Definitions

1. Inner Joins

- Join two tables on a column
- End result is the overlap of all rows included in both tables

2. Outer Joins

    - Join two tables on a column
    - End result is the overlap of all rows included in both tables and the rest of the tables including nulls

3. Update

    - Update an attribute on a table
    - Ex: UPDATE PRODUCT // Table
      SET PROD_QUANT = PROD_QUANT + 1
      WHERE PROD_CODE = 'ABC'

Concepts

Normalization

Sources

- Lecture 9/20
- DBS Chapter 6

Definitions

1. Normalization

    - Evaluating and correcting table structures to minimize data redundancies
    - Reduce data anomalies
    - Assign attributes to tables based on determination
    - Used for designing a new database structure
    - Improves existing data structure and creates an appropriate database design

2. Denormalization

    - Produce a lower form of normalization to improve performance and increase data redundancy

3. First Normal Form

    - Table Format
    - No repeating groups
    - Primary Keys identified

4. Second Normal Form

    - 1NF
    - No partial dependencies

5. Third Normal Form

    - 2NF
    - No transitive dependencies

6. Functional Dependence

    - Attribute A determines attribute B if all of the rows in the table that agree in value for attribute A also agree in value for attribute B

7. Partial Dependency

    - Functional dependence in which the determinant is a subset of the primary key.
    - Assumption - one candidate key

8. Transitive Dependency

    - An attribute functionally depends on another non-key attribute

9. Determinant

    - Any attribute whose value determines the other values within a row

Concepts

Process of Normalization

- Objective is to ensure that each table conforms to the concept of well-formed relations
    - Each table represents a single subject
    - No data item will be unnecessarily stored in more than one table
    - All non-prime attributes are dependent on the primary key
    - Each table is void of insertion, update and deletion anomalies

First Normal Form (1NF)

- Double PK relationships should connect both PKs before hitting dependencies
- Table Format
- No Repeating groups
- Primary Keys identified Steps:
    - Eliminate repeating groups
    - Identify primary key
    - Identify all dependencies

Second Normal Form (2NF)

- 1NF
- No partial dependencies Steps:
    - Make new tables to eliminate partial dependencies
    - Reassign corresponding dependent attributes

Third Normal Form (3NF)

- 2NF
- No transitive dependencies Steps:
    - Make new tables to eliminate transitive dependencies
    - Reassign corresponding dependent attributes

Dependency Diagrams

- Depicts all dependencies found within the given table structure
- Helps give an overview of all relationships

Transaction Management

Sources

- Lecture 9/25
- DBS Chapter 10

Definitions

1. Transaction

    - A logical unit of work that must be entirely completed or aborted
    - Consists of:
    - SELECT statement
    - Series of related UPDATE statements

    - Series of INSERT statements
    - Combination of SELECT, UPDATE and INSERT statements
    - Must maintain a consistent database state, i.e. all data integrity constraints must be satisfied
    - Formed by two or more database requests

2. Transaction Log

    - Maintained by the DBMS to keep track of all transactions that update the database
    - Used by the DBMS to recover from:
    - ROLLBACK
    - System Failure
    - Program's Abnormal Termination

3. Scheduler

    - Establishes the order in which the operations are executed within concurrent transactions. Interleaves the execution of database operations to ensure serializability and isolation of transactions

4. Deadlocks

    - Occur when two transactions wait indefinitely for each other to unlock data

5. Transaction Schema

6. Locking/Non-locking

7. Writing to Log

Concepts

ACIDS Properties

- ACIDS properties are a set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, etc.
- Atomicity, Consistency, Isolation, Durability, Serializability

Transaction Managment with SQL

- In SQL, the COMMIT and ROLLBACK statements provide support for transactions
- Transactions sequence will continue until:
    - COMMIT statement is reached

    - ROLLBACK statement is reached
    - End of program is reached
    - Program abnormally terminates

Concurrency Control

- Coordination of the execution of simultaneous transactions in a multi-user database
- Problems:
    - Lost Update:
      Occurs when the same data element is updated in two transactions but one of the updates is lost
    - Uncommitted Data:
      Occurs when two transactions are executed concurrently, and the first one is rolled back, but the second one accesses uncommitted data
    - Inconsistent Retrievals:
      Occurs when a transaction accesses some data before and after one or more transactions finish working with that data

Concurrency Control with Locking Methods

- Facilitate isolation of data items used in concurrently executing transactions
- Locks guarantee exclusive use of a data item to a current transaction
- Pessimistic locking assume that conflicts between transactions is likely
- Lock manager is responsible for assigning and policing the locks used by transactions

Locks are Granular:

- Database Locks
- Table Locks
- Page Locks
- Row Locks
- Field Locks

Types of Locks:

- Binary Locks are either locked or unlocked
- Exclusive locks are reserved for the transaction that locked the object
- Shared locks allow concurrent transactions to have read access on a common lock

Problems with Locks:

- Resulting transaction schedule might not be serializable, resulting in deadlocks

Two-Phase Locking

Method of acquiring locks that guarantees serializability but does not prevent deadlocks

Phases:

