Introduction to SQL (Part II)

SQL Data Manipulation Commands

- The basic data manipulation commands are: INSERT, SELECT UPDATE, and DELETE.
- **INSERT INTO** – adds new rows/records to a table
  - *Syntax*: **INSERT INTO *table_name (columns)* VALUES *(values)*;**
  - *Example*: INSERT INTO Students (LastName, Section) VALUES ('Reyes', 'IT102');
  - To add new records to all the columns of a table
    - *Syntax*: **INSERT INTO** *table_name* **VALUES** *(values)*;
    - *Example*: INSERT INTO Students VALUES ('Reyes', 'IT102');
- **SELECT** – retrieves values of all rows or a subset of rows in a table
  - *Syntax*: **SELECT** *columns* **FROM** *table_name*;
  - *Example*: **SELECT** LastName, Section **FROM** Students;
  - To select all columns:
    - *Syntax*: **SELECT * FROM** *table_name*;
    - *Example*: SELECT * FROM Students;
- **DISTINCT** – an operator used with SELECT to retrieve unique values from columns in a table
  - *Syntax*: **SELECT DISTINCT** *columns* **FROM** *table_name*;
  - *Example*: SELECT DISTINCT Section FROM Students;
- **WHERE** – an option used with SELECT to filter the rows of data based on provided criteria
  - *Syntax*: **SELECT** *columns* **FROM** *table_name* **WHERE** *condition*;
  - *Example*: SELECT * FROM Students WHERE Section = 'IT101';
  - To select numeric fields, do not enclose in quotation marks. *Example:* SELECT * FROM Students WHERE Age >= 18;
- **IS NULL** – an operator used with SELECT to determine whether a field is empty or not
  - *Syntax*: **SELECT** *columns* **FROM** *table_name* **WHERE** *column* **IS NULL**;
  - *Example*: SELECT LastName, Section FROM Students WHERE Section IS NULL;
- **LIKE** – an operator used with WHERE to determine whether a value matches a given string pattern
  - *Syntax*: **SELECT** *columns* **FROM** *table_name* **WHERE** *column* **LIKE** *pattern*;
  - *Wildcards*: % represents zero, one, or multiple characters while _ represents a single character
  - *Example*: SELECT * FROM Students WHERE LastName LIKE '_b%';
  - *Meaning*: All students with last names that have 'b' in the second position.
- **IN** – an operator used with WHERE to check whether a value matches any value within a given list
  - *Syntax*: **SELECT** *columns* **FROM** *table_name* **WHERE** *column* **IN** *(values)*;
  - *Example*: SELECT * FROM Students WHERE Section IN ('IT101', 'IT102', 'IT103');
- **BETWEEN** – an operator used with WHERE to check whether a value is within a range
  - *Syntax*: **SELECT** *columns* **FROM** *table_name* **WHERE** *column* **BETWEEN** *value1* **AND** *value2*;
  - *Example*: SELECT * FROM Students WHERE Age BETWEEN 13 AND 15;
- **ORDER BY** – An option used with SELECT to sort retrieved values in ascending or descending order
  - *Syntax*: **SELECT** *columns* **FROM** *table_name* **ORDER BY** *columns*;
  - *Example*: SELECT * FROM STUDENTS ORDER BY LastName
  - To sort values in descending order: SELECT * FROM *table_name* ORDER BY *columns* DESC;
- **UPDATE** – modifies existing records in a table
  - *Syntax*: **UPDATE** *table_name* **SET** *column1 = value1, …* **WHERE** *condition*;
  - *Example*: UPDATE Students SET Section = 'IT202', Status = 'Irregular' WHERE StudentID = '2018-100013'*;*
- **DELETE** – removes existing records in a table
  - Syntax: **DELETE FROM** *table_name* **WHERE** *condition*;
  - DELETE FROM Students WHERE StudentID = '2018-100013';
  - To delete all records: DELETE FROM *table_name*;

Reference:
Coronel, C. and Morris, S. (2017). *Database systems: design, implementation, and management* (12th ed.). USA: Cengage Learning.