

ΑΝΑΦΟΡΑ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

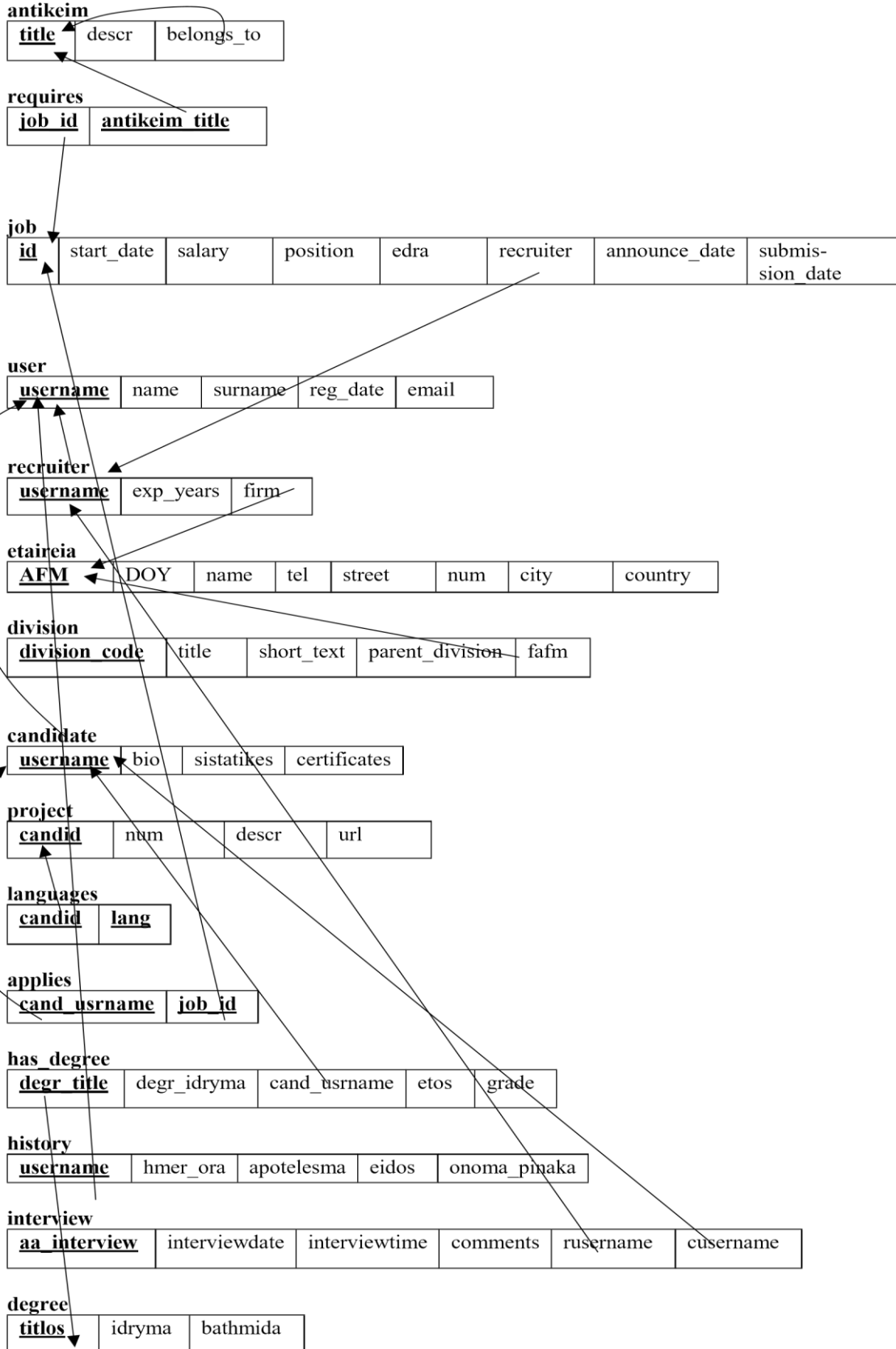
ΓΑΤΟΥ ΠΑΡΑΣΚΕΥΗ ΑΜ 1054370

Μέρος Α:ΒΔ και SQL

1.

Απάντηση

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ



2. Κατασκευή πίνακα Division

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `erecruit`
--
```

```
-- -----
--
-- Table structure for table `division`
--
```

```
DROP TABLE IF EXISTS `division`;
CREATE TABLE IF NOT EXISTS `division` (
  `division_code` int(11) NOT NULL,
  `title` text,
  `short_text` text,
  `parent_division` int(11) DEFAULT NULL,
  `fafm` char(9) DEFAULT NULL,
  PRIMARY KEY (`division_code`),
  KEY `cs3` (`parent_division`),
  KEY `cs4` (`fafm`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `division`
--
```

```
INSERT INTO `division` (`division_code`, `title`, `short_text`, `parent_division`, `fafm`) VALUES
(1, 'Software', 'Genikos Tomeas', NULL, '023453344'),
(2, 'Databases', 'Eidikeyetai stis baseis dedomenon', 1, '023451232'),
(3, 'Data Structures', 'Eidikeyetai stis domes dedomenon', 1, '123432211');
```

```
--
-- Constraints for dumped tables
--
```

```
-- Constraints for table `division`
--
```

```
ALTER TABLE `division`
  ADD CONSTRAINT `cs3` FOREIGN KEY (`parent_division`) REFERENCES `division` (`division_code`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `cs4` FOREIGN KEY (`fafm`) REFERENCES `etaireia` (`AFM`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Κατασκευή πίνακα Evaluation