- Growing Phase: transaction acquires all required locks without unlocking any data
- Shrinking Phase: transaction releases all locks and cannot obtain any new lock

Rules:

- Two transactions cannot have conflicting locks
- No unlock operation can precede a lock operation
- No data are affected until all locks are obtained

Concurrency Control with Timestamping

- Assigns global, unique timestamp to each transaction
  - Produces explicit order in which transactions are submitted to DBMS

Properties

- Uniqueness: no equal time stamp values exist
- Monotonicity: time stamp values always increase

Disadvantages

- Each value stored in the database requires two additional stamp fields
- Increased memory needs
- Increased demand on system resources

Concurrency Control with Optimistic Methods

- Assumes that majority of database operations do not conflict
- Does not require locking or time stamping techniques
- Transaction is executed without restrictions until it is committed

Phases

1. Read
   - Read database
   - Executes the needed computations
   - Makes the updates to a private copy of the database values

2. Validation
    - Transaction is validated to ensure that the changes made will not affect the integrity and the consistency of the database
3. Write
    - Changes are permanently applied to the database

Database Recovery Management

- Restore database from a given state to a previously consistent state
- Recovery transactions are based on the atomic transaction property
    - All portions of a transaction must be treated as a single logical unit of work
    - If a transaction operation cannot be completed
        - Transaction must be aborted
        - Changes to database must be rolled back

Ensuring Effectiveness of Transaction Recovery

- Write ahead log protocol
    - Ensure that transaction logs are always written before the data are updated
- Redundant Transaction Logs
- Buffers
    - Temporary storage areas in primary memory
- Checkpoints
    - Allows DBMS to write all its updated buffers in memory to disk

Deferred Write Recovery Technique

- When only transaction log is updated
- Identify the last check point in the transaction log
    i. If transaction was committed before the last checkpoint, do nothing
    ii. If transaction was committed after the last checkpoint, transaction log is used to redo the transaction
    iii. If transaction had a ROLLBACK operation after the last checkpoint, do nothing

Write-through Recovery Technique

- When database is immediately updated by transaction operations during transaction's execution
- Identify the last check point in the transaction log
    i. If transaction was committed before the last checkpoint, do nothing

ii. If transaction was committed after the last checkpoint, transaction must be redone
iii. If transaction had a ROLLBACK operation after the last checkpoint, transaction log is used to ROLLBACK the operations

---

Optimization

Sources

- Lecture 9/27
- DBS Chapter 11

Definitions

Concepts

DBMS Architecture

- Data Files
  - store data in predefined increments
- Table Space
  - group of data files
- Data Cache
  - shared, reserved memory area
  - faster than datafiles
- SQL Cache
  - stores most recently executed SQL statements
- I/O Request
  - low-level data access operation that reads or writes data to/from a device
  - main focus of performance-tuning activities

Query Optimization Modes

1. Selection of the optimum order to achieve the fastest execution time
2. Selection of the sites to be accessed to minimize communication costs

Types of Operation Modes

1. Automatic Query Optimization
   - DBMS finds the most cost-effective access path without user intervention
2. Manual Query Optimization
   - Requires that the optimization be selected and scheduled by the end user

Types of Timing Modes

1. Static Query Optimization
    - Best optimization strategy is selected when the query is compiled by the DBMS
2. Dynamic Query Optimization
    - Access strategy is dynamically determined by the DBMS at runtime

Types of Information Modes

1. Statistically Based Query Optimization Algorithm
    - Statistics are used by the DBMS to determine the best access strategy
2. Rule-Based Query Optimization Algorithm
    - Based on a set of user-defined rules to determine best access strategy

Types of Optimizer Choices

1. Rule-Based Optimizer
    - Use preset rules and points to determine the best approach to execute a query
2. Cost-Based Optimizer
    - Use algorithms based on statistics about objects being accessed to determine the best approach to execute a query

Query Processing

Parsing

- DBMS parses the SQL query and chooses the most efficient access plan
- Query is broken down into smaller units, then transformed to a more efficient version
- Query optimizer analyzes query and finds most efficient way to access data
- Access plans translate SQL query into I/O operations, and stores plan in cache

Execution

- DBMS executes the SQL query using the chosen execution plan
- All I/O operations are executed

Fetching

- DBMS fetches the data and sends the result set back to the client

Bottlenecks

- Delay introduced in the processing of an I/O operation may be caused by the CPU, RAM, Hard disk, Network, Application

Indexing

- Helps speed up data access
- Facilitates searching, sorting, aggregating, join
- Ordered sets of values that contain index keys and pointers
- More efficient than a full table scan

Data Structures

- Hash Indexes
- B-Tree Indexes
- Bitmap Indexes

SQL Performance Tuning

Index Selectivity

- Measure of likelihood that an index will be used in query processing
- Indexes are used when a subset of rows from a large table is to be selected based on a given condition
- Indexes cannot always be used to improve performance
- A function-based index can be used to index values for a function

Conditional Expressions

- Use simple columns or literals as operands
- Numeric field comparisons are faster than character, date and NULL comparisons
- Equality comparisons are faster than inequality comparisons
- Transform conditional expressions to use literals
- Write equality conditions first when using multiple conditional expressions
- When using multiple AND conditions, write the condition most likely to be false first
- When using multiple OR conditions, write the condition most likely to be true first
- Avoid the use of NOT

