

-- BASIC 1

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

```
select * from cd.facilities
```

-- 1.2 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?

```
SELECT NAME, MEMBERCOST FROM CD.FACILITIES
```

-- 1.3 How can you produce a list of facilities that charge a fee to members?

```
SELECT * FROM CD.FACILITIES  
WHERE MEMBERCOST > 0
```

--1.4 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.

```
SELECT FACID,NAME, MEMBERCOST, MONTHLYMAINTENANCE FROM CD.FACILITIES  
WHERE  
MEMBERCOST > 0 AND MEMBERCOST < MONTHLYMAINTENANCE/50.0
```

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

```
SELECT * FROM CD.FACILITIES  
WHERE NAME LIKE '%Tennis%'
```

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

```
SELECT * FROM CD.FACILITIES
```

```
WHERE FACID IN (1,5);
```

-- 1.7 How can you produce a list of facilities, with each labelled as 'cheap' or 'expensive' depending on if their monthly maintenance cost is more than \$100? Return the name and monthly maintenance of the facilities in question

```
SELECT NAME,  
CASE WHEN MONTHLYMAINTENANCE > 100 THEN  
    'expensive'  
else  
    'cheap'  
end as cost  
from cd.facilities
```

-- 1.8 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.

```
SELECT MEMID,SURNAME,FIRSTNAME,JOINDATE FROM CD.MEMBERS  
WHERE JOINDATE >= '2012-09-01'
```

--1.9 How can you produce an ordered list of the first 10 surnames in the members table? The list must not contain duplicates.

```
SELECT DISTINCT SURNAME FROM CD.MEMBERS  
ORDER BY SURNAME  
LIMIT 10
```

--1.10 You, for some reason, want a combined list of all surnames and all facility names. Yes, this is a contrived example :-). Produce that list!

```
SELECT SURNAME FROM CD.MEMBERS  
UNION  
SELECT NAME FROM CD.FACILITIES;
```

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

```
SELECT JOINDATE AS LATEST FROM CD.MEMBERS  
ORDER BY JOINDATE DESC  
LIMIT 1
```

--1.12 You'd like to get the first and last name of the last member(s) who signed up - not just the date. How can you do that?

```
SELECT FIRSTNAME,SURNAME,JOINDATE FROM CD.MEMBERS  
ORDER BY JOINDATE DESC  
LIMIT 1
```

-- 2 .Joins and Subqueries

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

```
SELECT B.STARTTIME FROM CD.BOOKINGS B  
INNER JOIN CD.MEMBERS M USING(MEMID)  
WHERE M.FIRSTNAME = 'David' and SURNAME = 'Farrell'
```

-- 2.2 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.

```
SELECT B.STARTTIME AS START,F.NAME FROM CD.BOOKINGS B  
INNER JOIN CD.FACILITIES F USING(FACID)  
WHERE F.NAME LIKE '%Tennis Court%'  
AND DATE(B.STARTTIME) = '2012-09-21'  
ORDER BY B.STARTTIME;
```

-- 2.3 How can you output a list of all members who have recommended another member?

-- Ensure that there are no duplicates in the list, and that results are ordered by (surname, firstname)

```
SELECT DISTINCT M2.FIRSTNAME,M2.SURNAME FROM CD.MEMBERS M1
```

```
INNER JOIN
```

```
CD.MEMBERS M2 ON M2.MEMID = M1.RECOMMENDED BY
```

```
ORDER BY SURNAME,FIRSTNAME
```

```
SELECT * FROM CD.MEMBERS
```

```
select distinct firstname as firstname,surname as surname
```

```
from
```

```
cd.members
```

```
WHERE MEMID IN (SELECT RECOMMENDED BY FROM CD.MEMBERS)
```

```
order by surname,firstname;
```

--2.4 How can you output a list of all members, including the individual who recommended them (if any)? Ensure that results are ordered by (surname, firstname).

```
SELECT MEM.FIRSTNAME AS MEMFNAME, MEM.SURNAME AS MEMSNAME, REF.FIRSTNAME AS  
RECFNAME, REF.SURNAME AS RECNAME
```

```
FROM CD.MEMBERS MEM
```

```
LEFT JOIN
```

```
CD.MEMBERS REF
```

```
ON REF.MEMID = MEM.RECOMMENDED BY
```

```
ORDER BY MEM.SURNAME , MEM.FIRSTNAME
```

-- 2.5 How can you produce a list of all members who have used a tennis court? Include in your output the name of the court, and the name of the member formatted as a single column. Ensure no duplicate data, and order by the member name followed by the facility name.

```
SELECT DISTINCT M.FIRSTNAME || '' || M.SURNAME AS MEMBER, F.NAME AS FACILITY FROM  
CD.MEMBERS M
```

```
INNER JOIN CD.BOOKINGS B USING(MEMID)
```

```
INNER JOIN CD.FACILITIES F USING(FACID)
WHERE F.NAME LIKE '%Tennis Court%'
ORDER BY MEMBER,FACILITY
```

-- 2.6 How can you produce a list of bookings on the day of 2012-09-14 which will cost the member (or guest) more than \$30? Remember that guests have different costs to members (the listed costs are per half-hour 'slot'), and the guest user is always ID 0. Include in your output the name of the facility, the name of the member formatted as a single column, and the cost. Order by descending cost, and do not use any subqueries.

```
SELECT M.FIRSTNAME || ' ' || M.SURNAME AS MEMBER , F.NAME AS FACILITY ,
CASE WHEN M.MEMID = 0 THEN
    B.SLOTS*F.GUESTCOST
    ELSE
    B.SLOTS*F.MEMBERCOST
END AS COST
FROM CD.MEMBERS M
INNER JOIN CD.BOOKINGS B USING(MEMID)
INNER JOIN CD.FACILITIES F USING(FACID)
WHERE DATE(B.STARTTIME) = '2012-09-14'
AND (M.MEMID = 0 AND B.SLOTS*F.GUESTCOST > 30 OR M.MEMID <> 0 AND
B.SLOTS*F.MEMBERCOST > 30)
ORDER BY COST DESC
```

-- 2.7 How can you output a list of all members, including the individual who recommended them (if any), without using any joins? Ensure that there are no duplicates in the list, and that each firstname + surname pairing is formatted as a column and ordered.

```
SELECT DISTINCT MEM.FIRSTNAME|| ' ' || MEM.SURNAME AS MEMBER ,
(SELECT REF.FIRSTNAME || ' ' || REF.SURNAME AS RECOMMENDER
FROM CD.MEMBERS REF
WHERE REF.MEMID = MEM.RECOMMENDEDBY
)
```

```
FROM CD.MEMBERS MEM
```

```
ORDER BY MEMBER
```

```
-- 2.8 Produce a list of costly bookings, using a subquery
```

```
select member, facility, cost from (
```

```
    select
```

```
        m.firstname || ' ' || m.surname as member,
```

```
        f.name as facility,
```

```
        case
```

```
            when m.memid = 0 then
```

```
                b.slots*f.guestcost
```

```
            else
```

```
                b.slots*f.membercost
```

```
        end as cost
```

```
    from
```

```
        cd.members m
```

```
        inner join cd.bookings b
```

```
            on m.memid = b.memid
```

```
        inner join cd.facilities f
```

```
            on b.facid = f.facid
```

```
    where
```

```
        bstarttime >= '2012-09-14' and
```

```
        bstarttime < '2012-09-15'
```

```
) as bookings
```

```
where cost > 30
```

```
order by cost desc;
```

```
-- 3 Modifying data
```

```
-- 3.1 Insert some data into a table
```

```
INSERT INTO CD.facilities VALUES(9,'Spa',20,30,100000,800)
```

-- 3.2 Insert multiple rows of data into a table

```
INSERT INTO CD.facilities VALUES
```

```
(9,'Spa',20,30,100000,800),
```

```
(10,'Squash Court 2',3.5,17.5,5000,80)
```

-- 3.3 Insert calculated data into a table

```
INSERT INTO  
CD.FACILITIES(FACID,NAME,MEMBERCOST,GUESTCOST,INITIALOUTLAY,MONTHLYMAINTENANCE)  
SELECT(SELECT MAX(FACID) FROM CD.FACILITIES)+1,  
'Spa', 20, 30, 100000, 800;
```

-- 3.4 Update some existing data

```
UPDATE CD.FACILITIES  
SET INITIALOUTLAY = 10000  
WHERE NAME = 'Tennis Court 2'
```

-- 3.5 Update multiple rows and columns at the same time

```
UPDATE CD.FACILITIES  
SET MEMBERCOST = 6 , GUESTCOST = 30  
WHERE NAME LIKE '%Tennis Court%'
```

-- 3.6 Update a row based on the contents of another row

```
update cd.facilities set  
membercost = (select membercost * 1.1 from cd.facilities where facid = 0),  
guestcost = (select guestcost * 1.1 from cd.facilities where facid = 0)
```

```
where facs.facid = 1;
```

```
-- 3.7 Delete all bookings
```

```
delete from cd.bookings;
```

```
-- 3.8 Delete a member from the cd.members table
```

```
delete from cd.members where memid = 37;
```

```
-- 3.9 Delete based on a subquery
```

```
delete from cd.members where memid not in (select memid from cd.bookings);
```

```
-- 4 Aggregation
```

```
-- 4.1 Count the number of facilities
```

```
SELECT COUNT(*) FROM CD.FACILITIES
```

```
-- 4.2 Count the number of expensive facilities
```

```
select count(*) from cd.facilities where guestcost >= 10;
```

```
SELECT * FROM CD. MEMBERS
```

```
-- 4.3 Count the number of recommendations each member makes.
```

```
SELECT RECOMMENDED BY,COUNT(*) FROM CD.MEMBERS
```

```
WHERE RECOMMENDED BY IS NOT NULL
```

```
GROUP BY RECOMMENDED BY
```

```
ORDER BY RECOMMENDED BY
```

```
-- 4.4 List the total slots booked per facility
```

```
SELECT FACID, SUM(SLOTS) AS "Total Slots" FROM CD.BOOKINGS
```

GROUP BY FACID

ORDER BY FACID

-- 4.5

```
select facid, sum(slots) as "Total Slots"
```

from cd.bookings

where

```
starttime >= '2012-09-01'
```

and starttime < '2012-10-01'

group by facid

order by sum(slots);

-- 4.6 List the total slots booked per facility per month

```
SELECT FACID, EXTRACT(MONTH FROM STARTTIME) AS MONTH, SUM(SLOTS) AS "TOTAL SLOTS"
```

FROM CD.BOOKINGS

WHERE EXTRACT(YEAR FROM STARTTIME) = 2012

GROUP BY FACID, MONTH

ORDER BY FACID, MONTH;

-- 4.7 Find the count of members who have made at least one booking

SELECT COUNT(DISTINCT MEMID) FROM CD.BOOKINGS

-- 4.8 List facilities with more than 1000 slots booked

```
SELECT FACID, SUM(SLOTS) AS "Total Slots" from CD.BOOKINGS
```

GROUP BY FACID

HAVING SUM(SLOTS) > 1000

ORDER BY FACID

-- 4.9 Find the total revenue of each facility

```
SELECT F.NAME, SUM(SLOTS * CASE
```

WHEN MEMID = 0 THEN F.GUESTCOST

```
        ELSE F.MEMBERCOST  
    END) AS REVENUE  
FROM CD.BOOKINGS B  
INNER JOIN CD.FACILITIES F  
    USING(FACID)  
GROUP BY F.NAME  
ORDER BY REVENUE;
```

-- 4.10

```
SELECT NAME, REVENUE FROM  
(  
SELECT FACS.NAME, SUM(CASE  
WHEN MEMID = 0 THEN SLOTS * FACS.GUESTCOST  
ELSE SLOTS * MEMBERCOST  
END) AS REVENUE  
FROM CD.BOOKINGS BKS  
INNER JOIN CD.FACILITIES FACS  
ON BKS.FACID = FACS.FACID  
GROUP BY FACS.NAME  
)  
AS AGG WHERE REVENUE < 1000  
ORDER BY REVENUE;
```

-- 4.11 Output the facility id that has the highest number of slots booked

```
SELECT FACID, SUM(SLOTS) AS "TOTAL SLOTS" FROM CD.BOOKINGS  
GROUP BY FACID  
ORDER BY SUM(SLOTS) DESC  
LIMIT 1;
```

-- 4.12 List the total slots booked per facility per month, part 2

```
SELECT FACID, EXTRACT(MONTH FROM STARTTIME) AS MONTH, SUM(SLOTS) SLOTS FROM
CD.BOOKINGS

WHERE STARTTIME >= '2012-01-01'

AND STARTTIME < '2013-01-01'

GROUP BY ROLLUP(FACID, MONTH)

ORDER BY FACID, MONTH;
```

-- 4.13 List the total hours booked per named facility

```
SELECT FACS.FACID, FACS.NAME,
TRIM(TO_CHAR(SUM(BKS.SLOTS)/2.0, '999999999999999D99')) AS "TOTAL HOURS"
FROM CD.BOOKINGS BKS
INNER JOIN CD.FACILITIES FACS
ON FACS.FACID = BKS.FACID
GROUP BY FACS.FACID, FACS.NAME
ORDER BY FACS.FACID;
```

