SQL continued!

Questions

- How do I limit the length of a text field in SQL?
- How should I name things when I create tables?
- How do I do multiple conditions in a `where` clause?
- How do I count the number of results?
- How do I pull information from two tables?
- How do I add a column to a table?
- How do I control the order that results are printed?
- How do I back up and restore my database?
- How do I do a subquery?
- How do I use a join with subqueries?
- What is a linking table and how do I use one?

How do I limit the length of a text field in SQL?

Use `varchar` (a.k.a. `character varying`)

```
create table players (
    id serial primary key,
    name varchar (100) -- limit the length to 100 chars
);
```

```
create table players (
    id serial primary key,
    name character (100) -- fills in spaces, up to 100 chars
);
```

How should I name things when I create tables?

Table names should be:

- a noun
- plural
- if needed add underscores for readibility

Column names should be:

- a noun
- singular
- if needed add underscores for readibility

How do I do multiple conditions in a `where` clause?

```
select * from games where player1_id=3 or player2_id=3;
```

How do I compare a field to multiple values?

```
select * from games where player1_id in (2, 3) or player2_id in (2, 3);
```

```
select * from games where player1_id in (1, 2, 3) or player2_id in (3, 4, 5);
```

How do I use the and operator?

```
select * from games
    where
        (player1_id=2 and player1_score=21)
        or
        (player2_id=2 and player2_score=21);
```

How do I count the number of results?

Counting all the records

```
select count(*) from games;
```

Counting based on a condition

```
select count(*) from games
    where
        (player1_id=2 and player1_score=21)
        or
        (player2_id=2 and player2_score=21);
```

How do I pull info from two tables?

Use a join!

A join creates a "virtual table" from two "concrete" tables.

```
select
    p1.name,        -- use the aliases for the tables!
    player1_score,    -- no aliases needed for columns in main table
    p2.name,
    player2_score
from
    games  -- this is my "main table"

    inner join -- find corresponding rows
        players p1
    on games.player1_id=p1.id -- when these values match

    inner join -- find corresponding rows
        players p2
    on games.player2_id=p2.id -- when these values match
;
```

How do I add a column to a table?

```
alter table games add column completed boolean;
```

How do I control the order that results are printed?

```
select player1_score, player2_score from games
order by
    player1_score;
```

```
select player1_score, player2_score from games
order by
    player1_score desc;
```

```
select player1_score, player2_score from games
order by
    player1_score desc,
    player2_score asc
;
```

How do I backup and restore my database?

Save to a sql file!

```
pg_dump -F p pingpong-app-db > backup_2018-11-01.sql
```

Restore from the sql file!

```
createdb pingpong-app-db
psql -f backup_2018-11-01.sql pingpong-app-db
```

How do I do a subquery?

```
select * from games
    where
        player1_id=(select id from players where name ilike 'jEff')
        or
        player2_id=(select id from players where name ilike 'jEff');
```

How do I use a join with subqueries?

```
select * from
    (
        games
        inner join
        players p1
        on games.player1_id=p1.id
        inner join
        players p2
        on games.player2_id=p2.id
    )
    where
        player1_id in (select id from players where name ilike '%c%')
        or
        player2_id in (select id from players where name ilike '%c%');
```

What is a linking table and how do I use one?

```
-- a user can like many posts
-- a post can have many likes
-- a post has many users through likes
create table likes (
  like_date timestamp,
  post_id integer references posts (id),
  user_id integer references users (id)
);

insert into likes (like_date, post_id, user_id)
values (now(), 5, 32);
```

How do I join onto a linking table?

```
select count(*) from likes where post_id=5;

select users.name from -- get the user names who liked the post
    likes
    inner join
    users
    on likes.user_id = users.id
where post_id=5;

select authors.name, users.name from -- get authors and user name
    likes
    inner join
    users
    on likes.user_id = users.id
    inner join
    posts
    on posts.id = likes.post_id
    inner join
    authors
    on posts.author_id = authors.id
where post_id=5;
```