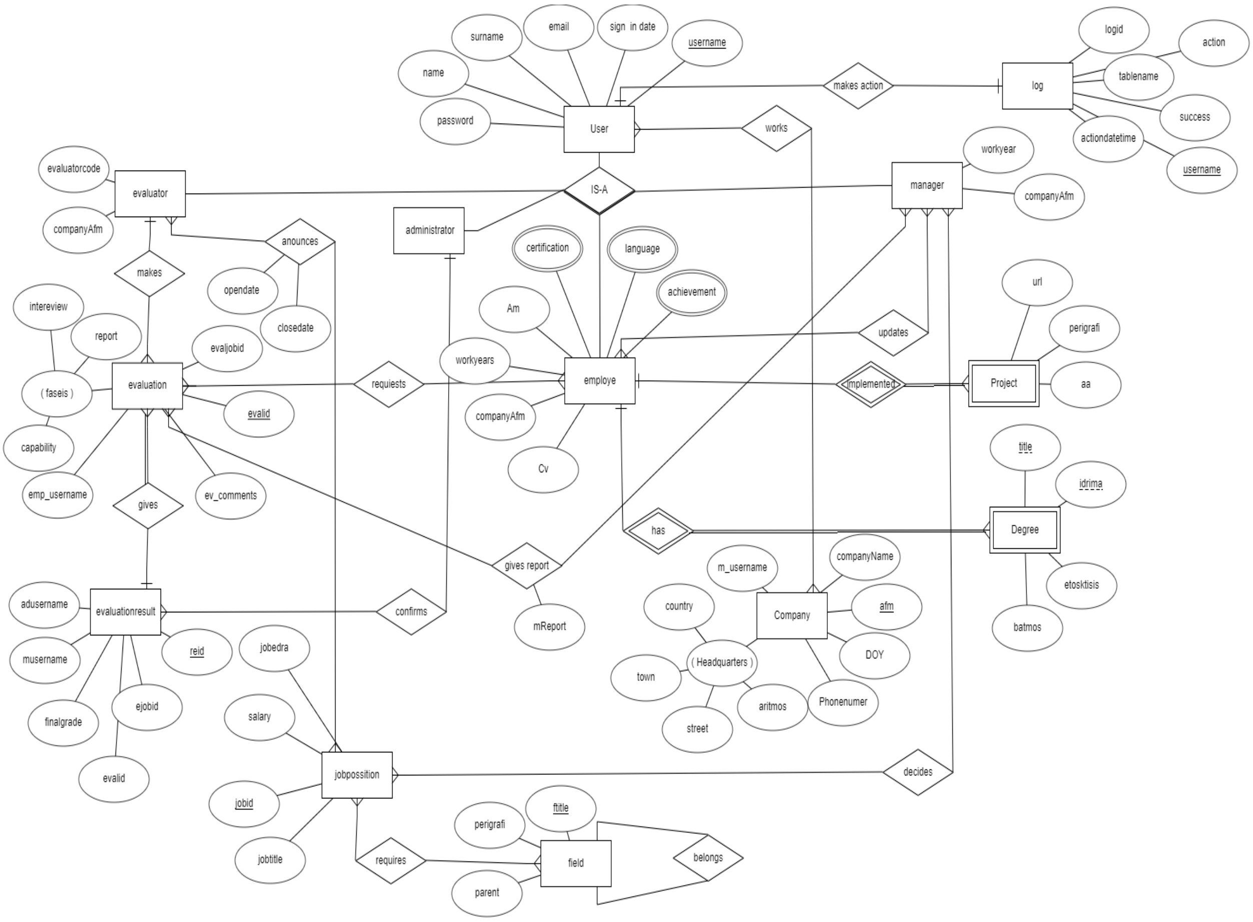
DATABASES

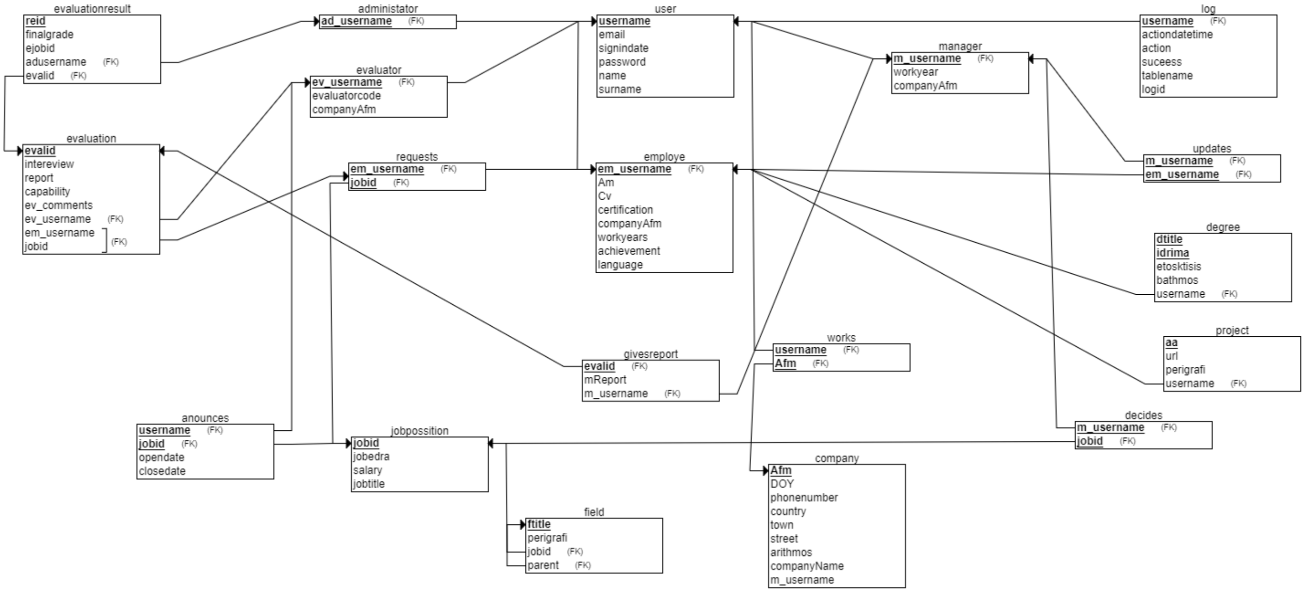
Employee Evaluation System

Contents

1.ER Diagram


surname
email
sign in date
username
name
password
User
makes action
log
logid
action
tablename
success
actiondatetime
username
works
workyear
manager
companyAfm
evaluatorcode
evaluator
companyAfm
IS-A
administrator
anounces
makes
opendate
closedate
certification
language
achievement
url
perigrafi
Am
updates
intereview
report
evaljobid
workyears
employe
Project
aa
(faseis)
evaluation
requiests
implemented
evalid
companyAfm
capability
Cv
title
idrima
emp_username
gives
ev_comments
has
Degree
gives report
m_username
companyName
etosktisis
adusername
evaluationresult
confirms
mReport
country
Company
afm
batmos
musername
reid
jobedra
(Headquarters)
DOY
town
aritmos
Phonenumer
finalgrade
ejobid
salary
street
evalid
jobpossition
decides
jobid
ftitle
perigrafi
jobtitle
requires
field
belongs
parent

2.Relational



3.SQL Implementation

3.1. Create Tables

20 tables have been created which are:

user, log, company, employe, administrator, evaluator, manager, updates, works, degree, project, evaluation, givesreport, evaluationresult, job_postition, announces, requests, field ,decides.

The tables are based on the description of the pronunciation, the ER and the relational one we have created. However, there are several things we have added that are closer to our personal understanding of the base.