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
```

```
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `erecruit`
--
```

```
-- Table structure for table `evaluation`
--
```

```
DROP TABLE IF EXISTS `evaluation`;
CREATE TABLE IF NOT EXISTS `evaluation` (
  `aa_evaluation` int(11) NOT NULL AUTO_INCREMENT,
  `personality` int(11) DEFAULT NULL,
  `education` int(11) DEFAULT NULL,
  `experience` int(11) DEFAULT NULL,
  `rusername` varchar(12) DEFAULT NULL,
  `cusername` varchar(12) DEFAULT NULL,
  `completed` int(11) NOT NULL,
  `job_id` int(11) NOT NULL,
  PRIMARY KEY (`aa_evaluation`),
  KEY `cs7` (`rusername`),
  KEY `cs8` (`cusername`),
  KEY `cs17` (`job_id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `evaluation`
--
```

```
INSERT INTO `evaluation` (`aa_evaluation`, `personality`, `education`, `experience`, `rusername`, `cusername`,
`completed`, `job_id`)
VALUES
(1, 2, 3, 4, 'bettyg', 'abrown', 1, 1),
(2, 5, 4, 2, 'bettyg', 'liagourma', 1, 1),
(3, 1, 1, 1, 'bettyg', 'cleogeo', 1, 1),
(4, 5, 0, 3, 'Giankost', 'mnikol', 1, 2),
(5, 2, 4, 2, 'msmith', 'zazahir23', 1, 2),
(6, 1, 2, 3, 'Giankost', 'mnikol', 1, 2),
(7, 0, 2, 3, 'n_tri', 'cleogeo', 1, 3),
(8, 4, 0, 0, 'n_tri', 'liagourma', 1, 3),
(9, 2, 1, 0, 'papad', 'mnikol', 0, 3),
(10, 0, 0, 4, 'papad', 'mnikol', 0, 4);
```

```
--
-- Constraints for dumped tables
--
```

```
-- Constraints for table `evaluation`
--
```

```
ALTER TABLE `evaluation`
  ADD CONSTRAINT `cs17` FOREIGN KEY (`job_id`) REFERENCES `job` (`id`) ON DELETE CASCADE ON UPDATE
  CASCADE,
  ADD CONSTRAINT `cs7` FOREIGN KEY (`rusername`) REFERENCES `recruiter` (`username`) ON UPDATE
  CASCADE,
  ADD CONSTRAINT `cs8` FOREIGN KEY (`cusername`) REFERENCES `candidate` (`username`) ON UPDATE
  CASCADE; COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Κατασκευή πίνακα History

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

--

-- Database: `erecruit`

--

-- -----

--

-- Table structure for table `history`

--

```
DROP TABLE IF EXISTS `history`;
CREATE TABLE IF NOT EXISTS `history` (
  `username` varchar(12) DEFAULT NULL,
  `hmer_ora` datetime DEFAULT NULL,
  `apotelesma` varchar(12) DEFAULT NULL,
  `eidos` varchar(12) DEFAULT NULL,
  `onoma_pinaka` varchar(12) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Κατασκευή πίνακα Interview

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

--

-- Database: `erecruit`

--

-- -----

--

-- Table structure for table `interview`

--

```

DROP TABLE IF EXISTS `interview`;
CREATE TABLE IF NOT EXISTS `interview` (
  `aa_interview` int(11) NOT NULL AUTO_INCREMENT,
  `interviewdate` date DEFAULT NULL,
  `intervietime` time DEFAULT NULL,
  `comments` text,
  `username` varchar(12) DEFAULT NULL,
  `cusername` varchar(12) DEFAULT NULL,
  PRIMARY KEY (`aa_interview`),
  KEY `cs1` (`username`),
  KEY `cs2` (`cusername`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `interview`
--

INSERT INTO `interview` (`aa_interview`, `interviewdate`, `intervietime`, `comments`, `username`, `cusername`)
VALUES (1, '2019-01-02', '00:00:12', 'Kala', 'bettyg', 'abrown'),
(2, '2019-01-01', '00:00:10', 'Polya kala', 'bettyg', 'cleogeo'),
(3, '2019-01-01', '09:00:00', 'Metria', 'bettyg', 'liagourma'),
(4, '2019-01-01', '12:00:00', 'Para poly kala', 'Giankost', 'mnikol'),
(5, '2019-01-03', '00:00:14', 'Poly Metria', 'msmith', 'zazahir23'),
(6, '2019-01-02', '00:00:15', 'Kalo apotelesma', 'Giankost', 'mnikol'),
(7, '2019-01-03', '00:00:15', 'Poly Kaka', 'n_tri', 'abrown'),
(8, '2019-01-02', '00:00:18', 'Etsi kai etsi', 'n_tri', 'cleogeo'),
(9, '2019-01-03', '00:00:19', 'Metria pros Kaka', 'papad', 'mnikol'),
(10, '2019-01-04', '00:00:22', 'Metria pros kala', 'papad', 'liagourma');

--
-- Constraints for dumped tables
--
--
-- Constraints for table `interview`
--
ALTER TABLE `interview`
  ADD CONSTRAINT `cs1` FOREIGN KEY (`username`) REFERENCES `recruiter` (`username`) ON UPDATE CASCADE,
  ADD CONSTRAINT `cs2` FOREIGN KEY (`cusername`) REFERENCES `candidate` (`username`) ON UPDATE CASCADE; COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

3.

