SQL

A relational database is a database that organizes information into one or more tables. Here, the relational database contains one table. A column is a set of data values of a particular type. A row is a single record in a table. A statement is text that the database recognizes as a valid command.  Clauses perform specific tasks in SQL. By convention, clauses are written in capital letters. Clauses can also be referred to as commands.

Constraints that add information about how a column can be used are invoked after specifying the data type for a column. They can be used to tell the database to reject inserted data that does not adhere to a certain restriction.

Examples: PRIMARY KEY, UNIQUE (have a different value for every row), NOT NULL, DEFAULT (take an additional argument).

The AND/ OR operators can be used to combine multiple conditions in a WHERE clause to make the result set more specific and useful.

If the BETWEEN operator is use with letters is not inclusive of the second letter. If it is used with numbers, the second number is included.

CREATE TABLE table\_name ( /creates a new table

column\_1 data\_type constraits\_example

column\_2 data\_type constraits\_example

);

INSERT INTO table\_name (column\_name, column\_name) /inserts new rows into the table

VALUES (value\_1, value\_2);

UPDATE table\_name /edits a row

SET column\_name = new\_value

WHERE column\_name = value; /indicates which row to update

ALTER TABLE table\_name /adds a new column

ADDS COLUMN column\_name data\_type;

DELETE FROM table\_name /deletes rows

WHERE column\_name = value;

SELECT column\_1, column\_2 / \* FROM table\_name; /query data from a table

SELECT column\_name AS “new\_name” FROM table\_name; /rename a column in the result only

SELECT DISTINCT clumn\_name FROM table\_name; /filters out all duplicate values

SELECT column\_name /filters the result where the condition is met

FROM table\_name

WHERE condition\_example AND/OR condition\_example; /can take comparisons operators

WHERE column\_name LIKE “str\_ng” / “%trin%”; /search for a specific pattern; LIKE isn’t case

sensitive.

WHERE column\_name BETWEEN example AND example; /filters the result within a certain range

SELECT column\_name

FROM table\_name

ORDER BY table\_name DESC / ASC; /sorts the result alphabetically or

numerically; if present, it goes after WHERE

LIMIT number; /specifies the maximum numbers of rows

SELECT column\_name

CASE /displays a new column with if/then logic

WHEN condition\_example THEN “example”

ELSE “example”

END AS “name” /renames this new column

FROM table\_name;

SELECT

COUNT(column\_name) /counts the number of non-empty values

SUM(column\_name) /returns the sum of all the values

MIN/MAX(column\_name) /returns the lowest/highest value

AVG(column\_name) /calculates the average value

ROUND (column\_name, integer) /rounds the value to the number of decimals

specified by the integer

FROM table\_name;

GROUP BY column\_name/number\_of\_selected\_column; /arrange identical data into groups; comes

after WHERE and before ORDER BY/LIMIT

HAVING condition\_axample; /limits the result based on an aggregate

property