# SQL (Structured Query Language)

## RDBMS (Relational Database Management System

Collection of tables which we link together with relationships between the data

Used for CRUD (Create, Read, Update, and Delete) operations

Primary Keys - A column which uniquely identifies each row (unique and not null)

* Composite Key: A combination of columns

Foreign Keys - Data/Record/Column which references data from one table which has been placed in another

Candidate Key – Keys that can also be used as Primary Keys but aren’t primary keys (can uniquely identify each row)

Referential Integrity – ensuring that relationships between database entities remain consistent and exist

* Updating or deleting an entry can result in an update or delete of any records referencing those entries

Domain Integrity - Concept of ensuring that the values we store in our database are the appropriate size and type

DDL (Data Definition Language) - Define the structure of our database, column names and types

* CREATE
  + CREATE [TABLE\_NAME] (COLUMN NAME TYPE)
* DROP
  + DROP [TABLE\_NAME]
* TRUNCATE
  + TRUNCATE [TABLE\_NAME]
* ALTER
  + DROP/ADD COLUMNS
  + ALTER COLUMN
  + EX: ADD COLUMN EMAIL VARCHAR2(50);

DML (Data Manipulation Language) – Used to perform our CRUD operations

* INSERT
  + EX: INSERT INTO EMPLOYEES (EMPLOYEE ID, NAME, DEPT) VALUES (4,’Sally’, 1);
* UPDATE
  + UPDATE EMPLOYEE SET DEPT\_ID = 3 WHERE ID = 4;

Updates EMPLOYEE DEPT\_ID if the ID is 4

* + MULTIPLE CONDITION UPDATE:

UPDATE EMPLOYEE SET DEPT\_ID = 3 WHERE ID = 4 AND NAME = ‘Sally’;

Updates EMPLOYEE DEPT\_ID if the ID is 4 and the NAME is Sally

* DELETE
  + DELETE FROM EMPLOYEE WHERE ID = 2; Deletes records where the ID = 2
  + DELETE FROM EMPLOYEE; Deletes records, but keeps table intact
* SELECT
  + Technically DQL (Data Query Language), but belongs to the DML
  + SELECT \* FROM EMPLOYEE;

Selects and reads all the information stored in the EMPLOYEE table to the screen

* + SELECT NAME, DEPT\_ID FROM EMPLOYEE;

Selects all the info in the columns NAME, DEPT\_ID from the EMPLOYEE table

* + SELECT \* FROM EMPLOYEE ORDER BY NAME ASC;

Selects all the information in the EMPLOYEE table but orders them by name ascending

* + - DESC: orders by descending
    - HAVING

Transaction Control Language (TCL)- Deals with Transactions or a Unit of work on the database

* + One single DML statement
  + A group of statements you want to execute together
* COMMIT – Saves changes permanently to our database
* ROLLBACK – Returns the database to the state before the last COMMIT or at the last SAVEPOINT
* SAVEPOINT – Indicates a specific point for ROLLBACK to go back to

DCL (Data Control Language) – allows you to specify permission each role has

* GRANT – Allows you to give permissions
* REVOKE – Allows you to take them away

Constraints

* Foreign Key – Uniquely identifies a row/record in another table
* Primary Key – A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* Unique - Ensures that all values in a column are different
* Not Null - Ensures that a column cannot have a NULL value
* Default - Sets a default value for a column when no value is specified
* Check - Ensures that all values in a column satisfies a specific condition
* Index - Used to create and retrieve data from the database very quickly

Scalar vs. Aggregate

Scalar functions accept a discreet number of inputs

Scalar Keywords:

* Date/time function
* Math functions sqrt(#)

Aggregate except a range, and gives you a value based on the evaluation of that range

Aggregate Keywords Ex:

* Average
* Sim
* Count
* Min
* Max

Multiplicity Relationships

* One to One – Two entities in separate tables directly relating to another
  + Ex: Person to SSN
* One to Many – One entity in a table can have multiple relationships with other entities
  + Ex: Department to Employees
* Many to One – multiple entities sharing relationships with one entity
  + Ex: Students to University
* Many to Many – Multiple entities relating to each other
  + Ex: Student to Courses
    - Multiple Students can take a course and multiple courses can be assigned to a student
  + Represented by a Junction Table