- a. select user.name,user.surname,etaireia.name,job.id as kodikos_thesis,job.salary,count(applies.job_id) as plithos_aithseon from user,recruiter,etaireia,requires,job,candidate,applies where user.username=recruiter.username and recruiter.username=job.recruiter and recruiter.firm=etaireia.afm and job.id=requires.job_id and candidate.username=applies.cand_usrname and applies.job_id=job.id group by job.id;
- b. select candidate.username,certificates,count(degree.titlos),avg(grade) from candidate inner join has_degree on candidate.username=has_degree.cand_usrname inner join degree on has_degree.degr_idryma=degree.idryma group by candidate.username having count(degree.titlos)>1;
- c. select candidate.username,count(applies.job_id) as plithos_aithseon from candidate inner join applies on candidate.username=applies.cand_usrname inner join job on applies.job_id=job.id group by candidate.username having avg(salary)>1800;

d. select etaireia.name,job.position,antikeim.title from etaireia, recruiter, job, requires, antikeim where etaireia.afm=recruiter.firm and recruiter.username=job.recruiter and job.id=requires.job_id and requires.antikeim_title=antikeim.title and antikeim.title like '%program%';

e.
select distinct a.username,b.plithos,c.plithoss,b.mesos from recruiter aLEFT JOIN (SELECT recruiter, count(id) as plithos, avg(salary) as mesos from job group by recruiter) b on a.username=b.recruiter LEFT JOIN (SELECT rusername, count(aa_interview) as plithoss from interview group by rusername) c on a.username=c.rusername having b.plithos>2 order by a.username;

4.

a. drop procedure if exists

```
askp1;
delimiter $
create procedure askp1(IN jid int)
begin
    declare nf int;
    declare temp int; declare
    fin int; declare cu
    varchar(12); declare tot
    int; declare av float(6,1);
    declare ed int; declare ex
    int; declare pers int;
    declare result text;

    declare bcursor cursor for select completed from evaluation where job_id=jid; declare bcursor2 cursor for select
    cusername,avg(personality),education,experience,(personality+education+experience)
    as total from evaluation where job_id=jid group by cusername order by total desc;

    declare continue handler for not found set nf=1;

    set nf=0;
    set fin=0;

    open bcursor;

    repeat
    fetch bcursor into temp;

    if(nf=0) then

        if (temp=0) then
            set fin=1;
        end if;
    end if; until
    (nf=1)
    end repeat;
    close bcursor;

    if (fin=1) then
        select 'den exei oloklhrothei h axiologhsh' as "";
    else

        set nf=0;

        open bcursor2;
        repeat
        fetch bcursor2 into cu,pers,ed,ex,tot;
```

```

        set result="";
    if(nf=0) then

        if (pers=0) then
            set result=(SELECT CONCAT(result," ","failed the interview") AS
result);
            end if;

            if (ed=0) then
                set result=(SELECT CONCAT(result," ","inadequate education") AS result);
                end if;

                if (ex=0) then
                    set result=(SELECT CONCAT(result," ","no prior experience") AS
result);
                    end if;

                    select cu,pers,ed,ex,tot,result;

                    end if;
                until (nf=1)
            end repeat;
            close bcursor2;

        end if;
    end$ delimiter
;
call askp1(1);

```

b.

```

drop trigger if exists candidatetrig1;
delimiter $ create trigger candidatetrig1
before insert on
candidate for each row begin
    DECLARE recordusername VARCHAR(12);
    SET recordusername = (SELECT username FROM candidate WHERE username = NEW.username);

    IF recordusername IS NOT NULL
    THEN
        insert into history values(NEW.username,now(),'Failure','insert','candidate');
        SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'INSERT on candidate failed due to duplicate candidates
username';
    ELSE
        insert into history
values(NEW.username,now(),'Success','insert','candidate');    END IF; end$
delimiter ;

```

insert into candidate values ('zazahir23','kalos','exei polles','10 certificates','zazahir23');-->FAIL insert into
candidate values ('zazahir24','kalos','exei polles','10 certificates','zazahir24');-->OK arkei na yparxei
kataxorimenso TO USERNAME STO recruiter

```

=====

drop trigger if exists candidatetrig2;
delimiter $
create trigger candidatetrig2 before UPDATE on
candidate for each row begin
    DECLARE recordusername VARCHAR(12);
    SET recordusername = (SELECT username FROM candidate WHERE username = NEW.username);

```

```

IF      recordusername      IS      NOT      NULL      and      new.bio=old.bio      and
      new.sistatikes=old.sistatikes      and new.certificates=old.certificates and new.password=old.password  THEN
insert into history values(NEW.username,now(),'Failure','update','candidate');
  SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'UPDATE on candidate failed due to duplicate candidates
username';
ELSEIF recordusername IS NOT NULL and (new.bio<>old.bio or new.sistatikes<>old.sistatikes or
new.certificates<>old.certificates or new.password<>old.password)then
      insert into history values(NEW.username,now(),'Success','update','candidate');
  SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'UPDATE on candidate failed due to duplicate candidates
username';  END IF; end$
delimiter ;

```

```

update candidate set username='zazahir23' where
username='zazahir23'; -->Fail update candidate set username='abrown'
where username='zazahir23'; -->Fail update candidate set bio='my bio'
where username='zazahir23'; -->OK
update candidate set bio='old bio' where username='zazahir230'; -->DEN DOYLEYEI
=====

```

```

= drop trigger if exists candidatetrig3; delimiter $ create trigger candidatetrig3 before
delete on
candidate for each row
begin
      declare res int;          set res=(select count(1) FROM candidate
WHERE username = 'abrownn');

      if res=0 then
            insert into history values(NEW.username,now(),'Success','update','candidate');
      SIGNAL SQLSTATE VALUE '45000' ;
      else          insert
into history
values(NEW.username,now(),'Success','update','candidate');      end if;
end$
delimiter ;