DBMS Performance Tuning

- Manage DBMS processes in primary memory and the structures in physical storage
- DBMS performance tuning at server end focusses on setting parameters for Data cache, sql cache, sort cache, optimizer modes
- In memory database can be used to store large portions of the database in primary storage
- Use RAID for performance improvement and fault tolerance
- Minimize disk contention
- Put high-usage tables in their own table spaces

- Use denormalized tables
- Store computed and aggregate attributes in tables
- Use index organized tables

Distributed Databases

Sources

- Lecture 10/2
- DBS Chapter 12

Defintions

1. Distributed Processing
    - Database's logical processing is shared among two or more physically independent sites via network
2. Distributed Database
    - Stores logically related database over two or more physically independent sites via computer network
3. Database Fragments
    - Database composed of many parts in distributed database system

Concepts

Single-Site Processing, Single-Site Data Systems

- Processing is done on a single host computer
- Data stored on host computer's local disk
- Processing restricted on end user's side
- DBMS is accessed by dumb terminals

Multiple-Site Processing, Multiple-Site Data Systems

- Multiple processes run on different computers sharing a single data repository
- Require Network file server running conventional applications
    - Accessed throuh LAN

- Client/Server architecture
  - Reduces network traffic
  - Processing is distributed

Multiple-Site Processing, Multiple-Site Data Systems

- Fully distributed database management system
- Classifications
  - Homogeneous: Integrate multiple instances of same DBMS
  - Heterogeneous: Integrate different types of DBMSs over a network
  - Fully Heterogeneous: Support different DBMSs, each supporting different data model running under different computer systems

Distributed Requests and Distributed Transactions

1. Remote Request
   - Single SQL statement accesses data processed by a single remote database processor
2. Remote Transaction
   - Accesses data at a single remote site composed of several requests
3. Distributed Transaction
   - Requests data from several different remote sites on network
4. Distributed Request
   - Single SQL statement references data at several DP sites

Two-Phase Commit Protocol

- Concurrency control is especially important in distributed database systems
- Guarantees if a portion of a transaction operation cannot be committed, all changes made at the other sites will be undone
- Requires that each DP's transaction log entry be written before database fragment is updated.

Do-Undo-Redo Protocol

- Roll transactions back and forward with the help of the system's transaction log entries

Write Ahead Protocol

- Forces the log entry to be written to permanent storage before actual operation takes place

Phases

- Preparation
- The final Commit

Data Fragmentation

- Breaks a single object into two or more segments

Strategies

- Horizontal Fragmentation - Division of a relation into subsets (fragments) of tuples (rows)
- Vertical Fragmentation - Division of a relation into attribute (column) subsets
- Mixed Fragmentation - Combination of horizontal and vertical strategies

Data Replication

- Storage of data copies at multiple sites served by a computer network
- All copies of data fragments must be identical
- Push replication focuses on maintaining data consistency
- Pull replication focuses on maintaining data availability

CAP Theorem

- Database Systems must have: Consistency, Availability, Partition Tolerance

BASE Theorem

- Database Systems must be: basically available, soft state, eventually consistent
- Trade off between availability and consistency

Business Intelligence, Database Administration and Security, Web Technologies

Sources

- Lecture 10/4
- Lecture 10/9
- Lecture 10/11
- DBS Chapter 13
- DBS Chapter 15
- DBS Chapter 16

Definitions

Concepts

Data Warehouse

- Separated from operational environments
- Integrated data
- Mainly read only

Data Admin vs Database Admin

1. Database Administrator
    - Responsible for control of the centralized and shared database
    - Support company operations
    - Produce query results
2. Data Administrator
    - Higher degree of responsibility than DBA
    - Enable strategic decision making and planning
    - Identify growth opportunities
    - Reduce costs, boost productivity

Web Technologies

- Databases must be able to connect to various web technologies including:
    - Java
    - .Net
    - APIs
    - Adapters

Star Schema

- Data Modeling Technique
- Maps multi-dimensional decision support data into a relational database

Components

- Facts - numeric values that represent a specific business aspect
- Dimensions - qualifying characteristics that provide additional perspectives to a given fact
- Attributes - used to search, filter, and classify facts

Representation

- Facts and dimensions represented by physical tables in data warehouse database
- Many to One relationship between fact table and each dimension table
- Facts and dimension tables are related by foreign keys
- Primary key of a fact table is composite primary key because fact table is related to many dimension tables

Snowflake Schema

- Performance improvement for star schema
- Dimension tables can have their own dimension tables

SQL Extensions for Online Analytical Processing (OLAP) Tools

Roll Up

- Used with GROUP BY clause to generate aggregates by different dimensions
- Enables subtotal for each column listed except for the last one, which gets a grand total
- Order of column list is important

Cube

- Used with GROUP BY clause to generate aggregates by the listed columns
- Includes the last column

Extra Credit? Best part of Documentary

- Jack Dorsey explaining how technology and data can let our voices be heard directly from the source of an event