-- 4.14 List each member's first booking after September 1st 2012

```
SELECT MEMS.SURNAME, MEMS.FIRSTNAME, MEMS.MEMID, MIN(BKS.STARTTIME) AS STARTTIME
FROM CD.BOOKINGS BKS
INNER JOIN CD.MEMBERS MEMS ON
MEMS.MEMID = BKS.MEMID
WHERE STARTTIME >= '2012-09-01'
GROUP BY MEMS.SURNAME, MEMS.FIRSTNAME, MEMS.MEMID
ORDER BY MEMS.MEMID;
```

-- 4.15 Produce a list of member names, with each row containing the total member count

```
SELECT COUNT(*) OVER(), FIRSTNAME, SURNAME FROM CD.MEMBERS
ORDER BY JOINDATE
```

-- 4.16 Produce a numbered list of members

```
SELECT ROW_NUMBER() OVER(ORDER BY JOINDATE), FIRSTNAME, SURNAME FROM CD.MEMBERS
```

```
ORDER BY JOINDATE
```

-- 4.17 Produce a numbered list of members

```
SELECT FACID, TOTAL FROM
```

```
(SELECT FACID, SUM(SLOTS) TOTAL, RANK() OVER (ORDER BY SUM(SLOTS) DESC) RANK FROM  
CD.BOOKINGS
```

```
GROUP BY FACID ) AS RANKED
```

```
WHERE RANK = 1
```

-- 4.18 Rank members by (rounded) hours used

```
SELECT FIRSTNAME, SURNAME,((SUM(BKS.SLOTS)+10)/20)*10 AS HOURS,
```

```
RANK() OVER (ORDER BY ((SUM(BKS.SLOTS)+10)/20)*10 DESC) AS RANK
```

```
FROM CD.BOOKINGS BKS
```

```
INNER JOIN CD.MEMBERS MEMS
```

```
ON BKS.MEMID = MEMS.MEMID
```

```
GROUP BY MEMS.MEMID
```

```
ORDER BY RANK, SURNAME, FIRSTNAME;
```

--4.19 Find the top three revenue generating facilities

```
SELECT NAME, RANK FROM
```

```
(SELECT F.NAME AS NAME, RANK() OVER (ORDER BY SUM(CASE
```

```
WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST
```

```
ELSE SLOTS * MEMBERCOST
```

```
END) DESC) AS RANK
```

```
FROM CD.BOOKINGS B
```

```
INNER JOIN CD.FACILITIES F
```

```
USING(FACID)
```

```
GROUP BY F.NAME) AS SUBQ
```

```
WHERE RANK <= 3
```

```
ORDER BY RANK;
```

```
-- 4.20 Classify facilities by value  
SELECT NAME, CASE WHEN CLASS=1 THEN 'high'  
WHEN CLASS=2 THEN 'average'  
ELSE 'low'  
END REVENUE  
FROM (SELECT F.NAME AS NAME, NTILE(3) OVER (ORDER BY SUM(CASE  
WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST  
ELSE SLOTS * MEMBERCOST  
END) DESC) AS CLASS  
FROM CD.BOOKINGS B  
INNER JOIN CD.FACILITIES F  
USING(FACID)  
GROUP BY F.NAME) AS SUBQ  
ORDER BY CLASS, NAME;
```

-- 4.21 Calculate the payback time for each facility

```
SELECT F.NAME AS NAME,  
F.INITIALOUTLAY/((SUM(CASE  
WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST  
ELSE SLOTS * MEMBERCOST  
END)/3) - F.MONTHLYMAINTENANCE) AS MONTHS  
FROM CD.BOOKINGS B  
INNER JOIN CD.FACILITIES F  
USING(FACID)  
GROUP BY F.FACID  
ORDER BY NAME;
```

-- 4.22 Calculate a rolling average of total revenue

```
SELECT DATEGEN.DATE,
```

```

(SELECT SUM(CASE
    WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST
    ELSE SLOTS * MEMBERCOST
END) AS REV
FROM CD.BOOKINGS B
INNER JOIN CD.FACILITIES F
USING(FACID)
WHERE B.STARTTIME > DATEGEN.DATE - INTERVAL '14 DAYS'
AND B.STARTTIME < DATEGEN.DATE + INTERVAL '1 DAY')/15 AS REVENUE
FROM
(SELECT CAST(GENERATE_SERIES(TIMESTAMP '2012-08-01','2012-08-31','1 DAY') AS DATE) AS DATE
AS DATEGEN
ORDER BY DATEGEN.DATE;

```

-- 5 Working with Timestamps Begin!

-- 5.1 Produce a timestamp for 1 a.m. on the 31st of August 2012

```
SELECT TIMESTAMP '2012-08-31 01:00:00';
```

-- 5.2 Subtract timestamps from each other

```
SELECT TIMESTAMP '2012-08-31 01:00:00' - TIMESTAMP '2012-07-30 01:00:00' AS INTERVAL;
```

-- 5.3 Generate a list of all the dates in October 2012

```
SELECT GENERATE_SERIES(TIMESTAMP '2012-10-01', TIMESTAMP '2012-10-31', INTERVAL '1 DAY')
AS TS;
```

-- 5.4 Get the day of the month from a timestamp

```
SELECT EXTRACT(DAY FROM TIMESTAMP '2012-08-31');
```

-- 5.5 Work out the number of seconds between timestamps

```
SELECT EXTRACT(EPOCH FROM (TIMESTAMP '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));
```

-- 5.6 Work out the number of days in each month of 2012

```
SELECT EXTRACT(MONTH FROM CAL.MONTH) AS MONTH,  
(CAL.MONTH + INTERVAL '1 MONTH') - CAL.MONTH AS LENGTH FROM  
(SELECT GENERATE_SERIES(TIMESTAMP '2012-01-01', TIMESTAMP '2012-12-01', INTERVAL '1  
MONTH') AS MONTH) CAL  
ORDER BY MONTH;
```

-- 5.7 Work out the number of days in each month of 2012

```
SELECT (DATE_TRUNC('MONTH',TS.TESTTS) + INTERVAL '1 MONTH') - DATE_TRUNC('DAY', TS.TESTTS)  
AS REMAINING  
FROM (SELECT TIMESTAMP '2012-02-11 01:00:00' AS TESTTS) TS
```

-- 5.8 Work out the end time of bookings

```
SELECT STARTTIME, STARTTIME + SLOTS*(INTERVAL '30 MINUTES') ENDTIME FROM CD.BOOKINGS  
ORDER BY ENDTIME DESC, STARTTIME DESC  
LIMIT 10
```

-- 5.9 Return a count of bookings for each month

```
SELECT DATE_TRUNC('MONTH', STARTTIME) AS MONTH, COUNT(*) FROM CD.BOOKINGS  
GROUP BY MONTH  
ORDER BY MONTH
```

-- 5.10 Work out the utilisation percentage for each facility by month

```
SELECT NAME, MONTH, ROUND((100*SLOTS)/  
CAST(25*(CAST((MONTH + INTERVAL '1 MONTH') AS DATE)- CAST (MONTH AS DATE)) AS  
NUMERIC),1) AS UTILISATION  
FROM (SELECT F.NAME AS NAME, DATE_TRUNC('MONTH', STARTTIME) AS MONTH, SUM(SLOTS) AS  
SLOTS FROM CD.BOOKINGS B
```

```
INNER JOIN CD.FACILITIES F
USING(FACID)
GROUP BY F.FACID, MONTH) AS INN
ORDER BY NAME, MONTH
```

-- 6 STRING

-- 6.1 Format the names of members

```
SELECT SURNAME || ',' || FIRSTNAME AS NAME FROM CD.MEMBERS
```

-- 6.2 Perform a case-insensitive search

```
SELECT * FROM CD.FACILITIES WHERE NAME LIKE 'Tennis%';
```

-- 6.3 Find telephone numbers with parentheses

```
SELECT MEMID, TELEPHONE FROM CD.MEMBERS WHERE TELEPHONE ~ '[()]';
```

-- 6.4 Pad zip codes with leading zeroes

```
SELECT LPAD(CAST(ZIPCODE AS CHAR(5)),5,'0') ZIP FROM CD.MEMBERS ORDER BY ZIP
```

-- 6.5 Count the number of members whose surname starts with each letter of the alphabet

```
SELECT SUBSTR (MEMS.SURNAME,1,1) AS LETTER, COUNT(*) AS COUNT FROM CD.MEMBERS MEMS
GROUP BY LETTER
ORDER BY LETTER
```

-- 6.6 Clean up telephone numbers

```
SELECT MEMID, TRANSLATE(TELEPHONE, '-() ', '') AS TELEPHONE
FROM CD.MEMBERS
ORDER BY MEMID;
```

-- 7 RECURSION

```
-- 7.1 Find the upward recommendation chain for member ID 27  
WITH RECURSIVE RECOMMENDERS(RECOMMENDER) AS (  
    SELECT RECOMMENDED_BY FROM CD.MEMBERS WHERE MEMID = 27  
    UNION ALL  
        SELECT M.RECOMMENDED_BY  
        FROM RECOMMENDERS R  
        INNER JOIN CD.MEMBERS M  
        ON M.MEMID = R.RECOMMENDER)  
    SELECT R.RECOMMENDER, M.FIRSTNAME, M.SURNAME  
    FROM RECOMMENDERS R  
    INNER JOIN CD.MEMBERS M  
    ON R.RECOMMENDER = M.MEMID  
    ORDER BY MEMID DESC
```

```
-- 7.2 Find the downward recommendation chain for member ID 1  
WITH RECURSIVE RECOMMENDED(S(MEMID) AS (  
    SELECT MEMID FROM CD.MEMBERS WHERE RECOMMENDED_BY = 1  
    UNION ALL  
        SELECT M.MEMID FROM RECOMMENDED R  
        INNER JOIN CD.MEMBERS M  
        ON M.RECOMMENDED_BY = R.MEMID)  
    SELECT R.MEMID, M.FIRSTNAME, M.SURNAME  
    FROM RECOMMENDED R  
    INNER JOIN CD.MEMBERS M  
    ON R.MEMID = M.MEMID  
    ORDER BY MEMID
```

```
-- 7.3 Produce a CTE that can return the upward recommendation chain for any member  
WITH RECURSIVE RECOMMENDERS(RECOMMENDER, MEMBER) AS (  
    SELECT RECOMMENDED_BY, MEMID  
    FROM CD.MEMBERS
```

UNION ALL

```
SELECT M.RECOMMENDED_BY, R.MEMBER
```

```
FROM RECOMMENDERS R
```

```
INNER JOIN CD.MEMBERS M
```

```
ON M.MEMID = R.RECOMMENDER )
```

```
SELECT R.MEMBER MEMBER, R.RECOMMENDER, M.FIRSTNAME, M.SURNAME
```

```
FROM RECOMMENDERS R
```

```
INNER JOIN CD.MEMBERS M
```

```
ON R.RECOMMENDER = M.MEMID
```

```
WHERE R.MEMBER = 22 OR R.MEMBER = 12
```

```
ORDER BY R.MEMBER ASC, R.RECOMMENDER DESC
```

The screenshot shows the pgAdmin 4 interface with the following details:

- File Bar:** File, Object, Tools, Help.
- Object Explorer:** Shows the database structure under the schema "cd".
 - Tables: bookings, facilities, members.
 - Triggers: table Tennis.
 - Views: Massage Room 1, Massage Room 2, Squash Court.
 - Sequences: facid.
- Query Editor:** The query `select * from cd.facilities` is run against PostgreSQL 16. The results are displayed in a Data Output tab.
- Data Output:** A table showing facility details:

facid [PK] Integer	name character varying (100)	membercost numeric	guestcost numeric	initialoutlay numeric	monthlymaintenance numeric
1	Tennis Court 1	5	25	10000	200
2	Tennis Court 2	5	25	8000	200
3	Badminton Court	0	15.5	4000	50
4	Table Tennis	0	5	320	10
5	Massage Room 1	35	80	4000	3000
6	Massage Room 2	35	80	4000	3000
7	Squash Court	3.5	17.5	5000	80
8	Snooker Table	0	5	450	15
9	Pool Table	0	5	400	15
- Messages:** Total rows: 9 of 9. Query complete 00:00:00.137.
- Notifications:** Successfully run. Total query runtime: 137 msec. 9 rows affected.
- System Bar:** Windows Start button, Taskbar icons (File Explorer, Mail, Edge, etc.), System tray showing Nifty smllcap -1.85%, 15:13, 19-07-2024.