The additional attributes that we have are the degree table that has the username attribute which inherits from the employe table .The Job_position table that holds the job title as the jobtitle attribute .The logid attribute in the log table that is its primary key as well as the evalid in the evaluation table and also the ev_username attribute that holds the name of the evaluator that does the evaluation. Finally, in the evaluationresult table where all the attributes were our own initiative we have the primary key reid , the finalgrade which is the total grade of the evaluation (i.e. the sum of the grades of interview,report and capability) and ad_username ,evalid which are inherited from the administrator and evaluator table respectively.

3.2. Insert

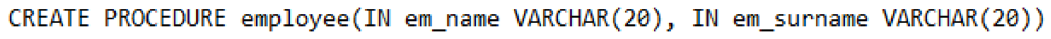
We entered values for each table attribute with INSERT INTO table_name VALUES(values);

To make the insertion process easier we relied on multiple values in the “The office” series.

3.3 Stored Procedures

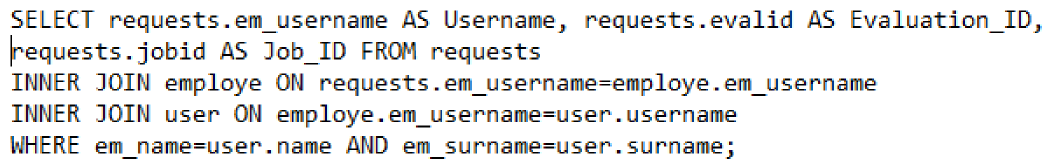
3.3.1

Procedure which accepts as input the first and last name of an employee.



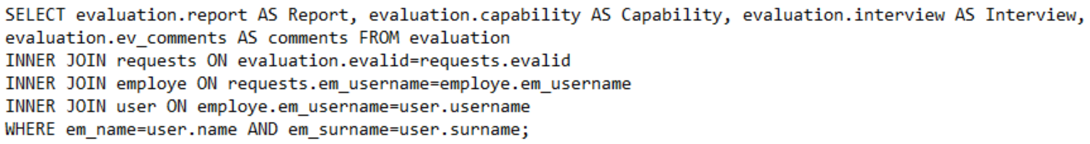
```
CREATE PROCEDURE employee(IN em_name VARCHAR(20), IN em_surname VARCHAR(20))
```

I choose with the Select command to display the username, rating number and job number and I give them an alias and with Inner Join I join them to the User table, so that the input I give finds the corresponding first and last name of the user I want.



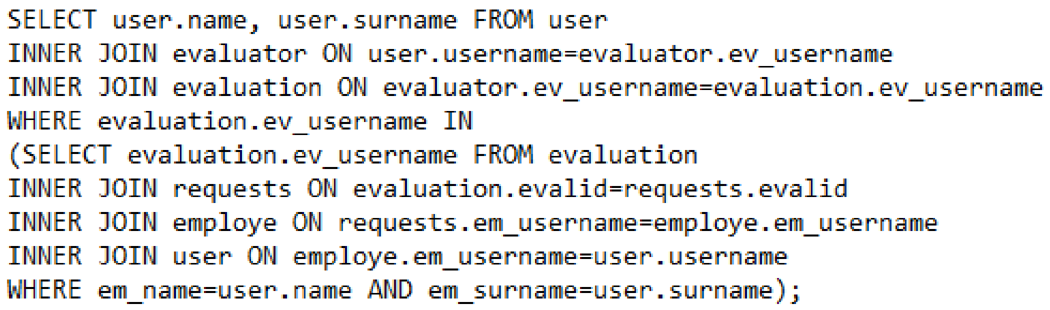
```
SELECT requests.em_username AS Username, requests.evalid AS Evaluation_ID,
requests.jobid AS Job_ID FROM requests
INNER JOIN employe ON requests.em_username=employe.em_username
INNER JOIN user ON employe.em_username=user.username
WHERE em_name=user.name AND em_surname=user.surname;
```

The manager's report, the employee's competency, the interviewed employee and the evaluator's comments are selected to be displayed. Inner Join joins them to the User table, so that the input given finds the corresponding first and last name of the user I want.



```
SELECT evaluation.report AS Report, evaluation.capability AS Capability, evaluation.interview AS Interview,
evaluation.ev_comments AS comments FROM evaluation
INNER JOIN requests ON evaluation.evalid=requests.evalid
INNER JOIN employe ON requests.em_username=employe.em_username
INNER JOIN user ON employe.em_username=user.username
WHERE em_name=user.name AND em_surname=user.surname;
```

