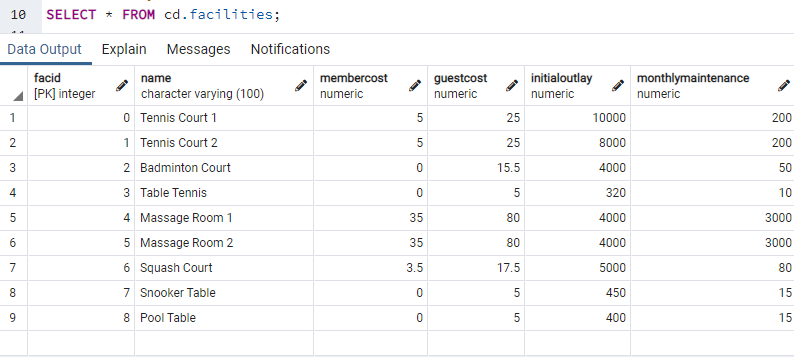
**SQL QUERIES**

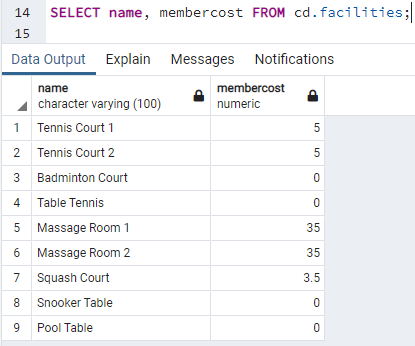
1.How can you retrieve all the information from the cd.facilities table?

**Solution:**



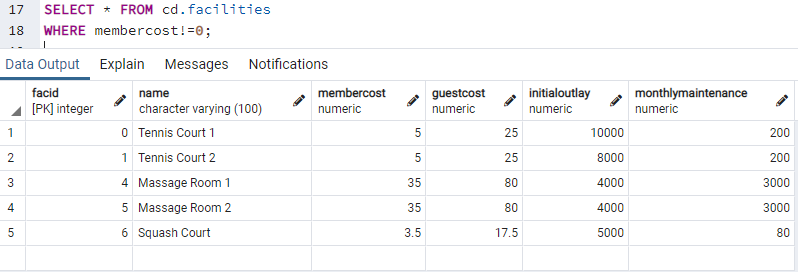
2.You want to print out a list of all of the facilities and their cost to members. How would you retrieve a list of only facility names and costs?

**Solution:**



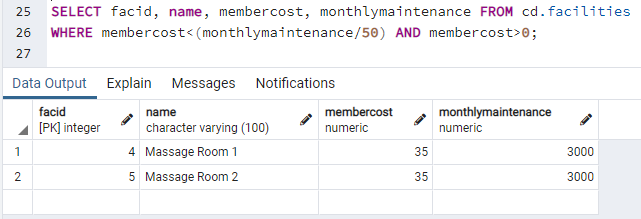
3.How can you produce a list of facilities that charge a fee to members?

**Solution:** The list should include all the facilities having corresponding cost other than zero. It can be done using ‘WHERE’ Clause.



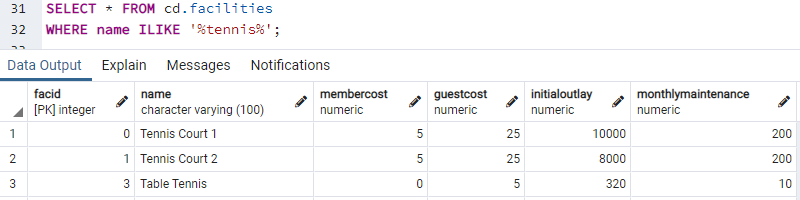
4.How can you produce a list of facilities that charge a fee to members, and that fee is less than 1/50th of the monthly maintenance cost? Return the facid, facility name, member cost, and monthly maintenance of the facilities in question.

**Solution:** To produce a list of facilities that charge a fee to members, and that fee is less than 1/50th of the monthly maintenance cost can simply be obtained by dividing monthly maintenance cost by 50.



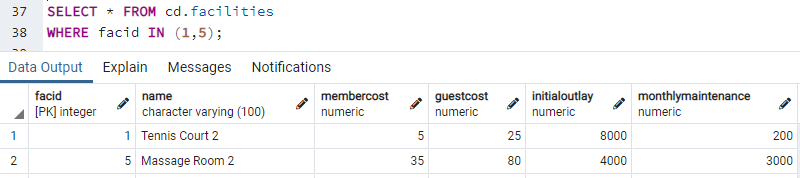
5.How can you produce a list of all facilities with the word 'Tennis' in their name?