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```

1 -- BASIC 1
2 -- 1.1 How can you retrieve all the information from the cd.facilities table?
3
4 select * from cd.facilities
5
6 -- 1.2 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 fac
7
8 SELECT NAME, MEMBERCOST FROM CD.FACILITIES

```

Data Output Messages Notifications

	name	membercost
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

Total rows: 9 of 9 Query complete 00:00:00.338

Successfully run. Total query runtime: 338 msec. 9 rows affected. Ln 8, Col 43

BSE smcap -2.02% 15:18 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```

1 -- BASIC 1
2 -- 1.1 How can you retrieve all the information from the cd.facilities table?
3
4 select * from cd.facilities
5
6 -- 1.2 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 fac
7
8 SELECT NAME, MEMBERCOST FROM CD.FACILITIES
9
10 -- 1.3 How can you produce a list of facilities that charge a fee to members?
11 SELECT * FROM CD.FACILITIES
12 WHERE MEMBERCOST > 0

```

Data Output Messages Notifications

facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
0	Tennis Court 1	5	25	10000	200
1	Tennis Court 2	5	25	8000	200
4	Massage Room 1	35	80	4000	3000
5	Massage Room 2	35	80	4000	3000
6	Squash Court	3.5	17.5	5000	80

Total rows: 5 of 5 Query complete 00:00:00.202

Successfully run. Total query runtime: 202 msec. 5 rows affected. Ln 11, Col 1

31°C Haze 15:22 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- Tables (3)
 - bookings
 - facilities
 - members
- Trigger Functions
- Types
- Views
- public
- Subscriptions
- postgres
- university_db
- Login/Group Roles
- Tablespaces

exercises/postgres@PostgreSQL 16

Query History

Execute/Refresh

```

6 -- 1.2 You want to print out a list of the facilities and their cost to members. How would you retrieve a list of only fac
7
8 SELECT NAME, MEMBERCOST FROM CD.FACILITIES
9
10 -- 1.3 How can you produce a list of facilities that charge a fee to members?
11
12 SELECT * FROM CD.FACILITIES
13 WHERE MEMBERCOST > 0
14
15 --1.4 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 main
16
17 SELECT FACID,NAME, MEMBERCOST,MONTHLYMAINTENANCE FROM CD.FACILITIES
18 WHERE
19 MEMBERCOST > 0 AND MEMBERCOST < MONTHLYMAINTENANCE/50.0
20
  
```

Data Output

facid	name	membercost	monthlymaintenance
1	Massage Room 1	35	3000
2	Massage Room 2	35	3000

Total rows: 2 of 2 Query complete 00:00:00.416

Successfully run. Total query runtime: 416 msec. 2 rows affected. Ln 16, Col 1

Nifty midcap -2.05% 15:29 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
- Tables (3)
 - bookings
 - facilities
 - members
- Trigger Functions
- Types
- Views
- public
- Subscriptions
- postgres
- university_db
- Login/Group Roles
- Tablespaces

exercises/postgres@PostgreSQL 16

Query History

Execute/Refresh

```

32 SELECT * FROM CD.FACILITIES
33 WHERE MEMBERCOST > 0
34
35 --1.4 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 main
36
37 SELECT FACID,NAME, MEMBERCOST,MONTHLYMAINTENANCE FROM CD.FACILITIES
38 WHERE
39 MEMBERCOST > 0 AND MEMBERCOST < MONTHLYMAINTENANCE/50.0
40
41 -- 1.5 How can you produce a list of all facilities with the word 'Tennis' in their name?
42 SELECT * FROM CD.FACILITIES
43 WHERE NAME LIKE '%Tennis%'
44
45
  
```

Data Output

facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
1	Tennis Court 1	5	25	10000	200
2	Tennis Court 2	5	25	8000	200
3	Table Tennis	0	5	320	10

Total rows: 3 of 3 Query complete 00:00:00.124

Successfully run. Total query runtime: 124 msec. 3 rows affected. Ln 22, Col 1

BSE midcap -2.06% 15:32 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
 - public
 - Subscriptions
 - postgres
 - university_db
- Login/Group Roles
- Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```
-- 1.6 How can you retrieve the details of facilities with ID 1 and 5? Try to do it without using the OR operator.
27
28 SELECT * FROM cd.FACILITIES
29 WHERE FACID IN (1,5);
30
31 -- 1.7 How can you produce a list of facilities, with each labelled as 'cheap' or 'expensive' depending on if their monthly maintenance cost is above or below 100?
32
33 SELECT NAME,
34 CASE WHEN MONTHLYMAINTENANCE > 100 THEN
35   'expensive'
36 else
37   'cheap'
38 end as cost
39 from cd.facilities
```

Data Output Messages Notifications

	name	cost
1	Tennis Court 1	expensive
2	Tennis Court 2	expensive
3	Badminton Court	cheap
4	Table Tennis	cheap
5	Massage Room 1	expensive
6	Massage Room 2	expensive
7	Squash Court	cheap
8	Snooker Table	cheap
9	Pool Table	cheap

Total rows: 9 of 9 Query complete 00:00:00.303

Successfully run. Total query runtime: 303 msec. 9 rows affected. Ln 39, Col 19

31°C Mostly cloudy 16:23 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
 - public
 - Subscriptions
 - postgres
 - university_db
- Login/Group Roles
- Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```
35   'expensive'
36 else
37   'cheap'
38 end as cost
39 from cd.facilities
40
41 -- 1.8 How can you produce a list of members who joined after the start of September 2012? Return the memid, surname, firstname, joindate,
42
43 SELECT MEMID,SURNAME,FIRSTNAME,JOINDATE FROM CD.MEMBERS
44 WHERE JOINDATE >= '2012-09-01'
```

Data Output Messages Notifications

	memid	surname	firstname	joindate
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

Total rows: 10 of 10 Query complete 00:00:00.168

Successfully run. Total query runtime: 168 msec. 10 rows affected. Ln 43, Col 1

31°C Mostly cloudy 16:30 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Exercises
- > Tables (3)
 - bookings
 - facilities
 - members
- > Types
- > Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

Execute/Refresh (F5)

```

from cd.facilities
-- 1.8 How can you produce a list of members who joined after the start of September 2012? Return the memid, surname, firstname,
-- WHERE JOINDATE >= '2012-09-01'
SELECT MEMID,SURNAME,FIRSTNAME,JOINDATE FROM CD.MEMBERS
WHERE JOINDATE >= '2012-09-01'

--1.9 How can you produce an ordered list of the first 10 surnames in the members table? The list must not contain duplicates.
SELECT DISTINCT SURNAME FROM CD.MEMBERS
ORDER BY SURNAME
LIMIT 10

```

Data Output 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	

Total rows: 10 of 10 Query complete 00:00:00.207 ✓ Successfully run. Total query runtime: 207 msec. 10 rows affected. Ln 48, Col 1

Type here to search

31°C Mostly cloudy 16:32 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Exercises
- > Tables (3)
 - bookings
 - facilities
 - members
- > Types
- > Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

Execute/Refresh (F5)

```

47
48 SELECT DISTINCT SURNAME FROM CD.MEMBERS
49 ORDER BY SURNAME
50 LIMIT 10
51
52 --1.10 You, for some reason, want a combined list of all surnames and all facility names. Yes, this is a contrived example :-). P
53
54 SELECT SURNAME FROM CD.MEMBERS
55 UNION
56 SELECT NAME FROM CD.FACILITIES;
57
58
59
60
61

```

Data Output Messages Notifications

	surname	character varying
1	Hunt	
2	Farrell	
3	Tennis Court 2	
4	Table Tennis	
5	Dare	
6	Rownam	
7	GUEST	
8	Badminton Court	
9	Smith	
10	Tupperware	

Total rows: 34 of 34 Query complete 00:00:00.271 ✓ Successfully run. Total query runtime: 271 msec. 34 rows affected. Ln 54, Col 1

Type here to search

31°C Haze 16:40 19-07-2024

pgAdmin 4

Object Explorer

```

Schemas (2)
  cd
    Aggregates
    Collations
    Domains
    FTS Configurations
    FTS Dictionaries
    FTS Parsers
    FTS Templates
    Foreign Tables
    Functions
    Materialized Views
    Operators
    Procedures
    Sequences
  Tables (3)
    bookings
    facilities
    members
  Trigger Functions
  Types
  Views
  public
  Subscriptions
  postres
  university_db
  Login/Group Roles
  Tablespaces

```

exercises/postgres@PostgreSQL 16

Query History

Execute/Refresh

```

LIMIT 10
--1.10 You, for some reason, want a combined list of all surnames and all facility names. Yes, this is a contrived example :-). P
51
52
53
54 SELECT SURNAME FROM CD.MEMBERS
55 UNION
56 SELECT NAME FROM CD.FACILITIES;
57
58
59 -- 1.11 You'd like to get the signup date of your last member. How can you retrieve this information?
60
61 SELECT JOINDATE AS LATEST FROM CD.MEMBERS
62 ORDER BY JOINDATE DESC
63 LIMIT 1
64

```

Data Output

	latest
	timestamp without time zone
1	2012-09-26 18:08:45

Total rows: 1 of 1 Query complete 00:00:00.142

Successfully run. Total query runtime: 142 msec. 1 rows affected. Ln 61, Col 1

Type here to search

Windows Taskbar: Weather ahead, 31°C Haze, 16:43, 19-07-2024

pgAdmin 4

Object Explorer

```

Schemas (2)
  cd
    Aggregates
    Collations
    Domains
    FTS Configurations
    FTS Dictionaries
    FTS Parsers
    FTS Templates
    Foreign Tables
    Functions
    Materialized Views
    Operators
    Procedures
    Sequences
  Tables (3)
    bookings
    facilities
    members
  Trigger Functions
  Types
  Views
  public
  Subscriptions
  postres
  university_db
  Login/Group Roles
  Tablespaces

```

exercises/postgres@PostgreSQL 16

Query History

Execute/Refresh

```

--1.12 You'd like to get the first and last name of the last member(s) who signed up - not just the date. How can you do that?
64
65
66
67 SELECT FIRSTNAME,SURNAME,JOINDATE FROM CD.MEMBERS
68 ORDER BY JOINDATE DESC
69 LIMIT 1
70
71
72 -- 2 .Joins and Subqueries
73 -- 2.1 How can you produce a list of the start times for bookings by members named 'David Farrell'?
74
75 SELECT B.STARTTIME FROM CD.BOOKINGS B
76 INNER JOIN CD.MEMBERS M USING(MEMID)
77 WHERE M.FIRSTNAME = 'David' AND SURNAME = 'Farrell'

```

Data Output

	starttime
	timestamp without time zone
1	2012-09-18 09:00:00
2	2012-09-18 13:30:00
3	2012-09-18 17:30:00
4	2012-09-18 20:00:00
5	2012-09-19 09:30:00
6	2012-09-19 12:00:00
7	2012-09-19 15:00:00
8	2012-09-20 11:30:00
9	2012-09-20 14:00:00
10	2012-09-20 15:30:00

Total rows: 34 of 34 Query complete 00:00:00.152

Successfully run. Total query runtime: 152 msec. 34 rows affected. Ln 74, Col 1

Type here to search

Windows Taskbar: 31°C Haze, 16:53, 19-07-2024

The screenshot shows the pgAdmin 4 interface with the following details:

- File Object Tools Help**
- Object Explorer** pane on the left listing database objects like Schemas, Tables, and Views.
- Query Editor** pane in the center displaying a SQL script with comments and two queries. The second query is highlighted in orange.

```
-- 2.1 How can you produce a list of the start times for bookings by members named 'David Farrell'?  
--  
SELECT B.STARTTIME FROM CD.BOOKINGS B  
INNER JOIN CD.MEMBERS M USING(MEMID)  
WHERE M.FIRSTNAME = 'David' and SURNAME = 'Farrell'  
  
-- 2.2 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  
SELECT B.STARTTIME AS START,F.NAME FROM CD.BOOKINGS B  
INNER JOIN CD.FACILITIES F USING(FACID)  
WHERE F.NAME LIKE '%Tennis Court%'  
AND DATE(B.STARTTIME) = '2012-09-21'  
ORDER BY B.STARTTIME;
```
- Data Output** pane below the query editor showing the results of the second query:

start	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	
7	2012-09-21 13:30:00	Tennis Court 1	
8	2012-09-21 14:00:00	Tennis Court 2	
9	2012-09-21 15:30:00	Tennis Court 1	
10	2012-09-21 16:00:00	Tennis Court 2	

- Status Bar** at the bottom right indicating "Successfully run. Total query runtime: 96 msec. 12 rows affected."
- System Icons** at the bottom left and bottom right.

