PROJECT IN DATABASES

TEAM DETAILS:

NAME: SOULI SPIRO NAME: METSI FEZO

AM:1070263 AM:1070727

YEAR: 4 YEAR:3

**Explanation of project implementation thinking:**

During the announcement of the project we decided to initially create 3 tables for the evaluation process where they will give their results in a separate table yp\_interview where in this table you become thecalculation of the mentioned evaluations and the result is recorded in the evaluationresult table. The 3 interview tables consist of interview1 which is the first evaluation phase, interview2 which is the second evaluation phase and interview3 which is the last evaluation phase. Then we made a table named applies for the application process. We also created a table named log to record the actions performed by any user. We added the administrator table, i.e. the administrators of the system who implement the evaluation of the personnel and we added some additional details to our relation (my\_afm,AM,Eexp\_years) for the employee table based on the statement. We created with the create.sql code the database with its tables and their contacts and gave values ​​with insert.sql code. Then we implemented the triggers.sql code where either they update the log table or they don't allow the change of some specific fields based on the statement. Finally we made stored\_procedures.sql codes for 3 functions requested by

the pronunciation.

Our codes

CREATE :

create database staffevaluation; use staffevaluation;

create table company(

AFM char(9) default "unknown" not null, DOY varchar(15) default "unknown" not null, name varchar(35) default "unknown" not null, phone bigint(16) not null,

street varchar(15) default "unknown" not null, num tinyint(4) default "0" not null,

city varchar(15) default "unknown" not null, country varchar(15) default "unknown" not null, primary key (AFM)

)engine=InnoDB;

create table user(

username varchar(12) default "unknown" not null, password varchar(10) default "unknown" not null, name varchar(25) default "unknown" not null, surname varchar(25) default "unknown" not null, reg\_date datetime not null,

email varchar(30) default "unknown" not null, primary key (username)

)engine=InnoDB;

create table administrator(

adminUsername varchar(12) default "unknown" not null, adminPassword varchar(10) default "unknown" not null, adminName varchar(25) default "unknown" not null, adminSurname varchar(25) default "unknown" not null, primary key (adminUsername)

)engine=InnoDB;

create table manager(

managerUsername varchar(12) default "unknown" not null, exp\_years tinyint(4) default "0" not null,

firm char(9) default "unknown" not null, primary key (managerUsername),

constraint mngu

foreign key (managerUsername) references user(username) on update cascade on delete cascade

)engine=InnoDB;

create table employee(

username varchar(12) default "unknown" not null, bio text ,

sistatikes varchar(35) default "no sistatikes found" not null,

certificates varchar(35) default "no certificates found" not null,

awards varchar(35) default "no rewards found" not null, my\_afm char(9) default "unknown" not null,

AM int(4) not null AUTO\_INCREMENT, Eexp\_years tinyint(4) default "0" not null, primary key (AM,username),

constraint usremp

foreign key (username) references user(username) on update cascade on delete cascade,

constraint afemp

foreign key (my\_afm) references company(AFM) on update cascade on delete cascade

)engine=InnoDB;

create table evaluator(

username varchar(12) default "unknown" not null, exp\_years tinyint(4) default "0" not null,

firm char(9) default "unknown" not null, primary key (username),

constraint usrev

foreign key (username) references user(username) on update cascade on delete cascade,

constraint firmev

foreign key (firm) references company(AFM) on update cascade on delete cascade

)engine=InnoDB;

create table job(

id int(4) default "0" not null, start\_date date,

salary float(6,1) default "0.00" not null,

position varchar(40) default "unknown" not null, edra varchar(45) default "unknown" not null, evaluator varchar(12) default "unknown" not null, announce\_date datetime,

submission\_date date,

primary key (id), constraint evjob

foreign key (evaluator) references evaluator(username) on update cascade on delete cascade

)engine=InnoDB;

create table antikeim(

title varchar(36) default "unknown" not null, descr tinytext,

belongs\_to varchar(36) default "unknown" not null, primary key (title),

constraint belant

foreign key (belongs\_to) references antikeim(title) on update cascade on delete cascade

)engine=InnoDB;

create table needs(

job\_id int(4) default "0" not null,

antikeim\_title varchar(36) default "unknown" not null, primary key (job\_id,antikeim\_title),

constraint jbne

foreign key (job\_id) references job(id) on update cascade on delete cascade, constraint antikm

foreign key (antikeim\_title) references antikeim(title) on update cascade on delete cascade