```

```

delete from candidate where username='abrownn';-->Fail
delete from candidate where username='abrown';-->OK
=====

```

```

----- drop
trigger if exists recruitertrig1;
delimiter $ create trigger recruitertrig1
before insert on
recruiter for each row begin
IF (EXISTS(SELECT 1 FROM user WHERE username = NEW.username))
THEN insert into history values(NEW.username,now(),'Failure','insert','recruiter');
  SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'INSERT failed due to duplicate recruiters
username'; else
      insert into history
values(NEW.username,now(),'Success','insert','recruiter'); END IF; end$
delimiter ;

```

```

insert into recruiter values ('bettyg',3,'beta','12345');
=====

```

```

drop trigger if exists recruitertrig2;
delimiter $
create trigger recruitertrig2 before update on recruiter for each row
begin
if ((select count(*) from recruiter where new.username = old.username)=0)
then      insert into history values(NEW.username,now(),'Failure','update','recruiter');

```



```

    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'UPDATE failed on recruiter due to non existent
username'; else
        insert into history
values(NEW.username,now(), 'Success', 'update', 'recruiter'); END IF; end$
delimiter ;
update recruiter set exp_years=27 where username='betttyg';

update recruiter set exp_years=27 where username='betttyg';
=====
=== drop trigger if exists recruitertrig3;
delimiter $ create trigger recruitertrig3
before delete on
recruiter for each row
begin
IF (NOT EXISTS(SELECT 1 FROM recruiter WHERE username = OLD.username)) THEN
    insert into history values(OLD.username,now(), 'Failure', 'delete', 'recruiter');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DFELETE failed on recruiter due to a non existent
username';

else
    insert into history
values(OLD.username,now(), 'Success', 'delete', 'recruiter'); END IF; end$
delimiter ;

delete from recruiter where username='my19001';
=====
----- drop
trigger if exists usertrig1;
delimiter $
create trigger usertrig1 before insert on
user for each row
begin
IF (EXISTS(SELECT 1 FROM user WHERE username = NEW.username))
THEN insert into history values(NEW.username,now(), 'Failure', 'insert', 'user');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'INSERT failed due to duplicate
username'; else
    insert into history
values(NEW.username,now(), 'Success', 'insert', 'user'); END IF; end$
delimiter ;

insert into user values ('mnikol','jones','Andrew','Smith','2018-01-27 16:02:56','andrewbr@yahoo.com');
ERROR 1644 (45000): INSERT failed due to duplicate username
=====
===
drop trigger if exists usertrig2; delimiter
$ create trigger usertrig2 before update
on
user for each row
begin
IF (EXISTS(SELECT 1 FROM user WHERE username = NEW.username))
THEN insert into history values(NEW.username,now(), 'Failure', 'update', 'user');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'UPDATE failed due to duplicate username'; elseif
NEW.username <> OLD.username or NEW.username <> username or NEW.password<>OLD.password or
NEW.NAME <> OLD.NAME or NEW.SURNAME <> OLD.SURNAME OR NEW.REG_DATE <> OLD.REG_DATE OR
NEW.EMAIL <>
OLD.EMAIL then
    insert into history
values(NEW.username,now(), 'Success', 'update', 'user'); else
    insert into history values(NEW.username,now(), 'Failure', 'update', 'user');

```

```

    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'UPDATE failed due to duplicate
username'; END IF; end$
delimiter ;
update user
set
username='
mnikool'
where
username='
betttyg';

```

```

update user
set
email='mnkp
pil@hotmail.
gr' where
username='
mnikool';

```

```

update user
set
username='
papad'
where
username='
mnikool';

```

```

mysql> update user set username='mnikool' where username='mnikool';
ERROR 1644 (45000): UPDATE failed due to duplicate username
mysql> update user set email='mnkppil@hotmail.gr' where
username='mnikool'; ERROR 1644 (45000): UPDATE failed due to duplicate
username mysql> update user set username='papad' where
username='mnikool';
ERROR 1644 (45000): UPDATE failed due to duplicate username
=====

```

```

drop trigger if exists usertrig3;
delimiter $
create trigger usertrig3 before delete on
user for each row
begin
IF (NOT EXISTS(SELECT * FROM user WHERE username = OLD.username)) THEN
    insert into history values(OLD.username,now(),'Failure','delete','user');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DFELETE on user failed due to a non existent
username';
else
    insert into history values(OLD.username,now(),'Success','delete','user');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DFELETE on user Succeded due to a non existent
username'; END IF; end$ delimiter ;

```

```

delete from user where username='my19001';
===== -----
-----
-

```

```

drop trigger if exists etaireiatrig1;
delimiter $ create trigger etaireiatrig1
before insert on
etaireia for each row
begin

```

```

IF (EXISTS(SELECT 1 FROM etaireia WHERE afm = NEW.afm)) THEN
    insert into history values('unknown',now(),'Failure','insert','etaireia');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'INSERT failed due to duplicate
afm'; else
    insert into history
values('unknown',now(),'Success','insert','etaireia'); END IF; end$
delimiter ;

```

```

insert into etaireia values ('023453344','B Patras',' Sofitel
A.E.','48385','Gounari',50,'Patras','Greece'); insert into etaireia values ('023453344','B Patras',' Sofitel
A.E.','48385','Gounari',50,'Patras','Greece');

```

=====

```

drop trigger if exists etaireiatrig2;
delimiter $
create trigger etaireiatrig2 after update on
etaireia for each row
begin