The screenshot shows the pgAdmin 4 interface. The left sidebar (Object Explorer) lists various database objects: Schemas (cd, public), Tables (bookings, facilities, members), and Views. The main area is a query editor titled 'exercises/postgres@PostgreSQL 16' containing a SQL script. The script includes comments for question 2.3 and a query to select distinct members recommended by others. Below the editor is a results grid for the 'members' table, showing columns 'firstname' and 'surname'. A status bar at the bottom indicates the query was successfully run with a runtime of 212 msec and 13 rows affected.

```
82 WHERE NAME LIKE '%Tennis Court%'  
83 AND DATE(B.STARTTIME) = '2012-09-21'  
84 ORDER BY B.STARTTIME;  
85  
86 -- 2.3 How can you output a list of all members who have recommended another member?  
87 -- Ensure that there are no duplicates in the list, and that results are ordered by (surname, firstname)  
88  
89 SELECT DISTINCT M2.FIRSTNAME,M2.SURNAME FROM CD.MEMBERS M1  
90 INNER JOIN  
91 CD.MEMBERS M2 ON M2.MEMID = M1.RECOMMENDEDBY  
92 ORDER BY SURNAME,FIRSTNAME  
93  
94 SELECT * FROM CD.MEMBERS  
95  
96 select distinct firstname as firstname,surname as surname
```

	firstname	surname
1	Florence	Bader
2	Timothy	Baker
3	Gerald	Butters
4	Jemima	Farrell
5	Matthew	Genting
6	David	Jones
7	Janice	Joplette
8	Millicent	Purview
9	Tim	Rownam
10	Darren	Smith

Total rows: 13 of 13 Query complete 00:00:00.212 Successfully run. Total query runtime: 212 msec. 13 rows affected. Ln 88, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - members
- > Tables (3)
 - bookings
 - facilities
 - members
- > Trigger Functions
- > Types
- > Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

```

98 cd.members
99 WHERE MEMID IN (SELECT RECOMMENDED BY FROM CD.MEMBERS)
100 ORDER BY surname, firstname;
101
102 --2.4 How can you output a list of all members, including the individual who recommended them (if any)? Ensure that results are c
103
104 SELECT MEM.FIRSTNAME AS MEMFNAME, MEM.SURNAME AS MEMSNAME, REF.FIRSTNAME AS RECNAME, REF.SURNAME AS RECN
105 FROM CD.MEMBERS MEM
106 LEFT JOIN
107 CD.MEMBERS REF
108 ON REF.MEMID = MEM.RECOMMENDED BY
109 ORDER BY MEM.SURNAME , MEM.FIRSTNAME
110
111
112

```

Data Output

	memname	memsname	recname	recline
1	Florence	Bader	Ponder	Stibbons
2	Anne	Baker	Ponder	Stibbons
3	Timothy	Baker	Jemima	Farrell
4	Tim	Boothe	Tim	Rownam
5	Gerald	Butters	Darren	Smith
6	Joan	Coplin	Timothy	Baker
7	Erica	Crumpet	Tracy	Smith
8	Nancy	Dare	Janice	Joplette
9	David	Farrell	[null]	[null]
10	Jemima	Farrell	[null]	[null]

Total rows: 31 of 31 Query complete 00:00:00.121

Successfully run. Total query runtime: 121 msec. 31 rows affected.

Ln 104, Col 1

31°C Haze 17:42 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - members
- > Tables (3)
 - bookings
 - facilities
 - members
- > Trigger Functions
- > Types
- > Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

```

105 FROM CD.MEMBERS MEM
106 LEFT JOIN
107 CD.MEMBERS REF
108 ON REF.MEMID = MEM.RECOMMENDED BY
109 ORDER BY MEM.SURNAME , MEM.FIRSTNAME
110
111 -- 2.5 How can you produce a list of all members who have used a tennis court? Include in your output the name of the court, and
112
113 SELECT DISTINCT M.FIRSTNAME || ' ' || M.SURNAME AS MEMBER, F.NAME AS FACILITY FROM CD.MEMBERS M
114 INNER JOIN CD.BOOKINGS B USING(MEMID)
115 INNER JOIN CD.FACILITIES F USING(FACID)
116 WHERE F.NAME LIKE '%Tennis Court%'
117 ORDER BY MEMBER,FACILITY
118

```

Data Output

	member	facility
1	Anne Baker	Tennis Court 1
2	Anne Baker	Tennis Court 2
3	Burton Tracy	Tennis Court 1
4	Burton Tracy	Tennis Court 2
5	Charles Owen	Tennis Court 1
6	Charles Owen	Tennis Court 2
7	Darren Smith	Tennis Court 2
8	David Farrell	Tennis Court 1
9	David Farrell	Tennis Court 2
10	David Jones	Tennis Court 1

Total rows: 46 of 46 Query complete 00:00:00.112

Successfully run. Total query runtime: 112 msec. 46 rows affected.

Ln 113, Col 1

31°C Mostly cloudy 18:00 19-07-2024

The screenshot shows the pgAdmin 4 interface with the following details:

- File Object Tools Help** are at the top.
- Object Explorer** on the left lists database objects: Foreign Data Wrappers, Languages, Publications, Schemas (2), cd, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (3), bookings, facilities, members, Trigger Functions, Types, Views, public, Subscriptions, postgres, university_db, Login/Group Roles, and Tablespaces.
- exercises/postgres@PostgreSQL 16** is the current connection.
- Query History** shows previous queries.
- Query** pane contains the following SQL code:

```
116 WHERE F.NAME LIKE '%ennis Courts%'  
117 ORDER BY MEMBER,FACILITY  
118 -- 2.6 How can you produce a list of bookings on the day of 2012-09-14 which will cost the member (or guest) more than $30? Remem  
119  
120  
121 SELECT M.FIRSTNAME || ' ' || M.SURNAME AS MEMBER , F.NAME AS FACILITY ,  
122 CASE WHEN M.MEMID = 0 THEN  
123 B.SLOTS+F.GUESTCOST  
124 ELSE  
125 B.SLOTS+F.MEMBERCOST  
126 END AS COST  
127 FROM CD.MEMBERS M  
128 INNER JOIN CD.BOOKINGS B USING(MEMID)  
129 INNER JOIN CD.FACILITIES F USING(FACID)  
130 WHERE DATE(B.STARTTIME) = '2012-09-14'
```
- Data Output**, **Messages**, and **Notifications** tabs are available.
- Results Grid** displays the following data:

member	facility	cost
GUEST GUEST	Massage Room 2	320
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Tennis Court 2	150
Jemima Farrell	Massage Room 1	140
GUEST GUEST	Tennis Court 1	75
GUEST GUEST	Tennis Court 1	75
GUEST GUEST	Tennis Court 2	75
Ponder Stibbons	Massage Room 1	70

- Total rows: 18 of 18** and **Query complete 00:00:00.280** are at the bottom.
- Status Bar** shows **Successfully run. Total query runtime: 280 msec. 18 rows affected.** and **Ln 134, Col 1**.
- System Icons** and **Taskbar** are at the bottom.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with various database objects like Schemas, Tables, and Views. The main area shows a query editor with a SQL script and its execution results.

Query Editor:

```
130 WHERE DATE(B.STARTTIME) = '2012-09-14'  
131 AND (M.MEMID = 0 AND B.SLOTS*F.GUESTCOST > 30 OR M.MEMID <> 0 AND B.SLOTS*F.MEMBERCOST > 30)  
132 ORDER BY COST DESC  
133  
-- 2.7 How can you output a list of all members, including the individual who recommended them (if any), without using any joins?  
134  
135  
136 SELECT DISTINCT MEM.FIRSTNAME|| ' ' || MEM.SURNAME AS MEMBER ,  
137 (SELECT REF.FIRSTNAME|| ' ' || REF.SURNAME AS RECOMMENDER  
138 FROM CD.MEMBERS REF  
139 WHERE REF.MEMID = MEM.RECOMMENDEDBY  
140 )  
141 FROM CD.MEMBERS MEM  
142 ORDER BY MEMBER  
143
```

Data Output:

member text	recommender text
1 Anna Mackenzie	Darren Smith
2 Anne Baker	Ponder Stibbons
3 Burton Tracy	[null]
4 Charles Owen	Darren Smith
5 Darren Smith	[null]
6 David Farrell	[null]
7 David Jones	Janice Joplette
8 David Pinker	Jemima Farrell
9 Douglas Jones	David Jones
10 Erica Crumpet	Tracy Smith

Total rows: 30 of 30 Query complete 00:00:00.098 Successfully run. Total query runtime: 98 msec. 30 rows affected.

18:37 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
 - public
 - Subscriptions
- postgres
- university_db
- Login/Group Roles
- Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```

139 WHERE REF.MEMID = MEM.RECOMMENDED_BY
140 )
141 FROM CD.MEMBERS MEM
142 ORDER BY MEMBER
143
-- 2.8 Produce a list of costly bookings, using a subquery
144 select member, facility, cost from (
145   select
146     m.firstname || ' ' || m.surname as member,
147     f.name as facility,
148     case
149       when m.memid = 0 then
150         b.slots*f.guestcost
151       else
152         b.slots*f.membercost
153
Data Output Messages Notifications
```

member	facility	cost
GUEST GUEST	Massage Room 2	320
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Massage Room 1	160
GUEST GUEST	Tennis Court 2	150
Jemima Farrell	Massage Room 1	140
GUEST GUEST	Tennis Court 1	75
GUEST GUEST	Tennis Court 2	75
GUEST GUEST	Tennis Court 1	75
Matthew Genting	Massage Room 1	70

Total rows: 18 of 18 Query complete 00:00:00.120 ✓ Successfully run. Total query runtime: 120 msec. 18 rows affected. Ln 145, Col 1

USD/INR +0.12% 18:52 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
 - public
 - Subscriptions
- postgres
- university_db
- Login/Group Roles
- Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```

156   on b.facid = f.facid
157   where
158     bstarttime >= '2012-09-14' and
159     bstarttime < '2012-09-15'
160   ) as bookings
161   where cost > 30
162   order by cost desc;
163
164   -- 3 Modifying data
165   -- 3.1 Insert some data into a table
166   INSERT INTO CD.facilities VALUES(9,'Spa',20,30,100000,800)
167
168
169
170
171
172
173
174
Data Output Messages Notifications
```

INSERT @ 1

Query returned successfully in 102 msec.

Total rows: 18 of 18 Query complete 00:00:00.102 Ln 173, Col 1

Earnings upcoming 18:56 19-07-2024

The screenshot shows the pgAdmin 4 interface with the following details:

- Title Bar:** pgAdmin 4
- File Menu:** File, Object, Tools, Help
- Object Explorer:** Shows the database schema structure:
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (2)
 - cd
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
 - public
 - Subscriptions
 - postgres
 - university_db
 - Login/Group Roles
 - Tablespaces
- Query Editor:** The current tab is "Query". The code entered is:

```
165      where cost > 30
166      order by cost desc;
167
168
169
170      -- 3 Modifying data
171      -- 3.1 Insert some data into a table
172      INSERT INTO CD.facilities VALUES(9,'Spa',20,30,100000,800)
173
174      -- 3.2 Insert multiple rows of data into a table
175      INSERT INTO CD.facilities VALUES
176      (9,'Spa',20,30,100000,800),
177      (10,'Squash Court 2',3.5,17.5,5000,80)
178
```
- Data Output:** Shows the message "INSERT 0 2".
- Messages:** Shows the message "Query returned successfully in 130 msec."
- Status Bar:** Total rows: 18 of 18, Query complete 00:00:00.130, 18:59, 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - > cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (3)
 - > bookings
 - > facilities
 - > members
 - > Trigger Functions
 - > Types
 - > Views
 - > public
 - > Subscriptions
- > postgres
- > university.db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

No limit

Query History

Query

```
169
170 -- 3 Modifying data
171 -- 3.1 Insert some data into a table
172 INSERT INTO CD.facilities VALUES(9,'Spa',20,30,100000,800)
173
174 -- 3.2 Insert multiple rows of data into a table
175 INSERT INTO CD.facilities VALUES
176 (9,'Spa',20,30,100000,800),
177 (10,'Squash Court ',3.5,17.5,5000,80)
178
179 INSERT INTO CD.FACILITIES(FACID,NAME,MEMBERCOST,GUESTCOST,INITIALAYOUT,MONTHLYMAINTENANCE)
180 SELECT(SELECT MAX(FACID) FROM CD.FACILITIES)+1,
181 'Spa', 20, 30, 100000, 800;
182
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 83 msec.

Total rows: 18 of 18 Query complete 00:00:00.083

Ln 179, Col 1

30°C Mostly cloudy

19:04 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - < cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - < Tables (3)
 - > bookings
 - > facilities
 - > members
 - > Trigger Functions
 - > Types
 - > Views
 - > public
 - > Subscriptions
 - > postres
 - > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```
176 (9,'Spa',20,30,100000,800),
177 (10,'Squash Court 2',3.5,17.5,5000,80)
178 -- 3.3 Insert calculated data into a table
181 INSERT INTO CD.FACILITIES(FACID,NAME,MEMBERCOST,GUESTCOST,INITIALOUTLAY,MONTHLYMAINTENANCE)
182 SELECT (SELECT MAX(FACID) FROM CD.FACILITIES)+1,
183 'Spa', 20, 30, 100000, 800;
184
185 -- 3.4
187 UPDATE CD.FACILITIES
188 SET INITIALOUTLAY = 10000
189 WHERE NAME = 'Tennis Court 2'
190
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 104 msec.

Total rows: 18 of 18 Query complete 00:00:00.104 ✓ Ln 186, Col 1

Type here to search

30°C Mostly cloudy 19:07 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - < cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - < Tables (3)
 - > bookings
 - > facilities
 - > members
 - > Trigger Functions
 - > Types
 - > Views
 - > public
 - > Subscriptions
 - > postres
 - > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query Query History

```
182 SELECT (SELECT MAX(FACID) FROM CD.FACILITIES)+1,
183 'Spa', 20, 30, 100000, 800;
184
185 -- 3.4 Update some existing data
186
187 UPDATE CD.FACILITIES
188 SET INITIALOUTLAY = 10000
189 WHERE NAME = 'Tennis Court 2'
190
191 -- 3.5 Update multiple rows and columns at the same time
192 UPDATE CD.FACILITIES
193 SET MEMBERCOST = 6 , GUESTCOST = 30
194 WHERE NAME LIKE '%Tennis Court%'
195
```

Data Output Messages Notifications

UPDATE 2

Query returned successfully in 231 msec.

Total rows: 18 of 18 Query complete 00:00:00.231 ✓ Ln 192, Col 1

Type here to search

30°C Mostly cloudy 19:10 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - > cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (3)
 - > bookings
 - > facilities
 - > members
- > Trigger Functions
- > Types
- > Views
- > public
- > Subscriptions
- > postgres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

```
191 -- 3.5 Update multiple rows and columns at the same time
192 UPDATE CD.FACILITIES
193 SET MEMBERCOST = 6 , GUESTCOST = 30
194 WHERE NAME LIKE '%Tennis Court'
195
196 -- 3.6 Update a row based on the contents of another row
197
198 update cd.facilities facts
199 set
200 membercost = (select membercost * 1.1 from cd.facilities where facid = 0),
201 guestcost = (select guestcost * 1.1 from cd.facilities where facid = 0)
202 where facts.facid = 1;
203
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 339 msec.