Selected to display the first and last name of the evaluator who rated the employee. Then there are embedded Select. In the internal one, it shows me the username of the corresponding evaluator. In the external, it displays the first and last name of the evaluator with the username displayed in the internal select.



```
SELECT user.name, user.surname FROM user
INNER JOIN evaluator ON user.username=evaluator.ev_username
INNER JOIN evaluation ON evaluator.ev_username=evaluation.ev_username
WHERE evaluation.ev_username IN
(SELECT evaluation.ev_username FROM evaluation
INNER JOIN requests ON evaluation.evalid=requests.evalid
INNER JOIN employe ON requests.em_username=employe.em_username
INNER JOIN user ON employe.em_username=user.username
WHERE em_name=user.name AND em_surname=user.surname);
```

3.3.2

The stored procedure sumofevaluation is created which takes as input the id of the evaluator and the id of job_position.

Variables are declared within the body of the procedure the total, id, inter, rep and cap and assigned the values of the attributes : the sum of the values of the attributes interview , report and capability ,the evalid ,the interview , report ,capability respectively by doing inner join from the evaluation table, to the evaluator table, to the announces table and to the job_position with limit 1 .

Then via if it is checked if the scores are greater than zero or not.If they are then the total is stored in finalgrade otherwise “Not all phases completed” is printed.

3.3.3

A procedure evaluationStatus is created which takes as input a jobid.

The store procedure evaluationStatus informs us about the status of evaluations for a job.

Two variables l and k are created in which the number of applications made, and the number of results finalized are stored respectively (for the specific job given in the input).So if l=k then all applications have been graded, and the grade and name of the employee are displayed. If l>k then pending grades are shown as many as have been completedWhile if l=0 there are no applications.

In each case, a message appears informing you of the status of the evaluations.

3.3.4

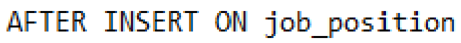
In the context of creating the GUI another procedure arose which we called mesos.

This procedure takes as input a username of an evaluator and displays the average of the scores generally given by the evaluator given by the user.

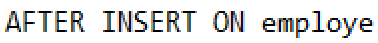
3.4 Triggers

3.4.1

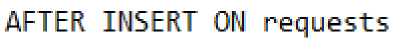
Selected after insertion in the appropriate table:



```
AFTER INSERT ON job_position
```

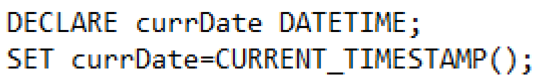


```
AFTER INSERT ON employe
```



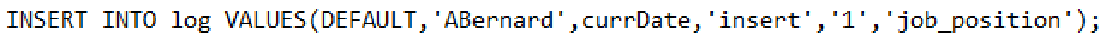
```
AFTER INSERT ON requests
```

A variable is created in which the date function is defined, so that it reports an exact date after each action.



```
DECLARE currDate DATETIME;
SET currDate=CURRENT_TIMESTAMP();
```

Finally, the values to be entered into the log table are entered to report the actions taken.



```
INSERT INTO log VALUES(DEFAULT,'ABernard',currDate,'insert','1','job_position');
```

For the other actions (update, delete) the code is similar. The only things that change are the names of the tables and the actions.

3.4.2

The companyUpdate trigger prevents the user from changing VAT, name and tax office. Whenever the user goes to update a company, the update is done except for these three traits. So, the trigger is done before update on company and using new and old. Specifically, each time it makes the NEW. equal to OLD. for each of the above.

3.4.3

The changeprofile trigger is activated when a user other than the administrator tries to make a change to the profile details. If it is checked if the username is ABernard (the username of our database administrator) if it is not then the new username,name,surname ,email and sing_in_date are set equal to the pre-existing ones.