Database Theory

# Chapter One (Introduction)

Database

* A collection of related data
* A collection of interrelated files with minimum redundancy
* Raw facts

Mini-world

* Some part of the real world about which data is stored in a database
  + ex. student grades and transcripts at a university

Query

* A request to the database to find a specific set of information

Integrated Database Environment Includes

* Database
* DBMS
* Users
* Applications
* [\*Figure 1.4](Figures/fig%201.4.pdf)

DBMS

* Database Management System
* Controls access to database
* Has facilities to
  + Set up database structure
  + Load the data
  + Retrieve requested data and format it for users
  + Hide sensitive data
  + Accept and perform updates
  + Handle concurrency
  + Perform backup and recovery
  + Many other functions

People in Integrated Database Environment

* End users
  + Casual users
    - use query language
  + Naïve users
    - use programs
  + Secondary users
    - use database output
* Applications programmers
  + write programs for other users
  + DBA (Database administrator)
* [\*Figure 1.5](Figures/fig%201.5.pdf)

# Chapter Two (Database Planning and Database Architecture)

Resource

* An asset that has value and incurs cost

Information

* Processed data
* Useful for decision-making

Four Levels of Data

* Real world data
* Conceptual model of database
* Logical model of database
* Physical model of database

Data Sublanguages

* Data Definition Language
  + Used to define the database
* Data Manipulation Language
  + Used to process the database

System Analysis Approach

* Preliminary Investigation
* Feasibility Study
* Preliminary Design
* Detailed Design
* System Implementation
* System Operation
* Eventually repeat

Stages in Database Design

* Analyze user Environment
* Develop conceptual data model
* Choose a DBMS
* Develop logical model, by mapping conceptual model to DBMS
* Develop physical model
* Evaluate physical model
* Perform tuning
* Implement physical model
* [\*Figure 2.3](Figures/fig%202.3.pdf)

CASE (Computer-Aided Software Engineering) tools

* Upper case
  + Used for collecting and analyzing data
  + Used for designing logical model
  + Used for designing applications
* Lower case
  + Used for implementing the database
  + Prototyping
  + Data conversion
  + Generating application code
  + Generating reports
  + Testing
* Integrated CASE tool
  + Covers both upper case and lower case

Data Dictionary

* Contains metadata
* Can be integrated (part of DBMS) or free-standing
* Useful for
  + Collecting information about data in a central location
  + Securing agreement on meanings of items
  + Communicating with users
  + Identifying inconsistencies – synonyms and homonyms
  + Keeping track of changes to DB structure
  + Determining impact of changes to DB structure
  + Identifying sources of/responsibility for items
  + Recording external/logical/physical models & mappings
  + Recording access control information
  + Providing audit information

Project Management Software

* Tools to help plan and manage projects
* GANTT chart
* PERT chart
* User specifies
  + Scope and objectives
  + Major tasks and phases
  + Task dependencies
  + Resources, including personnel
  + Timelines for phases and completion of project
* Software can
  + Generate calendars
  + Produce graphs with different views of projects
  + Provide means of communication for staff

Database Administrator (DBA)

* Designs, creates, and maintains the database
* Planning and Design Stage
  + Preliminary planning
  + Identifying user requirements
  + Developing and maintaining the data dictionary
  + Designing the conceptual model
  + Choosing a DBMS
  + Developing the logical model
  + Developing the physical model
* Development Phase
  + Creating and loading the database
  + Developing user views
  + Writing and maintaining documentation
  + Developing and enforcing data standards
  + Developing and enforcing application program standards
  + Developing operating procedures
* Database Management Phase
  + Monitoring performance
  + Tuning and reorganizing
  + Keeping current on database improvements

Entity-Relationship Model

* Entities
  + Real-world objects about which we collect data
* Attributes
  + Describe the entities
* Relationships
  + Associations among entities
* Entity set
  + Set of entities of the same type
* Relationship set
  + Set of relationships of same type
  + May have descriptive attributes
* Represented by E-R diagrams

