**Joins and subqueries**

**1.Retrieve the start times of memebers’ bookings**

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

**SQLOUTPUT:**

****

**2. Work out the start times of bookings for tennis courts**

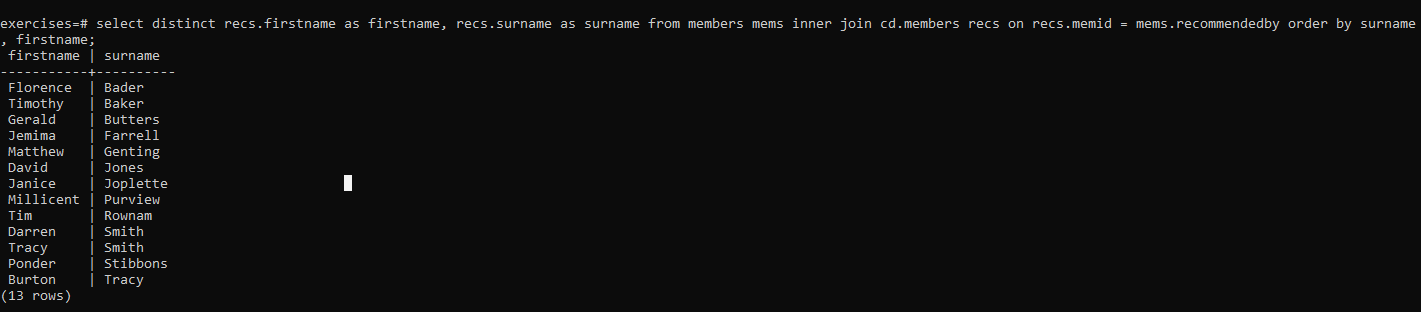
**Query:** select bks.starttime as start, facs.name as name from facilities facs inner join bookings bks on facs.facid = bks.facid where facs.name in ('Tennis Court 2','Tennis Court 1') and bks.starttime >= '2012-09-21' and bks.starttime < '2012-09-22' order by bks.starttime;

**SQLOUTPUT:**

**3. Produce a list of all members who have recommended another member**

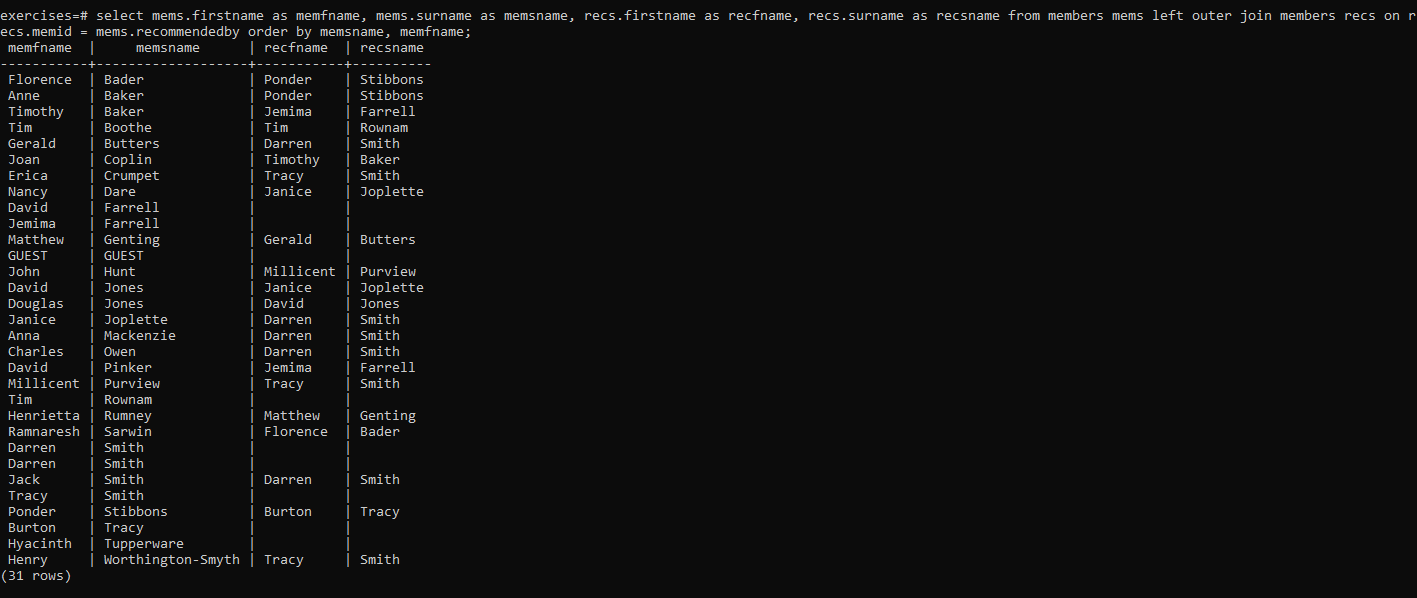
**Query:** select distinct recs.firstname as firstname, recs.surname as surname from members mems inner join cd.members recs on recs.memid = mems.recommendedby order by surname, firstname;