)engine=InnoDB;

create table project(

empl varchar(12) default "unknown" not null, num tinyint(4) not null AUTO\_INCREMENT, descr text ,

url varchar(60) default "unknown" not null, primary key (num,empl),

constraint empro

foreign key (empl) references employee(username) on delete cascade on update cascade

)engine=InnoDB;

create table languages(

employee varchar(12) default "unknown" not null, lang set ("EN","FR","SP","GR"),

primary key (employee,lang), constraint emplan

foreign key (employee) references employee(username) on update cascade on delete cascade

)engine=InnoDB;

create table degree(

titlos varchar(50) default "unknown" not null, idryma varchar(40) default "unknown" not null,

bathmida enum("LYKEIO","UNIV","MASTER","PHD"),

primary key (titlos), unique (idryma)

)engine=InnoDB;

create table has\_degree(

degr\_title varchar(50) default "unknown" not null, degr\_idryma varchar(40) default "unknown" not null, empl\_usrname varchar(12) default "unknown" not null, etos year(4) not null,

grade float(3,1) default "0.00" not null,

primary key (degr\_title,degr\_idryma,empl\_usrname), constraint titdeg

foreign key (degr\_title) references degree(titlos) on update cascade on delete cascade,

constraint idrdeg

foreign key (degr\_idryma) references degree(idryma) on update cascade on delete cascade,

constraint usrdeg

foreign key (empl\_usrname) references employee(username) on update cascade on delete cascade

)engine=InnoDB;

create table requestsevaluation(

empl\_usrname varchar(12) default "unknown" not null, job\_id int(4) default "0" not null,

primary key (empl\_usrname,job\_id), constraint requsr

foreign key (empl\_usrname) references employee(username) on update cascade on delete cascade,

constraint reqjob

foreign key (job\_id) references job(id) on update cascade on delete cascade

)engine=InnoDB;

create table applies(

job\_id int(4) default "0" not null,

employee\_username varchar(12) default "unknown" not null, num\_applies tinyint(4) default "0" not null,

constraint appjob

foreign key (job\_id) references job(id) on update cascade on delete cascade, constraint appusr

foreign key (employee\_username) references employee(username) on update cascade on delete cascade

)engine=InnoDB;

create table interview1(

evaluator\_username varchar(12) default "unknown" not null, employee\_username varchar(12) default "unknown" not null, int\_grade enum("0","1","2","3","4") not null, int\_comments varchar(255) default "unknown" not null, primary key (evaluator\_username,employee\_username), constraint intvev

foreign key (evaluator\_username) references evaluator(username) on update cascade on delete cascade,

constraint intvempl

foreign key (employee\_username) references employee(username) on update cascade on delete cascade

)engine=InnoDB;

create table interview2(

manager\_name varchar(12) default "unknown" not null, employee\_username varchar(12) default "unknown" not null, evaluator\_username varchar(12) default "unknown" not null, report varchar(255) default "unknown" not null,

int2\_grade enum("0","1","2","3","4") not null, primary key (manager\_name,employee\_username,

evaluator\_username), constraint intvman

foreign key (manager\_name) references manager(managerUsername) on update cascade on delete cascade,

constraint intvem

foreign key (employee\_username) references employee(username) on update cascade on delete cascade,

constraint intveva

foreign key (evaluator\_username) references evaluator(username) on update cascade on delete cascade

)engine=InnoDB; create table interview3(

grade\_ptyxiou float(3,1) unsigned default "0.00" not null, systatikes\_ep varchar(35) default "no sistatikes found" not null, num\_projects tinyint(4) not null AUTO\_INCREMENT, employee\_username varchar(12) default "unknown" not null, awards\_employee varchar(35) default "no rewards found" not null, int3\_grade enum("0","1","2") not null,

primary key (num\_projects,employee\_username)

)engine=InnoDB;

create table yp\_interview(

aksiologhsh1 enum("0","1","2","3","4") not null,

aksiologhsh2 enum("0","1","2","3","4") not null, aksiologhsh3 enum("0","1","2") not null,

result int(4) not null ,

comments varchar(255) default "unknown" not null, primary key (aksiologhsh1,aksiologhsh2,aksiologhsh3)

