* What is Data?
  + Facts and entities such as names, numbers, etc.
* What is a database?
  + A container of data.
* Why use a database?
  + Storage for large numbers of records.
  + Can store and update data in an efficient way.
* What constitutes a database?
  + Tables
  + Each table consists of columns and rows.
* What is a relational database?
  + Collective set of multiple data sets organized by table, records, and columns.
* What is SQL?
  + Structured Query Language
  + Used to communicate with a database, such as retrieving, storing, and manipulating data in databases.
* Database Types:
  + Relational Databases: Store data in a tabular form
    - Normalization is present.
    - RDBMS defines the integrity constraint for the purpose of ACID property.
    - Supports Distributed Databases
    - Handle large amount of data.
    - Support multiple users.
    - Examples:
      * MySQL
      * Access
      * Oracle
      * SQL Server
      * MariaDB
      * PostgreSQL
  + Non-relational Databases (Non-SQL databases): Stores data as files
    - Normalization is not present.
    - DMBS does not apply any security regarding data manipulation.
    - No relations between tables.
    - Does not support distributed databases.
    - Deals with small data.
    - Supports one single user.
    - Examples:
      * MongoDB
      * XML
      * Flat file
      * OODB
* Relational Database Storage
  + Table: Rows and columns
    - Tables -> Entity
    - Columns -> Fields
    - Rows -> Record
* Database Table’s Components
  + Primary Key: At least one column that uniquely identifies each row in a table.
  + Composite Primary Key: When two or more columns are used are primary keys.
* Relationship Types
  + One-to-Many
    - Publisher publishes book.
  + Many-to-Many
    - Book is written by Author.
  + One-to-One
    - The person has a Social Security Number.
* Relationship Between Two Tables:
  + Foreign Key: Same key as a primary key in a One-to-Many relationship where the primary key is on the table with one, and foreign is on the table with many.
* Column Properties
  + Data Type: Type of information stored in the column.
* SQL Statement Categories
  + DML (Data Manipulation Language): It lets you work with data in the database.
    - SELECT, INSERT, UPDATE, DELETE
  + DDL (Data Definition Language): It lets you work with the objects in the database.
    - CREATE DATABASE, TABLE, INDEX, ALTER TABLE, INDEX, DROP DATABASE, TABLE, INDEX
* SELECT Statement
  + Selects and retrieves data from a database.
  + The data returned is stored in a result table, called the result-set
  + Syntax:
    - SELECT select\_list
    - FROM table\_source
    - [WHERE search\_condition
    - [ORDER BY order\_of\_list]
  + The four clauses:
    - SELECT
    - FROM
    - WHERE
      * Filters the rows in the table so only those rows that match the condition are included in the result-set.
    - ORDER BY
      * A keyword used to sort the result-set in ascending/descending order.
  + Examples:
    - SELECT \* FROM Customers
      * Retrieve all columns from the table Customers
      * \* means all
    - SELECT CompanyName, ContactTitle
    - FROM Customers;
      * Specifies which columns to return
* SELECT Statement that JOIN Data
  + A SELECT that JOIN data from multiple tables using the Primary and Foreign Keys
  + INNER JOIN is the same as JOIN; INNER is optional. It selects records that have matching values.
* SQL Syntax (Clean vs Sloppy)
  + SELECT statement that is easy to read.
    - Start each new clause on a new line.
    - Break long clauses into multiple lines and indent continued lines.
    - Capitalize the first letter of each keyword and each word in column and table names.
    - End each statement with a semicolon.
    - Use comments only for portions of code that are difficult to understand.
* SQL VIEWS
  + A virtual table based on the result-set of an SQL statement.
  + It contains rows and columns, just like a real table.
  + Benefits:
    - Data restriction for certain users (Protect sensitive data)
    - Data access on through Views.
  + CREATE VIEW
* STORED PROCEDURE
  + A prepared SQL code that you can save, so the code can be reused repeatedly.
  + Benefits:
    - SQL statements in each procedure are only compiled and optimized the first time they are executed
    - Database performance improvement
  + Syntax:
    - CREATE PROCEDURE procedure\_name
    - AS
    - Sql\_statement;
  + Execute a Stored Procedure:
    - EXEC procedure\_name;