**Solution**: ILIKE is case-insensitive and using ‘%tennis%’ produces the desired outcome.



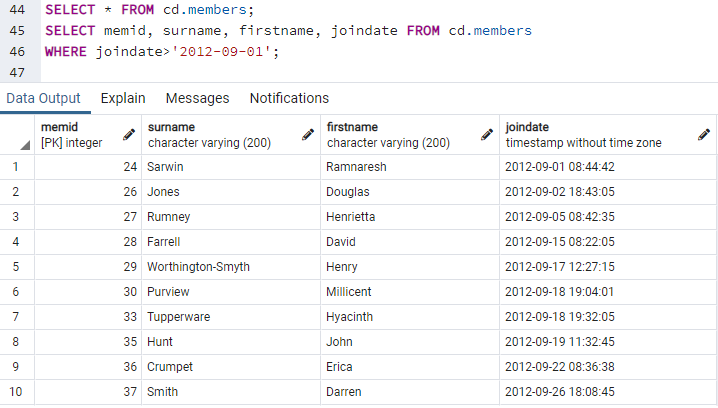
6.How can you retrieve the details of facilities with ID 1 and 5? Try to do it without using the OR operator.

**Solution:** The details of facilities with ID 1 and 5 can be found using ‘IN’ operator instead of ‘OR’.



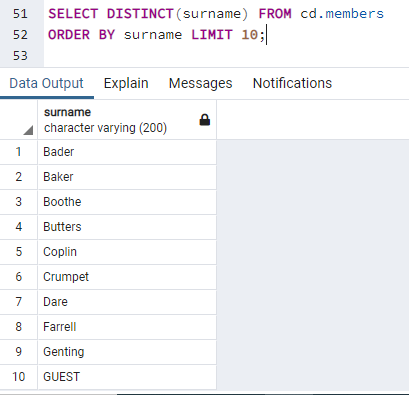
7.How can you produce a list of members who joined after the start of September 2012? Return the memid, surname, firstname, and joindate of the members in question.

**Solution:** The desired output can easily be obtained by using ‘>’ condition in where clause.



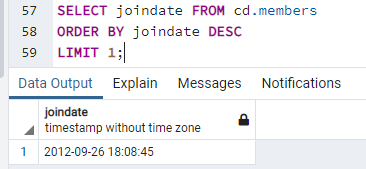
8.How can you produce an ordered list of the first 10 surnames in the members table? The list must not contain duplicates.

**Solution:** We use DISTINCT command so that our output does not have duplicate values. ORDER BY helps to obtain a ordered list and to obtain 10 surnames we use LIMIT.



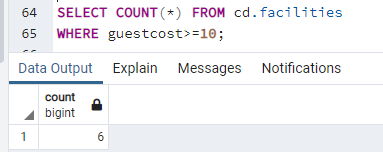
9.You'd like to get the signup date of your last member. How can you retrieve this information?

**Solution**: To get a single value from end we make use of ORDER BY DESC and LIMIT 1.



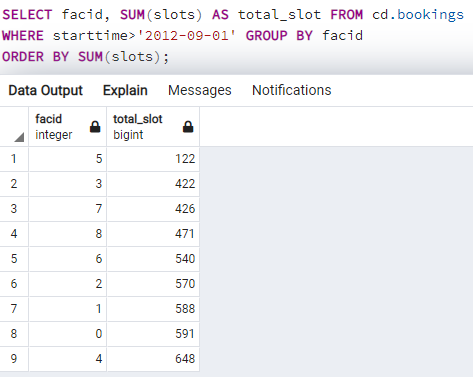
10.Produce a count of the number of facilities that have a cost to guests of 10 or more.

**Solution:** Simply using COUNT and ‘>=’ condition in WHERE gives desired result.



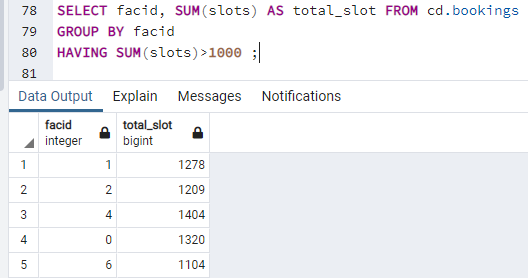
11.Produce a list of the total number of slots booked per facility in the month of September 2012. Produce an output table consisting of facility id and slots, sorted by the number of slots.

**Solution:** To get total slots by faculty id we need to use ‘GROUP BY facid’ and we apply ‘>’ operator to filter date. We wish to find the values sorted by number of slots so ORDER BY helps here.