)engine=InnoDB;

create table evaluationresult( Evld int(4) not null,

empl\_usrname varchar(12) default "unknown" not null, job\_id int(4) default "0" not null,

grade int(4) not null ,

commentsev varchar(255) default "unknown" not null, primary key (empl\_usrname),

constraint evaempl

foreign key (empl\_usrname) references employee(username) on update cascade on delete cascade,

constraint evajb

foreign key (job\_id) references job(id) on update cascade on delete cascade

)engine=InnoDB;

create table log(

ins\_usernameJ varchar(12), upd\_usernameJ varchar(12), del\_usernameJ varchar(12), ins\_conditionJ enum("YES","NO"), upd\_conditionJ enum("YES","NO"), del\_conditionJ enum("YES","NO"), date\_insJ datetime,

date\_updJ datetime, date\_delJ datetime,

on1 enum("JOB","EMPLOYEE","REQUESTSEVALUATION") ,

inserted\_job int(4),

updated\_job int(4), deleted\_job int(4), ins\_usernameE varchar(12), upd\_usernameE varchar(12), del\_usernameE varchar(12),

ins\_conditionE enum("YES","NO"), upd\_conditionE enum("YES","NO"), del\_conditionE enum("YES","NO"), date\_insE datetime,

date\_updE datetime , date\_delE datetime,

on2 enum("JOB","EMPLOYEE","REQUESTSEVALUATION"),

inserted\_employee varchar(12), updated\_employee varchar(12), deleted\_employee varchar(12), ins\_usernameR varchar(12), upd\_usernameR varchar(12), del\_usernameR varchar(12), ins\_conditionR enum("YES","NO"), upd\_conditionR enum("YES","NO"), del\_conditionR enum("YES","NO"), date\_insR datetime,

date\_updR datetime , date\_delR datetime ,

on3 enum("JOB","EMPLOYEE","REQUESTSEVALUATION") ,

inserted\_requestev int(4), updated\_requestev int(4), deleted\_requestev int(4)

)engine=InnoDB;

INSERT :

insert into company(AFM,DOY,name,phone,street, num,city,country) VALUES("102030300","292929292019223","KALO KREAS", "6986473820","THARRALEOU PASSA",8,"PATRA","GRECCE"), ("198762563","726353637837536","PIPIS AND SONS AE"," 2611124178","EIRINIS KAI FILIAS",9,"PATRA","GREECE"), ("384758490","635467635647891","SPIRT AE", "00306938748591","POSNJE",2,"BERAT","ALBANIA"), ("837638198","748391029273847","LONDONINO AE",

"0690982828291","LONDONS SQUARE",12,"LONDON","AMERICA"), ("039282910","627281910182919","KARELIAS AND SONS AE", "2610819201","CHARILAOU TRIKOYPI",6,"ATHENS","GREECE");

