

# Assignment 3

EECS341 Spring 2019

Yue Shu

Due: Thursday, Feb 21, 2019

---

## 1. Find the ID and name for all users who are not associated with a role.

For this problem, what we need to do is to subtract users who are associated with a role from all the users and project their name and id.

```
SELECT DISTINCT t.name, t.id
FROM user t
WHERE t.id NOT IN (SELECT u.id
                  FROM user u, user_role b
                  WHERE u.id = b.user_id);
```

---

## 2. Find names that are used by more than fifty (50) users. For each name, also return the count of how many times the name is used, and sort the list of names so that the most popular name is at the top of the list.

For this problem, we can apply aggregation with the `HAVING` function to filter names with repetition more than 50 times, and then order the count of names in descending order with the `ORDER BY ... DESC` function so that the most popular name is at the top.

```
SELECT user.name, COUNT(user.name)
FROM user
GROUP BY user.name
HAVING COUNT(user.name) > 50
ORDER BY COUNT(user.name) DESC;
```

---

## 3. For each user, return the user ID, user name, and count of distinct areas that the user is able to access based on the roles assigned to the user.

For this problem, what we need to do is basically to use `SELECT ... WHERE ...` function to make sure that the distinct `area_id` is corresponding to the specific user, and then present the list with aggregation.

```
SELECT u.id, u.name, COUNT(DISTINCT ra.area_id)
FROM user u, user_role ur, role_area ra
WHERE u.id = ur.user_id
      AND ur.role_id = ra.role_id
GROUP BY u.id
ORDER BY COUNT(DISTINCT ra.area_id) DESC;
```

---

## 4. Translate from relational algebra to sql

$$\pi_{\text{user.id, user.name, role.id, role.name}} \left( \sigma_{\text{user.id=user\_role.user\_id} \wedge \text{role.id=user\_role.role\_id} \wedge \text{role.name='Ward 6'}} \left( \text{user} \times \text{role} \times \text{user\_role} \right) \right)$$

For this problem, we are simply retrieving a list of users who's associated with a role named "Ward 6".

```
SELECT DISTINCT ur.user_id, u.name user_name, ur.role_id, r.name role_name
FROM user u, role r, user_role ur
WHERE u.id = ur.user_id
      AND ur.role_id = r.id
      AND r.name = 'Ward 6'
```

---

## 5. Translate from relational algebra to sql

$$\pi_{\text{user.id, user.name}} \left( \sigma_{\text{user\_role.user\_id is null}} \left( \text{user} \bowtie_{\text{user.id = user\_role.user\_id}} \text{user\_role} \right) \right)$$

For this problem, if we take a deeper look at the relational algebra equation, we can conclude that the final result we get is simply the list of distinct users without any role assigned, which is just the same as the solution of problem one. So let's do this one last time:

```
SELECT DISTINCT t.name, t.id
FROM user t
WHERE t.id NOT IN (SELECT u.id
                  FROM user u, user_role b
                  WHERE u.id = b.user_id);
```