**SQLOUTPUT:**

****

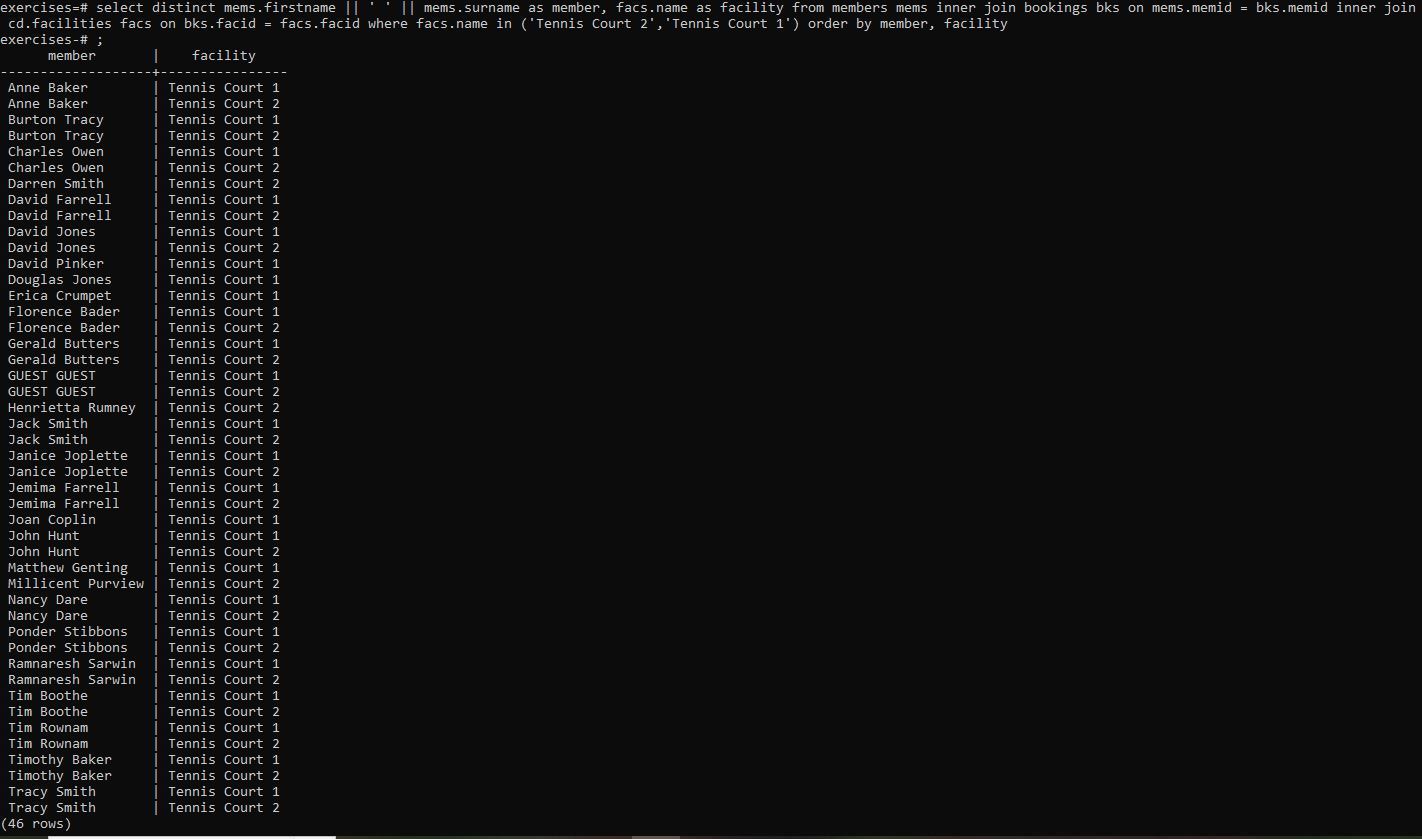
**4. Produce a list of all members, along with their recommender**