Relational Model

* Relations
  + Represented as tables
* Attributes
  + Represented as columns

# Chapter Three (Entity Relationship model)

Role

* Function that an entity plays in a relationship

Existence Dependency

* Entity Y is existence dependent on entity X if each instance of Y must have a corresponding instance of X
  + Y must have total participation in its relationship with X
  + if Y does not have its own candidate key, Y is called weak entity, and X is strong entity
* Weak entity
  + Discriminator
    - Partial key
    - Distinguishes instances of the weak entity that are related to the same strong entity
  + Represented by double rectangle in E-R diagram
    - Followed by a double diamond for relationship connecting it to its associated strong entity

Entities

* Specific objects or things in the mini-world that are represented in the database

Attributes

* Properties used to describe an entity
* Simple
  + Each entity has a single atomic value for the attribute
* Composite
  + The attribute may be composed of several components
  + May form a hierarchy where some components are themselves composite
* Multi-valued
  + An entity may have multiple values for that attribute

Specific Entity

* Has a value for each of its attributes
* Entity Type

Entity Type

* Entities with the same basic attributes

Key Attribute

* An attribute that participate in a key

Composite Key

* A composite contains more than one components

Alternate Key

* An entity type may have more than one candidate key

Degree

* Number of participating entity types

Cardinality Ratio (Of a Binary Relationship

* 1:1
* 1:N
* N:1
* M:N
* M/N = Many

Participation constraint

* Total
  + All entities participate
* Partial
  + Some entities do not participate

ER Diagram

* [\*Figure 3.12](Figures/fig%203.12.pdf)

# Chapter Four (The Relational Model)

Tables

* Represent relations
* Are related to other tables
* Holds information about objects
* Rows
  + Tuples
  + Correspond to individual records
  + Each tuple is distinct, no duplicates
  + Order doesn’t matter
* Columns
  + Correspond to attributes
  + Contains values from one domain
  + Has a distinct name, the name of the attribute it represents
  + Values all come from the same domain
* Domains
  + consist of atomic values
* Cell
  + contains at most one value
* [\*Figure 4.1](Figures/fig%204.1.pdf)

Database Relations

* Is very confusing
* [\*Figure 4.2](Figures/fig%204.2.png)

Degree

* The number of attributes
* Binary
  + two attributes
* Ternary
  + Three attributes
* N-ary
  + N attributes
* Property of intension
  + Does not change often

Cardinality

* The number of tuples
* Changes as tuples are added or deleted
* Property of extension
  + Changes often

Superkey

* Set of attributes that uniquely identifies tuples

Candidate key

* Minimal superkey
* Superkey such that no proper subset of it is also a superkey
* Has no unnecessary attributes

Primary key

* Candidate key chosen for unique identification of tuples

Foreign key

* Is an attribute or combination of attributes that is the primary key of some other relation

Integrity

* Correctness and internal consistency of data

Integrity Constraints

* Are rules or restrictions that apply to all instances of the database

Types of Constraints

* Domain
  + Limits set of values for attribute
* Entity integrity
  + No attribute of a primary key can have a null value
* Referential integrity
  + Each foreign key value must match the primary key value of some tuple in its home relation or be completely null
* General constraints or business rules
  + Statements indicating certain predicate must be true
  + May be expressed as table constraints or SQL assertions

Representing Relational Database Schemas

* Can have any number of relation schemas
* For each relation schema list a name of relation followed by list of attributes in parentheses
* Underline primary key in each relation schema
* Indicate foreign keys (italics or arrows)
* Database schema includes
  + Domains
  + Views
  + Character sets
  + Constraints
  + Stored procedures
  + Authorizations
  + Etc.
* Example
  + University database schema

Student (stuId, lastName, firstName, major, credits)

Class (classNumber, *facId,* schedule, room)

Faculty (facId, name, department, rank)

Enroll (*stuId,classNumber,* grade)

Relational Algebra

* Both operands and results are tables
* Can assign name to resulting table (rename)
* Contains operations to manipulate relations
* Used to specify retrieval requests (queries)
* Query result is in the form of a relation