INSERT INTO user(username,password,name,surname,reg\_date,email) VALUES("AGG","HFFD323G41","NTANOS","AGGELOPOULOS","1999-03-23 15:00:00","[NTANOSAGGEL@GMAIL.COM](mailto:NTANOSAGGEL@GMAIL.COM)"), ("ANTE","KJJA9V4337","GIANNIS","ANTETOKOUMPO","1999-06-14 12:32:12","[GIANNISANTE@HOTMAIL.COM](mailto:GIANNISANTE@HOTMAIL.COM)"), ("AGR","T553GDB910","TOLIS","AGREPIS","1999-07-03 12:34:23","[TOLISAGREP@GMAIL.COM](mailto:TOLISAGREP@GMAIL.COM)"), ("PER","KSVR124GKA","TAKHS","PERISTERHS","1998-01-17 12:00:00","[TAKHSPER@YAHOO.GR](mailto:TAKHSPER@YAHOO.GR)"), ("RIO","VNNNX9201K","ALEX","RIOTHS","1998-06-19 13:42:23","[ALEXRIOTHS@GMAIL.COM](mailto:ALEXRIOTHS@GMAIL.COM)"), ("MET","0123456789","GIORGOS","METSI","2015-01-29 11:41:12","[GIORGOSMETSI@GMAIL.COM](mailto:GIORGOSMETSI@GMAIL.COM)"), ("SIA","987654321","KONSTANTINOS","SIAMAS","2016-08-02 16:43:12","[KONSTANTINOSSIAMAS@GMAIL.COM](mailto:KONSTANTINOSSIAMAS@GMAIL.COM)"), ("MPAK","ik738hid81","GIORGOS","MPAKOPOULOS","2019-06-13 12:42:33","[GIORGOSMPAKOPOULOS@GMAIL.COM](mailto:GIORGOSMPAKOPOULOS@GMAIL.COM)"), ("SOULI","6DG4675HD7","BOBOS","SOULI","2012-03-01 23:32:43","[BOBOSSOULI@GMAIL.COM](mailto:BOBOSSOULI@GMAIL.COM)"), ("KAOU","9SHD782HD6","RETOS","KAOURHS","2011-11-02 12:42:12","[RETOSKAOURHS@HOTMAIL.COM](mailto:RETOSKAOURHS@HOTMAIL.COM)"), ("STAMO","83HS7382H2","MPAKHS","STAMOPOULOS","2014-04-24 23:43:43","[MPAKHSSTAMOPOULOS@YAHOO.COM](mailto:MPAKHSSTAMOPOULOS@YAHOO.COM)"), ("MAKO","MKGF123381","MAKHS","MAKOPOULOS","2011-05-24 12:12:00","[MAKHSMAK@HOTMAIL.COM](mailto:MAKHSMAK@HOTMAIL.COM)"), ("SPA","TGDV268S29","LEWNIDAS","SPARTAS","2017-02-12 15:00:00","[LEWNSPART@GMAIL.COM](mailto:LEWNSPART@GMAIL.COM)"), ("MHTSO","GBB1G94N81","KONSTANTINOS","MHTSOTAKHS","2018-10-19 13:42:23","[KONMHTSOTAKHS@YAHOO.GR](mailto:KONMHTSOTAKHS@YAHOO.GR)"), ("FOFH","45GDNSS823","FOFH","GENHMATA","2019-04-11 12:54:42","[FOFHGENH@GMAIL.COM](mailto:FOFHGENH@GMAIL.COM)"), ("PELET","GFFFK14574","KWSTAS","PELETIDHS","2007-11-08 22:43:12","[KWSTASPELE@HOTMAIL.COM](mailto:KWSTASPELE@HOTMAIL.COM)"), ("FLORO","G4HGFC327","SPYROS","FLOROPOULOS","2017-01-19 12:43:31","[SPYFLORO@YAHOO.GR](mailto:SPYFLORO@YAHOO.GR)"), ("SIOYT","KIRJ2712J3","VAGGELHS","SIOUTHS","2000-06-12 12:32:12","[VAGGSIOYT@HOTMAIL.COM](mailto:VAGGSIOYT@HOTMAIL.COM)"), ("GRECO","BESTTRAPPA","IVAN","GRECO","2018-05-05 12:34:32","RIOULTRAS@GMAIL,COM"), ("HUSTLA","SNIKSNIK87","SNIK","HUSTLA","2015-02-03 23:34:43","[SNIKTHEBOSS@GMAIL.COM](mailto:SNIKTHEBOSS@GMAIL.COM)"), ("YPO","ZNZNZN","YPO","XTHONIOS","2007-12-03 12:32:21","[YPO@GMAIL.COM](mailto:YPO@GMAIL.COM)"), ("TAKI","ZN1","TAKI","TAKOPOULO","2007-11-02 13:42:43","[TAKITSAN@YAHOO.GR](mailto:TAKITSAN@YAHOO.GR)"), ("ILEOO","ILEOO1","GIORGO","ASLANIDH","2017-03-03 12:42:44","[ILEOOMINATI@HOTMAIL.COM](mailto:ILEOOMINATI@HOTMAIL.COM)");

INSERT INTO administrator(adminUsername,adminPassword,adminName,adminSurname) VALUES("AGG","HFFD323G41","NTANOS","AGGELOPOULOS"), ("ANTE","KJJA9V4337","GIANNIS","ANTETOKOUMPO"), ("AGR","T553GDB910","TOLIS","AGREPIS"),

("PER","KSVR124GKA","TAKHS","PERISTERHS"),

("RIO","VNNNX9201K","ALEX","RIOTHS");

INSERT INTO manager(managerUsername,exp\_years,firm) VALUES("MET",5,"102030300"),

("SIA",4,"198762563"),

("MPAK",7,"384758490"),

("SOULI",8,"837638198"),

("KAOU",9,"837638198"),

("STAMO",6,"039282910");

INSERT INTO employee(username,bio,sistatikes,certificates,awards,my\_afm,AM,Eexp\_years) VALUES("GRECO","APO MIKROS STHN ALANA EMATHE THN DOYLEIA APO MIKROS KAI EGINE EPITYXHMENOS",

"MAD HOUSE","PTYXIOSE TRAP EPISTHMES PANEPISTHMIO RIOY","MAD AWARDS 2020","102030300",0001,2),

("HUSTLA","H TRAP SKHNH EINAI DIKIA MOU","SNIK","YPOXTHONIOS","BACHELORS FILOSOFIKOU STO LIVERPOOL","198762563",0002,1),

("YPO","O THEIOS OLWN","MAD HOUSE","PTYXIO MHXANOLOGON MHXANHKON METSOVIO POLYTEXNEIO","NO AWARDS","384758490",0003,4),

("TAKI","DEN EXW FILOUS EXW MONO GNWSTOUS","ZN","NO certificates","YOUTUBE BEST TRAPPER AWARD 2020","837638198",0004,5), ("ILEOO",DEFAULT,DEFAULT,DEFAULT,DEFAULT,"039282910",DEFAULT,DEFAULT);

