* I chose to use the Postgres resource because I have it downloaded when I was trying to learn myself how to use it.
* The stuff I learned before and now include a variety of things that include creating databases, creating tables, inserting/deleting columns, inserting/deleting rows, join two tables based off certain commonalities, etc.
* There are different types of joins you can perform such as a natural join based off common columns. A natural join can be an inner or left or right outer join.
* I jumped straight into the exercises to start working on Postgres. I learned all the basic things which includes querying certain things from the database using different clauses.
* I learned how to essentially give an if condition and to display a string if true and another string if it is false. The output would be displayed in a table.
* I learned the *distinct* keyword which removes duplicates when performing a query.
* You can also join a table with another matching column with the same type.
* I got up to the part where complex queries are performed. The commands are really long and complex and they use joins. These complex queries use subqueries in which some are easy to do, however, others require a great amount of detail.
* For example, one of the queries I came across was lengthy and detailed which is shown on the following bullet.

**select** member, facility, cost **from** (

**select**

mems.firstname || ' ' || mems.surname **as** member,

facs.name **as** facility,

**case**

**when** mems.memid = 0 **then**

bks.slots\*facs.guestcost

**else**

bks.slots\*facs.membercost

**end** **as** cost

**from**

cd.members mems

**inner** **join** cd.bookings bks

**on** mems.memid = bks.memid

**inner** **join** cd.facilities facs

**on** bks.facid = facs.facid

**where**

bks.starttime >= '2012-09-14' **and**

bks.starttime < '2012-09-15'

) **as** bookings

**where** cost > 30

**order** **by** cost **desc**;

* As you can see this requires a great deal of thinking to come up with to get the required output.
* I tried some commands myself on my Postgres, I will include screenshots below.

When creating a table you have to specify the character type for each field such as int or varchar, then you have to insert data which takes quite a while.

Text

Description automatically generated

Joined two tables using employeeid as the common reference or foreign key is the term used I believe

A screen shot of a computer

Description automatically generated

No duplicates using keyword distinct

Text

Description automatically generated

Shows all the fields from student table

A picture containing diagram

Description automatically generated

Arithmetic is used to calculate grades from student table

Text

Description automatically generated

\*END OF LAB 9\*