Relational Operations

* SELECT σ
  + Selects the tuples (rows) from relation R that satisfy certain selection condition c
    - σc(R)
  + c is an arbitrary Boolean Expression on the attributes of R
  + Resulting relation has the same attributes as R
  + Resulting relation includes each tuple in r(R) whose attribute values satisfy the condition c
  + Examples
    - σDNO=4(EMPLOYEE)
    - σSALARY>30000(EMPLOYEE)
    - σ(DNO=4 AND SALARY>25000) OR DNO=5 (EMPLOYEE)
* PROJECT π
  + Choose certain attributes (columns) from a relation R to put in a list L
  + Resulting relation has only those attributes of R specified in L from every tuple of R
  + Eliminates duplicate tuples
* UNION U
* INTERSECTION ∩
* DIFFERENCE -
* CARTESIAN PRODUCT X.
  + Is the set of all pairs from two or more combined tuples
  + D1 X D2
  + Can combine related tuples from two relations if followed by the appropriate select operation
  + [Figure 6.5](Figures/fig%206.5.png)
* JOIN | |
  + Theta-join
    - A |X| B
* Other relational operations:
* DIVISION
* OUTER JOIN
* AGGREGATE FUNCTIONS
* RENAMED
  + [Figure 4.21](Figures/fig%204.21.png)

Query

* A combination of relational operations
* Relational algebra expression
* [Figure 6.1](Figures/fig%206.1.png)

# Chapter Five (Relational DBMS and SQL)

Data Definition Language

* Commands
  + CREATE TABLE
  + CREATE INDEX
  + ALTER TABLE
  + RENAME TABLE
  + DROP TABLE
  + DROP INDEX
  + CREATE VIEW

Constraints

* Rules that regulate what data can be where

Column Constraints

* Constraints specified for values in individual columns, cannot be adjusted

Table Constraints

* Constraints specified for values in individual tables, they can be updated and turned off and on

Indexes

* Facilitate fast retrieval of records with specific values in a column
* Stored in same file as base table
* B+\_trees or B\_trees used
* Graphical user interface, text, application

  Description automatically generated

Basic Commands

* SELECT
* UPDATE
* INSERT
* DELETE

Relational Views

* Can be subsets of base tables, or subsets of joins, or contain calculated data
* Allow different users to see the data in different forms
* If database is restructured, view can keep the user’s model constant

Active Database

* DBMS monitors database to prevent illegal states, using constraints and triggers
* Constraints are checked whenever a change is made to the database to ensure DB is in a valid state

Triggers

* Essentially if else statements
* Event
* Condition
* Action

:OLD

* Refers to values in a tuple deleted or to values replaced in an update

:NEW

* Refers to values in a tuple just inserted or to the new values in an update

Project

* removes duplicates

Select/Project

* inner query creates temporary table
* outer query creates permanent table

Query union

* both tables must have the same domain

DDL Commands

* Text

  Description automatically generated

Create Table

* Text

  Description automatically generated

Create Table Example

* Text

  Description automatically generated

New Datatypes

* Can create new datatypes that can be used to make attributes that can only be compared with the same attributes of the same new datatype
* Text

  Description automatically generated

Column-level constraints

* A picture containing text

  Description automatically generated

Table-Level Constraints

* Text, letter

  Description automatically generated

# Chapter Seven (Object Oriented and Object Relational Database)

Why OO

* OO database provides persistent objects to correspond to temporary objects in programs
* Container for each class

OO Data Concepts

* Can only get data through methods

Object Classes

* Objects of the same class form an object class
* Extent
  + Set of objects in a class
* Objects in class are like entity instances

Defining a class

Writing and Using Methods

* Written in OO programing language

Class Hierarchies

* Inheritance
* Hierarchy diagram
  + Rectangles are classes
  + Subclasses and superclasses are connected via triangles

UML

# Chapter Eight (Database Security)

Securing your database

* Lock everything
* Make it difficult for people to steal

Security

* Protecting the database from unauthorized
  + Access
  + Modification
  + Destruction

Privacy

* The right of individuals to have some control over information about themselves
* Right to privacy of data can be protected by database security