INSERT INTO evaluator(username,exp\_years,firm) VALUES("MAKO",9,"102030300"),

("SPA",3,"102030300"),

("MHTSO",2,"198762563"),

("FOFH",1,"384758490"),

("PELET",13,"384758490"),

("FLORO",3,"837638198"),

("SIOYT",20,"039282910");

INSERT INTO job(id,start\_date,salary,position,edra, evaluator,announce\_date,submission\_date)

VALUES(1,"2019-06-12",1000.0,"YPEY8YNOS POLHSEON",

"PAKISTAN","MAKO","2019-06-18 10:34:09",

"2019-04-01"),

(2,"2011-02-19",599.0,"YPEY8YNOS POLHSEON","ALBANIA", "MHTSO","2010-02-11 12:34:06",

"2011-01-01"),

(3,"2017-08-09",801.0,"YPEY8YNOS PROSOPIKOU","ENGLAND", "FOFH","2016-08-09 12:54:32","2017-09-09"),

(4,"2020-12-01",900.0,"YPEY8YNOS AGORON","GREECE",

"FLORO","2020-01-02 12:23:53","2020-11-01"),

(5,"2018-06-01",700.0,"YPEY8YNOS PROSOPIKOU","GREECE", "SIOYT","2018-01-01 16:54:54","2018-05-01");

INSERT INTO antikeim(title,descr,belongs\_to) VALUES("TOURISTIKA","OTI NANAI","TOURISTIKA"), ("POL","MANAGING OIKONOMIKA ETAIRIAS KAI POLHSEON","POL"),

("PROSOPIKO","MANAGING TOY PROSOPIKOY THS ETAIRIAS", "PROSOPIKO"),

("AGORES","MANAGING AGORON ETAIRIAS","AGORES"), ("PROSOPIKOY","MANAGING TOY PROSOPIKOY THS ETAIRIAS", "PROSOPIKOY");

INSERT INTO needs(job\_id,antikeim\_title) VALUES(1,"TOURISTIKA"),

(2,"POL"),

(3,"PROSOPIKO"),

(4,"AGORES"),

(5,"PROSOPIKOY");

INSERT INTO project(empl,num,descr,url) VALUES("GRECO",3,"MOYSIKH STO YOUTUBE", "[WWW.YOUTUBE.COM](http://WWW.YOUTUBE.COM/)"),

("HUSTLA",1,"MOUSIKH MAD","[WWW.MAD.COM](http://WWW.MAD.COM/)"),

("YPO",2,"LYRIKH SKHNH","[WWW.LYRIKH.COM](http://WWW.LYRIKH.COM/)"),

("TAKI",4,"RAP SKHNH","[WWW.RAP.COM](http://WWW.RAP.COM/)"), ("ILEOO",DEFAULT,DEFAULT,DEFAULT);

INSERT INTO languages(employee,lang) VALUES("GRECO","SP"),

("HUSTLA","GR"),

("YPO","FR"),

("TAKI","EN"),

("ILEOO","FR");

INSERT INTO degree(titlos,idryma,bathmida) VALUES("TRAP","RIOY","UNIV"),

("EKPA","RIO","UNIV"),

("MHXANHKON","AGIOY","MASTER"),

("APOLYTHRIO","TOU","LYKEIO"), (DEFAULT,"MOUN",DEFAULT);

INSERT INTO has\_degree(degr\_title,degr\_idryma, empl\_usrname,etos,grade) VALUES("TRAP","RIOY","GRECO", "2012",5.5),

("EKPA","RIO","HUSTLA", "2007",9.2),

("MHXANHKON","AGIOY",

"YPO","2002",6.7),

("APOLYTHRIO","TOU","TAKI", "2003",16.2),

(DEFAULT,"MOUN","ILEOO",DEFAULT,DEFAULT);

INSERT INTO requestsevaluation(empl\_usrname,job\_id) VALUES("GRECO",1),

("HUSTLA",2),

("YPO",3),

("TAKI",4),

("ILEOO",5);

INSERT INTO applies(job\_id,employee\_username,num\_applies) VALUES(1,"GRECO",1),

(2,"HUSTLA",1),

(3,"YPO",2),

(4,"TAKI",1),

(5,"ILEOO",1);

INSERT INTO interview1(evaluator\_username,employee\_username, int\_grade,int\_comments)