Total rows: 18 of 18 Query complete 00:00:00.339

✓ Query returned successfully in 339 msec. Ln 198, Col 1

This screenshot shows the pgAdmin 4 interface with the 'exercises/postgres@PostgreSQL 16' connection selected. The 'Schemas' tree on the left shows the 'cd' schema expanded, containing various objects like tables, functions, and triggers. The main window displays a SQL query for updating the 'facilities' table in the 'cd' schema. The query uses a self-update approach to increase member and guest costs by 10% for the first facility. After executing the query, a message indicates it ran successfully in 339 milliseconds, updating 1 row out of 18 total. The status bar at the bottom right shows the date and time as 19-07-2024 19:13.

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - > cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (3)
 - > bookings
 - > facilities
 - > members
- > Trigger Functions
- > Types
- > Views
- > public
- > Subscriptions
- > postgres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

```
198 update cd.facilities facts
199 set
200 membercost = (select membercost * 1.1 from cd.facilities where facid = 0),
201 guestcost = (select guestcost * 1.1 from cd.facilities where facid = 0)
202 where facts.facid = 1;
203
204 -- 3.7 Delete all bookings
205
206 delete from cd.bookings;
207
208 -- 3.8 Delete a member from the cd.members table
209
210 delete from cd.members where memid = 37;
211
212
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 136 msec.

Total rows: 18 of 18 Query complete 00:00:00.136

✓ Query returned successfully in 136 msec. Ln 211, Col 1

This screenshot shows the pgAdmin 4 interface with the same connection and schema structure as the previous one. It displays a SQL query for deleting data from the 'bookings' and 'members' tables. The 'bookings' table is completely deleted, and a single member with memid 37 is deleted from the 'members' table. Both operations are shown as successful with execution times of 136 msec each. The status bar at the bottom right shows the date and time as 19-07-2024 19:16.

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - > cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (3)
 - > bookings
 - > facilities
 - > members
 - > Trigger Functions
 - > Types
 - > Views
 - > public
 - > Subscriptions
- > postgres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

```
204 -- 3.7 Delete all bookings
205 delete from cd.bookings;
206
207 -- 3.8 Delete a member from the cd.members table
208
209 delete from cd.members where memid = 37;
210
211 -- 3.9 Delete based on a subquery
212 delete from cd.members where memid not in (select memid from cd.bookings);
213
214
215
216
```

Data Output Messages Notifications

DELETE 30

Query returned successfully in 78 msec.

Total rows: 18 of 18 Query complete 00:00:00.078 ✓ Query returned successfully in 78 msec. Ln 214, Col 1

Type here to search

30°C Mostly cloudy 19:16 19-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - > cd
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (3)
 - > bookings
 - > facilities
 - > members
 - > Trigger Functions
 - > Types
 - > Views
 - > public
 - > Subscriptions
- > postgres
- > university_db
- > Login/Group Roles
- > Tablespaces

exercises/postgres@PostgreSQL 16

Query History

```
208 -- 3.8 Delete a member from the cd.members table
209
210 delete from cd.members where memid = 37;
211
212 -- 3.9 Delete based on a subquery
213 delete from cd.members where memid not in (select memid from cd.bookings);
214
215 -- 4 Aggregation
216 -- 4.1 Count the number of facilities
217 SELECT COUNT(*) FROM CD.FACILITIES
218
219
220
221
```

Data Output Messages Notifications

count	bigint
1	12

Total rows: 1 of 1 Query complete 00:00:00.095 ✓ Successfully run. Total query runtime: 95 msec. 1 rows affected. Ln 219, Col 1

Type here to search

29°C Mostly cloudy 08:29 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (2)
 - cd
 - Exercises
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Trigger Functions
 - Types
 - Views
- > public
- > Subscriptions
- > postres
- > university_db
- > Login/Group Roles
- > Tablespaces

Query Query History

```

210 delete from cd.members where memid = 37;
211 -- 3.9 Delete based on a subquery
212 delete from cd.members where memid not in (select memid from cd.bookings);
213
214
215
216 -- 4 Aggregation
217 -- 4.1 Count the number of facilities
218 SELECT COUNT(*) FROM CD.FACILITIES
219
220 -- 4.2 Count the number of expensive facilities
221 select count(*) from cd.facilities where guestcost >= 10;
222
223
  
```

Data Output

count	bigint
1	9

Total rows: 1 of 1 Query complete 00:00:00.099 Ln 222, Col 1

✓ Successfully run. Total query runtime: 99 msec. 1 rows affected. X

Type here to search

High winds soon 08:30 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

- > Servers
- > PostgreSQL 16
 - > Databases (4)
 - dvrdrental
 - exercises
 - Casts
 - Catalogs
 - > Event Triggers
 - > Extensions
 - > Foreign Data Wrappers
 - > Languages
 - > Publications
 - > Schemas (2)
 - cd
 - Exercises
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (3)
 - bookings
 - facilities
 - members
 - Columns (8)

Query Query History

```

217 -- 4.1 Count the number of facilities
218 SELECT COUNT(*) FROM CD.FACILITIES
219
220 -- 4.2 Count the number of expensive facilities
221 select count(*) from cd.facilities where guestcost >= 10;
222
223 SELECT * FROM CD. MEMBERS
224
225 -- 4.3 Count the number of recommendations each member makes.
226 SELECT RECOMMENDEDBY,COUNT(*) FROM CD.MEMBERS
227 WHERE RECOMMENDEDBY IS NOT NULL
228 GROUP BY RECOMMENDEDBY
229 ORDER BY RECOMMENDEDBY
230
  
```

Data Output

recommendedby	count	bigint
1	1	5
2	2	3
3	3	1
4	4	2
5	5	1
6	6	1
7	9	2
8	11	1
9	13	2
10	15	1

Total rows: 13 of 13 Query complete 00:00:00.219 Ln 226, Col 1

✓ File saved successfully. X

Type here to search

29°C Mostly cloudy 08:35 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query Query History

```

223 SELECT * FROM CD.MEMBERS
224
225 -- 4.3 Count the number of recommendations each member makes.
226 SELECT RECOMMENDED_BY,COUNT(*) FROM CD.MEMBERS
227 WHERE RECOMMENDED_BY IS NOT NULL
228 GROUP BY RECOMMENDED_BY
229 ORDER BY RECOMMENDED_BY
230
231 SELECT * FROM CD.BOOKINGS
232
233 -- 4.4 List the total slots booked per facility
234 SELECT FACID, SUM(SLOTS) AS "Total Slots" FROM CD.BOOKINGS
235 GROUP BY FACID
236 ORDER BY FACID
237

```

Data Output Messages Notifications

facid	Total Slots
1	0
2	1278
3	1209
4	830
5	1404
6	228
7	1104
8	908
9	911

✓ Successfully run. Total query runtime: 159 msec. 9 rows affected. Ln 234, Col 1

✓ Query returned successfully in 814 msec. Ln 234, Col 1

Type here to search Today's weather 08:44 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query Query History

```

238
239 -- 4.4 List the total slots booked per facility
240 SELECT FACID, SUM(SLOTS) AS "Total Slots" FROM CD.BOOKINGS
241 GROUP BY FACID
242 ORDER BY FACID
243
244 -- 4.5
245 select facid, sum(slots) as "Total Slots"
246   from cd.bookings
247   where
248     starttime >= '2012-09-01'
249     and starttime < '2012-10-01'
250   group by facid
251   order by sum(slots);
252

```

Data Output Messages Notifications

facid	Total Slots
1	5
2	3
3	7
4	8
5	6
6	2
7	1
8	0
9	4

✓ Query returned successfully in 814 msec. Ln 237, Col 1

Type here to search Earnings upcoming 08:47 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query History

```

237 select facid, sum(slots) as "Total Slots"
238   from cd.bookings
239   where
240     starttime >= '2012-09-01'
241     and starttime < '2012-10-01'
242   group by facid
243   order by sum(slots);
244
-- 4.6 List the total slots booked per facility per month
245 SELECT FACID, EXTRACT(MONTH FROM STARTTIME) AS MONTH, SUM(SLOTS) AS "TOTAL SLOTS"
246   FROM CD.BOOKINGS
247   WHERE EXTRACT(YEAR FROM STARTTIME) = 2012
248   GROUP BY FACID, MONTH
249
250 ORDER BY FACID, MONTH;

```

Data Output Messages Notifications

facid	month	TOTAL SLOTS
1	0	7
2	0	8
3	0	9
4	1	7
5	1	8
6	1	9
7	2	7
8	2	8
9	2	9
10	3	7
		104

Total rows: 27 of 27 Query complete 00:00:00.211

Successfully run. Total query runtime: 211 msec. 27 rows affected. Ln 246, Col 1

Query returned successfully in 814 msec. Ln 246, Col 1

30°C Mostly cloudy 08:52 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query History

```

241   and starttime < '2012-10-01'
242   group by facid
243   order by sum(slots);
244
-- 4.6 List the total slots booked per facility per month
245 SELECT FACID, EXTRACT(MONTH FROM STARTTIME) AS MONTH, SUM(SLOTS) AS "TOTAL SLOTS"
246   FROM CD.BOOKINGS
247   WHERE EXTRACT(YEAR FROM STARTTIME) = 2012
248   GROUP BY FACID, MONTH
249
250 ORDER BY FACID, MONTH;
251
252 -- 4.7
253 SELECT COUNT(DISTINCT MEMID) FROM CD.BOOKINGS
254

```

Data Output Messages Notifications

count
1
30

Total rows: 1 of 1 Query complete 00:00:00.162

Successfully run. Total query runtime: 162 msec. 1 rows affected. Ln 254, Col 1

Query returned successfully in 814 msec. Ln 254, Col 1

30°C Mostly cloudy 08:53 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s...

University_db.sql

clubdata.sql*

clubdata.sql*

clubdata.sql*

exercises/post...

Clubdata_Assignment_PostgresSql.sql*

exercises/postgres@PostgreSQL 16

No limit

Query History

```
257 GROUP BY FACID
258 HAVING SUM(SLOTS) > 1000
259 ORDER BY FACID
260
261 -- 4.9 Find the total revenue of each facility
262 SELECT F.NAME, SUM(SLOTS * CASE
263 WHEN MEMID = 0 THEN F.GUESTCOST
264 ELSE F.MEMBERCOST
265 END) AS REVENUE
266 FROM CD.BOOKINGS B
267 INNER JOIN CD.FACILITIES F
268 USING(FACID)
269 GROUP BY F.NAME
270 ORDER BY REVENUE;
```

Data Output

	name	revenue
1	Table Tennis	180
2	Snooker Table	240
3	Pool Table	270
4	Badminton Court	1906.5
5	Squash Court	13468.0
6	Massage Room 2	15810
7	Tennis Court 1	16632
8	Tennis Court 2	18889.2
9	Massage Room 1	72540

Total rows: 9 of 9 Query complete 00:00:00.174

✓ Query returned successfully in 814 msec. X

Ln 261, Col 48

Type here to search

30°C Mostly cloudy 09:04 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s...

University_db.sql

clubdata.sql*

clubdata.sql*

clubdata.sql*

exercises/post...

Clubdata_Assignment_PostgresSql.sql*

exercises/postgres@PostgreSQL 16

No limit

Query History

```
268 USING(FACID)
269 GROUP BY F.NAME
270 ORDER BY REVENUE;
271
272 -- 4.10
273 SELECT NAME, REVENUE FROM
274 (
275 SELECT FACS.NAME, SUM(CASE
276 WHEN MEMID = 0 THEN SLOTS * FACS.GUESTCOST
277 ELSE SLOTS * MEMBERCOST
278 END) AS REVENUE
279 FROM CD.BOOKINGS BKS
280 INNER JOIN CD.FACILITIES FACS
281 ON BKS.FACID = FACS.FACID
282 GROUP BY FACS.NAME
```

Data Output

	name	revenue
1	Table Tennis	180
2	Snooker Table	240
3	Pool Table	270

Total rows: 3 of 3 Query complete 00:00:00.337

✓ Query returned successfully in 814 msec. X

Ln 273, Col 1

Type here to search

30°C Mostly cloudy 09:07 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

248 WHERE EXTRACT(YEAR FROM STARTTIME) = 2012
249 GROUP BY FACID, MONTH
250 ORDER BY FACID, MONTH;
251
252 -- 4.7 Find the count of members who have made at least one booking
253 SELECT COUNT(DISTINCT MEMID) FROM CD.BOOKINGS
254
255 -- 4.8 List facilities with more than 1000 slots booked
256 SELECT FACID, SUM(SLOTS) AS "Total Slots" FROM CD.BOOKINGS
257 GROUP BY FACID
258 HAVING SUM(SLOTS) > 1000
259 ORDER BY FACID
260
261 -- 4.9 Find the total revenue of each facility
262 SELECT F.NAME, SUM(SLOTS * CASE

```

Data Output Messages Notifications

facid	Total Slots
1	0
2	1
3	2
4	4
5	6

✓ Successfully run. Total query runtime: 289 msec. 5 rows affected.