Threats

* Accidental
  + User error
  + Communications system errors
    - User sent a message that should be sent to another user
    - System connects a user to a session that belongs to another user with different access privileges
  + OS errors
    - Accidentally overwrites files and destroys part of database
    - Fetches the wrong files and sends them to the user
    - Fails to erase files that should be erased
* Deliberate
  + User intentionally gains unauthorized access and/or performs unauthorized operations on the database
  + Disgruntled employee who is familiar with the organization’s computer system seeks revenge
  + Industrial spies seek information for competitors
  + Methods
    - Wiretapping of communication liens
    - Electronic eavesdropping[picking up electronic signals
    - Reading display screens or printouts left unsupervised
    - Impersonating authorized users or users with greater access
    - Writing programs to bypass the DBMS and access database data directly
    - Writing applications programs that perform unauthorized operations
    - Deriving information about hidden data by clever querying
    - Removing physical storage devices from the computer facility
    - Making copies of stored files without going through the DBMS
    - Bribing, blackmailing, or influencing authorized users to obtain information or damage the database

Security Plan

* Should begin with physical security measures for the building
  + physical barriers
  + control access
  + require badges
  + sign-in, etc.
* Should have more physical security for the computer facilities
  + e.g. locked door
* Additional security control for database

Authentication

* User authentication
  + verifying the identity of users
* Operating system uses user profiles:
  + User ids and passwords
  + Authentication procedures
  + Badges
  + Keys
  + Or physical characteristics of the user
* Additional authentication can be required to access the database

Authorization

* DBMSs designed for multiple users have a security subsystem

Access Control

* Covers he mechanisms for implementing authorizations
* matrix
  + Planning tool identify operations different users are permitted perform on various database objects

User Profile

* System has a user profile for each id, giving information about the user
* Stored profiles should be kept secure, possibly in encrypted for
* Profile normally includes a password, allegedly known only to the user
* Passwords should be kept secret and changed frequently
* System should never display passwords at sign-in time

Security Mechanisms

* Views
  + simple method for access control
* Security log
  + Journal for storing records of attempted secuirity violations
  + Audit trail
    - records all access to the database requestor, operation performed, workstation used, time, data items and values involved
  + Triggers
    - can be used to set up an audit trail
  + Encryption
    - Choose a function that alters the data and can inverse to get it back

Encryption

* Uses a cipher system that consists of

SQL Authorization Language

* Grant
  + statement used for authorization
* Revoke
  + Statement used to retract authorization
* Privileges can be given to users directly

Oracle Privileges

* Object Privilege
  + the right to use a DML command to perform an action on an object
* System Privilege

Cascade

* affect the parent entity and all of their child entities

Statistical Databases

* Support statistical analysis on populations
* Data itself may contain facts about individuals, but is not meant to be retrieved on an individual basis
* Users are permitted to access statistical information-totals, counts. or averages. but not information about individuals

Statistical DB Security

* Need special precautions to ensure users are not able to deduce data about individuals

Need for DB security on the internet

* Messages transmitted in plaintext can be read by intruders using packet sniffing software
* Customers need assurance that their credit card info is kept private when sent over the Internet Companies that allow web connections to their internal networks for access to their database need to protect it from attack
* Receivers and senders of messages need to be sure that the site they are communicating with is genuine and trustworthy

Techniques for Internet Security

* Firewalls
* Certification authorities such as Verisign

Firewalls

* A hardware/software barrier that protects an organizations’ intranet from unauthorized access
* Ensures that messages entering or leaving intranet meet the organizations’ standards
* May use a proxy server that intercepts all messages in both directions-hides the actual network address

# Chapter Nine (Transaction Management)

Transaction

* A logical unit of work that takes the database from one consistent state to another
* A logical unit is in a consistent state before and after transaction
* steps
  + Locate the record to be updated
  + Bring the block into the buffer
  + Write the update to buffer
  + Write the modified block t=out to disk
* Can terminate successfully and commit
  + everything that should transfers to disk transfers
* Can terminate unsuccessfully and be aborted
* Either everything needs to be done or nothing
  + if you made changes to disk and then some of the work fails you need to undo all the work that you have done on the disk
* Aborted transaction must be undone

Protecting database during transactions

* Recovery
  + backup
  + restoring the database to a correct state after a failure
* Concurrency control