```

```

    IF NEW.AFM <> OLD.AFM or NEW.DOY<>OLD.DOY or NEW.NAME <> OLD.NAME then
insert into history values('admin',now(),'Failure','update','etaireia');
    SIGNAL SQLSTATE VALUE '45000'; SET MESSAGE_TEXT = 'UPDATE failed due to invalid modification on
AFM/DOY/NAME';

```

```

    ELSE
        insert into history
values('ADMIN',now(),'Success','update','etaireia');      end if;
end$
delimiter ;

```

```

update etaireia set afm='023451232' where afm='fffff';
update etaireia set doy='gdfgfd' where doy='C Patras';
update etaireia set name='aaaaddfd' where name='EXPENDITURE Ltd';

```

=====

```

drop trigger if exists etaireiatrig3;
delimiter $
create trigger etaireiatrig3 before delete on
etaireia; for each row
begin

```

```

IF (EXISTS(SELECT 1 FROM etaireia WHERE AFM = OLD.AFM)) THEN
insert into history values(OLD.AFM,now(),'Success','delete','etaireia'); else
    insert into history values(OLD.AFM,now(),'Failure','delete','etaireia');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DELETE failed due to a non existent etaireia
AFM'; END IF; end$
delimiter ;

```

```

-----
- drop trigger if exists jobtrig1; delimiter $
create trigger jobtrig1 before insert on
job for each row
begin

```

```

    if new.announce_date > curdate() then
        insert into history values('admin',now(),'failure','insert','job');
        SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'INSERT failed due to announce day> current
day afm';      else
        insert into history
values('admin',now(),'success','insert','job');      end if; end$
delimiter ;

```

```

insert into job values (14,'2019-01-21',1000,'Java Developer','Patra','msmith','2020-10-2','2018-1220');
insert into job values (14,'2019-01-01',1000,'Java Developer','Patra','msmith','2018-102','2018-12-20');
=====
=== drop trigger if exists jobtrig2;
delimiter $ create trigger jobtrig2
before update
on job for each row
begin if new.announce_date > curdate() then
    insert into history values('admin',now(),'failure','update','job');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'UPDATE failed due to announce day> current
day'; else
    insert into history
values('admin',now(),'sucess','update','job');    end if;
end$
delimiter ;

update job set announce_date='2019-01-31' where id=1;
=====
===
= drop trigger if exists
jobtrig3;
delimiter $
create trigger jobtrig3 before delete on
job for each row
begin
    IF (EXISTS(SELECT 1 FROM job WHERE id = old.id)) THEN
        insert into history values('admin',now(),'Success','delete','job');
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DELETE hit due to a non existent job'; else
        insert into history values('admin',now(),'Failure','delete','job');
        SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DELETE failed due to a non existent job';
    END IF;
end$
delimiter ;

delete from job where id=2;
=====

```

C.

```

drop trigger if exists applies_job;
delimiter $
create trigger applies_job before delete on
applies for each row
begin
    IF (EXISTS(select 1 from job inner join applies on applies.job_id=job.id and OLD.job_id=job.id AND
SUBMISSION_DATE<CUR-
DATE())) then
        SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'DELETE of APPLICATION FAILED due to
SUBMISSION_DATE<CURDATE()';    insert into history
values('cand_username',now(),'Failure','delete','applies'); else
    insert into history

```

```
values('cand_username',now()),'Success','delete','applies');  
end if; end$
```

```
delimiter ;
```

```
delete from applies where job_id=2;
```

Μέρος B: GUIs

```
package project;
```

```
import java.awt.event.*;  
import java.sql.Connection;  
import  
java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.Vector;  
import javax.swing.*;  
import javax.swing.table.DefaultTableModel;  
import java.util.Calendar;
```

```
public class Login extends JFrame
```

```
{  
    JLabel JL_username,JL_password;  
    JTextField  
JT_password,JT_username;  
JButton btn_insert;    public String orisma;  
    int choice;
```

```
    public Login()  
    {  
        super("LOGIN");
```

```
        choice=Integer.parseInt(JOptionPane.showInputDialog("Δώστε 1 για είσοδο Recruiter, 2 για είσοδο Candidate και 3  
για είσοδο admin"));
```

```
        if (choice==1)  
            JOptionPane.showMessageDialog(null,"Δώστε username και password Recruiter");
```

```
        else if (choice==2)  
            JOptionPane.showMessageDialog(null,"Δώστε username και password Candidate");
```

```
        else if (choice==3)  
            JOptionPane.showMessageDialog(null,"Δώστε username και password Admin");
```

```
    else  
    {  
        JOptionPane.showMessageDialog(null,"Λάθος Επιλογή");
```

```
return;
```

```
    }  
    JL_username = new JLabel("Username:");  
    JL_password = new JLabel("Password:");
```

```
    JL_username.setBounds(20, 50, 100, 20);  
    JL_password.setBounds(20, 20, 100, 20);
```

```

JT_username = new JTextField(20);
JT_password = new JTextField(20);

JT_username.setBounds(130, 50, 150, 20);
JT_password.setBounds(130,20,150,20);

btn_insert = new JButton("Login");
btn_insert.setBounds(300, 50, 80, 20);

setLayout(null);

add(JL_username);
add(JL_password);

add(JT_username);
add(JT_password);

add(btn_insert);

btn_insert.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
try
{
            if
(choice==1)
                theQuery("select * from recruiter where password ="+JT_password.getText()+" and username
="+JT_username.getText()+"");

            if (choice==2)
                theQuery("select * from candidate where password ="+JT_password.getText()+" and username
="+JT_username.getText()+"");

            if (choice==3)
                theQuery("select * from user where password ="+JT_password.getText()+" and username
="+JT_username.getText()+"");

        } catch(Exception
ex){}

    }

});

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
setLocationRelativeTo(null);
setSize(500,200);

}

String getorisma()
{
    return orisma;
}

//function to execute the insert update delete query
public void theQuery(String query)
{
    Connection con = null;
    Statement st = null;

try

