



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ

UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ
ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

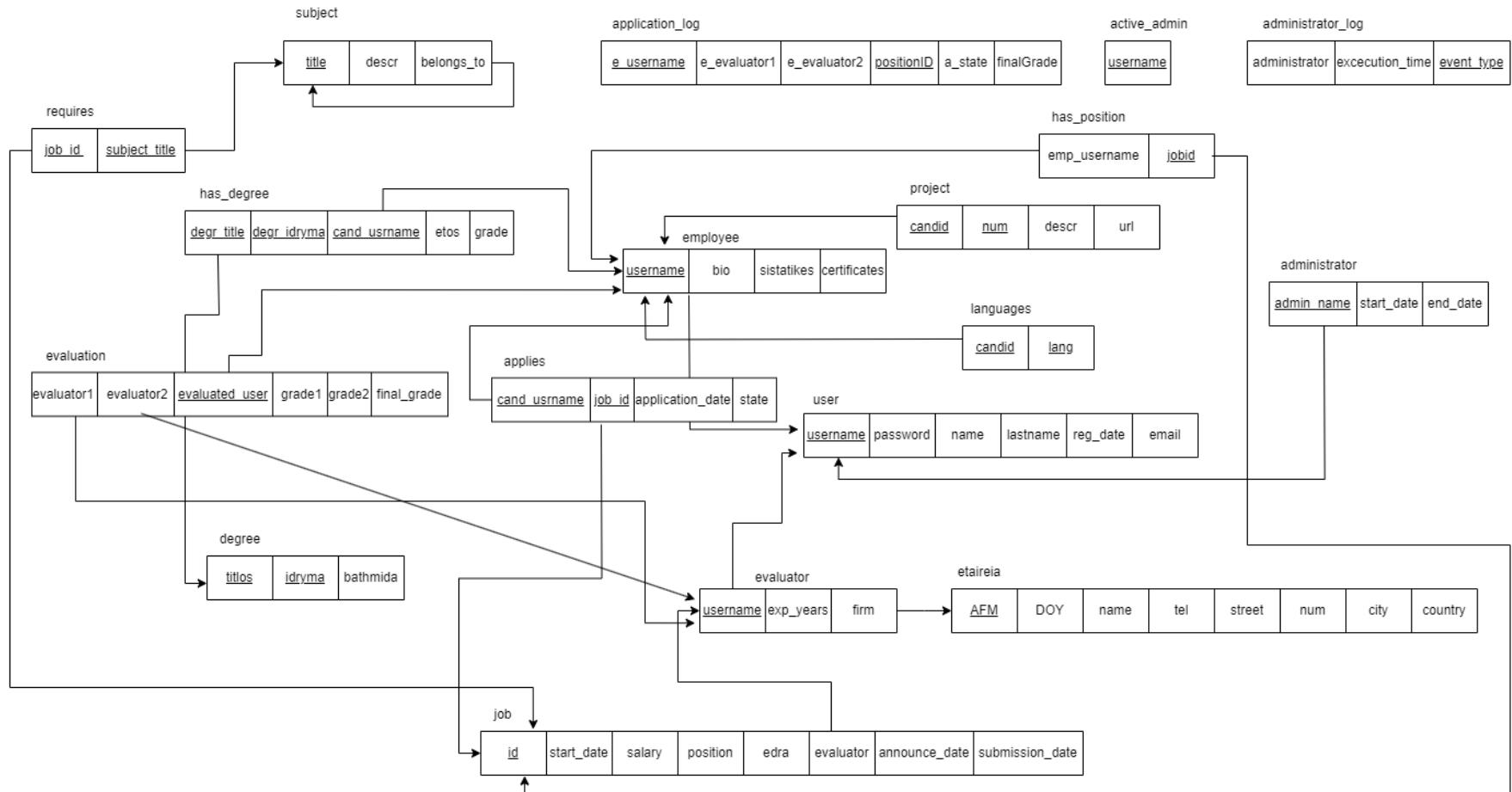
PROJECT ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Ακαδημαϊκό Έτος 2023-2024

ΚΩΝΣΤΑΝΤΙΝΟΣ ΑΝΑΣΤΑΣΟΠΟΥΛΟΣ	1093320
ΦΙΛΙΠΠΟΣ ΜΕΡΚΟΥΡΙΟΣ ΔΑΛΑΣ	1096314
ΧΑΡΑΛΑΜΠΟΣ ΑΝΑΣΤΑΣΙΟΥ	1093316

ΚΕΦΑΛΑΙΟ 1

Σχεσιακό Διάγραμμα ΒΔ



Εντολές για τη δημιουργία της ΒΔ

Εντολές Create

```

DROP DATABASE IF EXISTS etaireia_aksiologisis;
CREATE DATABASE etaireia_aksiologisis;
USE etaireia_aksiologisis;

CREATE TABLE IF NOT EXISTS etaireia(
    AFM CHAR(9) NOT NULL,
    DOY VARCHAR(30) DEFAULT 'unknown' NOT NULL,
    name VARCHAR(35) DEFAULT 'unknown' NOT NULL,
    tel VARCHAR(10) NOT NULL,
    street VARCHAR(15) DEFAULT 'unknown' NOT NULL,
    num INT(11) DEFAULT '0' NOT NULL,
    city VARCHAR(45) DEFAULT 'unknown' NOT NULL,
    country VARCHAR(15) DEFAULT 'unknown' NOT NULL,
    PRIMARY KEY(AFM),
    UNIQUE(tel)
);

```

```

CREATE TABLE IF NOT EXISTS user(
    username VARCHAR(30) NOT NULL,
    password VARCHAR(20) DEFAULT 'unknown' NOT NULL,
    name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
    lastname VARCHAR(35) DEFAULT 'unknown' NOT NULL,
    reg_date DATETIME DEFAULT '1900-01-01 00-00-00' NOT NULL,
    email VARCHAR(30) NOT NULL,
    PRIMARY KEY(username),
    UNIQUE(email)
);

CREATE TABLE IF NOT EXISTS evaluator(
    username VARCHAR(30) NOT NULL,
    exp_years tinyint(4) DEFAULT '0' NOT NULL,
    firm CHAR(9) NOT NULL,
    PRIMARY KEY(username),
    CONSTRAINT EVALUSER FOREIGN KEY(username) REFERENCES user(username)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT EVALFIRM FOREIGN KEY(firm) REFERENCES etaireia(AFM)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS employee(
    username VARCHAR(30) NOT NULL,
    bio TEXT NOT NULL,
    sistatikes VARCHAR(35) DEFAULT 'unknown',
    certificates VARCHAR(35) DEFAULT 'unknown',
    PRIMARY KEY(username),
    CONSTRAINT EMPLUSER FOREIGN KEY(username) REFERENCES user(username)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS languages(
    candid VARCHAR(30) NOT NULL,
    lang SET('EN', 'FR', 'SP', 'GE', 'CH', 'GR') NOT NULL,
    PRIMARY KEY(candid, lang),
    CONSTRAINT LANGEMPL FOREIGN KEY(candid) REFERENCES employee(username)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS project(
    candid VARCHAR(30) NOT NULL,
    num TINYINT(4) NOT NULL,
    descr TEXT NOT NULL,
    url VARCHAR(60) DEFAULT 'unknown' NOT NULL,
    PRIMARY KEY(candid, num),
    CONSTRAINT PROJEMPL FOREIGN KEY(candid) REFERENCES employee(username)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS job(
    id INT(11) AUTO_INCREMENT NOT NULL,
    start_date DATE DEFAULT '1900-01-01' NOT NULL,
    salary FLOAT DEFAULT '0' NOT NULL,
    position VARCHAR(60) DEFAULT 'unknown' NOT NULL,

```

```

edra VARCHAR(60) DEFAULT 'unknown' NOT NULL,
evaluator VARCHAR(30) NOT NULL,
announce_date DATETIME DEFAULT '1900-01-01 00-00-00' NOT NULL,
submission_date DATE DEFAULT '1900-01-01' NOT NULL,
PRIMARY KEY(id),
CONSTRAINT JOBEVAL FOREIGN KEY(evaluator) REFERENCES evaluator(username)
ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS applies(
    cand_usrname VARCHAR(30) NOT NULL,
    job_id INT(11) NOT NULL,
    application_date DATE DEFAULT '1900-01-01' NOT NULL,
    state ENUM('active', 'completed', 'canceled') DEFAULT 'active' NOT NULL,
    PRIMARY KEY(cand_usrname, job_id),
    CONSTRAINT APPEMPL FOREIGN KEY(cand_usrname) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT APPLJOB FOREIGN KEY(job_id) REFERENCES job(id)
    ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS degree(
    titlos VARCHAR(150) NOT NULL,
    idryma VARCHAR(140) NOT NULL,
    bathmida ENUM('BSc', 'MSc', 'PhD') NOT NULL,
    PRIMARY KEY(titlos, idryma)
);

```

```

CREATE TABLE IF NOT EXISTS has_degree(
    degr_title VARCHAR(150) NOT NULL,
    degr_idryma VARCHAR(140) NOT NULL,
    cand_usrname VARCHAR(30) NOT NULL,
    etos YEAR(4) DEFAULT '0000' NOT NULL,
    grade FLOAT DEFAULT '0' NOT NULL,
    PRIMARY KEY(degr_title, degr_idryma, cand_usrname),
    CONSTRAINT DEGEMPL FOREIGN KEY(cand_usrname) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT DEGDEG FOREIGN KEY(degr_title, degr_idryma) REFERENCES degree(titlos, idryma)
    ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS subject(
    title VARCHAR(36) NOT NULL,
    descr TINYTEXT NOT NULL,
    belongs_to VARCHAR(36),
    PRIMARY KEY(title),
    CONSTRAINT SUBSUB FOREIGN KEY(belongs_to) REFERENCES subject(title)
    ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS requires(
    job_id INT(11) NOT NULL,
    subject_title VARCHAR(36) NOT NULL,
    PRIMARY KEY(job_id, subject_title),

```

```

CONSTRAINT REQJOB FOREIGN KEY(job_id) REFERENCES job(id)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT REQSUB FOREIGN KEY(subject_title) REFERENCES subject(title)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS has_position(
    emp_username VARCHAR(30) NOT NULL,
    jobid INT(11) NOT NULL,
    PRIMARY KEY(jobid),
    CONSTRAINT POSEMPL FOREIGN KEY(emp_username) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT POSJOB FOREIGN KEY(jobid) REFERENCES job(id)
    ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS evaluation(
    evaluator1 VARCHAR(30),
    evaluator2 VARCHAR(30),
    evaluated_user VARCHAR(30) NOT NULL,
    grade1 INT DEFAULT '0',
    grade2 INT DEFAULT '0',
    final_grade INT DEFAULT '0',
    PRIMARY KEY(evaluated_user),
    CONSTRAINT EVALEmpl FOREIGN KEY(evaluated_user) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT EVALEVAL1 FOREIGN KEY(evaluator1) REFERENCES evaluator(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT EVALEVAL2 FOREIGN KEY(evaluator2) REFERENCES evaluator(username)
    ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS application_log(
    e_username VARCHAR(30) NOT NULL,
    e_evaluator1 VARCHAR(30),
    e_evaluator2 VARCHAR(30),
    positionID INT NOT NULL,
    a_state ENUM('active', 'completed', 'canceled') DEFAULT 'completed' NOT NULL,
    finalGrade INT NOT NULL,
    PRIMARY KEY(e_username, positionID)
    #CONSTRAINT LOGEMPL FOREIGN KEY(e_username) REFERENCES employee(username)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGJOB FOREIGN KEY(positionID) REFERENCES job(id)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGSTA FOREIGN KEY(a_state) REFERENCES applies(state)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGGRA FOREIGN KEY(finalGrade) REFERENCES evaluation(final_grade)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGEVAL1 FOREIGN KEY(e_evaluator1) REFERENCES evaluator(username)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGEVAL2 FOREIGN KEY(e_evaluator2) REFERENCES evaluator(username)
    #ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS administrator(
    admin_name VARCHAR(30) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE DEFAULT NULL,
    PRIMARY KEY(admin_name),
    CONSTRAINT ADMINUSER FOREIGN KEY(admin_name) REFERENCES user(username)
    ON UPDATE CASCADE ON DELETE CASCADE
);


```

```

CREATE TABLE active_admin(
    username VARCHAR(30) NOT NULL,
    PRIMARY KEY(username)
);

```

```

CREATE TABLE IF NOT EXISTS administrator_log(
    administrator VARCHAR(30) NOT NULL,
    excecution_time DATETIME DEFAULT '2024-01-01 00:00:00',
    event_type TEXT(256) NOT NULL,
    PRIMARY KEY (event_type(256))
);

```

Εντολές Insert

```

USE etaireia_aksiologisis;

DELIMITER $

DROP TRIGGER IF EXISTS autoIncrementProject$;

CREATE TRIGGER autoIncrementProject
BEFORE INSERT ON project
FOR EACH ROW
BEGIN
    DECLARE pnum TINYINT(4);
    SELECT MAX(num) INTO pnum
    FROM project
    WHERE candid = NEW.candid;

    IF pnum IS NULL THEN
        SET pnum = 0;
    END IF;

    SET NEW.num = pnum + 1;
END$;

DELIMITER ;

INSERT INTO etaireia (AFM, DOY, name, tel, street, num, city, country)

```

VALUES

```
('111234798', 'ATHENS', 'ZeroCom', '2102625215', 'Androutsou', 13, 'Athens', 'Greece'),  
('234987234', 'PATRAS', 'Interday', '2610102343', 'Karaiskaki', 90, 'Patras', 'Greece'),  
('890345879', 'KALAMATA', 'Tranzitzone', '2763098765', 'Eleftherias', 101, 'Kalamata', 'Greece'),  
('784512369', 'PATRAS', 'ConnectWave Communications', '2610123456', 'Androutsou', 58, 'Patras', 'Greece'),  
('630974182', 'THEBES', 'SwiftLink TeleSystems', '2262032550', 'Amfionos', 15, 'Thebes', 'Greece'),  
('215689347', 'THESSALONIKI', 'NexusNet Telecom Technologies', '2310123456', 'Egnatia', 88, 'Thessaloniki', 'Greece'),  
('10AM56789', 'ATHENS', 'PEPSI', '2102516782', 'Ermou', 7, 'Athens', 'Greece'),  
('10AM62718', 'PATRAS', 'NIKE', '2610827165', 'Ermou', 27, 'Patras', 'Greece'),  
('10AM32176', 'KOZANI', 'ADIDAS', '2461022516', 'Nikolaou', 17, 'Kozani', 'Greece');
```

INSERT INTO user (username, password, name, lastname, reg_date, email)

VALUES

```
('maria123', 'sidufhnjsdf', 'Maria', 'Alexopoulou', '2023-01-15 08:30:00', 'maria123@gmail.com'),  
('giorgospet', '93hg98grh', 'Giorgos', 'Petropoulos', '2023-02-20 10:45:00', 'georgepetropoulos@gmail.com'),  
('mark_ion', '3908ifdjfh', 'Markos', 'Ioannou', '2023-03-25 12:15:00', 'mark.j@gmail.com'),  
('sara84', '1234567890', 'Sara', 'Andreou', '2023-04-30 14:00:00', 'sara.999@gmail.com'),  
('peter_wilson', 'lolokoko', 'Peter', 'Wilson', '2023-05-05 16:20:00', 'peterwil@gmail.com'),  
('egwdean', '0irfje9rifj', 'Kostas', 'Anastasopoulos', '2023-06-10 18:00:00', 'egwegw@gmail.com'),  
('anastasis9090', 'lmlmlma88yu', 'Anastasis', 'Kyriakopoulos', '2023-07-15 20:30:00', 'anast.kyre@gmail.com'),  
('tzinaaa', 'foegoeirjg', 'Tzina', 'Lykourgou', '2023-08-20 22:45:00', 'tzinlyk@gmail.com'),  
('michalis123', 'aa890890', 'Michalis', 'Mpikos', '2023-09-25 00:15:00', 'michalis123@gmail.com'),  
('nicnic', 'aaaaakokokok', 'Nicole', 'Aslanidi', '2023-10-30 02:00:00', 'nicoleaslanidi@gmail.com'),  
('manos1978', 'p9ifrje9ruifj', 'Manos', 'Asioglou', '2023-11-05 04:20:00', 'manos123@gmail.com'),  
('emma_ross', 'woeidjweoidjioej', 'Emma', 'Ross', '2023-12-10 06:00:00', 'emma.ross@gmail.com'),  
('emily_johnson84', 'duyfgwuef78', 'Emily', 'Johnson', '2023-03-13 14:00:00', 'emily_johnson84@gmail.com'),  
('alex_smithson22', 'hduwdh6786', 'Alex', 'Smithson', '2023-04-26 18:45:00', 'alexsmithson@gmail.com'),  
('olivia_brown92', '3748jdscsj', 'Olivia', 'Brown', '2023-05-01 15:35:00', 'oliviabrown@gmail.com'),  
('jack_robinson77', '556783420', 'Jack', 'Robinson', '2023-07-28 19:31:00', 'jackrobinson@gmail.com'),  
('chloe_davis89', 'grinder567', 'Chloe', 'Davis', '2023-07-05 09:20:00', 'chloedav@gmail.com'),  
('ethan_miller64', 'isyoboy3434', 'Ethan', 'Miller', '2023-06-19 11:26:00', 'ethanmiller@gmail.com'),  
('lily_wilson78', 'grgerg454', 'Lily', 'Wilson', '2023-07-05 21:13:00', 'lily_wilson@gmail.com'),  
('noah_thompson55', 'fgeft4t545', 'Noah', 'Thompson', '2023-07-04 19:45:00', 'noah_thompson@gmail.com'),  
('ava_harrison86', 'aaoiokjin458', 'Ava', 'Harrison', '2023-08-16 14:15:00', 'ava_harrison@gmail.com'),  
('james_mitchell71', 'isoiswni798', 'James', 'Mitchell', '2023-09-28 01:00:00', 'james_mitchell@gmail.com'),  
('sophia_baker79', 'osideienkscn90', 'Sophia', 'Baker', '2023-10-01 03:43:00', 'sophia_baker@gmail.com'),  
('daniel_parker88', 'weywureosxcn870', 'Daniel', 'Parker', '2023-10-15 09:00:00', 'daniel_parker@gmail.com'),  
('user1','gsjhahgsa','Giorgos','Papadopoulos','2023-12-30 07:27:00','us1@email.com'),  
('user2','gsjhaadqw','Kostas','Papadopoulos','2023-09-27 09:29:00','us2@email.com'),  
('user3','gswwwgsa','Roula','Gogka','2023-04-21 11:17:00','us3@email.com'),  
('user4','yauagdahgsa','Maki','Toge','2023-03-30 13:02:00','us4@email.com'),  
('user5','QAWJSgsa','Itadori','Yugi','2022-11-08 20:20:00','us5@email.com'),  
('user6','gshagshhgsa','Gojo','Satoru','2023-12-28 14:21:00','us6@email.com'),  
('user7','hahahhgsa','Mei','Zenin','2022-09-30 11:27:00','us7@email.com'),  
('user8','hahahsaahgsa','Foivos','Delivorias','2023-01-01 01:27:00','us8@email.com'),  
('user9','gsjhjjjaGG','Despoina','Vndi','2023-09-02 11:30:00','us9@email.com'),  
('user10','1999wwzz','Lionel','Messi','2023-07-29 08:13:00','us10@email.com'),  
('user11','goat10hgsa','Cristiano','Ronaldo','2022-11-27 17:27:00','us11@email.com'),  
