Relational Databases

Concepts

Relational Database

- Invented by E.F. Codd in 1970
- Collection of tables
    - Tables
        - Rows
        - Columns—name, data type, size
- Unlimited parent/child relationships between tables
    - Major improvement over the previous databases of 1960s which limited parent/child to a hierarchy
    - Relational algebra—forms the basis for Structured Query Language (SQL)

SQL: Structured Query Language

- Pronounced two ways
    1. “See-kwel”
    2. Speak the letters “S Q L”
- De facto standard query language for relational databases
- ANSI standard
    - Basic, commonly used constructs are the same
- Variations among various vendors

DML vs. DDL

SQL statements are classified two ways:

1. DDL (data definition language)
   - Create or drop tables, views, indices, etc.
   - Change configuration settings
   - Security: users, roles, permissions, etc.
   - Typically only used by the DBA (database administrator)
2. DML (data manipulation language)
   - Select, insert, update, delete
   - What we will typically use

Procedural vs. Nonprocedural

- Procedural programming languages
  - Specify exactly how to do something step by step
  - “If” statements, loops, etc.
  - Examples: Python, C/C++, Java, JavaScript, Assembly
- Nonprocedural programming languages
  - Also known as declarative programming languages
  - Specify what we want done
  - Programming language generates an execution plan to do what we specified
  - SQL is a nonprocedural (declarative) programming language

Functional Programming

- Procedural code wrapped around nonprocedural (declarative) code
- Best of both worlds
- Allows us to add procedural constructs, such as “if” statements, loops, etc. to SQL
- Example: Python making an SQL query and looping through and processing the results

Relation

- Table
- Result of an SQL select query
- Result of an SQL select subquery
- Not to be confused with relationship

Primary Key

- Primary key—column (or group of columns) that uniquely identifies a row in a table
  - Composite key—primary key with more than one column
- Natural key—choose the primary key to match the business
- Surrogate key—for performance reasons, we use a meaningless integer for the key instead of the natural key

Foreign Key

- Parent/child relationship
  - Joins match the parent’s primary key to the child’s foreign key
- Foreign key could be part of the primary key
  - Sometimes they are, sometimes they are not
- Don’t confuse relation with relationship

Data Model

ERD (entity relationship diagram) show us the following:

- Tables
- Columns
- Primary keys
- Foreign keys
- Parent/child relationships between tables
- How to join a parent table to a child table

Normalization

- Normalized
  - 3NF (third normal form)
  - Each element of data is only stored once
  - Complicated subject—takes weeks to learn how to normalize a database
- Denormalized
  - Each piece of data is stored multiple times for ease of use in analytics

Transactional Databases

Transactional databases

- Execute the business
- OLTP (online transaction processing)
- Normalized, 3NF
- Focus on current data
- ACID (atomicity, consistency, isolation, durability)
  aka immediate consistently
- Row headers—tables stored in row major order
- Scale up very limited

Analytical Databases

Analytical databases

- Evaluate the execution of the business
- OLAP (online analytical processing)
- Denormalized
- Focus on both current and historical data
- BASE (basically available, soft state, eventual consistency) aka eventual consistency
- Columnar—column header—tables stored in column major order
- Scale up very well

Concepts: Relational Databases

The End

Relational Databases

Business Cases

Obvious Relational Databases

Almost every software system uses a relational database to store data.

- Accounting systems
- Reservation systems
- Timecard systems
- Point of sale systems
- Card key access systems
- Etc.

Less Obvious Relational Databases

Small embedded relational database

- Phone apps
- Tablet apps
- Smart watches
- IoT devices
- Vehicles

Databases at Several Levels

- Data center has a large database for the POS (point of sale) server
- Each store has a smaller computer with a database to cache semi-static data that changes only a few times a day
- Actual POS device has a small embedded database
- Every night we dump the data from the POS server’s database into an analytical database for deep analytics.
- The POS system writes live data to a streaming process that has a database for speed analytics

Business Cases: Relational Databases

The End

SQL: Single Table Queries

Concepts

Select Clause

- Column list
- Derived columns
  - Created by calculations on columns
- Column aliases

Order by Clause

- Sorts the results of a query
  - Column, derived column, column alias
  - Each column can be sorted in ascending (default) or descending order
- SQL does not guarantee any order on query results unless an order by clause is specified

Where Clause

- Filter the results of a query
- Applied before any aggregations

Aggregation

- Aggregation
  - Aka summarization of data
  - Aka roll up of data
- Aggregation at the table level
  - Default behavior
- Aggregation on a column or list of columns
  - Group by clause

Where vs. Having

- Where clause
  - Filters row pre-aggregation
- Having clause
  - Filters rows post-aggregation

SQL: Single Table Queries

The End

SQL: Multiple Table Queries

Concepts

Set Operations

- Combine rows from two queries
- Typical operations
  - Union: combines removing duplicates
  - Union all: combines without removing duplicated
  - Intersect: in both queries
  - Minus: in first query, not in second query
  - Somewhat vendor dependent
- Union compatibility
  - Number of columns must match exactly
  - Each column’s data type must match (or be convertible)

Join Operations

- Combine columns from two tables
- Parent/child
  - Parent’s primary key
  - Child’s foreign key
  - ERD-specified parents, primary keys, child tables, foreign keys, etc.