VALUES("MAKO","GRECO",4,"POLY

SYNERGATIKOS ME EMPEIRIA KAI FILIKOS ETOIMOS GIA THN 8ESH"), ("MHTSO","HUSTLA",1,

"OXI TOSO SYNERGATIKOS ALLA ARKETA EMPEIROS EXEI PROOPTIKES"), ("MHTSO","YPO",2,"SWSTO

YFOS KAI ARKETA SYNERGATIKOS"), ("FOFH","TAKI",3,"POLY SYNERGATIKOS

ME OREKSH GIA DOULEIA OXI TOSO EMPEIROS"), ("SIOYT","ILEOO",0,"KATHOLOY EMPEIROS XWRIS GNWSEIS APARAITHTES GIA THN DOULEIA");

INSERT INTO interview2(manager\_name,employee\_username, evaluator\_username,report,int2\_grade) VALUES("MET","GRECO","MAKO",

"EKSYPNOS KAI EMPEIROS",4), ("SIA","HUSTLA","MHTSO","METRIES GNWSEIS TOY ANTIKEIMENOY ALLA ARKETH DIA8ESH GIA DOULEIA",2), ("MPAK","YPO","MHTSO","ARKETH EMPEIRIA KAI ARKETOS

XWROS GIA VELTIOSH",2),

("KAOU","TAKI","FOFH","POLY KALOS SE 8EMATA EMPEIRIAS KAI GNOSEON",3), ("STAMO","ILEOO","SIOYT","

ANTIPA8HTIKOS KAI XWRIS GNOSEIS",0);

INSERT INTO interview3(grade\_ptyxiou,systatikes\_ep, num\_projects,employee\_username,awards\_employee, int3\_grade)

VALUES(5.5,"MAD HOUSE",3,"GRECO","MAD AWARDS 2020",

2), (9.2,"SNIK",1,"HUSTLA",

"BACHELORS FILOSOFIKOU STO LIVERPOOL",1), (6.7,"MAD HOUSE",2,"YPO","NO AWARDS",1),

(16.2,"ZN",4,"TAKI","YOUTUBE BEST TRAPPER AWARD 2020", 3), (DEFAULT,DEFAULT,DEFAULT,"ILEOO",DEFAULT,DEFAULT);

INSERT INTO yp\_interview(aksiologhsh1,aksiologhsh2, aksiologhsh3,result,comments)

VALUES(4,4,2,10,"POLY SYNERGATIKOS ME EMPEIRIA KAI FILIKOS ETOIMOS GIA THN 8ESH"),

(1,2,1,4,"OXI TOSO SYNERGATIKOS ALLA ARKETA EMPEIROS EXEI PROOPTIKES"), (2,2,1,5,"SWSTOYFOS KAI ARKETA SYNERGATIKOS"),

(3,2,1,6,"POLY SYNERGATIKOS ME OREKSH GIA DOULEIA OXI TOSO EMPEIROS"),

(0,0,0,0,"KATHOLOY EMPEIROS XWRIS GNWSEIS APARAITHTES GIA THN DOULEIA");

INSERT INTO evaluationresult(Evld,empl\_usrname,job\_id, grade,commentsev)

VALUES(1,"GRECO",1,10,"POLY SYNERGATIKOS ME EMPEIRIA KAI FILIKOS ETOIMOS GIA THN 8ESH"),

(2,"HUSTLA",2,4,"OXI TOSO SYNERGATIKOS ALLA ARKETA EMPEIROS EXEI PROOPTIKES"),

(3,"YPO",3,5,"SWSTOYFOS KAI ARKETA SYNERGATIKOS"), (4,"TAKI",4,6,"POLY SYNERGATIKOS ME OREKSH GIA DOULEIA OXI TOSO EMPEIROS"),

(5,"ILEOO",5,0,"KATHOLOY EMPEIROS XWRIS GNWSEIS APARAITHTES GIA THN DOULEIA");

STORED PROCEDURES :

delimiter $

drop procedure if exists proc1$

create procedure proc1(usernameEmpl varchar(12)) begin

declare nameEmpl varchar(12); declare surnameEmpl varchar(25); declare appliesEmpl tinyint(4); declare grade1Empl int(4);

declare grade2Empl int(4); declare grade3Empl int(4); declare gradeEmpl int(4); declare evaluator varchar(12);