ACID Properties

* Atomicity
  + Entire transaction is considered as one unit of operation
* consistency
  + database before the transaction starts and after it finishes it is in a consistent state
  + Concurrency control subsystem must ensure this for multiple transactions
* Isolation
  + When transactions execute simultaneously, DBMS ensures that final effect is as if the transactions were executed one after another
* Durability

Concurrency Problems

* Needed when transactions can process simultaneously, if at least one includes an update
* potential problems
  + Lost update problem
    - can use lock and key to fix, but adds time
  + Uncommitted update problem
    - Person 1 changes data Person 2 reads changed data, Person 1 rolls back his changes. Person 2 writes and the data is once again changed even though it should’ve been rolled back
  + Inconsistent analysis problem
    - Reading data after someone has already changed it
  + Non-repeatable read problem
    - you read the data twice and you get different results because someone changed it
  + Phantom data problem
    - during your work someone has made changes without your knowledge, and it affects your results

Conflict in Transaction

* Only reading items cannot conflict, order is unimportant
* If two transactions operate on completely separate data items, they do not conflict and order is not important
* If one transaction writes to a data item and another either reads or writes to the same data item, then the order of execution is important
* Conflict only occurs if
  + they belong to different transaction
  + …
  + …

Serial Execution

* Execute one transaction at a time with no interleaving of operations
* Can have n! possible serial executions
* May produce different results all are considered correct

Serializable Schedules

* Find serializable schedule to maximize concurrency while maintaining the correctness of the data
* Order the data
  + uses order of operations
* Create multiple critical sections, makes it so one transaction waits till another transactions finishes a section before it starts on the same section
* Result needs to be the same as at least one Serial Execution

Precedence Graph

* Used to determine whither a schedule S, is conflict serializable
* S is conflict serializable if graph has no cycles
* If S is serializable, can use the graph to find an equivalent serial schedule by examining the edges

Phases in Validation Techniques

* Read phase
* Validation phase
  + Check for errors
  + Examines reads and writes of other transactions, T, that may cuase interference
  + Has three
* Write phase

# Chapter Ten (Distributed Database)

# Chapter Eleven (Data Warehouse and Data Mining)

Data Warehouses

* Gather data to analyze to predict the future
* Subject oriented
  + data about specific subject
  + ex. sales
    - who was best customer for this item last year
* Integrated
  + Gathered from multiple sources, shapes, sizes
  + Need to be edited so that they can be combined into a consistent format
    - naming conflicts
    - inconsistencies among units
      * cm to inches
* Non-volatile
  + Is data that won’t change over time
  + Historical data
* Time-varying
  + Overtime new data will be different from old data
  + Data warehouse focus on change over time is what is meant by the term time variant
* Collection of data that is used for decision making
* Stores MASSIVE amount of data
* Typical uses
  + data mining
  + decision support systems
  + on-line analytical processing

Database vs Data Warehouses

* Database
  + Online transactions
  + Optimized for write operation
  + Daily changes
* Data Warehouse
  + Online Analytical Processing
  + No normalization
  + One table
  + Optimized for read operations
    - high performance
  + Not changed very often
  + Includes a database in its structure

Operational Database

* used in day-to-day business
* Support online transaction processing
* serve the information needs of end users
* Support day-to-day business operations
* Require high availability and efficient performance
* Handle large volume of transactions

Data Warehouse

* Support online analytical processing
* Used for decision making
* Contain very large amount of data
* Have data from multiple operational databases, taken at different periods of time
* May include data from other sources, summarized data, metadata
* Optimized for efficient query processing and presentation of results
* Updates done periodically, not in real time
* Support data mining
  + discover patterns or trends in the data over long periods of time

Data Warehouse Architecture

* Must support unanticipated types of requests, ad-hoc queries
* Input data
  + Taken from various data sources
  + Extracted using back-end system tools-accommodate differences among heterogeneous sources
  + Reformatted into a consistent form
  + Checked for integrity and validity – data cleaning

