



Getting Started

It's pretty simple to get going with the exercises: all you have to do is [open the exercises](#), take a look at the questions, and try to answer them!

The dataset for these exercises is for a newly created country club, with a set of members, facilities such as tennis courts, and booking history for those facilities. Amongst other things, the club wants to understand how they can use their information to analyse facility usage/demand. **Please note:** this dataset is designed purely for supporting an interesting array of exercises, and the database schema is flawed in several aspects - please don't take it as an example of good design. We'll start off with a look at the Members table:

```
CREATE TABLE cd.members
(
    memid integer NOT NULL,
    surname character varying(200) NOT NULL,
    firstname character varying(200) NOT NULL,
    address character varying(300) NOT NULL,
    zipcode integer NOT NULL,
    telephone character varying(20) NOT NULL,
    recommendedby integer,
    joindate timestamp not null,
    CONSTRAINT members_pk PRIMARY KEY (memid),
    CONSTRAINT fk_members_recommendedby FOREIGN
KEY (recommendedby)
        REFERENCES cd.members(memid) ON DELETE
SET NULL
);
```

Each member has an ID (not guaranteed to be sequential), basic address information, a reference to the member that recommended them (if any), and a timestamp for when they joined. The addresses in the dataset are entirely (and unrealistically) fabricated.

```

CREATE TABLE cd.facilities
(
    facid integer NOT NULL,
    name character varying(100) NOT NULL,
    membercost numeric NOT NULL,
    guestcost numeric NOT NULL,
    initialoutlay numeric NOT NULL,
    monthlymaintenance numeric NOT NULL,
    CONSTRAINT facilities_pk PRIMARY KEY (facid
)
);

```

The facilities table lists all the bookable facilities that the country club possesses. The club stores id/name information, the cost to book both members and guests, the initial cost to build the facility, and estimated monthly upkeep costs. They hope to use this information to track how financially worthwhile each facility is.

```

CREATE TABLE cd.bookings
(
    bookid integer NOT NULL,
    facid integer NOT NULL,
    memid integer NOT NULL,
    starttime timestamp NOT NULL,
    slots integer NOT NULL,
    CONSTRAINT bookings_pk PRIMARY KEY (bookid),
    CONSTRAINT fk_bookings_facid FOREIGN KEY (facid) RE
FERENCES cd.facilities(facid),
    CONSTRAINT fk_bookings_memid FOREIGN KEY (memid) RE
FERENCES cd.members(memid)
);

```

Finally, there's a table tracking bookings of facilities. This stores the facility id, the member who made the booking, the start of the booking, and how many half hour 'slots' the booking was made for. This idiosyncratic design will make certain queries more difficult, but should provide you with some interesting challenges - as well as prepare you for the horror of working with some real-world databases :-).

Okay, that should be all the information you need. You can select a category of query to try from the menu above, or alternatively [start from the](#)

beginning.

I want to use my own Postgres system

No problem! Getting up and running isn't too hard. First, you'll need an install of PostgreSQL, which you can get from [here](#). Once you have it started, [download the SQL](#).

Finally, run `psql -U <username> -f clubdata.sql -d postgres -x -q` to create the 'exercises' database, the Postgres 'pgexercises' user, the tables, and to load the data in. Note that you may find that the sort order of your results differs from those shown on the web site: that's probably because your Postgres is set up using a different locale to that used by PGExercises (which uses the C locale)

When you're running queries, you may find psql a little clunky. If so, I recommend trying out pgAdmin or the Eclipse database development tools.

Site content licensed under [CC BY-SA 3.0](#)