select username

into usernameEmpl from user

where nameEmpl=name AND surnameEmpl=surname; select num\_applies

into appliesEmpl from applies

where employee\_username=usernameEmpl; select grade

into gradeEmpl

from evaluationresult

where empl\_usrname=usernameEmpl; select evaluator\_username

into evaluator from interview1

where employee\_username=usernameEmpl; select int\_grade

into grade1Empl from interview1

where employee\_username=usernameEmpl; select int2\_grade

into grade2Empl from interview2

where employee\_username=usernameEmpl; select int3\_grade

into grade3Empl from interview3

where employee\_username=usernameEmpl; if (gradeEmpl=0) then

select "Aksiologisi se Ekseliksi";

select gradeEmpl,grade1Empl,grade2Empl,grade3Empl;

else

select num\_applies from applies

where employee\_username=usernameEmpl; select name

from user

where username=evaluator; select surname

from user

where username=evaluator;

select gradeEmpl,grade1Empl,grade2Empl,grade3Empl; end if;

end $

drop procedure if exists proc2$

create procedure proc2(in myid int(4), in evaluatorId varchar(12)) begin

declare aksOlikh int(4); declare aks1 int(4); declare aks2 int(4); declare aks3 int(4);

declare employeeId varchar(12); select int\_grade

into aks1

from interview1

where evaluator\_username=evaluatorId; select int2\_grade

into aks2

from interview2

where evaluator\_username=evaluatorId; select employee\_username

into employeeId from interview1

where evaluator\_username=evaluatorId; select int3\_grade

into aks3

from interview3

where employee\_username=employeeId; if(aks1!=0 and aks2!=0 and aks3!=0) then

select result

into aksOlikh from yp\_interview

where aksiologhsh1=aks1 and aksiologhsh2=aks2 and aksiologhsh3=aks3 ;

select "H aksiologhsh oristikopoih8hke"; update evaluationresult

set grade=aksOlikh where job\_id=myid; select aksOlikh;

else

select "Den yparxei va8mos se oles tis aksiologhseis"; end if;

end $

drop procedure if exists proc3$

create procedure proc3(in thesi int(4)) begin

declare ekremoun int(4); declare flag int(4); declare mygrade int(4);

declare ypopshf varchar(12); declare statement int(4); DECLARE finishedFlag INT;

DECLARE my\_empl\_usrname varchar(12); DECLARE my\_result int(4);

DECLARE maincur cursor FOR

select empl\_usrname,grade from evaluationresult where job\_id=thesi;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finishedFlag=1;

OPEN maincur; SET finishedFlag=0; REPEAT

FETCH maincur INTO my\_empl\_usrname,my\_result; IF (finishedFlag=0) THEN

SELECT my\_empl\_usrname AS 'EMPLOYEE\_USERNAME',my\_result AS 'FINAL\_RESULT';

END IF;

UNTIL (finishedFlag=1) END REPEAT;

CLOSE maincur;

set @x=0; select num\_applies

into flag from applies

where job\_id=thesi; while flag>0 do

select grade

into mygrade

from evaluationresult where job\_id=thesi;

if mygrade!=null then

set @x=@x+1; else

set ekremoun=ekremoun+1;

end if;

set flag=flag-1;

end while;

select num\_applies

into flag from applies

where job\_id=thesi; if @x=flag then

select "Oristikopoihmenoi pinakes"; select empl\_usrname

from evaluationresult where job\_id=thesi order by grade desc;

select grade

from evaluationresult where job\_id=thesi order by grade desc;

end if;

if ekremoun!=0 then select grade

from evaluationresult where job\_id=thesi order by grade desc;

select "Aksiologisi se ekseliksi ";

select "Ekremoun ";

select ekremoun;

end if;

if flag=0 then

select "Den vre8hkan ypopshfioi gia thn thesi";

end if;

end $

TRIGGERS :

delimiter $

create trigger logs1

after insert on job for each row begin

declare insJ int(4);

declare ins\_usrJ varchar(12); declare currdate1 datetime; select id into insJ from job;

select username into ins\_usrJ from user; insert into log(inserted\_job,ins\_usernameJ,

ins\_conditionJ,on1)

values (insJ,ins\_usrJ,"YES", "JOB"); insert into log(date\_insJ)

values (NOW()); end$

create trigger logs2

after update on job for each row

begin

declare updJ int(4);

declare upd\_usrJ varchar(12);

declare upd\_condJ enum("YES","NO");

declare o1 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select id into updJ from job;

select username into upd\_usrJ from user; if (old.id!=updJ) then

set upd\_condJ="YES";

else

set upd\_condJ="NO"; end if;

insert into log(updated\_job,upd\_usernameJ

,on1)

values (updJ,upd\_usrJ,"JOB"); insert into log(date\_updJ) values (NOW());

end$

create trigger logs3

after delete on job for each row begin

declare delJ int(4);

declare del\_usrJ varchar(12);