Data Mining

* No normalization
* Best format is a flat file
* Discovering hidden information from very large sets of data
* Uses techniques from statistics and artificial intelligences
* Needs a large database or data warehouse
  + more data means more accuracy
* Formats
  + Should be considered in the original design of the warehouse
  + Requires summarized data as well as raw data from original data sources
  + Requires knowledge of the domain and also knowledge of data mining process
    - ex. medicine has a lot of terminology that CS major likely won’t know
  + If flat not used, data must be prepared and reformatted for data mining
    - de-normalize
* Purpose
  + Predict the future behavior of attributes
  + Classify items, placing them in the proper categories
    - Supervise classification
      * Number of classes predefined
    - Unsupervised
      * Number of classes not defined
      * Determine based off of information
  + Identify the existence of an activity or an event
  + Optimize the use of the organization’s resources

Sequential patterns

* prediction that a customer who buys a particular product in one transaction will purchase a related product in a later transaction
* Can involve a set of products
* Patterns are represented as sequences {S1}, {S2}
* First subsequence {S1} is a predictor of the second subsequence {S2}
* Support is the percentage of times such a sequence occurs in the set of transactions
* Confidence is the probability that when {S1} occurs, {S2} will occur on a subsequent transaction – can calculate from observed data

Time Series Patterns

* A sequence of events that are all of the same type
* Sales figures, stock prices, interest rates, inflation rates, and many other quantities can be analyzed using time series
* Time series data can be studied to discover patterns and sequences

Data Mining Methods

* Decision tree
  + a method of developing classification rules
  + Developed by examining past data to determine how significant attributes and values are related to outcomes
  + Nodes of the tree represent partitioning attributes, which allow the set of training instances to be partitioned into disjoint classes
  + The partitioning conditions are shown on the branches
* Tree is then used to classify new cases

Regression

* A statistical method for predicting the value of an attribute, Y, (the dependent variable), given the values of attributes X1, X2, …, Xn (the independent variables)
* Statistical packages allow users to identify potential factors for predicting the value of the dependent variable
* Using linear regression, the package fins the contribution or weight of each independent variable, as coefficients for a linear function

Clustering

* grouping by specific attributes

Application of Data Mining

* Retailing
  + Customer relations management
  + Advertising campaign management
* Banking and Finance
  + Credit scoring
  + Fraud detection and prevention
* Manufacturing
  + Optimizing use of resources
  + Manufacturing process optimization
  + Product design
* Medicine
  + Determining effectiveness of treatments
  + Analyzing effects of drugs
  + Finding relationships between patient care and outcomes

# Chapter Twelve (Big Data and No-SQL)

Motivation for Big Data

* Data collected is not typical of the type found in relational tables
  + Unstructured
  + Generated in real-time
  + May not have a well-defined schema
  + Varying types: pictures, video, social media posts, sensor data, etc.

Defining Big Data

* Volume
  + Massive volumes of data
  + Sensors
  + Monitors
  + Social Media Log files
* Variety
  + Data in a wide variety of forms: structured, unstructured, or semi-structured
  + Text
  + Video
* Velocity
  + Streaming Data
  + Fast Rate of Speed
  + Real-time
* Veracity
  + Correctness
  + Validity
  + Trustworthiness
* Value
  + Knowledge
  + The advantage that the successful analysis of big data can bring to organizations
  + Can gain new knowledge by understanding data
  + Core of Big Data
  + Insight
* Needs to be edited to be usable

Big Data Pipeline

* Collect data
* Extract from data
* Clean Data
* Integrate, aggregate, and represent data
* Analyze data
* Interpret data
  + needs an expert in the field of the data to interpret it

Hadoop

* Provides storage an analytics for companies such as Facebook, LinkedIn, Twitter, Netflix, Etsy, and Disney
* A system for distributing large data sets across a network of commodity computers
* Can be complex to manage distributed file components and metadata
* Provides a high level of fault tolerance
* Supports parallel processing for faster computation

MapReduce

* parallel programing
* Separates tasks to multiple computers and re-combines the end result

Node

* is a computer

Racks

* a group of nodes

Storage Formats in Hive

* Row format or file format
* SerDe interface to read and write data
  + serialize file to work on it
  + de-serialize file once done working on it

