This file contains exercise assignments for us to solve in the first exercise session. Prepare by reading this document and at least think of how to solve the tasks.

## Exercise 1A: The electoral system

As you may have noticed, it's that time where everyone re-learns how the US electoral system works.

Make a table with a column for state names (or abbreviations), columns for Biden and Trump votes respectively and a column for number of electors.

Then create a view called StateResults that shows for each state: The name of the state, the name of the winning candidate (Biden or Trump) and the number of electors.

Finally make a query that shows the total number of electoral votes of both candidates (the result should have two rows for the two candidates).

You can use this data (inspired by actual events) to test your solution:

```
-- Test data
INSERT INTO States VALUES ('NV', 588252,  580605,  6);
INSERT INTO States VALUES ('AZ', 3215969, 3051555, 11);
INSERT INTO States VALUES ('GA', 2406774, 2429783, 16);
INSERT INTO States VALUES ('PA', 3051555, 3215969, 20);
-- Slightly simplified data
INSERT INTO States VALUES ('Red states',  0,  1,  232);
INSERT INTO States VALUES ('Blue states', 1, 0, 253);
```

**Hint**: Start by writing a query showing only the states Biden wins, along with a column containing the text value 'Biden'. You're more than half-way done!

**Hint**: Use the view to create the winner-query.

## Exercise 1B: The online RPG

Below is a draft for a relational schema for part of the database of an online role-playing game. It is missing primary keys and other constraints.

***Players(name, level, money)***

***Items(id, itemname, value)***

***Equippable(id, equipslot)***

***PlayerInventory(player, item)***

***Equipped(player, item, equipslot)***

**Players** contains the username, levels and in-game money of all players.

**Items** describe all the items players can find in the game, each has an ID-number, a name (several items can have the same name) and an in-game monetary value.

**Equippable** further describes the subset of items that can be worn as equipment, each such item has an equipslot attribute (e.g. 'armor', 'weapon' or such) indicating where players can equip the item.

**PlayerInventory** contains the items all players are carrying (not including equipped items).

**Equipped** contains the equipment all players are wearing, note that a player can equip at most one item in each equipslot.

**Fixing the schema**: Your first task is to add sensible reference and primary key constraints to the schema. Apart from ensuring that equipped items really exist etc. you should ensure that a player can not equip multiple items in the same equipment slot, and prevent players from equipping items in incorrect slots. Also, make sure all monetary values are non-negative.

Hint: Around five reference constraints seem right for this task.

**Implementing the schema in SQL**: Next you should make CREATE TABLE statements from your schema.

**Hint**: A foreign key can only reference keys, meaning either the complete primary key or a set of columns with a UNIQUE constraint, sometimes you have to add extra UNIQUE constraint to facilitate this.

**Insert data**: Write inserts to fill the table with this data (add missing values as needed):

1) An item called “Vorpal sword”, worth 10000 and equippable in the weapon slot.
2) An unequippable item called “Java direkt med Swing” worth 542.
3) An item called “Databases: The complete book” worth 495 and equippable in the weapon slot.
4) An item called “Oven mitts +1”, worth 999 and equippable in the hands slot.
5) A user called jonas who is level 80, wields the mighty vorpal sword, and carries Databases: The complete book and oven mitts in its inventory.
6) A user called “Leeroy Jenkins” who is level 14, wears oven mitts, and carries Databases: The complete book in its inventory.

**Test your constraints**: Write inserts for each of the operations below, they should all fail if you have implemented your constraints correctly.

1) Equipping Java direct med Swing in jonas hands-slot (not equippable).
2) Equipping Databases: The complete book in jonas hands-slot (wrong slot).
3) Equipping Databases: The complete book in jonas weapon slot (slot collision).
4) Create another user named jonas.

**Queries of various difficulty**:

1) List the complete inventory with item names for player ‘jonas’.
2) Find the average value of all items in the game.
3) Find for each item, the number of players that carry it in their inventories. The result should contain two columns, for item id's and number of players. Make sure to show 0 for items that no one has in their inventory.
4) Find for each player the name of the item in their weapon slot, or “Nothing” for players that do not have any weapon equipped. The result should have player and item names columns.
5) Find the names of all items with a value of at least 500 and equipped by at least one player with a level above 75. The result should have a single column containing item names.
6) Find for each player, the total combined value of all items in its equipment and inventory. The result should have two columns, one for player names and the other for total value. You may exclude players that have no items.
7) Find all equippable items in Leeroy Jenkins inventory, whose equipslot is currently not occupied (i.e. every items Leeroy is carrying that Leeroy can equip without removing any current equipment). The result should have two columns, for item id and for equipslot.

## Extra assignments

If we have spare time, we will solve these exam questions from 2019 (available on the course webpage):

- 2019-01-16, question 3
- 2019-06-10, question 3
- 2019-08-29, question 3

## Solutions

Will be posted on the course webpage.