declare del\_condJ enum("YES","NO");

declare o1 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select id into delJ from job;

select username into del\_usrJ from user;

if (old.id=delJ) then

set del\_condJ="YES"; else

set del\_condJ="NO"; end if;

insert into log(deleted\_job,del\_usernameJ,on1) values (delJ,del\_usrJ,"JOB");

end$

create trigger logs4

after insert on employee for each row

begin

declare insE varchar(12); declare currdate4 datetime;

declare ins\_condE enum("YES","NO");

declare o2 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select username into insE from employee; set currdate4=GETDATE();

insert into log(inserted\_employee,ins\_usernameE, ins\_conditionE,date\_insE,on2)

values (insE,insE,"YES",currdate4,"EMPLOYEE"); end$

create trigger logs5

after update on employee for each row

begin

declare updE varchar(12); declare currdate5 datetime;

declare upd\_condE enum("YES","NO");

declare o2 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select username into updE from employee; set currdate5=GETDATE();

if (old.username!=updE) then set upd\_condE="YES";

else

set upd\_condE="NO"; end if;

insert into log(updated\_employee,upd\_usernameE

,date\_updE,on2)

values (updE,updE,currdate5,"EMPLOYEE"); end$

create trigger logs6

after delete on employee for each row

begin

declare delE varchar(12);

declare currdate6 datetime;

declare del\_condE enum("YES","NO");

declare o2 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select username into delE from employee; set currdate6=GETDATE();

if (old.username=delE) then set del\_condE="YES";

else

set del\_condE="NO"; end if;

insert into log(deleted\_employee,del\_usernameE, date\_delE,on2)

values (delE,delE,currdate6,"EMPLOYEE"); end$

create trigger logs7

after insert on requestsevaluation for each row

begin

declare insR int(4);

declare ins\_userR varchar(12); declare currdate7 datetime;

declare ins\_condR enum("YES","NO");

declare o3 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select job\_id into insR from requestsevaluation;

select empl\_usrname into ins\_userR from requestsevaluation; set currdate7=GETDATE();

insert into log(inserted\_requestev,ins\_usernameR, ins\_conditionR,date\_insR,on3)

values(insR,ins\_userR,"YES",currdate7, "REQUESTSEVALUATION");

end $

create trigger logs8

after update on requestsevaluation for each row

begin

declare updR int(4);

declare upd\_userR varchar(12); declare currdate8 datetime;

declare upd\_condR enum("YES","NO");

declare o3 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select job\_id into updR from requestsevaluation;

select empl\_usrname into upd\_userR from requestsevaluation; set currdate8=GETDATE();

if (old.job\_id!=updR) then set upd\_condR="YES"; else

set upd\_condR="NO";

end if;

insert into log(updated\_requestev,upd\_usernameR, date\_updR,on3) values(updR,upd\_userR,currdate8,"REQUESTSEVALUATION"); end $

create trigger logs9

after delete on requestsevaluation for each row

begin

declare delR int(4);

declare del\_userR varchar(12); declare currdate9 datetime;

declare del\_condR enum("YES","NO");

declare o3 enum("JOB","EMPLOYEE","REQUESTSEVALUATION");

select job\_id into delR from requestsevaluation;

select empl\_usrname into del\_userR from requestsevaluation; set currdate9=GETDATE();

if (old.job\_id=delR) then set del\_condR="YES"; else

set del\_condR="NO"; end if;

insert into log(deleted\_requestev,del\_usernameR, date\_delR,on3)

values(delR,del\_userR,currdate9,"REQUESTSEVALUATION"); end $

create trigger logs10

before update on company for each row

begin

declare oldAFM char(9); declare oldDOY varchar(15); declare oldNAME varchar(35); declare newphone bigint(16); declare newstreet varchar(15); declare newnum tinyint(4); declare newcity varchar(15);

declare newcountry varchar(15); select old.AFM into oldAFM; select old.DOY into oldDOY; select old.name into oldNAME; select new.phone into newphone; select new.street into newstreet; select new.num into newnum; select new.city into newcity;

select new.country into newcountry;

insert into company(AFM,DOY,name,phone,street,num,city,

country) values(oldAFM,oldDOY,oldNAME,newphone,newstreet,

newnum,newcity,newcountry);

end $

create trigger logs11

before insert on user for each row

begin

declare user1 varchar(12); declare userif varchar(12);

select username into user1 from user;

select adminUsername into userif from administrator; if (user1!=userif) then

signal sqlstate value "45000"

set message\_text= "You have no admin privilages"; end if;

end$

TΕΛΟΣ