

PROJECT -1

1. SQL QUERIES

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

Exercise /postgres@PostgreSQL 13 ▾

Query Editor Query History


Sc

1 **SELECT** * **FROM** cd.facilities;

Data Output Explain Messages Notifications

	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
7	6	Squash Court	3.5	17.5	5000	80
8	7	Snooker Table	0	5	450	15
9	8	Pool Table	0	5	400	15



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?

 Exercise /postgres@PostgreSQL 13 ▾


Query Editor Query History

1 **SELECT** name,membercost **FROM** cd.facilities;

Data Output Explain Messages Notifications

	name character varying (100)	 membercost numeric	
1	Tennis Court 1	5	
2	Tennis Court 2	5	
3	Badminton Court	0	
4	Table Tennis	0	
5	Massage Room 1	35	
6	Massage Room 2	35	
7	Squash Court	3.5	
8	Snooker Table	0	
9	Pool Table	0	

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

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

1

SELECT * FROM cd.facilities

2

WHERE membercost>0;

Data Output


Explain

Messages

Notifications

	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	4	Massage Room 1	35	80	4000	3000
4	5	Massage Room 2	35	80	4000	3000
5	6	Squash Court	3.5	17.5	5000	80

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.


Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

```

1 SELECT facid, name, membercost,monthlymaintenance
2 FROM cd.facilities
3 WHERE membercost>0 AND membercost<monthlymaintenance/50;
4 -----or-----
5 SELECT facid, name, membercost,monthlymaintenance
6 FROM cd.facilities
7 WHERE membercost>0 AND membercost<monthlymaintenance*0.02;---since 1/50=0.02;---
```

Data Output

Explain

Messages

Notifications

	facid [PK] integer	name character varying (100)	membercost numeric	monthlymaintenance numeric
1	4	Massage Room 1	35	3000
2	5	Massage Room 2	35	3000

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

Exercise /postgres@PostgreSQL 13 ▾							
Query Editor Query History							St
<pre>1 SELECT * FROM cd.facilities 2 WHERE name ILIKE '%tennis%';</pre>							
Data Output Explain Messages Notifications							
	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	3	Table Tennis	0	5	320	10	


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

Exercise /postgres@PostgreSQL 13 ▾							
Query Editor Query History							St
<pre>1 SELECT * FROM cd.facilities 2 WHERE facid IN (1, 5);;</pre>							
Data Output Explain Messages Notifications							
	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.

Exercise /postgres@PostgreSQL 13 ▾				
Query Editor Query History				
<pre> 1 SELECT memid, surname, firstname, joindate 2 FROM cd.members 3 WHERE joindate >='2012-09-01'; </pre>				
Data Output Explain Messages Notifications				
	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
6	30	Purview	Millicent	2012-09-18 19:04:01
7	33	Tupperware	Hyacinth	2012-09-18 19:32:05
8	35	Hunt	John	2012-09-19 11:32:45
9	36	Crumpet	Erica	2012-09-22 08:36:38
10	37	Smith	Darren	2012-09-26 18:08:45

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

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

1

2

3

4


SELECT DISTINCT surname
FROM cd.members
ORDER BY surname **LIMIT 10**;

Data Output


Explain

Messages

Notifications



surname
character varying (200)



1

Bader

2

Baker

3

Boothe

4

Butters

5

Coplin

6

Crumpet

7

Dare

8

Farrell


9

Genting

10

GUEST

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

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

1

2

3

4


```
SELECT MAX(joindate)
FROM cd.members;
```

Data Output


Explain

Messages

Notifications




max
timestamp without time zone



12012-09-26 18:08:45

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

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

1

2

3

4


```
SELECT COUNT(*)
FROM cd.facilities
WHERE guestcost >= 10;
```

Data Output


Explain

Messages

Notifications




count
bigint



1

6

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.

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

```
1 SELECT facid, SUM(slots) AS slots
2 FROM cd.bookings
3 WHERE starttime<='2012-09-30'AND starttime>='2012-09-01'
4 GROUP BY facid
5 ORDER BY SUM(slots);
6
```

Data Output


Explain

Messages

Notifications

	facid integer	slots bigint
1	5	118
2	3	410
3	7	412
4	8	453
5	6	522
6	2	546
7	0	567
8	1	567
9	4	628

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.

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor

Query History

```
1 SELECT facid, SUM(slots) AS total_slots
2 FROM cd.bookings
3 GROUP BY facid
4 HAVING SUM(slots)>1000
5 ORDER BY facid;
6
```

Data Output


Explain

Messages

Notifications

	facid integer		total_slots bigint	
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.