```

```

    {
con =
DriverManager.getConnection("jdbc:mysql://localhost/erecruit","root","");    st =
con.createStatement();

        ResultSet result=st.executeQuery(query);
            if
(result.next())
{
            JOptionPane.showMessageDialog(null,"Query Executed");
orisma=JT_username.getText();
            if
(choice==1)
            {
                int choice2=Integer.parseInt(JOptionPane.showInputDialog("1.Επεξεργασία στοιχείων
εταιρείας\n2.Επεξεργασία στοιχείων recruiter\n3.Προσθήκη νέας θέσης εργασίας\n4.Επεξεργασία θέσεων εργασίας του
recruiter\n5.Επεξεργασία θέσεων εργασίας Εταιρείας (εκτός του recruiter)"));
                    if
                        (choice2==1)
new FirmUpdate();
else
                        if (choice2==2)
new RecruiterUpdate(orisma);
else
                        if (choice2==3)
new InsertJob();
else
                        if (choice2==4)
new
JobsRecruiter(orisma);
else
if (choice2==5)
new
JobsFirmNotRecruiter(orisma);

            }
            else if (choice==2)
            {
                int choice2=Integer.parseInt(JOptionPane.showInputDialog("1.Επεξεργασία στοιχείων candidate"));

                    if (choice2==1)
new CandidateUpdate(orisma);

            }

        }
    }
else
        JOptionPane.showMessageDialog(null, "Wrong username/password");
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex.getMessage());
    }
}

public static void main(String[] args)
{
    new Login();
}

public class JobsRecruiter extends javax.swing.JFrame
{

```

```

public JobsRecruiter(String orisma)
{
    initComponents();

}

@SuppressWarnings("unchecked")

private void initComponents()
{
    jScrollPane1 = new
javax.swing.JScrollPane();    tableResult = new
javax.swing.JTable();    jButton1 = new
javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    tableResult.setModel(new
javax.swing.table.DefaultTableModel(    new Object [][] {
        {null,null, null, null, null, null, null, null}, new String[] { "id", "start date", "salary", "Position", "Edra",
"Recruiter", "announce date", "submission date" }
    ));
    jScrollPane1.setViewportView(tableResult);

    jButton1.setText("Show Jobs");
    jButton1.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(java.awt.event.ActionEvent evt)
        {
            jButton1ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(97, 97, 97)
            .addComponent(jScrollPane1, 890, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(37, 37, 37)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 120, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(33, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addComponent(jButton1)
            .addContainerGap(29, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1)
            .addGap(20, 20, 20)
            .addComponent(jButton1)
            .addContainerGap(36, Short.MAX_VALUE))
    );
}

```



```
setVisible(true);
pack();
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    Connection getcon=null;
    Connection con = null;
    Statement st = null;

try
    {
        con =
DriverManager.getConnection("jdbc:mysql://localhost/erecruit","root","");
con.createStatement();
        JOptionPane.showMessageDialog(null,"Query Executed");
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex.getMessage());
    }

    Vector col = new Vector();
    Vector dat= new Vector();
    ResultSet rs = null;
```

```
try
{
col.add("
id");
col.add("
start_dat
e");
col.add("
salary");
col.add("
position"
);
col.add("
edra");
col.add("
recruiter"
);
col.add("
announc
e_date");
col.add("
submissi
on_date"
);
System.
out.print(
orisma);
rs=con.createSt
atement().execut
eQuery("select *
from job where
recruiter="+oris
ma+""");
dat.clear();
```

```

while(rs.next())
{
    Vector v =new Vector();

    v.add(rs.getString("id").trim());
    v.add(rs.getString("start_date").trim());
    v.add(rs.getString("salary").trim());
    v.add(rs.getString("position").trim());
    v.add(rs.getString("edra").trim());
    v.add(rs.getString("recruiter").trim());
    v.add(rs.getString("announce_date").trim());
    v.add(rs.getString("submission_date").trim());

dat.add(v);
    }
    tableResult.setModel(new DefaultTableModel(dat, col));
}
catch(Exception ex)
{
    JOptionPane.showMessageDialog(null, ex.getMessage());
}
}

private javax.swing.JButton jButton1;
private javax.swing.JScrollPane
jScrollPane1; private javax.swing.JTable
tableResult;
}

public class JobsFirmNotRecruiter extends javax.swing.JFrame
{
    public JobsFirmNotRecruiter(String orisma)
    {
        initComponents();
    }

    @SuppressWarnings("unchecked")

    private void initComponents()
    {
        jScrollPane1 = new javax.swing.JScrollPane();
tableResult = new javax.swing.JTable();        tableResult.setEnabled(false);
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        tableResult.setModel(new
javax.swing.table.DefaultTableModel(            new Object [][] {
                {null,null, null, null, null, null, null, null}, new String[] {"id", "start date", "salary", "Position", "Edra",
"Recruiter", "announce date", "submission date"}
            ) );
        jScrollPane1.setViewportView(tableResult);

        jButton1.setText("Show Jobs");
        jButton1.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton1ActionPerformed(evt);
            }
        });
    }

```

```

    }
    });

    javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(97, 97, 97)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 890,
    javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(37, 37, 37)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
    javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(33, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(29, 29, 29)
                    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
    javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(20, 20, 20)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 336,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(36, Short.MAX_VALUE))
    );

