PostgreSQL Exercises

Back to listing

Home / Joins / Simplejoin

Next Exerci

Retrieve the start times of members' booking

Question

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

Schema reminder ▲

cd.members

memid	integer
surname	character varying(200)
firstname	character varying(200)
address	character varying(300)
zipcode	integer
telephone	character varying(20)
recommendedby	integer
joindate	timestamp

cd.bookings

facid	integer
memid	integer
starttime	timestamp
slots	integer

cd.facilities

facid	integer
name	character varying(100)
membercost	numeric
guestcost	numeric
initialoutlay	numeric
monthlymaintenance	numeric

Expected Results

starttime
2012-09-18 09:00:00
2012-09-18 17:30:00
2012-09-18 13:30:00
2012-09-18 20:00:00
2012-09-19 09:30:00
2012-09-19 15:00:00
2012-09-19 12:00:00
2012-09-20 15:30:00
2012-09-20 11:30:00
2012-09-20 14:00:00

Your Answer

Hint Help Save Run Query

```
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';
```

starttime
2012-09-18 09:00:00
2012-09-18 17:30:00
2012-09-18 13:30:00
2012-09-18 20:00:00
2012-09-19 09:30:00
2012-09-19 15:00:00

Answers and Discussion Hide

```
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';
```

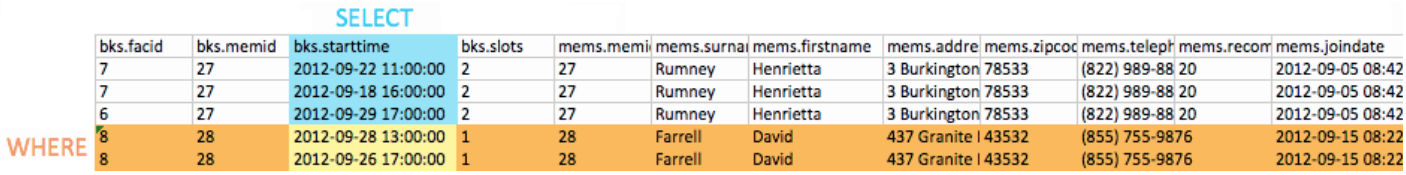
The most commonly used kind of join is the `INNER JOIN`. What this does is combine two tables based on a join expression - in this case, for each member id in the members table, we're looking for matching values in the bookings table. Where we find a match, a row combining the values for each table is returned. Note that we've given each table *alias* (bks and mems). This is used for two reasons: firstly, it's convenient, and secondly we might join to the same table several times, requiring us to distinguish between columns from each different time the table was joined in.

Let's ignore our select and where clauses for now, and focus on what the `FROM` statement produces. In all our previous examples, FROM has just been a simple table. What is it now? Another table! This time, it's produced as a composite of bookings and members. You can see a subset of the output of the join below:

bks.facid	bks.memid	bks.starttime	bks.slots	mems.memi	mems.surna	mems.firstname	mems.addre	mems.zipcoc	mems.teleph	mems.recom	mems.joindate
...	...	...	...	...	...	...	...	...	...	...	...
7	27	2012-09-22 11:00:00	2	27	Rumney	Henrietta	3 Burkington	78533	(822) 989-88	20	2012-09-05 08:42:3
7	27	2012-09-18 16:00:00	2	27	Rumney	Henrietta	3 Burkington	78533	(822) 989-88	20	2012-09-05 08:42:3
6	27	2012-09-29 17:00:00	2	27	Rumney	Henrietta	3 Burkington	78533	(822) 989-88	20	2012-09-05 08:42:3
8	28	2012-09-28 13:00:00	1	28	Farrell	David	437 Granite I	43532	(855) 755-9876		2012-09-15 08:22:0
8	28	2012-09-26 17:00:00	1	28	Farrell	David	437 Granite I	43532	(855) 755-9876		2012-09-15 08:22:0
...	...	...	...	...	...	...	...	...	...	...	...

For each member in the members table, the join has found all the matching member ids in the bookings table. For each match, it's then produced a row combining the row from the members table, and the row from the bookings table.

Obviously, this is too much information on its own, and any useful question will want to filter it down. In our query, we use the start of the SELECT clause to pick columns, and the WHERE clause to pick rows, as illustrated below:



bks.facid	bks.memid	bks.starttime	bks.slots	mems.memi	mems.surna	mems.firstname	mems.addre	mems.zipcoc	mems.teleph	mems.recom	mems.joindate
7	27	2012-09-22 11:00:00	2	27	Rumney	Henrietta	3 Burkington	78533	(822) 989-88	20	2012-09-05 08:42
7	27	2012-09-18 16:00:00	2	27	Rumney	Henrietta	3 Burkington	78533	(822) 989-88	20	2012-09-05 08:42
6	27	2012-09-29 17:00:00	2	27	Rumney	Henrietta	3 Burkington	78533	(822) 989-88	20	2012-09-05 08:42
8	28	2012-09-28 13:00:00	1	28	Farrell	David	437 Granite I	43532	(855) 755-9876		2012-09-15 08:22
8	28	2012-09-26 17:00:00	1	28	Farrell	David	437 Granite I	43532	(855) 755-9876		2012-09-15 08:22

That's all we need to find David's bookings! In general, I encourage you to remember that the output of the FROM clau is essentially one big table that you then filter information out of. This may sound inefficient - but don't worry, under t covers the DB will be behaving much more intelligently :-).

One final note: there's two different syntaxes for inner joins. I've shown you the one I prefer, that I find more consister with other join types. You'll commonly see a different syntax, shown below:

```
select bks.starttime
        from
                cd.bookings bks,
                cd.members mems
        where
                mems.firstname='David'
                and mems.surname='Farrell'
                and mems.memid = bks.memid;
```

This is functionally exactly the same as the approved answer. If you feel more comfortable with this syntax, feel free to use it!

Back to listing

Home / Joins / Simplejoin

Next Exerci