SQL REVIEW

* SQL (Structured Query Language)
  + Programming language
  + Scripting language (you run commands and it tells a machine what to do. For SQL that is a database)
* Sub-languages of SQL
  + DML (Data Manipulation Language) – adds or removes/ edits data
    - Insert
    - Set
    - Delete
  + DDL (Data Definition Language) – create tables or drop them
    - Drop
    - Create
    - Alter
  + DQL (Data Query Language)
    - SELECT
  + DCL (Data Control Language) – grant permissions for users to your database
    - Grant
    - REVOKE
  + TCL (Transaction Control Language)
    - COMMIT
    - ROLLBACK
* Databases
  + Persist information in some sort of physical medium
    - Turn off the power information is still there
  + Relational databases
    - Store information in tables
    - These tables are connected to each other via foreign keys
  + Schema
    - Organization and rules of your database
    - Database design
* PL/SQL
  + Procedural Language /SQL
  + Procedure
    - Stored SQL script
    - Procedures have no restriction in what they are able to do
      * DML, DQL, DDL
    - Can have parameters
  + Functions
    - Cannot alter your database in anyway
      * NO DDL, NO DML
    - They always have at least one input
    - They always have an output
    - Aggregate functions
      * Average, sum, stdev
    - Scalar functions
      * Upper, round
    - You can create custom functions
* Normalization (the process of eliminating redundancy in your database)
  + 1nf
    - All fields are atomic (You cannot break down the field into more specific fields. No array like information in a field)
    - Primary key to uniquely identify every record
  + 2nf
    - In 1nf
    - No functional dependencies
      * You cannot calculate a field using other fields
  + 3nf
    - In 2nf
    - No transitive dependencies
      * Information stored in one table cannot be found in another table in the database
* Constraints
  + Rules on your fields
  + Prevent inconsistent values
    - Primary key – the unique identifier for every record in the table. Pk = unique + not null
    - Unique- every record must have a unique value in this field
    - Not null
    - Default
    - Check
    - Foreign key – Serves as a reference to the parent table
      * The foreign is placed on a child table
      * The foreign key field is not unique. But what it references must be unique.
* Multiplicities
  + One -one
  + One – many
  + Many- many (requires a junction table)
* JOINS and UNIONS/set operators
  + Joins
    - Stitch two tables horizontally based on a join predicate
    - It denormalizes your tables for easier querying
    - Varieties
      * LEFT JOIN
      * RIGHT JOIN
      * INNER JOIN
      * NATURAL JOIN
        + Inner join except you do not need a predicate because it automatically tries to join on a column named the same in each table.
      * CROSS JOIN
      * FULL JOIN
      * OUTER JOIN
  + Set operators
    - Stacks two tables on top of each other vertically
    - Varieties
      * UNION ALL
      * UNION
      * INTERSECT
* Transactions
  + Transaction is a logical set of SQL statements
  + Should follow ACID properties
    - Atomic
      * A transaction either completely works or it does not
    - Consistent
      * There is no situation where a database is saved/committed midway through a transaction
    - Isolated
      * Concurrent transactions do not step on each other
        + Similar to threading problems
    - Durable
      * If a transaction fails it fails gracefully. No loss of data or corruption
* DBMS
  + Database Management System
  + The software responsible for maintaining your tables and updating the information
    - MariaDB
    - MySQL
    - Oracle
    - Postgres

JDBC (Java Database Connectivity)

* Main Java SQL library for interaction with databases
* Core Interfaces
  + Connection – Represents a connection. Allows us to interact with database
  + PreparedStatement – Interface that allows us to safely create SQL statements
  + Statement- Version of PreparedStatement that is vulnerable to SQL injection. DO NOT USE
  + ResultSet – stores the result of a query
  + CallableStatement – used to call procedures