 Exercise /postgres@PostgreSQL 13 ▾

Query Editor Query History

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

Data Output Explain Messages Notifications

	starttime timestamp without time zone	name character varying (100)
1	2012-09-21 08:00:00	Tennis Court 1
2	2012-09-21 09:30:00	Tennis Court 1
3	2012-09-21 12:00:00	Tennis Court 1
4	2012-09-21 13:30:00	Tennis Court 1
5	2012-09-21 15:30:00	Tennis Court 1
6	2012-09-21 17:00:00	Tennis Court 1
7	2012-09-21 08:00:00	Tennis Court 2
8	2012-09-21 10:00:00	Tennis Court 2
9	2012-09-21 11:30:00	Tennis Court 2
10	2012-09-21 14:00:00	Tennis Court 2
11	2012-09-21 16:00:00	Tennis Court 2
12	2012-09-21 18:00:00	Tennis Court 2

2. SQL DATABASE

*Creating database “student ”

Create - Database

General

Definition

Security

Parameters

Advanced

SQL

Database

School

Owner

postgres

Comment

i

?

Cancel

Reset

Save

*Creating students table

The screenshot shows a PostgreSQL query editor interface. At the top, there is a toolbar with icons for database operations. Below the toolbar, the connection name 'School/postgres@PostgreSQL 13' is displayed. The interface has two tabs: 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, showing a SQL query to create a table named 'students'. The query is as follows:

```
1 CREATE TABLE students(  
2 student_id serial PRIMARY KEY,  
3 first_name VARCHAR(50) NOT NULL,  
4 last_name VARCHAR(50) NOT NULL,  
5 homeroom_number integer,  
6 phone VARCHAR(255) NOT NULL UNIQUE,  
7 email VARCHAR(50) UNIQUE,  
8 graduation_year integer  
9 );
```

Below the query editor, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the output of the query:

```
CREATE TABLE
```

Below the output, a message states: 'Query returned successfully in 240 msec.'

*Creating teachers table

School/postgres@PostgreSQL 13 ▾

Query Editor 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 integer,  
6 department VARCHAR(50),  
7 phone VARCHAR(255) UNIQUE,  
8 email VARCHAR(50) UNIQUE  
9 );
```

Data Output Explain Messages Notifications

CREATE TABLE

Query returned successfully in 180 msec.

Output:

Query Editor Query History Scratch Pad Fn Ctr

```
1 SELECT * FROM students;
```

Data Output Explain Messages Notifications

student_id [PK] integer	first_name character varying (50)	last_name character varying (50)	homeroom_number integer	phone character varying (255)	email character varying (50)	graduation integer
----------------------------	--------------------------------------	-------------------------------------	----------------------------	----------------------------------	---------------------------------	-----------------------

School/postgres@PostgreSQL 13

Query Editor

Query History

Scratch Pad

1 **SELECT** * **FROM** teachers;

Data Output

Explain

Messages

Notifications

teacher_id	first_name	last_name	homeroom_number	department	phone	email
[PK] integer	character varying (50)	character varying (50)	integer	character varying (50)	character varying (255)	characte

*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.

School/postgres@PostgreSQL 13

Query Editor

Query History

Scratch Pad

1 **INSERT INTO** students (student_id, first_name, last_name, homeroom_number, phone, grad_year)
2 **VALUES** (1, 'Mark', 'Watney', 5, '777-555-1234', 2035);
3 **SELECT** * **FROM** students;

Data Output

Explain

Messages

Notifications

student_id	first_name	last_name	homeroom_number	phone	email	graduation_year
[PK] integer	character varying (50)	character varying (50)	integer	character varying (255)	character varying (50)	integer
1	Mark	Watney	5	777-555-1234	[null]	2035

*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.

School/postgres@PostgreSQL 13

Query EditorQuery HistoryScratch Pad

```
1 INSERT INTO teachers (teacher_id, first_name, last_name, homeroom_number, department, phone, email)
2 VALUES (1, 'Jonas', 'Salk', 5, 'Biology', '777-555-4321', 'jsalk@school.org');
3 SELECT * FROM teachers;
```

Data OutputExplainMessagesNotifications

teacher_id [PK] integer	first_name character varying (50)	last_name character varying (50)	homeroom_number integer	department character varying (50)	phone character varying (255)	email character varying (50)
1	Jonas	Salk	5	Biology	777-555-4321	jsalk@school.org