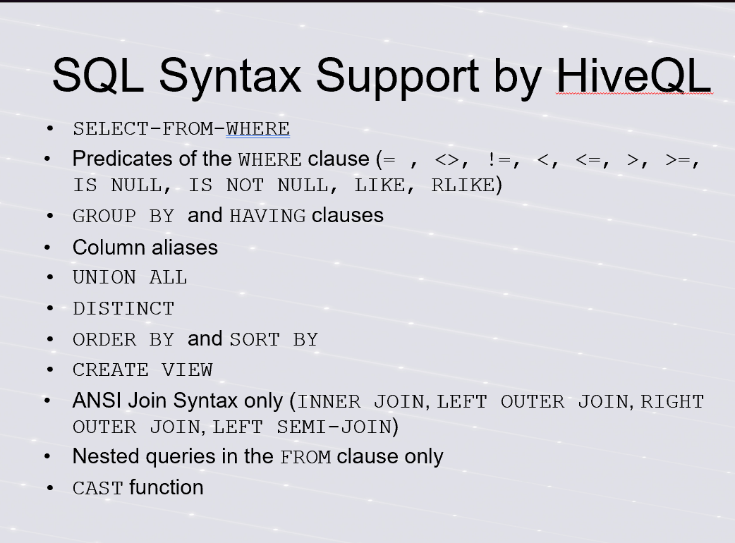
Deserialization

* transforms a row of data into an internal object representation

HiveQL

* SQL interface that supports ad-hoc queries over Hive tables
  + Ad-hoc means you don’t know what the information is going to be
* Designed for batch processing and not real-time processing
* Simple to write, will eventually be transferred to MapReduce

Features supported by HiveQL

* 

Features NOT supported by HiveQL

* No row level inserts, updates, or deletes
* No update views
* No stored procedures
* Note: SORT BY will only sort the output of a single reducer
* Use ORDER BY to get a total ordering of the output from all reducers

Explain

* Shows program code generated in map reduce for a query

NoSQL Systems

* Certain features of SQL need to be added to Hadoop
* Built on top of Hadoop system
* Means “Not only SQL”
* Designed for large scale projects

Features of NoSQL

* Flexible data schemas
* Weaker concurrency model
* Simple call level interface
* Parallel processing

ACID

* Atomicity
* Consistency
* Integrity
* Makes data slower

BASE

* Basically Available all the time but eventually consistent
* Soft-State (does not have to be consistent all the time)
* Eventually Consistent
* The data will be consistent in the end

Types of NoSQL Systems

* Key-Value Pair Systems
* Column-Oriented Table Systems
* Document-Oriented Systems
  + documents may not all have same shape so you may need to add tags
* Graph-Oriented Systems

HBase

* Large, sparsely populated tables
* Tables contain rows and columns
* Differences
  + Cell values are uninterpreted byte arrays (no notion of different data types)
  + Individual cells of a table can be versioned, storing the history of values for a cell
    - uses time stamps
  + Each row has a row key, with the table sorted by the row key
  + Atomicity is only guaranteed at the row level
  + Columns in Hbase are not the same as columns in relational tables

Columns in Hbase

* Column qualifier
* Groups of columns are called column families
  + conceptually organize column qualifiers into groups that have the same access patterns
* Shouldn’t have more than two or three families
  + Lots of duplicates
* Keep names as small as possible

Hbase Maps

* Hbase can be viewed as a multi-dimensional map
  + dictionary of dictionaries

Hbase design issues

* More complex than relational data
* lots of duplicates

NewSQL Systems

* Developing alternative to NoSQL and RDBMS technology
* Exploit distributed database technology together with cloud computing to handle big data together with transactional capabilities that support ACID properties

# Chapter Thirteen

Steps in executing SQL query-DBMS

* Checks query syntax
* Validates query-checks data dictionary; verifies objects referred to are database objects and requested operations are valid
* Translates query inot relational algebra
  + come up with the most effectient query

