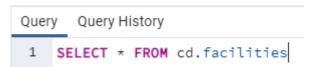
Database Management with PostgreSQL

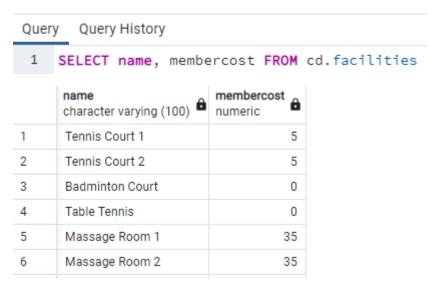
By J. Jaya Prakash

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



	facid [PK] integer	name character varying (100)	membercost numeric	guestcost numeric	initialoutlay numeric	monthlymaintenance numeric
1	0	Tennis Court 1	5	25	10000	200
2	1	Tennis Court 2	5	25	8000	200
3	2	Badminton Court	0	15.5	4000	50
4	3	Table Tennis	0	5	320	10
5	4	Massage Room 1	35	80	4000	3000
6	5	Massage Room 2	35	80	4000	3000

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?



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

```
Query Query History

1 SELECT name, membercost FROM cd.facilities
2 WHERE membercost > 0
```

	name character varying (100)	membercost numeric
1	Tennis Court 1	5
2	Tennis Court 2	5
3	Massage Room 1	35
4	Massage Room 2	35
5	Squash Court	3.5

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.

```
Query Query History
     SELECT facid, name, membercost, monthlymaintenance FROM cd.facilities
 1
     WHERE membercost > 0 AND membercost < monthlymaintenance/50
      facid
                                         membercost
                                                       monthlymaintenance
      [PK] integer
                   character varying (100) *
                                                       numeric
1
                   Massage Room 1
                                                                     3000
                                                  35
2
                5
                   Massage Room 2
                                                   35
                                                                     3000
```

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



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

```
Query Query History

1 SELECT * FROM cd.facilities
2 WHERE facid in (1, 5)
```

	facid [PK] integer	name character varying (100)	membercost numeric	guestcost numeric	initialoutlay numeric	monthlymaintenance numeric
1	1	Tennis Court 2	5	25	8000	200
2	5	Massage Room 2	35	80	4000	3000

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.

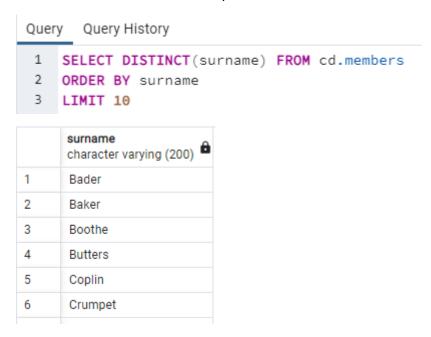
```
Query Query History

1 SELECT memid, surname, firstname, joindate FROM cd.members
2 WHERE joindate > '2012-09-01'

memid . surname . firstname . joindate
```

	memid [PK] integer	surname character varying (200)	firstname character varying (200)	joindate timestamp without time zone
1	24	Sarwin	Ramnaresh	2012-09-01 08:44:42
2	26	Jones	Douglas	2012-09-02 18:43:05
3	27	Rumney	Henrietta	2012-09-05 08:42:35
4	28	Farrell	David	2012-09-15 08:22:05
5	29	Worthington-Smyth	Henry	2012-09-17 12:27:15

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



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



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



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.



	facid integer	â	sum bigint	â
1		4		648
2		0		591
3		1		588
4		2		570
5		6		540
6		8		471

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.

Query Query History 1 SELECT facid, SUM(slots) FROM cd.bookings 2 GROUP BY facid 3 HAVING SUM(slots) > 1000 4 ORDER BY facid

	facid integer	â	sum bigint	â
1		0		1320
2		1		1278
3		2		1209
4		4		1404
5		6		1104

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.

```
Query Query History

1 SELECT starttime, name FROM cd.facilities
2 INNER JOIN cd.bookings
3 ON cd.facilities.facid = cd.bookings.facid
4 WHERE starttime >= '2012-09-21' AND starttime < '2012-09-22'
5 AND name ILIKE 'Tennis%'
6 ORDER BY starttime
```

	starttime timestamp without time zone	name character varying (100)
1	2012-09-21 08:00:00	Tennis Court 1
2	2012-09-21 08:00:00	Tennis Court 2
3	2012-09-21 09:30:00	Tennis Court 1
4	2012-09-21 10:00:00	Tennis Court 2
5	2012-09-21 11:30:00	Tennis Court 2
6	2012-09-21 12:00:00	Tennis Court 1

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

Query Query History 1 SELECT starttime, firstname, surname FROM cd.bookings 2 INNER JOIN cd.members 3 ON cd.bookings.memid = cd.members.memid 4 WHERE firstname = 'David' AND surname = 'Farrell' 5 ORDER BY starttime

starttime timestamp without time zone	firstname character varying (200)	surname character varying (200)
2012-09-18 09:00:00	David	Farrell
2012-09-18 13:30:00	David	Farrell
2012-09-18 17:30:00	David	Farrell
2012-09-18 20:00:00	David	Farrell
2012-09-19 09:30:00	David	Farrell
2012-09-19 12:00:00	David	Farrell
	timestamp without time zone 2012-09-18 09:00:00 2012-09-18 13:30:00 2012-09-18 17:30:00 2012-09-18 20:00:00 2012-09-19 09:30:00	timestamp without time zone character varying (200) 2012-09-18 09:00:00 David 2012-09-18 13:30:00 David 2012-09-18 17:30:00 David 2012-09-18 20:00:00 David 2012-09-19 09:30:00 David

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.

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.

```
CREATE TABLE students(
  1
  2
                       student_id SERIAL PRIMARY KEY,
  3
                       first_name VARCHAR(50) NOT NULL,
                       homeroom number SMALLINT UNIQUE NOT NULL,
  4
  5
                       phone INTEGER UNIQUE NOT NULL,
  6
                       email VARCHAR(100) UNIQUE NOT NULL,
  7
                       graduation_year SMALLINT NOT NULL
  8
         );
Query Query History
  1
            CREATE TABLE teachers(
   2
                       teacher_id SERIAL PRIMARY KEY,
   3
                       first_name VARCHAR(50) NOT NULL,
   4
                       last_name VARCHAR(50) NOT NULL,
   5
                       homeroom number SMALLINT UNIQUE NOT NULL,
   6
                       department VARCHAR(50) NOT NULL,
  7
                       email VARCHAR(100) UNIQUE NOT NULL,
   8
                       phone INTEGER UNIQUE NOT NULL
   9
            );
Query Query History
  1 ALTER TABLE students
  2 ADD last_name VARCHAR(50) NOT NULL;
Query Query History
  1 ALTER TABLE students ALTER COLUMN phone SET DATA TYPE VARCHAR(20);
Query Query History
         INSERT INTO students (first_name, last_name, homeroom_number,
  1
  2
                                                                          phone, graduation_year, email)
        VALUES ('Mark', 'Watney', 5, '7775551234', 2035, 'unknown');
Query Query History
        INSERT INTO teachers (first_name, last_name, homeroom_number,
  2
                                                                  phone, department, email)
  3 VALUES ('Jonas', 'Salk', 5, 777-555-4321, 'Biology', 'jsalk@school.org');
       student_id [PK] integer | first_name | homeroom_number | phone | character varying (50) | character varying (50) | character varying (20) | character varying (100) | character varying (100) | character varying (50) | char
                                                                              5 7775551234
                    3 Mark
                                                                                                               unknown
                                                                                                                                                              2035 Watney
```