✓ Query returned successfully in 814 msec.

Total rows: 5 of 5 Query complete 00:00:00.289

30°C Mostly cloudy 09:08 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

286
287 -- 4.11 Output the facility id that has the highest number of slots booked
288 SELECT FACID, SUM(SLOTS) AS "TOTAL SLOTS" FROM CD.BOOKINGS
289 GROUP BY FACID
290 ORDER BY SUM(SLOTS) DESC
291 LIMIT 1;
292
293 -- 4.12 List the total slots booked per facility per month, part 2
294
295 SELECT FACID, EXTRACT(MONTH FROM STARTTIME) AS MONTH, SUM(SLOTS) SLOTS FROM CD.BOOKINGS
296 WHERE STARTTIME >= '2012-01-01'
297 AND STARTTIME < '2013-01-01'
298 GROUP BY ROLLUP(FACID, MONTH)
299 ORDER BY FACID, MONTH;

```

Data Output Messages Notifications

facid	month	slots
1	0	7
2	0	8
3	0	9
4	[null]	1320
5	1	7
6	1	8
7	1	9
8	1	1278
9	2	7
10	2	8

✓ Query returned successfully in 814 msec.

Total rows: 37 of 37 Query complete 00:00:00.133

30°C Mostly cloudy 09:15 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query History

```

295 SELECT FACID, EXTRACT(MONTH FROM STARTTIME) AS MONTH, SUM(SLOTS) SLOTS FROM CD.BOOKINGS
296 WHERE STARTTIME >= '2012-01-01'
297 AND STARTTIME < '2013-01-01'
298 GROUP BY ROLLUP(FACID, MONTH)
299 ORDER BY FACID, MONTH;
300
301 -- 4.13 List the total hours booked per named facility
302 SELECT FACS.FACID, FACS.NAME,
303 TRIM(TO_CHAR(SUM(BKS.SLOTS)/2.0, '99999999999999D99')) AS "TOTAL HOURS"
304 FROM CD.BOOKINGS BKS
305 INNER JOIN CD.FACILITIES FACS
306 ON FACS.FACID = BKS.FACID
307 GROUP BY FACS.FACID, FACS.NAME
308 ORDER BY FACS.FACID;

```

Data Output

facid	name	TOTAL HOURS
1	0 Tennis Court 1	660.00
2	1 Tennis Court 2	639.00
3	2 Badminton Court	604.50
4	3 Table Tennis	415.00
5	4 Massage Room 1	702.00
6	5 Massage Room 2	114.00
7	6 Squash Court	552.00
8	7 Snooker Table	454.00
9	8 Pool Table	455.50

Total rows: 9 of 9 Query complete 00:00:00.568

Successfully run. Total query runtime: 568 msec. 9 rows affected. Ln 302, Col 1

Query returned successfully in 814 msec. Ln 302, Col 1

Type here to search

30°C Mostly cloudy 09:18 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query History

```

303 TRIM(TO_CHAR(SUM(BKS.SLOTS)/2.0, '99999999999999D99')) AS "TOTAL HOURS"
304 FROM CD.BOOKINGS BKS
305 INNER JOIN CD.FACILITIES FACS
306 ON FACS.FACID = BKS.FACID
307 GROUP BY FACS.FACID, FACS.NAME
308 ORDER BY FACS.FACID;
309
310 -- 4.14 List each member's first booking after September 1st 2012
311 SELECT MEMS.SURNAME, MEMS.FIRSTNAME, MEMS.MEMID, MIN(BKS.STARTTIME) AS STARTTIME FROM CD.BOOKINGS BKS
312 INNER JOIN CD.MEMBERS MEMS ON
313 MEMS.MEMID = BKS.MEMID
314 WHERE STARTTIME >= '2012-09-01'
315 GROUP BY MEMS.SURNAME, MEMS.FIRSTNAME, MEMS.MEMID
316 ORDER BY MEMS.MEMID;

```

Data Output

surname	firstname	memid	starttime
GUEST	GUEST	0	2012-09-01 08:00:00
Smith	Darren	1	2012-09-01 09:00:00
Smith	Tracy	2	2012-09-01 11:30:00
Rownam	Tim	3	2012-09-01 16:00:00
Joplette	Janice	4	2012-09-01 15:00:00
Butters	Gerald	5	2012-09-02 12:30:00
Tracy	Burton	6	2012-09-01 15:00:00
Dare	Nancy	7	2012-09-01 12:30:00
Bootho	Tim	8	2012-09-01 08:30:00
Stibbons	Ponder	9	2012-09-01 11:00:00

Total rows: 30 of 30 Query complete 00:00:00.539

Query returned successfully in 814 msec. Ln 311, Col 1

Type here to search

Nifty midcap -2.11% 09:19 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query Query History

```

307 GROUP BY FACS.FACID, FACS.NAME
308 ORDER BY FACS.FACID;
309
310 -- 4.14 List each member's first booking after September 1st 2012
311 SELECT MEMS.SURNAME, MEMS.FIRSTNAME, MEMS.MEMID, MIN(BKS.STARTTIME) AS STARTTIME FROM CD.BOOKINGS BKS
312 INNER JOIN CD.MEMBERS MEMS ON
313 MEMS.MEMID = BKS.MEMID
314 WHERE STARTTIME >= '2012-09-01'
315 GROUP BY MEMS.SURNAME, MEMS.FIRSTNAME, MEMS.MEMID
316 ORDER BY MEMS.MEMID;
317
318 -- 4.15 Produce a list of member names, with each row containing the total member count
319 SELECT COUNT(*) OVER(), FIRSTNAME, SURNAME FROM CD.MEMBERS
320 ORDER BY JOINDATE;

```

Data Output Messages Notifications

count	firstname	surname
1	31 GUEST	GUEST
2	31 Darren	Smith
3	31 Tracy	Smith
4	31 Tim	Rownam
5	31 Janice	Joplette
6	31 Gerald	Butters
7	31 Burton	Tracy
8	31 Nancy	Dare
9	31 Tim	Boothe
10	31 Ponder	Stibbons

Total rows: 31 of 31 Query complete 00:00:00.162

✓ Successfully run. Total query runtime: 162 msec. 31 rows affected. ✘

✓ Query returned successfully in 814 msec. ✘

Nifty midcap -2.11% 09:20 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query Query History

```

318 -- 4.15 Produce a list of member names, with each row containing the total member count
319 SELECT COUNT(*) OVER(), FIRSTNAME, SURNAME FROM CD.MEMBERS
320 ORDER BY JOINDATE;
321
322 -- 4.16 Produce a numbered list of members
323 SELECT ROW_NUMBER() OVER(ORDER BY JOINDATE), FIRSTNAME, SURNAME FROM CD.MEMBERS
324 ORDER BY JOINDATE;
325
326 -- 4.17 Produce a numbered list of members
327 SELECT FACID, TOTAL FROM
328 (SELECT FACID, SUM(SLOTS) TOTAL, RANK() OVER (ORDER BY SUM(SLOTS) DESC) RANK FROM CD.BOOKINGS
329 GROUP BY FACID ) AS RANKED
330
331 WHERE RANK = 1

```

Data Output Messages Notifications

facid	total
1	4

Total rows: 1 of 1 Query complete 00:00:00.223

✓ Successfully run. Total query runtime: 223 msec. 1 rows affected. ✘

✓ Query returned successfully in 814 msec. ✘

BSE midcap -2.17% 09:23 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query History

```

327 -- 4.1.7 Produce a numbered list of members
328 SELECT FACID, TOTAL FROM
329 (SELECT FACID, SUM(SLOTS) TOTAL, RANK() OVER (ORDER BY SUM(SLOTS) DESC) RANK FROM CD.BOOKINGS
330 GROUP BY FACID ) AS RANKED
331 WHERE RANK = 1
332
333 -- 4.1.8 Rank members by (rounded) hours used
334 SELECT FIRSTNAME, SURNAME,((SUM(BKS.SLOTS)+10)/20)*10 AS HOURS,
335 RANK() OVER (ORDER BY ((SUM(BKS.SLOTS)+10)/20)*10 DESC) AS RANK
336 FROM CD.BOOKINGS BKS
337 INNER JOIN CD.MEMBERS MEMS
338 ON BKS.MEMID = MEMS.MEMID
339 GROUP BY MEMS.MEMID
340 ORDER BY RANK, SURNAME, FIRSTNAME;

```

Data Output

	firstname	surname	hours	rank
	character varying (200)	character varying (200)	bigint	bigint
1	GUEST	GUEST	1200	1
2	Darren	Smith	340	2
3	Tim	Rownam	330	3
4	Tim	Boothe	220	4
5	Tracy	Smith	220	4
6	Gerald	Butters	210	6
7	Burton	Tracy	180	7
8	Charles	Owen	170	8
9	Janice	Joplette	160	9
10	Anne	Baker	150	10

Total rows: 30 of 30 Query complete 00:00:00.481

✓ Query returned successfully in 814 msec. Ln 334, Col 1

BSE midcap -2.17% 09:25 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query History

```

349 ORDER BY RANK, SURNAME, FIRSTNAME;
350
351 --4.1.9 Find the top three revenue generating facilities
352 SELECT NAME, RANK FROM
353 (SELECT F.NAME AS NAME, RANK() OVER (ORDER BY SUM(CASE
354 WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST
355 ELSE SLOTS * MEMBERCOST
356 END) DESC) AS RANK
357 FROM CD.BOOKINGS B
358 INNER JOIN CD.FACILITIES F
359 USING(FACID)
360 GROUP BY F.NAME) AS SUBQ
361 WHERE RANK <= 3
362
363 ORDER BY RANK;

```

Data Output

	name	rank
	character varying (100)	bigint
1	Massage Room 1	1
2	Tennis Court 2	2
3	Tennis Court 1	3

Total rows: 3 of 3 Query complete 00:00:00.419

✓ Query returned successfully in 814 msec. Ln 343, Col 1

BSE smicap -2.22% 09:28 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

369
370 -- 4.21 Calculate the payback time for each facility
371
372 SELECT F.NAME AS NAME,
373 F.INITIALOUTLAY/(SUM(CASE
374 WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST
375 ELSE SLOTS * MEMBERCOST
376 END)/3) - F.MONTHLYMAINTENANCE) AS MONTHS
377 FROM CD.BOOKINGS B
378 INNER JOIN CD.FACILITIES F
379 USING(FACID)
380 GROUP BY F.FACID
381 ORDER BY NAME;
382

```

Data Output

	name	months
1	Badminton Court	6.8317677198975235
2	Massage Room 1	0.18885741265344664778
3	Massage Room 2	1.7621145374449339
4	Pool Table	5.3333333333333333
5	Snooker Table	6.9230769230769231
6	Squash Court	1.1339582703356516
7	Table Tennis	6.4000000000000000
8	Tennis Court 1	1.8712574850299401
9	Tennis Court 2	1.6403123154648645

Total rows: 9 of 9 Query complete 00:00:00.144

Watchlist ideas 09:34 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

356 SELECT NAME, CASE WHEN CLASS=1 THEN 'high'
357 WHEN CLASS=2 THEN 'average'
358 ELSE 'low'
359 END REVENUE
360 FROM (SELECT F.NAME AS NAME, NTILE(3) OVER (ORDER BY SUM(CASE
361 WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST
362 ELSE SLOTS * MEMBERCOST
363 END) DESC) AS CLASS
364 FROM CD.BOOKINGS B
365 INNER JOIN CD.FACILITIES F
366 USING(FACID)
367 GROUP BY F.NAME) AS SUBQ
368 ORDER BY CLASS, NAME;
369
370 -- 4.21 Calculate the payback time for each facility

```

Data Output

	name	revenue
1	Massage Room 1	high
2	Tennis Court 1	high
3	Tennis Court 2	high
4	Badminton Court	average
5	Massage Room 2	average
6	Squash Court	average
7	Pool Table	low
8	Snooker Table	low
9	Table Tennis	low

Successfully run. Total query runtime: 120 msec. 9 rows affected.

Query returned successfully in 814 msec. Ln 356, Col 1

Nifty bank -0.67% 09:44 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

383 -- 4.22 Calculate a rolling average of total revenue
384
385     SELECT DATEGEN.DATE,
386             (SELECT SUM(CASE
387                 WHEN MEMID = 0 THEN SLOTS * F.GUESTCOST
388                 ELSE SLOTS * MEMBERCOST
389             END) AS REV
390         FROM CD.BOOKINGS B
391         INNER JOIN CD.FACILITIES F
392             USING(FACID)
393         WHERE B.STARTTIME > DATEGEN.DATE - INTERVAL '14 DAYS'
394         AND B.STARTTIME < DATEGEN.DATE + INTERVAL '1 DAY')/15 AS REVENUE
395
396     (SELECT CAST(GENERATE_SERIES(TIMESTAMP '2012-08-01', '2012-08-31', '1 DAY') AS DATE) AS DATE) AS DATEGEN
397     ORDER BY DATEGEN.DATE

```

Data Output Messages Notifications

	date	revenue
	date	numeric
1	2012-08-01	1191.7933333333333333
2	2012-08-02	1219.3200000000000000
3	2012-08-03	1229.4600000000000000
4	2012-08-04	1245.206666666666666667
5	2012-08-05	1228.9333333333333333
6	2012-08-06	1254.3600000000000000
7	2012-08-07	1249.186666666666666667
8	2012-08-08	1238.6400000000000000
9	2012-08-09	1218.506666666666666667
10	2012-08-10	1240.6733333333333333

Total rows: 31 of 31 Query complete 00:00:00.110

✓ Successfully run. Total query runtime: 110 msec. 31 rows affected.

✓ Query returned successfully in 814 msec.

Ln 385, Col 1

Nifty bank -0.67% 09:44 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

396     (SELECT CAST(GENERATE_SERIES(TIMESTAMP '2012-08-01', '2012-08-31', '1 DAY') AS DATE) AS DATE) AS DATEGEN
397     ORDER BY DATEGEN.DATE;
398
399
400 -- 5 Working with Timestamps Begin!
401 -- 5.1 Produce a timestamp for 1 a.m. on the 31st of August 2012
402
403     SELECT TIMESTAMP '2012-08-31 01:00:00';
404
405 -- 5.2 Subtract timestamps from each other
406     SELECT TIMESTAMP '2012-08-31 01:00:00' - TIMESTAMP '2012-07-30 01:00:00' AS INTERVAL;
407
408 -- 5.3
409

```

Data Output Messages Notifications

	timestamp	timestamp without time zone
1	2012-08-31 01:00:00	

Total rows: 1 of 1 Query complete 00:00:00.421

✓ Successfully run. Total query runtime: 421 msec. 1 rows affected.

✓ Query returned successfully in 814 msec.

Ln 404, Col 1

Nifty bank -0.67% 09:44 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s...

exercises/postgres@PostgreSQL 16

Query History

```
396 [SELECT CAST(GENERATE_SERIES(TIMESTAMP '2012-08-01','2012-08-31','1 DAY') AS DATE) AS DATEGEN
397 ORDER BY DATEGEN.DATE;
398
399 -- 5 Working with Timestamps Begin!
400 -- 5.1 Produce a timestamp for 1 a.m. on the 31st of August 2012
401
402
403 SELECT TIMESTAMP '2012-08-31 01:00:00';
404
405 -- 5.2 Subtract timestamps from each other
406 SELECT TIMESTAMP '2012-08-31 01:00:00' - TIMESTAMP '2012-07-30 01:00:00' AS INTERVAL;
407
408 -- 5.3
409
```

Data Output

	timestamp	timestamp without time zone
1	2012-08-31 01:00:00	

Total rows: 1 of 1 Query complete 00:00:00.421

Ln 407, Col 1

Query returned successfully in 814 msec. X

Nifty bank -0.67% 09:44 ENG 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s...

exercises/postgres@PostgreSQL 16

Query History

```
396 [SELECT CAST(GENERATE_SERIES(TIMESTAMP '2012-08-01','2012-08-31','1 DAY') AS DATE) AS DATEGEN
397 ORDER BY DATEGEN.DATE;
398
399 -- 5 Working with Timestamps Begin!
400 -- 5.1 Produce a timestamp for 1 a.m. on the 31st of August 2012
401
402
403 SELECT TIMESTAMP '2012-08-31 01:00:00';
404
405 -- 5.2 Subtract timestamps from each other
406 SELECT TIMESTAMP '2012-08-31 01:00:00' - TIMESTAMP '2012-07-30 01:00:00' AS INTERVAL;
407
408 -- 5.3
409
```

Data Output

	interval	interval
1	32 days	

Total rows: 1 of 1 Query complete 00:00:00.138

Ln 407, Col 1

Query returned successfully in 814 msec. X

Nifty bank -0.67% 09:44 ENG 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s...

University_db.sql

clubdata.sql*

clubdata.sql*

clubdata.sql*

exercises/post...

Clubdata_Assignment_PostgresSql.sql*

exercises/postgres@PostgreSQL 16

Query History

397 ORDER BY DATEGEN.DATE;

398

399 -- 5 Working with Timestamps Begin!

400 -- 5.1 Produce a timestamp for 1 a.m. on the 31st of August 2012

401

402

403 SELECT TIMESTAMP '2012-08-31 01:00:00';

404

405 -- 5.2 Subtract timestamps from each other

406 SELECT TIMESTAMP '2012-08-31 01:00:00' - TIMESTAMP '2012-07-30 01:00:00' AS INTERVAL;

407

408 -- 5.3

409 SELECT GENERATE_SERIES(TIMESTAMP '2012-10-01', TIMESTAMP '2012-10-31', INTERVAL '1 DAY') AS TS;

410

Data Output Messages Notifications

ts
timestamp without time zone
1 2012-10-01 00:00:00
2 2012-10-02 00:00:00
3 2012-10-03 00:00:00
4 2012-10-04 00:00:00
5 2012-10-05 00:00:00
6 2012-10-06 00:00:00
7 2012-10-07 00:00:00
8 2012-10-08 00:00:00
9 2012-10-09 00:00:00
10 2012-10-10 00:00:00

Total rows: 31 of 31 Query complete 00:00:00.275

✓ Successfully run. Total query runtime: 275 msec. 31 rows affected.

✓ Query returned successfully in 814 msec.

Ln 410, Col 1

Type here to search

30°C Mostly cloudy 09:45 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s...

University_db.sql

clubdata.sql*

clubdata.sql*

clubdata.sql*

exercises/post...

Clubdata_Assignment_PostgresSql.sql*

exercises/postgres@PostgreSQL 16

Query History

403 SELECT TIMESTAMP '2012-08-31 01:00:00';

404

405 -- 5.2 Subtract timestamps from each other

406 SELECT TIMESTAMP '2012-08-31 01:00:00' - TIMESTAMP '2012-07-30 01:00:00' AS INTERVAL;

407

408 -- 5.3 Generate a list of all the dates in October 2012

409 SELECT GENERATE_SERIES(TIMESTAMP '2012-10-01', TIMESTAMP '2012-10-31', INTERVAL '1 DAY') AS TS;

410

411 -- 5.4 Get the day of the month from a timestamp

412 SELECT EXTRACT(DAY FROM TIMESTAMP '2012-08-31');

413

414 -- 5.5

415 SELECT EXTRACT(EPOCH FROM (TIMESTAMP '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));

416

417

Data Output Messages Notifications

extract
numeric
1 169200.000000

Total rows: 1 of 1 Query complete 00:00:00.120

✓ Successfully run. Total query runtime: 120 msec. 1 rows affected.

✓ Query returned successfully in 814 msec.

Ln 416, Col 1

Type here to search

Earnings upcoming 09:47 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Aa FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> t.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

410
411 -- 5.4 Get the day of the month from a timestamp
412 SELECT EXTRACT(DAY FROM TIMESTAMP '2012-08-31');
413
414 -- 5.5 Work out the number of seconds between timestamps
415
416 SELECT EXTRACT(EPOCH FROM (TIMESTAMP '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));
417
418 -- 5.6 Work out the number of days in each month of 2012
419
420 SELECT EXTRACT(MONTH FROM CAL.MONTH) AS MONTH,
421 (CAL.MONTH + INTERVAL '1 MONTH') - CAL.MONTH AS LENGTH FROM
422 (SELECT GENERATE_SERIES(TIMESTAMP '2012-01-01', TIMESTAMP '2012-12-01', INTERVAL '1 MONTH') AS MONTH) CAL
423 ORDER BY MONTH;

```

Data Output Messages Notifications

month	length	interval
numeric		
1	1	31 days
2	2	29 days
3	3	31 days
4	4	30 days
5	5	31 days
6	6	30 days
7	7	31 days
8	8	31 days
9	9	30 days
10	10	31 days

Total rows: 12 of 12 Query complete 00:00:00.159 Ln 420, Col 1

Query returned successfully in 814 msec. X

Type here to search

Cloud 30°C Mostly cloudy 09:50 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Aa FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> t.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

415
416 SELECT EXTRACT(EPOCH FROM (TIMESTAMP '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));
417
418 -- 5.6 Work out the number of days in each month of 2012
419
420 SELECT EXTRACT(MONTH FROM CAL.MONTH) AS MONTH,
421 (CAL.MONTH + INTERVAL '1 MONTH') - CAL.MONTH AS LENGTH FROM
422 (SELECT GENERATE_SERIES(TIMESTAMP '2012-01-01', TIMESTAMP '2012-12-01', INTERVAL '1 MONTH') AS MONTH) CAL
423 ORDER BY MONTH;
424
425 -- 5.7 Work out the number of days in each month of 2012
426
427 SELECT (DATE_TRUNC('MONTH', TS.TESTTS) + INTERVAL '1 MONTH') - DATE_TRUNC('DAY', TS.TESTTS) AS REMAINING
428 FROM (SELECT TIMESTAMP '2012-02-11 01:00:00' AS TESTTS) TS

```

Data Output Messages Notifications

month	length	interval
numeric		
1	1	31 days
2	2	29 days
3	3	31 days
4	4	30 days
5	5	31 days
6	6	30 days
7	7	31 days
8	8	31 days
9	9	30 days
10	10	31 days

Total rows: 12 of 12 Query complete 00:00:00.115 Ln 420, Col 1

Query returned successfully in 814 msec. X

Type here to search

Cloud Morning news 09:52 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

415
416 SELECT EXTRACT(EPOCH FROM (TIMESTAMP '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));
417
418 -- 5.6 Work out the number of days in each month of 2012
419
420 SELECT EXTRACT(MONTH FROM CAL.MONTH) AS MONTH,
421 (CAL.MONTH + INTERVAL '1 MONTH') - CAL.MONTH AS LENGTH
422 FROM (SELECT GENERATE_SERIES(TIMESTAMP '2012-01-01', TIMESTAMP '2012-12-01', INTERVAL '1 MONTH') AS MONTH) CAL
423 ORDER BY MONTH;
424
425 -- 5.7 Work out the number of days in each month of 2012
426
427 SELECT (DATE_TRUNC('MONTH',TS.TESTTS) + INTERVAL '1 MONTH') - DATE_TRUNC('DAY', TS.TESTTS) AS REMAINING
428 FROM (SELECT TIMESTAMP '2012-02-11 01:00:00' AS TESTTS) TS

```

Data Output Messages Notifications

remaining interval
1 19 days

Successfully run. Total query runtime: 303 msec. 1 rows affected.

Query returned successfully in 814 msec.

Total rows: 1 of 1 Query complete 00:00:00.303

Ln 427, Col 1

Windows Taskbar: Type here to search, Morning news, 09:52, 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

428 SELECT EXTRACT(MONTH FROM CAL.MONTH) AS MONTH,
429 (CAL.MONTH + INTERVAL '1 MONTH') - CAL.MONTH AS LENGTH
430 FROM (SELECT GENERATE_SERIES(TIMESTAMP '2012-01-01', TIMESTAMP '2012-12-01', INTERVAL '1 MONTH') AS MONTH) CAL
431 ORDER BY MONTH;
432
433 -- 5.7 Work out the number of days in each month of 2012
434
435 SELECT (DATE_TRUNC('MONTH',TS.TESTTS) + INTERVAL '1 MONTH') - DATE_TRUNC('DAY', TS.TESTTS) AS REMAINING
436 FROM (SELECT TIMESTAMP '2012-02-11 01:00:00' AS TESTTS) TS
437
438 -- 5.8 Work out the end time of bookings
439 SELECT STARTTIME, STARTTIME + SLOTS*(INTERVAL '30 MINUTES') ENDTIME FROM CD.BOOKINGS
440 ORDER BY ENDTIME DESC, STARTTIME DESC
441 LIMIT 10

```

Data Output Messages Notifications

starttime timestamp without time zone	endtime timestamp without time zone
2013-01-01 15:30:00	2013-01-01 16:00:00
2012-09-30 19:30:00	2012-09-30 20:30:00
2012-09-30 19:00:00	2012-09-30 20:30:00
2012-09-30 19:30:00	2012-09-30 20:00:00
2012-09-30 19:00:00	2012-09-30 20:00:00
2012-09-30 19:00:00	2012-09-30 20:00:00
2012-09-30 18:30:00	2012-09-30 20:00:00
2012-09-30 18:30:00	2012-09-30 19:30:00
2012-09-30 18:30:00	2012-09-30 19:30:00

Successfully run. Total query runtime: 125 msec. 10 rows affected.

Query returned successfully in 814 msec.

Total rows: 10 of 10 Query complete 00:00:00.125

Ln 431, Col 1

Windows Taskbar: Type here to search, Morning news, 09:54, 22-07-2024

The screenshot shows the pgAdmin 4 interface with the following details:

- File Object Tools Help** menu at the top.
- Object Explorer** sidebar on the left listing database configurations, functions, tables, and other objects.
- Query History** tab selected in the main pane.
- SQL Editor** containing the following query:

```
-- 5.7 Work out the number of days in each month of 2012
SELECT (DATE_TRUNC('MONTH', TS.TESTTS) + INTERVAL '1 MONTH') - DATE_TRUNC('DAY', TS.TESTTS) AS REMAINING
FROM (SELECT TIMESTAMP '2012-02-11 01:00:00' AS TESTTS) TS

-- 5.8 Work out the end time of bookings
SELECT STARTTIME, STARTTIME + SLOTS*(INTERVAL '30 MINUTES') ENDTIME FROM CD.BOOKINGS
ORDER BY ENDTIME DESC, STARTTIME DESC
LIMIT 10

-- 5.9 Return a count of bookings for each month
SELECT DATE_TRUNC('MONTH', STARTTIME) AS MONTH, COUNT(*) FROM CD.BOOKINGS
GROUP BY MONTH
ORDER BY MONTH
```

Data Output tab shows the results of the last query:

month	timestamp without time zone	count
1	2012-07-01 00:00:00	658
2	2012-08-01 00:00:00	1472
3	2012-09-01 00:00:00	1913
4	2013-01-01 00:00:00	1

Status Bar at the bottom indicates:
Total rows: 4 of 4 Query complete 00:00:00.379
✓ Successfully run. Total query runtime: 379 msec. 4 rows affected.
✓ Query returned successfully in 814 msec.
Ln 436, Col 1

The screenshot shows the pgAdmin 4 interface with the following details:

- File Object Tools Help**
- Object Explorer** pane on the left showing database structures like FTS Configurations, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (3), and members.
- Current Connection**: exercises/postgres@PostgreSQL 16
- Query Editor** pane containing the following SQL code:

```
-- 5.9 Return a count of bookings for each month
SELECT DATE_TRUNC('MONTH', STARTTIME) AS MONTH, COUNT(*) FROM CD.BOOKINGS
GROUP BY MONTH
ORDER BY MONTH

-- 5.10 Work out the utilisation percentage for each facility by month
SELECT NAME, MONTH, ROUND((100*SLOTS)/
CAST(25*(CAST((MONTH + INTERVAL '1 MONTH') AS DATE)- CAST (MONTH AS DATE)) AS NUMERIC),1) AS UTILISATION
FROM (SELECT F.NAME AS NAME, DATE_TRUNC('MONTH', STARTTIME) AS MONTH, SUM(SLOTS) AS SLOTS FROM CD.BOOKINGS B
INNER JOIN CD.FACILITIES F
USING(FACID)
GROUP BY F.FACID, MONTH) AS INN
ORDER BY NAME, MONTH
```
- Data Output** pane showing the results of the query:

	name	month	utilisation
1	Badminton Court	2012-07-01 00:00:00	23.2
2	Badminton Court	2012-08-01 00:00:00	59.2
3	Badminton Court	2012-09-01 00:00:00	76.0
4	Massage Room 1	2012-07-01 00:00:00	34.1
5	Massage Room 1	2012-08-01 00:00:00	63.5
6	Massage Room 1	2012-09-01 00:00:00	86.4
7	Massage Room 2	2012-07-01 00:00:00	3.1
8	Massage Room 2	2012-08-01 00:00:00	10.6
9	Massage Room 2	2012-09-01 00:00:00	16.3
10	Pool Table	2012-07-01 00:00:00	15.1

- Messages** pane showing "Successfully run. Total query runtime: 218 msec. 28 rows affected."
- Notifications** pane showing no notifications.
- System Bar** at the bottom with search, file, and system status icons.

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query History

```

440 -- 5.10 Work out the utilisation percentage for each facility by month
441 SELECT NAME, MONTH, ROUND((100*SLOTS)/
442 CAST(25*(CAST((MONTH + INTERVAL '1 MONTH') AS DATE)- CAST (MONTH AS DATE)) AS NUMERIC),1) AS UTILISATION
443 FROM (SELECT F.NAME AS NAME, DATE_TRUNC('MONTH', STARTTIME) AS MONTH, SUM(SLOTS) AS SLOTS FROM CD.BOOKINGS B
444 INNER JOIN CD.FACILITIES F
445 USING(FACID)
446 GROUP BY F.FACID, MONTH) AS INN
447 ORDER BY NAME, MONTH
448
449
450 -- 6 STRING
451 -- 6.1 Format the names of members
452 SELECT SURNAME || ' ' || FIRSTNAME AS NAME FROM CD.MEMBERS
453

```

Data Output Messages Notifications

	name	text
1	GUEST	GUEST, GUEST
2	Smith, Darren	
3	Smith, Tracy	
4	Rownam, Tim	
5	Joplette, Janice	
6	Butters, Gerald	
7	Tracy, Burton	
8	Dare, Nancy	
9	Booth, Tim	
10	Stibbons, Ponder	

Total rows: 31 of 31 Query complete 00:00:00.108 ✓ Successfully run. Total query runtime: 108 msec. 31 rows affected. Ln 453, Col 1

Type here to search Morning news 10:09 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

Query History

```

449
450 -- 6 STRING
451 -- 6.1 Format the names of members
452 SELECT SURNAME || ' ' || FIRSTNAME AS NAME FROM CD.MEMBERS
453
454 -- 6.2 Perform a case-insensitive search
455 SELECT * FROM CD.FACILITIES WHERE NAME LIKE 'Tennis%';
456
457 -- 6.3 Find telephone numbers with parentheses
458 SELECT MEMID, TELEPHONE FROM CD.MEMBERS WHERE TELEPHONE ~ '[(())]';
459
460 -- 6.4 Pad zip codes with leading zeroes
461 SELECT LPAD(CAST(ZIPCODE AS CHAR(5)),5,'0') ZIP FROM CD.MEMBERS ORDER BY ZIP
462

```

Data Output Messages Notifications

	zip	text
1	00000	
2	00234	
3	00234	
4	04321	
5	04321	
6	10383	
7	11986	
8	23423	
9	28563	
10	33862	

Total rows: 31 of 31 Query complete 00:00:00.366 ✓ Successfully run. Total query runtime: 366 msec. 31 rows affected. Ln 462, Col 1

Type here to search NASDAQ -0.81% 10:14 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

454 -- 6.2 Perform a case-insensitive search
455 SELECT * FROM CD.FACILITIES WHERE NAME LIKE 'Tennis%';
456
457 -- 6.3 Find telephone numbers with parentheses
458 SELECT MEMID, TELEPHONE FROM CD.MEMBERS WHERE TELEPHONE ~ '[(())]';
459
460 -- 6.4 Pad zip codes with leading zeroes
461 SELECT LPAD(CAST(ZIPCODE AS CHAR(5)),5,'0') ZIP FROM CD.MEMBERS ORDER BY ZIP
462
463 -- 6.5 Count the number of members whose surname starts with each letter of the alphabet
464 SELECT SUBSTR (MEMS.SURNAME,1,1) AS LETTER, COUNT(*) AS COUNT FROM CD.MEMBERS MEMS
465 GROUP BY LETTER
466 ORDER BY LETTER

```

Data Output Messages Notifications

letter	text	count
1	B	5
2	C	2
3	D	1
4	F	2
5	G	2
6	H	1
7	J	3
8	M	1
9	O	1
10	P	2

Total rows: 14 of 14 Query complete 00:00:00.159 Ln 464, Col 1

Successfully run. Total query runtime: 159 msec. 14 rows affected.

NASDAQ -0.81% 10:15 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query Query History

```

459
460 -- 6.4 Pad zip codes with leading zeroes
461 SELECT LPAD(CAST(ZIPCODE AS CHAR(5)),5,'0') ZIP FROM CD.MEMBERS ORDER BY ZIP
462
463 -- 6.5 Count the number of members whose surname starts with each letter of the alphabet
464 SELECT SUBSTR (MEMS.SURNAME,1,1) AS LETTER, COUNT(*) AS COUNT FROM CD.MEMBERS MEMS
465 GROUP BY LETTER
466 ORDER BY LETTER
467
468 -- 6.6 Clean up telephone numbers
469
470 SELECT MEMID, TRANSLATE(TELEPHONE, '-(.) ', '') AS TELEPHONE
471 FROM CD.MEMBERS
472 ORDER BY MEMID;

```

Data Output Messages Notifications

memid	telephone
1	0 000000000
2	1 5555555555
3	2 5555555555
4	3 8446930723
5	4 8339424710
6	5 8440784130
7	6 8223549973
8	7 8337764001
9	8 8114332547
10	9 8331603900

Total rows: 31 of 31 Query complete 00:00:00.435 Ln 470, Col 1

Successfully run. Total query runtime: 435 msec. 31 rows affected.

30°C Mostly cloudy 10:16 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query History

```

474 -- / RECURSION
475 -- 7.1 Find the upward recommendation chain for member ID 27
476 WITH RECURSIVE RECOMMENDERS(RECOMMENDER) AS (
477   SELECT RECOMMENDED_BY FROM CD.MEMBERS WHERE MEMID = 27
478 UNION ALL
479   SELECT M.RECOMMENDED_BY
480   FROM RECOMMENDERS R
481   INNER JOIN CD.MEMBERS M
482   ON M.MEMID = R.RECOMMENDER
483   SELECT R.RECOMMENDER, M.FIRSTNAME, M.SURNAME
484   FROM RECOMMENDERS R
485   INNER JOIN CD.MEMBERS M
486   ON R.RECOMMENDER = M.MEMID
487 ORDER BY MEMID DESC

```

Data Output

recommender	firstname	surname
1	20	Matthew Genting
2	5	Gerald Butters
3	1	Darren Smith

Total rows: 3 of 3 Query complete 00:00:00.298 Ln 474, Col 15

Type here to search

BSE midcap +0.72% ENG IN 22-07-2024

pgAdmin 4

File Object Tools Help

Object Explorer

```

> FTS Configurations
> FTS Dictionaries
> Ae FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> i.3 Sequences
> Tables (3)
> bookings
> facilities
> members
> Columns (8)
  memid
  surname
  firstname
  address
  zipcode
  telephone
  recommended
  joindate
> Constraints
> Indexes
> RLS Policies
> Rules
> Triggers
> Trigger Functions
> Types
> Views

```

exercises/postgres@PostgreSQL 16

Query History

```

488
489 -- 7.2 Find the downward recommendation chain for member ID 1
490 WITH RECURSIVE RECOMMENDED(MEMID) AS (
491   SELECT MEMID FROM CD.MEMBERS WHERE RECOMMENDED_BY = 1
492 UNION ALL
493   SELECT M.MEMID FROM RECOMMENDED R
494   INNER JOIN CD.MEMBERS M
495   ON M.RECOMMENDED_BY = R.MEMID
496   SELECT R.MEMID, M.FIRSTNAME, M.SURNAME
497   FROM RECOMMENDED R
498   INNER JOIN CD.MEMBERS M
499   ON R.MEMID = M.MEMID
500 ORDER BY MEMID

```

Data Output

memid	firstname	surname
1	4	Janice Joplette
2	5	Gerald Butters
3	7	Nancy Dare
4	10	Charles Owen
5	11	David Jones
6	14	Jack Smith
7	20	Matthew Genting
8	21	Anna Mackenzie
9	26	Douglas Jones
10	27	Henrietta Rumney

Total rows: 10 of 10 Query complete 00:00:00.174 Ln 490, Col 1

pgAdmin 4

File Object Tools Help

Object Explorer

University_db.s... University_db.sql clubdata.sql* clubdata.sql* clubdata.sql* exercises/post... Clubdata_Assignment_PostgresSql.sql*

exercises/postgres@PostgreSQL 16

Query History

```
494 INNER JOIN CD.MEMBERS M
495 ON M.RECOMMENDED_BY = R.MEMID
496 SELECT R.MEMID, M.FIRSTNAME, M.SURNAME
497 FROM RECOMMENDS R
498 INNER JOIN CD.MEMBERS M
499 ON R.MEMID = M.MEMID
500 ORDER BY MEMID
501
502 -- 7.3 Produce a CTE that can return the upward recommendation chain for any member
503 WITH RECURSIVE RECOMMENDERS(RECOMMENDER, MEMBER) AS (
504     SELECT RECOMMENDED_BY, MEMID
505     FROM CD.MEMBERS
506     UNION ALL
507     SELECT M.RECOMMENDED_BY, R.MEMBER
508     FROM RECOMMENDERS R
```

Data Output Messages Notifications

member	recommended	firstname	surname
1	12	9	Ponder
2	12	6	Burton
3	22	16	Timothy
4	22	13	Jemima
			Farrell

Total rows: 4 of 4 Query complete 00:00:00.145 Ln 517, Col 1

Type here to search

Windows Taskbar: File Explorer, Mail, Google Chrome, Microsoft Edge, File, Start, Task View, Taskbar Icons, Weather (32°C), System (Mostly cloudy), Language (ENG IN), Date (22-07-2024)