Relational Algebra Translation

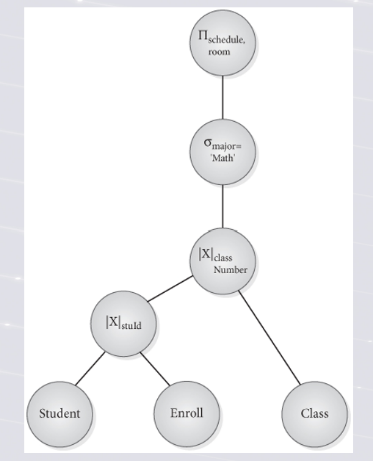
* select = project
* select uses where
* ‘from’ when used on many things is ‘join’

Text

Description automatically generated

Query Tree

* Leaves are tables
* Unary or binary operations are represented by internal nodes (inner query)
  + join
  + select
    - should be part of inner query
  + project
  + etc.
* AN internal node can be executed when its operands are available
* Node is replaced by the result of the operation it represents
* Root node is executed last, and is replaced by the result of the entire tree
* Executed from leaves on up



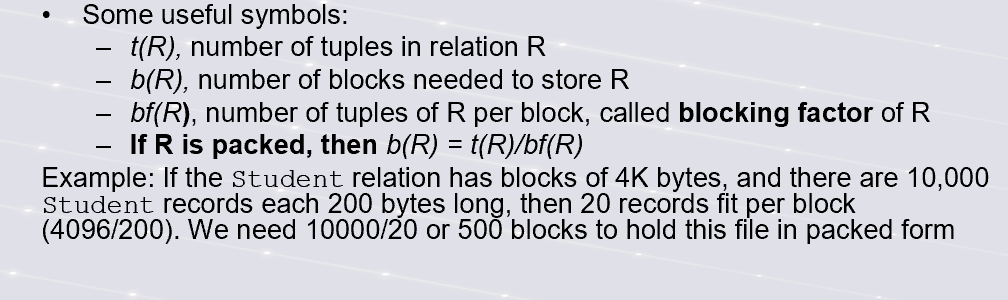
Selection Rule

* Do selection as early as possible

Cost Factors

* cost of reading files
* processing costs once data is in memory
* cost of writing and storing results
* communication costs
* cost of writing final result to storage

Estimating Access Cost

* Number of blocks brought into main memory for reading or written to secondary storage as results
* Tables may be store in
  + Packed form – blcoks contain only tuples from table
  + unpacked form- tuples are interspersed with tuples from other tables
* 

Indexes

* clustered index
* dense or non-dense
* B+ tree usually used
* Index access cost is usually small compared to the cost of accessing the data records

# Review

Chapter Seven

* Enhanced entity relationship
  + Supertypes and subtypes
  + Specializations
  + For each box create a type and create a table of that type
    - Type name used as constructor
    - Table becomes object table
* Uses object query language
* Value in table can be anything
* System generates an object Id for each object
  + eliminates need for foreign key
* Object oriented table
  + No table
  + UML Diagram
    - classes
  + Create objects of classes
  + Each class has extent

Chapter Eight

* Dangers
  + People may want to damage/gain unauthorized access
  + System may crash
* Ways to protect database
  + physical barrier
    - locks
    - cameras
    - log in passwords
    - encryption
  + views created by DBA
  + Authentications
  + Log of everything happening in the system
  + Backup of system periodically
  + Authorization Language
    - granting authority to others
  + Filters

Chapter Nine

* Database is created and shared among multiple users
* Many people want to use the same item
  + only one person can use an item at a time
* Protocols for determining who gets to use what when
* Locking
  + Lock data before you read or write on the data
  + Easy to implement
  + Reduces degree of concurrency
  + Grants Serializability
  + Can create Deadlock
* Deadlock
  + Create wait for graph and cycles can create deadlock
  + System has to chooses a victim to release resources
* Database Hierarchy
  + database
  + tables
  + blocks
  + records
  + data items
* Lock an item on a parent in the hierarchy and all children are locked as well
  + have to check entire subtree
* intention locking
* Timestamping
* Multi-versioning
  + When you write you have to create a new copy to write on
  + Reading becomes more difficult

Chapter Ten

* Data stored on multiple computers
* May have local database and distrub

# Stuff to do for next time

* Look into scholarship from [Minaieaf@uvu.edu](mailto:Minaieaf@uvu.edu) in engineering department
* Try to solve pointer problem from senati