

Project task

You have been given a database to manage. The database contains data of customers, projects, employees, user groups and roles. You can see the ERD of the database below.

Your task is to do the following:

- (8 %) Create four views to provide for the company superiors and management.
 - The views should contain information that is important for the management of the company
 - Idea of views is to combine data from various tables to provide an easier access to the combined information.
 - The view does not have to be the final query result (i.e. View is used for easier access and then queried again for more detailed information)
 - Each view should join at least two tables (not including linking tables)
- (15 %) Create three triggers for the database:
 - One for before inserting a new skill, make sure that the same skill does not already exist
 - One for after inserting a new project, check the customer country and select three employees from that country to start working with the project (i.e. create new project roles)
 - One for before updating the employee contract type, make sure that the contract start date is also set to the current date and end date is either 2 years after the start date if contract is of Temporary type, NULL otherwise. (Temporary contract in Finnish is "määräaikainen". It's a contract that has an end date specified).
- (9 %) Create three procedures for the database:
 - Procedure that sets all employees salary to the base level based on their job title
 - Procedure that adds 3 months to all temporary contracts
 - Procedure that increases salaries by a percentage based on the given percentage. You can also specify the highest salary to be increased (give limit X and salaries that are below X are increased).
EDIT 20.04.2023: The user can specify the salary limit when calling the procedure. If user doesn't specify one (or gives 0 or null), then the limit is not considered. The percentage can be given in decimals or numbers or what ever you specify, as long as the procedure works.
- (8 %) Partition two of the following tables to at least three partitions (excluding default partition):

- Employee table
- Customer table
- Project table
- Note! You may have to create partitions based on the primary key unless you come up with another method
- (6 %) Create access rights:
 - Create three roles - admin, employee, trainee.
 - Give admin all administrative rights (same rights as postgres superuser would have)
 - Give employee rights to read all information but no rights to write
 - Give trainee rights to read ONLY project, customer, geo_location, and project_role tables as well as limited access to employee table (only allow reading employee id, name, email)
- (4 %) Do the following changes to the database:
 - Add zip_code column to Geo_location (you don't have to populate it with data)
 - Add a NOT NULL constraint to customer email and project start date
 - Add a check constraint to employee salary and make sure it is more than 1000. You may have to update the salary information to be able to add the constraint (unless you have already done so)

The project is worth 50 %

Additional tasks IF done in pairs

- A procedure that calculates the correct salary based on the acquired skills (skills may give a salary bonus and it is indicated in the database)
- A trigger after insert on employee.
 - If employee's job title is HR secretary, add them to the HR user group.
 - If employee's job title is any of the admin related, add them to the Administration group.
 - Everyone else is added to the employee group
- A function that returns all projects that were ongoing based on the given date (end date is after the given date)
 - e.g. get_running_projects(date) that returns the project table data joined with the customer data
- Two additional views
- An additional role called views_only and give them read access to all created views (and nothing else)
-