Rell Lab assignment

1. Aim

The purpose of this lab is to learn the syntax of Rell and how to use it in conjunction with a frontend. If you are not yet familiar with Rell or if you need help to set up the developing environment, this powerpoint may be helpful.

2. Task

The task is to create a company management system on the blockchain. The database will contain information about employees, supervisors, departments and projects. The user should be able to make queries and change the content of the database from the client. For this assignment, the frontend javascript file will be given to you. The frontend file can be found at github. Your task is to create a project in Rell that can handle all the operations called in this frontend and execute the test function correctly.

3. Tips

- Employee, Department and Project should each be their own tables in the database and therefore their own Entities in the Rell backend.

- As you will see in the javascript file an initialization function will be called to create a “super” Employee. When we add the first admin, the “super” employee will be the transaction signer. The corresponding init operation in Rell can look like this:

```
operation init(supervisor_pubkey:pubkey){
    require ((employee @* {} limit 1).size() == 0);
    create employee(
    pubkey = supervisor_pubkey,
    name = "admin",
    age = 100,
    salary = 0,
    is_admin = 1

    );

}
```

- In the client, transactions are signed by an admin employee when we:

    - Register a new employee

    - Register a new department
    - Register a new project
    - Change the salary of an employee.

  Therefore it can be reasonable to have an admin attribute in the backend employee entity. For simplicity this can be an integer with value 1 if the employee is an admin and 0 if not. Each time an admin is the signer, the backend must also check that the signer actually is an admin.

- When we make queries, we use the employee pubkey to find the employee in the database. From the client, we send the pubkey as a string but in Rell we have to convert it to a byte array in order to query. It can be written in Rell like this:

```
val byte_pubkey = byte_array(employee_pubkey);
```

- The keyword “mutable” should be used for the salary attribute in the employee entity. It is the only employee attribute that can be changed.

- A lot of valuable information about entities, operations and queries can be found in the Rell documentation.