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

**SQLOUTPUT: **

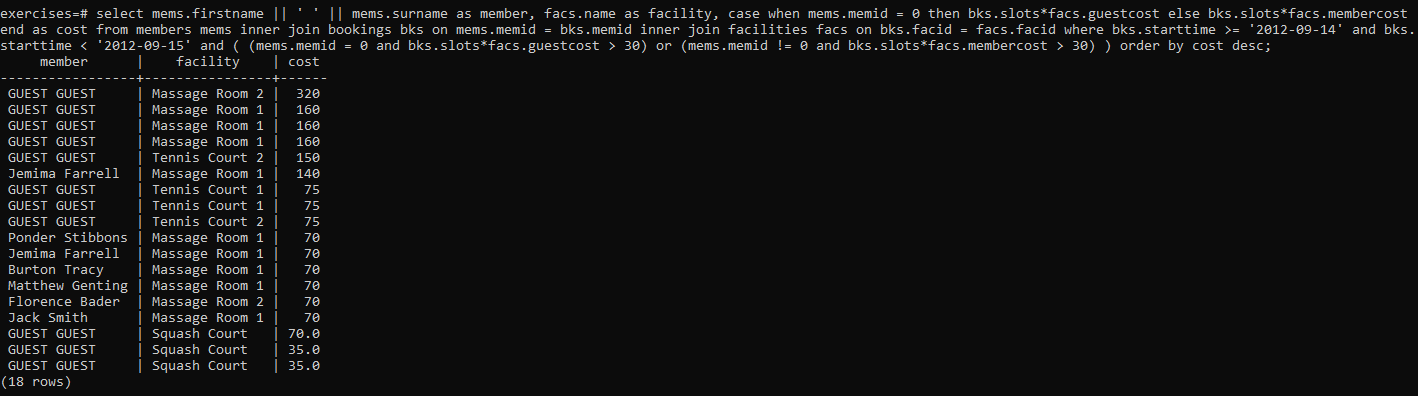
# 5. Produce a list of all members who have used a tennis court

**Query:** select distinct mems.firstname || ' ' || mems.surname as member, facs.name as facility from members mems inner join 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 **SQLOUTPUT:**