```

```

    setVisible(true);
    pack();
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // setEnabled(false);
    Connection getcon=null;
    Connection con = null;
    Statement st = null;
    try
    {
        con =
        DriverManager.getConnection("jdbc:mysql://localhost/erecruit","root","");    st =
        con.createStatement();
        JOptionPane.showMessageDialog(null,"Query Executed");
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex.getMessage());
    }
}

```

```

Vector col = new Vector();

```

```

Vector dat= new Vector();
ResultSet rs = null;

```

```

        try    {
col.add("id");

col.add("start_date");
col.add("salary");
col.add("position");
col.add("edra");
col.add("recruiter");
col.add("announce_date");
        col.add("submission_date");

        rs=con.createStatement().executeQuery("select * from job where recruiter<>'"+orisma+"'");
dat.clear();

        while(rs.next())
        {
            Vector v =new Vector();

            v.add(rs.getString("id").trim());
            v.add(rs.getString("start_date").trim());
            v.add(rs.getString("salary").trim());
            v.add(rs.getString("position").trim());
            v.add(rs.getString("edra").trim());
            v.add(rs.getString("recruiter").trim());
            v.add(rs.getString("announce_date").trim());
            v.add(rs.getString("submission_date").trim());

dat.add(v);
        }
        tableResult.setModel(new DefaultTableModel(dat, col));
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null, ex.getMessage());
    }
}

private javax.swing.JButton jButton1;
private javax.swing.JScrollPane
jScrollPane1;    private javax.swing.JTable
tableResult;
}

```

```

public class RecruiterUpdate extends JFrame
{
    JLabel JL_username,JL_exp_years,JL_firm;
    JTextField JT_username,JT_exp_years,JT_firm;

    JButton btn_insert,btn_update,btn_delete;

    public RecruiterUpdate(String orisma)
    {
        super("UPDATE RECRUITER");

        JL_username = new JLabel("Username:");
        JL_exp_years = new JLabel("Exp Years:");
        JL_firm = new JLabel("Εταιρεία:");

        JL_username.setBounds(20, 20, 100, 20);
        JL_exp_years.setBounds(20, 50, 100, 20);
        JL_firm.setBounds(20, 80, 100, 20);
    }
}

```

```

JT_username = new JTextField(20);
JT_exp_years= new JTextField(20);
JT_firm = new JTextField(20);

JT_username.setBounds(130,20,150,20);
JT_exp_years.setBounds(130, 50, 150, 20);
JT_firm.setBounds(130, 80, 150, 20);

btn_update = new JButton("Update");

btn_update.setBounds(300, 80, 80, 20);

setLayout(null);

add(JL_username);
add(JL_exp_years);
add(JL_firm );

add(JT_username);
JT_username.setText(orisma);
JT_username.setEditable(false);
add(JT_exp_years);
add(JT_firm);

add(btn_update);

btn_update.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
try
        {
            theQuery("update recruiter set exp_years="+JT_exp_years.getText()+"",firm="+JT_firm.getText()+" where
username = "+JT_username.getText()+"");
        }
        catch(Exception ex){}
    }
});

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
setLocationRelativeTo(null);
setSize(450,280);

}
//function to execute the insert update delete query
public void theQuery(String query)
{
    Connection con = null;
    Statement st = null;

try
    {
con =
DriverManager.getConnection("jdbc:mysql://localhost/erecruit","root","");
        st
= con.createStatement();
        st.executeUpdate(query);
        JOptionPane.showMessageDialog(null,"Query Executed");
    }
    catch(Exception ex)
    {

```

```

        JOptionPane.showMessageDialog(null,ex.getMessage());
    }
}
}

public class CandidateUpdate extends JFrame
{
    JLabel JL_username,JL_biografiko,JL_sistatikes, JL_certificates;
    JTextField JT_username,JT_biografiko,JT_sistatikes, JT_certificates;

    JButton btn_insert,btn_update,btn_delete;

    public CandidateUpdate(String orisma)
    {
        super("UPDATE CANDIDATE");

        JL_username = new JLabel("Username:");
        JL_biografiko = new JLabel("Βιογραφικό:");
        JL_sistatikes = new JLabel("Συστατικές:");
        JL_certificates = new JLabel("certificates:");

        JL_username.setBounds(20, 20, 100, 20);
        JL_biografiko.setBounds(20, 50, 100, 20);
        JL_sistatikes.setBounds(20, 80, 100, 20);
        JL_certificates.setBounds(20, 110, 100, 20);

        JT_username = new JTextField(20);
        JT_biografiko= new JTextField(20);
        JT_sistatikes = new JTextField(20);
        JT_certificates = new JTextField(20);

        JT_username.setBounds(130,20,150,20);
        JT_biografiko.setBounds(130, 50, 150, 20);
        JT_sistatikes.setBounds(130, 80, 150, 20);
        JT_certificates.setBounds(130, 110, 150, 20);

        btn_update = new JButton("Update");

        btn_update.setBounds(300, 80, 80, 20);

        setLayout(null);

        add(JL_username);
        add(JL_biografiko);    add(JL_sistatikes);
        add(JL_certificates);

        add(JT_username);
        JT_username.setText(orisma);
        JT_username.setEditable(false);
        add(JT_biografiko);    add(JT_sistatikes);
        add(JT_certificates);

        add(btn_update);

        btn_update.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
try
                {

```

```

        theQuery("update candidate      set
bio='"+JT_biografiko.getText()+"',sistatikes='"+JT_sistatikes.getText()+"',certificates='"+JT_certificates.getText()+"' where
username = '"+JT_username.getText()+"'");
    }
    catch(Exception ex){}
}
});

```

```

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true); setLocationRelativeTo(null);
setSize(450,280);
}

