**JOINS**

**Part 1:** In SQL, we have the capability to execute Inner Joins, which will return only the values that the two tables have in common. Left Joins will return all values from the left table, as well as values that both tables have in common. Right Joins operate similarly but will include all values from the right table. Full Outer Joins return all values from both tables. If there are not matching rows in the tables when the join is executed, the resulting table will show NULL values for these.

**Part 2:** a LEFT JOIN was executed to get the resulting table

**ALTER vs. UPDATE**

**Part 1:** ALTER TABLE changes the structure of a table. It can be used for many purposes, such as adding/dropping a column, renaming a column/table, changing the data type of a column, and adding constraints to a column. UPDATE allows you to change the data within a table, or manually input new values for a certain field.

**Part 2:**

ALTER TABLE table\_name

RENAME COLUMN department\_id TO dept\_id;

---------------------------------------------------------------

ALTER TABLE table\_name

ADD COLUMN annual\_salary INT;

**DML and DDL**

**Part 1:** DML is Data Manipulation Language and is used to insert, update, or delete records. Examples include the INSERT, UPDATE, and DELETE commands. DDL is Data Definition Language and is used to create/alter the tables or database. Examples of DDL include the CREATE, DROP, RENAME and ALTER commands.

**Part 2:**

UPDATE table\_name

SET vendor\_country = ‘France’

WHERE id = 3;

**Duplicates**

**Part 1:** To locate a duplicate record, you use the HAVING COUNT command to see which fields occur in the table more than once. For example, with the table provided, to find duplicated within one field:

SELECT yarn\_name,

COUNT(yarn\_name)

FROM duplicated\_yarn

GROUP BY yarn\_name,

HAVING COUNT(yarn\_name) > 1

---------------------------------------------------------------------------------------------------------------------------

For finding duplicate values using more than one field, you could use the following query:

SELECT yarn\_name,

vendor\_id,

COUNT(yarn\_name),

COUNT(vendor\_id)

FROM duplicated\_yarn

GROUP BY yarn\_name,

vendor\_id

HAVING COUNT(yarn\_name) > 1

AND COUNT(vendor\_id) > 1

**GROUPBY**

SELECT state,

AVG(duration) as mean\_duration

FROM usa\_ufo

GROUPBY state