****

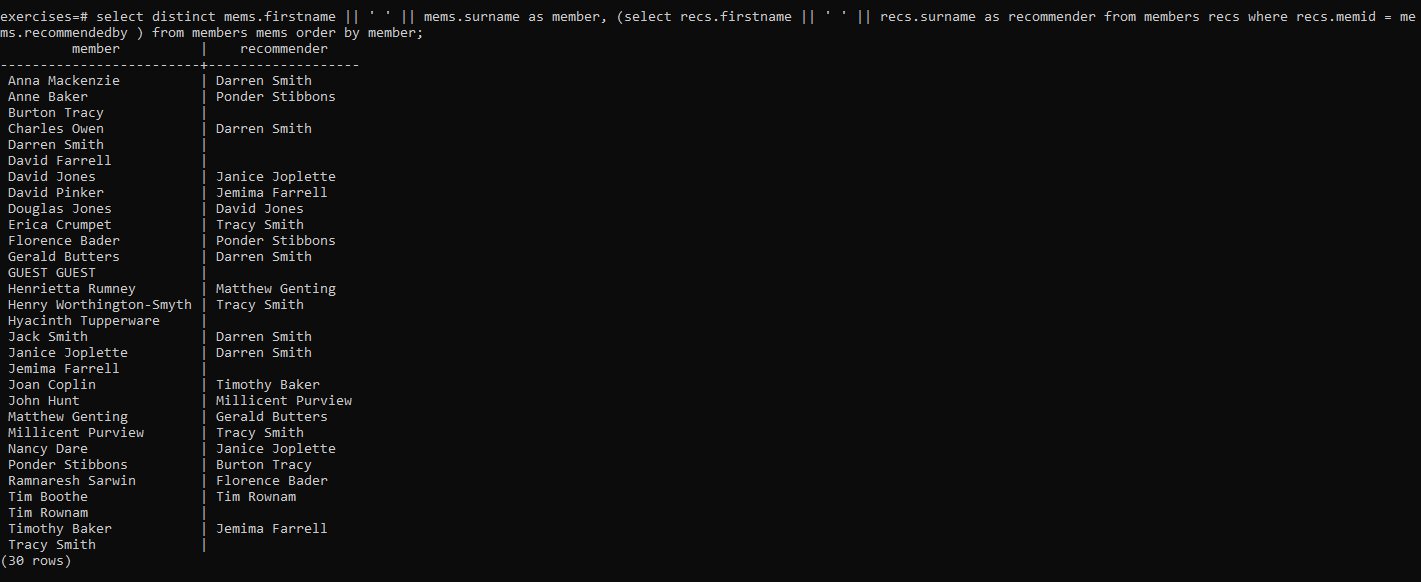
# 6. Produce a list of costly bookings

**Query:** 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 members mems inner join bookings bks on mems.memid = bks.memid inner join 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;

**SQLOUTPUT: **

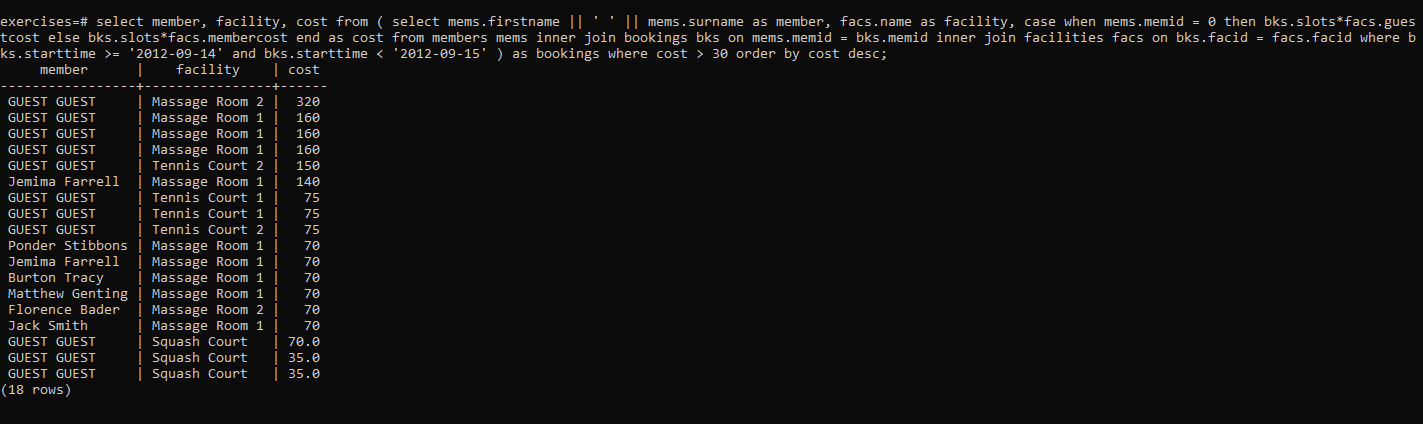
**7. Produce a list of all members, along with their recommender, using no joins.**

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

**SQLOUTPUT: **

# 8. Produce a list of costly bookings, using a subquery

**Query:** 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 members mems inner join bookings bks on mems.memid = bks.memid inner join 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;

**SQLOUTPUT: **