Inner vs. Outer Joins

- Inner: only parent rows with child rows
- Outer: parent rows without child rows are also included
- Left outer: parent table on left side of query
- Right outer: parent table on right side
  - Rare: typically, we place the parent on the left side of the query
- Full outer includes:
  - Parent rows without child rows
  - Child rows without parent rows (aka orphans)

Joins on Sibling Tables

- Join a primary key to a primary key
- Since primary keys are unique, always a one-to-one relationship
- Typically found when we join tables from a primary dataset with a secondary dataset that are not using the same ERD

Dangerous Joins

- Joins that do not fall into:
  - Parent/child relationship—join parent’s primary key to child’s foreign key
  - Sibling relationship—join on primary keys
- Issues
  - Extra rows problem
  - Missing rows problem
- Typically found when we join tables from a primary dataset with a secondary dataset that are not using the same ERD

Subqueries

- Queries inside of other queries
- Type 1 subqueries
  - No relationship between inner query and outer query
  - Can be pulled out and run by itself
  - Scales up
- Type 2 subqueries
  - Relationship between inner query and outer query
  - Cannot be pulled out and run by itself—generates syntax error
  - Does not scale up

Views

- Act just like a type 1 subquery
- No storage
  - Some databases support a materialized view with cached storage and a refresh rate
- Hide complicated set and join operations from users
  - Good DBA will have views for all set and join operations
- Denormalized layers on top of a normalized 3NF database
  - Heavily used in analytics

SQL: Multiple Table Queries

The End

SQL: Transaction Processing

Concepts

Transaction Processing

- Transactions
  - Changes to the database are only present in the current transaction until we commit
  - Once we commit, all queries from all users will see our changes
- Transaction processing cycle
  - Begin a transaction
  - Make changes
  - Commit the changes if the transaction is successful
  - Roll back the changes if the transaction is not successful

Bulk Loads and Changes

- Transaction space is limited—assumes small amount of changes per transaction
- Bulk loads and bulk changes—run out of transaction space
- Solution: add logic to commit every 1,000 or so rows

Inserting Data

Insert statement

- Inserts a new row into a table
- Primary key must be unique
- Foreign keys must validate against the parent table’s primary key

Updating Data

Update statement

- Updates existing row(s) in a table
- Most databases do not allow an update of any column in the primary key
- Updates to foreign keys must validate against the parent table’s primary key

Deleting Data

Delete statement

- Deletes existing row(s) in a table
- Cannot delete a row if a child row exists
  - Must delete the child rows first

SQL: Transaction Processing

The End

Basic Data Visualization of Data Pulled From SQL

Concepts

Data Visualization

- Creation and study of the visual representation of data
- Interdisciplinary
  - Art
  - Statistics
  - Computer science
  - Geography
  - Etc.
- Old cliché: “A picture is worth a thousand words”

Tips

- Keep it as simple to understand as possible
  - Know your audience
- Make large datasets easy to understand
- Make it easy to compare
- Make it easy to see the level of aggregation
- Be sure data visualizations are consistent with other data you have presented

Most Important Tip

When presenting a data visualization

- Always have the data to back it up
- Be ready to explain and prove

Labs

- Pull data from SQL and present it using data visualizations
- Focus on simple, basic data visualizations needed to complete the projects
    - Pie charts, grids, scatter plots, line plots, bar charts, histograms, box plots, violin plots

Basic Data Visualization of Data Pulled From SQL

The End

Basic Geographic Data Visualization of Data Pulled From SQL

Concepts

Map Applications

Assume everyone is familiar with using map applications.

Basic Attributes

- Type
  - Road
  - Terrain
  - Satellite
  - Hybrid
- Size
  - Height
  - Width
- Zoom level

Markers, Heatmaps, Choropleths

- Markers
  - Marks a location on a map
  - Often users can click on for more information
- Heatmaps
  - Size, color, etc. of the marker adds meaning to the location
- Choropleths
  - Heatmaps that use a geographic area instead of a single location

Additional Common Layers

- Driving directions
- Traffic layers
- Transit layers

Latitude and Longitude

- Latitude and longitude can give us a specific location anywhere in the world.
- Using geometry, we can calculate:
  - Distance between two locations
  - Direction between two locations
  - Given a location, a direction, a distance, find a new location
  - All locations within a given distance

Sphere vs. Ellipsoid

- Earth is not an exact sphere; it’s an ellipsoid
  - Radius at equator is 3,963 miles (6,378 km)
  - Radius at poles is 3,950 miles (6,357 km)
- Great circle calculations
  - Assume Earth is a sphere
  - Use average radius
- Geodesic calculations
  - Earth is an ellipsoid
  - WGS84 ellipsoid used for Earth

Great Circle vs. Geodesic

- Berkeley to Miami
  - Great circle: 2,573.8 miles
  - Geodesic: 2,577.8 miles
  - Difference: 4 miles, about 0.15%
- Geodesic popular
  - Drones
  - Robots
  - Deliveries that need specific driveways, etc.

Locations and Databases

- Locations stored as a latitude column and longitude column in a database table
- Given a location and a distance, find all locations
    - Create a box
        - Locations 0, 90, 180, 270 degrees at the distance
    - Query locations that fit in the box
    - Loop through results and refine using individual distance calculations (or just use the box as an approximation)

Basic Geographic Data Visualization of Data Pulled From SQL

The End