# EPIC REPORT

## Roles:

Made roles:

* Admin
* Employee
* Trainee
* Views\_only

We gave admin superuser status.

We gave employee predefined role ‘pg\_read\_all\_data’ which gives: “Read all data (tables, views, sequences), as if having SELECT rights on those objects, and USAGE rights on all schemas, even without having it explicitly. This role does not have the role attribute BYPASSRLS set. If RLS is being used, an administrator may wish to set BYPASSRLS on roles which this role is GRANTed to.” – postgresql documentation, e-documentation, referenced [21/04/2024], available at <https://www.postgresql.org/docs/current/predefined-roles.html>

Trainee has been granted ability to read tables project, customer, geo\_location, and project\_role. It also has permission to see traineeView which consists information from employee table rows id, name and email.

Views\_only is W.I.P

## Triggers

Made triggers:

* skillCheck
* assignTrigger
* contractTrigger
* groupTrigger

With each having their own procedure

* skillChecking
* assignEmployees
* contractCheck
* groupCheck

Each trigger only calls the corresponding procedure. Here’s table to show when they trigger and what is the corresponding procedure

|  |  |  |
| --- | --- | --- |
| Trigger | Procedure | when |
| skillCheck | skillCheck | before insert on skills |
| assignTrigger | assignTrigger | after insert on project |
| contractTrigger | contractTrigger | before update of contract\_type on employee |
| groupTrigger | groupCheck | After insert on employee |

Explanation what each trigger does:

skillCheck:

Checks that is there skill with same name than the new input.

assignTrigger:

finds all employees that are in same country than the customer and chooses 3 on the top found to be in the project.

contractTrigger

checks that contract start date is today. Checks that there is an end date for temporary contract and that it is exactly two years after current day. Checks that there is no end date for non temporary contracts.

groupTrigger

checks if given job title ‘HR secretary’ and insert that employee to HR group. Checks if job title contains world admin and inserts that into admin group. Else they go employee group.

## Procedures

Made procedures:

* salaryBase()
* temporaryIncrease()
* percentSalaryIncrease(percentValue numeric, maximumValue numeric)
* correctSalary()

SalaryBase sets all employees salary to the salary given by their job title.

TemporaryIncrease gives all employees with temporary contract 3 months more contract time

percentSalaryIncrease takes in any numeric values. percentValue proceeds with integers being per cents like 20 = 20% and so on. It also takes maximum value and if the original value was higher than the given value then the value wasn’t increased. percentSalaryIncrease increased current salary value by given per cent value.

correctSalary first call salaryBase procedure to give them their salary a base value and then gives them additional salary for each benefit salary marked in the skills they have.

## Functions

Made function:

* getProjects(givenDate date)

getProjects takes a date and returns in a table all projects which end date is later than given date. In the table are information about the project and the customer information.

## Constraints

Added following constraints:

* *customer*: *email* not null
* *project*: *p\_start\_date* not null
* *employee*: *salary* > 1000

## Partitions

PostgreSQL does not support partitioning an existing table so instead the tables were copied except they were partitioned by the corresponding column and then renaming the tables.

The following partitions were made this way:

* *customer* by *c\_id*
* *project* by *p\_id*

Each table has 3 partitions and a default partition.

## Views

Created the following views:

* employees\_on\_project

lists all employees on a project

* employees\_in\_department\_in\_hq

lists all employees by department and headquarter

* employees\_by\_skill

lists all employees by their skills

* customers\_by\_location

lists all customers by their location

* employees\_by\_group

lists all employees by the group(s) they belong to

* employees\_by\_title

lists all employees by the job title(s) they have

## Zip\_code

Added a *zip\_code* column to *geo\_location*. It holds an integer and has no constraints.