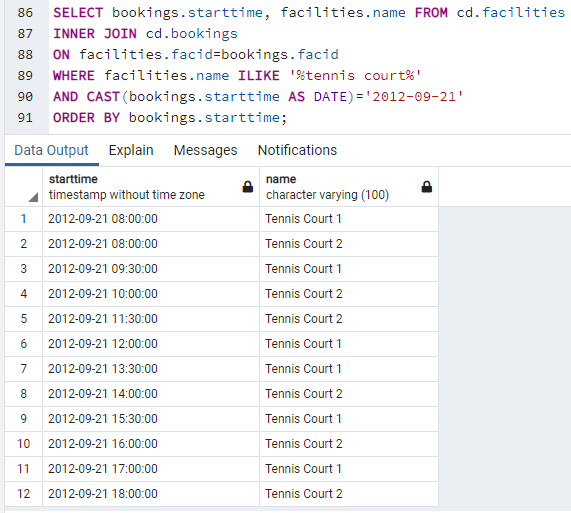
12.Produce a list of facilities with more than 1000 slots booked. Produce an output table consisting of facility id and total slots, sorted by facility id.

**Solution**: To Produce a list of facilities with more than 1000 slots booked sorted by facility id we use GROUP BY along with HAVING.



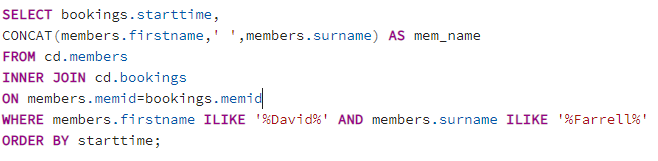
13.How can you produce a list of the start times for bookings for tennis courts, for the date '2012-09-21'? Return a list of start time and facility name pairings, ordered by the time.

**Solution**: An ordered list of the start times for bookings for tennis courts, for the date '2012-09-21' can be obtained by query as shown. INNER JOIN is applied on two tables facilities and bookings, both contains common column facid. To obtain the result only for tennis court we use ILIKE command. We may need to convert starttime to DATE format before directly equating to the given date.

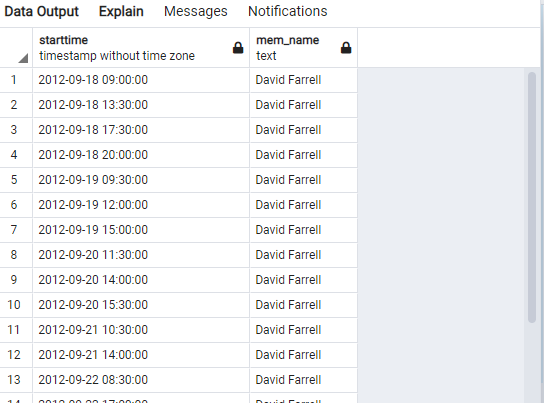


14.How can you produce a list of the start times for bookings by members named 'David Farrell'?

**Solution:** The starttime column is in bookings table but the bookings table does not contain first name and surname therefore we use INNER JOIN on tables members and bookings. Then using ILIKE, ORDER BY along WHERE helps.



The output has 34 rows.



**SQL DATABASE**

Complete the following task:

Create a new database called "School" this database should have two tables: teachers and students.

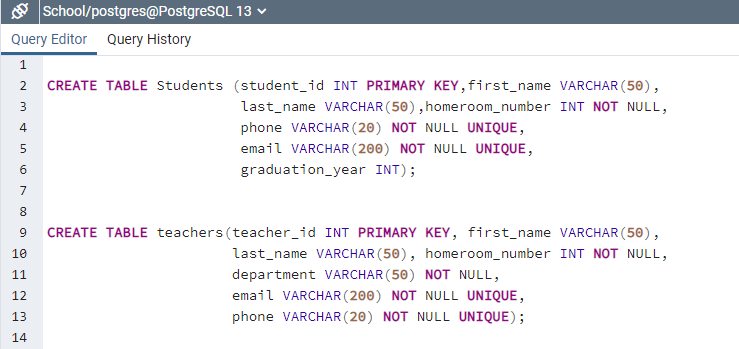
The students table should have columns for student\_id, first\_name, last\_name, homeroom\_number, phone, email, and graduation year.

The teachers table should have columns for teacher\_id, first\_name, last\_name, homeroom\_number, department, email, and phone.

The constraints are mostly up to you, but your table constraints do have to consider the following:

* 1. We must have a phone number to contact students in case of an emergency.
  2. We must have ids as the primary key of the tables
  3. Phone numbers and emails must be unique to the individual.

**Solution:**



Once you've made the tables, insert a student named Mark Watney (student\_id=1) who has a phone number of 777-555-1234 and doesn't have an email. He graduates in 2035 and has 5 as a homeroom number.

Then insert a teacher names Jonas Salk (teacher\_id = 1) who as a homeroom number of 5 and is from the Biology department. His contact info is: jsalk@school.org and a phone number of 777-555-4321.

Keep in mind that these insert tasks may effect your constraints!

**SOLUTION:**

