Mid-Semester Review

Ryan Nixon

Some Definitions

- Table - Holds multiple items with common attributes
- Column - A single attribute for the table's items
- Row - A single item in a table
- Query - Written text to retrieve items from tables
- Transaction - A collection of multiple queries, all being executed at once
- Clause - Major components to a query (SELECT, FROM, WHERE)
- Boolean - Conditional (True/False) requirements to filter items
- Resultset - A tabular list of items resulting from the query
- Connection - The length of time that an individual is using the database

Some Definitions

- Data Type - A limitation placed on a table's columns to require data to be formatted in a specific manner
- Constraint - A limitation placed on a table's columns to require data to be set a particular way
- Key - A constraint that is placed in order to allow table relationships to be formed
- Primary Key - The identifying column for a table; must be unique
- Foreign Key - A reference to a primary key in a *different* table
- NULL - The absence of a value
- String - A series of characters (A-Z, 0-9, !, @, #, $, etc.)
- Integer - A numeric value

Some Definitions

- Function - An action that will collect or manipulate data from multiple rows
- Aggregate - A function that collects data from multiple rows and calculates a single result
- Row-Level Function - A function that manipulates individual rows in place or generates new data
- Calculated field - Row-level functions represented as basic arithmetic operations

Query Format

```
CREATE TEMPORARY TABLE temp_name AS
SELECT FUNCTION(column1) AS alias1
   , column2 AS alias2
FROM table AS table1
INNER JOIN table AS table2
   ON (table2.pkey = table1.fkey)
OUTER JOIN table AS table3
   ON (table3.fkey = table2.pkey)
WHERE column2 = 'string' OR column1 = number
GROUP BY column1, column2
HAVING FUNCTION(column1) > number
ORDER BY alias1 ASC, alias2 DESC
LIMIT rowcount;
```

Inner Joins

- Connect 2 or more tables together
- Append additional columns to the resultset
- Connected by the primary & foreign keys in those tables
- Foreign keys formatted as "Table_Column"
  - For example, "person_id" = the id column in the person table

Outer Joins

- Exactly like Inner joins, but with the added requirement that data stays intact
- If no data can be added to a row in the left, the new columns will contain NULLs
- Because only one side is retained, direction *matters*. You must join in the proper order

Grouping

- "Categorizes" or "Groups" items
- Partitions data into smaller groups
- Also splits aggregates into smaller groups
  - The split is determined by the value of the column in the GROUP BY

Views & Temp Tables

- Views - Saved queries; using the view runs the query
- Temp Tables - Saved resultsets; using the table pulls data from the saved results
- Temp Tables can also be considered “copies” of the original table(s)
- Temp Tables are deleted when the connection is closed

Upcoming Material

- Data Types
- Creating Tables
- Defining Constraints
- Inserting/Updating/Deleting Data

Data Types

- Example: Integer, Float, Varchar, Datetime
- There are many data types that can be used for table columns. The type needs to match the data that will be held in them
- It is difficult to change these after they are defined. Choose wisely when using them
- TIP: Be very aware of the differences between Strings, Numbers, and Dates. Use the example database as a reference

Creating Tables

- ```
  CREATE TABLE table_name (
  id INTEGER PRIMARY KEY,
  col1 VARCHAR(50),
  col2 INTEGER NOT NULL DEFAULT 1,
  FOREIGN KEY (col2)
      REFERENCES other_table(pkey)
  );
  ```

- TIP: Look over Unit 1 again. Focus on the definitions for the keys, constraints and table relationships
- PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, DEFAULT
- These will all be required in some way in the SQL project
  ```

Inserting Data

- `INSERT INTO table_name (col1, col2)`
  `VALUES ('col1 data', 2);`
- Inserts a single row into the specified table
- Defaults can be provided, as well as NULLs
- TIP: Refresh your memory on when quotations should or should not be used around values

Updating Data

- ```
  UPDATE table_name SET
       col1 = 'New data',
       col2 = 3
  WHERE col1 = 'col1 data'
  ```
- Updates multiple columns with new information; cannot insert
- Able to edit multiple rows based off of the WHERE clause
- TIP: Try out some SELECTS using WHERE. Understand why the resultset has those rows and not others
  ```

Deleting Data

- `DELETE FROM table_name`
  `WHERE id = 1`
- Deletes individual rows according to the WHERE condition
- There is no undelete; always double check before running the query
- This will not be required for the project

Project Preview - The Example Database

Reminders

- Assignment 8 up tonight regarding the SQL Project, due 11/19
- Lab time in SSB 172 on Wednesday