```

```

//function to execute the insert update delete query
public void theQuery(String query)
{
    Connection con = null;
    Statement st = null;
    try
    {
        con =
        DriverManager.getConnection("jdbc:mysql://localhost/erecruit","root","");
        st
        = con.createStatement();
        st.executeUpdate(query);
        JOptionPane.showMessageDialog(null,"Query Executed");
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex.getMessage());
    }
}
}

```

```

public class FirmUpdate extends JFrame
{
    JLabel JL_afm,JL_doy,JL_name,JL_tel,JL_street,JL_num,JL_city,JL_country;
    JTextField JT_afm,JT_doy,JT_name,JT_tel,JT_street,JT_num,JT_city,JT_country;

    JButton btn_insert,btn_update,btn_delete;

    public FirmUpdate()
    {
        super("UPDATE FIRM");

        JL_afm = new JLabel("ΑΦΜ:");
        JL_doy = new JLabel("ΔΟΥ:");
        JL_name = new JLabel("Όνομα Εταιρείας:");
        JL_tel = new JLabel("Τηλέφωνο:");
        JL_street = new JLabel("Οδός:");
        JL_num = new JLabel("Αριθμός:");
        JL_city = new JLabel("Πόλη:");
        JL_country = new JLabel("Χώρα:");

        JL_afm.setBounds(20, 20, 100, 20);
        JL_doy.setBounds(20, 50, 100, 20);
        JL_name.setBounds(20, 80, 100, 20);
        JL_tel.setBounds(20, 110, 100, 20);
        JL_street.setBounds(20, 140, 100, 20);
        JL_num.setBounds(20, 170, 100, 20);
        JL_city.setBounds(20, 200, 100, 20);
    }
}

```

```

JL_country.setBounds(20, 230, 100, 20);

JT_afm = new JTextField(20);
JT_doy = new JTextField(20);
JT_name = new JTextField(20);
JT_tel = new JTextField(20);
JT_street = new JTextField(20);
JT_num = new JTextField(20);
JT_city = new JTextField(20);
JT_country = new JTextField(20);

JT_afm.setBounds(130,20,150,20);
JT_doy.setBounds(130, 50, 150, 20);
JT_name.setBounds(130, 80, 150, 20);
JT_tel.setBounds(130, 110, 150, 20);
JT_street.setBounds(130,140,150,20);
JT_num.setBounds(130, 170, 150, 20);
JT_city.setBounds(130, 200, 150, 20);
JT_country.setBounds(130, 230, 150, 20);

btn_insert = new JButton("Insert");
btn_update = new JButton("Update");      btn_delete
= new JButton("Delete");

btn_insert.setBounds(300, 50, 80, 20);
btn_update.setBounds(300, 80, 80, 20);
btn_delete.setBounds(300, 110, 80, 20);

setLayout(null);
add(JL_afm);      add(JL_doy);
add(JL_name);
    add(JL_tel);

    add(JL_street);
add(JL_num);      add(JL_city);
    add(JL_country);

    add(JT_afm);
add(JT_doy);
add(JT_name);
    add(JT_tel);

    add(JT_street);
add(JT_num);      add(JT_city);
add(JT_country);

    add(btn_update);

    btn_insert.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
try
            {
                //      theQuery("insert      into      users (fname,lname,age)
values(""+JT_fname.getText()+"", ""+JT_lname.getText()+"", ""+JT_age.getText()+"");
            }
            catch(Exception ex){}
        }
    });

```



```

        //button update
btn_update.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
try{
    //theQuery("update users set fname = '"+JT_fname.getText()+"',lname = '"+JT_lname.getText()+"', age =
    '"+JT_age.getText()+"' where id = '"+JT_id.getText());

    theQuery("update etaireia set tel= '"+JT_tel.getText()+"', street = '"+JT_street.getText()+"',num =
    '"+JT_num.getText()+"', city = '"+JT_city.getText()+"',country = '"+JT_country.getText()+"' where afm =
    '"+JT_afm.getText());
    }
    catch(Exception ex){}
    }
    });

        //button delete
btn_delete.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
try{
    // theQuery("delete from users where id = '"+JT_id.getText());
    }
    catch(Exception ex){}
    }
    });

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
        setLocationRelativeTo(null);
setSize(600,300);

    }
    //function to execute the insert update delete query
    public void theQuery(String query)
    {
        Connection con = null;


        Statement st = null;

try
    {
con =
DriverManager.getConnection("jdbc:mysql://localhost/erecruit","root","");
        st
        = con.createStatement();
        st.executeUpdate(query);
        JOptionPane.showMessageDialog(null,"Query Executed");
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex.getMessage());
    }
    }
}
}

```

Ενδεικτικά Screenshots εκτέλεσης κώδικα Β' Μέρος

Input

 Δώστε 1 για είσοδο Recruiter, 2 για είσοδο Candidate και 3 για είσοδο admin

OK Cancel

Message

 Δώστε username και password Recruiter


OK

LOGIN

Password:

Username: Login

Input

 1.Επεξεργασία στοιχείων εταιρείας
2.Επεξεργασία στοιχείων recruiter
3.Προσθήκη νέας θέσης εργασίας
4.Επεξεργασία θέσεων εργασίας του recruiter
5.Επεξεργασία θέσεων εργασίας Εταιρείας (εκτός του recruiter)

OK Cancel

UPDATE FIRM

ΑΦΜ:

ΔΟΥ:

Όνομα Εταιρείας:

Τηλέφωνο:

Οδός:

Αριθμός:

Πόλη:

Χώρα:

UPDATE RECRUITER

Username:

Exp Years:

Εταιρεία: