**Data base:**

**Basic**

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

Answer: **select** \* **from** cd.facilities;

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?

Ans: **select** name, membercost **from** cd.facilities;

# 3. **Control which rows are retrieved**

### Question

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

Answ: **select** \* **from** cd.facilities **where** membercost > 0;

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.

Answer: **select** facid, name, membercost, monthlymaintenance **from** cd.facilities **where** membercost > 0 **and** (membercost < monthlymaintenance/50.0);

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%';

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

**Join ans subquareyselect b.starttime,f.name from cd.facilities f**

**left join cd.bookings b on b.facid= f.facid**

**where f.name like '%Tennis Court%' and**

**date(b.starttime)='2012-09-21'**

**order by b.starttime**

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

Ans : **select** bks.starttime **from** cd.bookings bks **inner** **join** cd.members mems **on** mems.memid = bks.memid **where** mems.firstname='David' **and** mems.surname='Farrell';

select starttime from cd.bookings f left join cd.members m on f.memid=m.memid

where m.firstname like '%David%' and surname like '%Farrell%';

10.   
**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.**

**Answers:** select b.starttime,f.name from cd.facilities f

left join cd.bookings b on b.facid= f.facid

where f.name like '%Tennis Court%' and

date(b.starttime)='2012-09-21'

order by b.starttime

;

11

**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).**

**Ans** select m.firstname,m.surname from cd.members m

inner join(select DISTINCT recommendedby from cd.members m )t

on m.memid=t.recommendedby

order by surname,firstname;

12. **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).**

Asw: **select** mems.firstname **as** memfname, mems.surname **as** memsname, recs.firstname **as** recfname, recs.surname **as** recsname **from** cd.members mems **left** **outer** **join** cd.members recs **on** recs.memid = mems.recommendedby **order** **by** memsname, memfname;

13. **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.**

**Ans:** **select** **distinct** mems.firstname || ' ' || mems.surname **as** member, facs.name **as** facility **from** cd.members mems **inner** **join** cd.bookings bks **on** mems.memid = bks.memid **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **where** facs.name **in** ('Tennis Court 2','Tennis Court 1') **order** **by** member, facility

14**. 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.**

**Answ: select** mems.firstname || ' ' || mems.surname **as** member, facs.name **as** facility, **case** **when** mems.memid = 0 **then** bks.slots\*facs.guestcost

**else**

bks.slots\*facs.membercost **end** **as** cost **from**

cd.members mems **inner** **join** cd.bookings bks **on**

mems.memid = bks.memid **inner** **join** cd.facilities facs

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

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

**and** bks.starttime < '2012-09-15' **and**

( (mems.memid = 0 **and** bks.slots\*facs.guestcost > 30) **or** (mems.memid != 0 **and** bks.slots\*facs.membercost > 30) ) **order** **by** cost **desc**;

15**. 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.**

**Answers:** **select** **distinct** mems.firstname || ' ' || mems.surname **as** member, (**select** recs.firstname || ' ' || recs.surname **as** recommender **from** cd.members recs **where** recs.memid = mems.recommendedby ) **from** cd.members mems **order** **by** member;

16. The [Produce a list of costly bookings](https://pgexercises.com/questions/joins/threejoin2.html) exercise contained some messy logic: we had to calculate the booking cost in both the WHERE clause and the CASE statement. Try to simplify this calculation using subqueries. For reference, the question was:

*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.*

**Answ:** select member, facility, cost from ( select mems.firstname || ' ' || mems.surname as member, facs.name as facility, case

when mems.memid = 0 then bks.slots\*facs.guestcost

else

bks.slots\*facs.membercost end as cost from cd.members mems

inner join cd.bookings bks on mems.memid = bks.memid

inner join cd.facilities facs on bks.facid = facs.facid

where bks.starttime >= '2012-09-14' and bks.starttime < '2012-09-15' ) as bookings where cost > 30 order by cost desc;

1**7.modifying the data**

The club is adding a new facility - a spa. We need to add it into the facilities table. Use the following values:

* facid: 9, Name: 'Spa', membercost: 20, guestcost: 30, initialoutlay: 100000, monthlymaintenance: 800.

**Answ:** **insert** **into** cd.facilities

(facid, name, membercost, guestcost, initialoutlay, monthlymaintenance)

**values** (9, 'Spa', 20, 30, 100000, 800);

18.   
In the previous exercise, you learned how to add a facility. Now you're going to add multiple facilities in one command. Use the following values:

* facid: 9, Name: 'Spa', membercost: 20, guestcost: 30, initialoutlay: 100000, monthlymaintenance: 800.
* facid: 10, Name: 'Squash Court 2', membercost: 3.5, guestcost: 17.5, initialoutlay: 5000, monthlymaintenance: 80.

**Answ:** **insert** **into** cd.facilities

(facid, name, membercost, guestcost, initialoutlay, monthlymaintenance) **values**

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

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

19.   
Let's try adding the spa to the facilities table again. This time, though, we want to automatically generate the value for the next facid, rather than specifying it as a constant. Use the following values for everything else:

* Name: 'Spa', membercost: 20, guestcost: 30, initialoutlay: 100000, monthlymaintenance: 800

**Answ:** **insert** **into** cd.facilities

(facid, name, membercost, guestcost, initialoutlay, monthlymaintenance) **select** (**select** max(facid) **from** cd.facilities)+1, 'Spa', 20, 30, 100000, 800;

20. We made a mistake when entering the data for the second tennis court. The initial outlay was 10000 rather than 8000: you need to alter the data to fix the error.

Answ: **update** cd.facilities

**set** initialoutlay = 10000

**where** facid = 1;

30. We want to increase the price of the tennis courts for both members and guests. Update the costs to be 6 for members, and 30 for guests.

Ans: **update** cd.facilities

**set** membercost = 6,

guestcost = 30

**where** facid **in** (0,1);

31.   
We want to alter the price of the second tennis court so that it costs 10% more than the first one. Try to do this without using constant values for the prices, so that we can reuse the statement if we want to.

Answ: **update** cd.facilities facs

**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;

33. As part of a clearout of our database, we want to delete all bookings from the cd.bookings table. How can we accomplish this?

**delete** **from** cd.bookings;

34. We want to remove member 37, who has never made a booking, from our database. How can we achieve that?

Ans: **delete** **from** cd.members **where** memid = 37;

35. In our previous exercises, we deleted a specific member who had never made a booking. How can we make that more general, to delete all members who have never made a booking?

Answ: **delete** **from** cd.members **where** memid **not** **in** (**select** memid **from** cd.bookings);

Or

**delete** **from** cd.members mems **where** **not** **exists** (**select** 1 **from** cd.bookings **where** memid = mems.memid);

36.