('user12','netflix2121','Nina','Simone','2023-12-11 07:27:00','us12@email.com');
```

```
INSERT INTO evaluator (username, exp_years, firm)
```

```
VALUES
```

```
('maria123', 5, '111234798'),  
('giorgospet', 3, '234987234'),  
('mark_ion', 8, '890345879'),  
('sara84', 6, '111234798'),  
('peter_wilson', 4, '234987234'),  
('egwdean', 7, '890345879'),  
('emily_johnson84', 4, '784512369'),  
('alex_smithson22', 2, '630974182'),  
('olivia_brown92', 5, '215689347'),  
('jack_robinson77', 9, '784512369'),  
('chloe_davis89', 3, '630974182'),  
('ethan_miller64', 4, '215689347'),  
('user1', 3, '10AM56789'),  
('user2', 7, '10AM56789'),  
('user3', 2, '10AM62718'),  
('user4', 8, '10AM62718'),  
('user5', 5, '10AM32176'),  
('user6', 9, '10AM32176');
```

```
INSERT INTO employee (username, bio, sistatikes, certificates)
```

```
VALUES
```

```
('anastasis9090', 'Experienced software engineer with a focus on web development.', 'Web development', 'Bachelor of Computer Science'),  
('tzinaaa', 'Passionate about data analysis and machine learning.', 'Data analysis, Machine learning', 'Master of Data Science'),  
('michalis123', 'Results-driven project manager with a proven track record.', 'Project management', 'Master of Business Administration'),  
('nicnic', 'Creative graphic designer with expertise in branding.', 'Graphic design, Branding', 'Bachelor of Fine Arts'),  
('manos1978', 'Skilled network administrator with a strong IT background.', 'Network administration', 'Bachelor of Information Technology'),  
('emma_ross', 'Dynamic marketing specialist with experience in digital campaigns.', 'Digital marketing', 'Bachelor of Marketing'),  
('lily_wilson78', 'A proficient network engineer adept at navigating the intricate tapestry of connectivity. Dedicated to optimizing network efficiency and resolving complexities.', 'Network Engineering', 'Bachelor of Computer Science'),  
('noah_thompson55', 'An astute data analyst adept at deciphering vast datasets to extract pivotal insights shaping the telecommunications landscape. Proficient in data analytics.', 'Data analysis', 'Master of Data Science'),  
('ava_harrison86', 'A dedicated customer support specialist adept at bridging the technical intricacies of telecommunications with user-friendly assistance.', 'Customer Relationship Management', 'Bachelor of Business Administration'),  
('james_mitchell71', 'An unwavering cybersecurity specialist dedicated to fortifying digital landscapes against evolving threats.', 'Master of Network Security', 'Bachelor of Computer Science'),  
('sophia_baker79', 'A seasoned telecommunications engineer adept at architecting robust and scalable network infrastructures.', 'Telecommunications Engineering', 'Bachelor of Telecommunications'),  
('daniel_parker88', 'A forward-thinking technology strategist specializing in telecommunications. Proficient in devising strategic roadmaps that align with organizational goals.', 'Bachelor of Technology', 'Digital marketing'),  
('user7', 'this is text', 'sistatiki1', 'BSc Physics'),  
('user8', 'this is text', 'sistatiki2', 'BSc Mathematics'),  
('user9', 'this is text', 'sistatiki3', 'BSc Data Analysis'),  
('user10', 'this is text', 'sistatiki4', 'MSc Physics'),  
('user11', 'this is text', 'sistatiki5', 'BSc Computer Science'),  
('user12', 'this is text', 'sistatiki6', 'PhD Computer Architecture');
```

```
INSERT INTO languages (candid, lang)
```

VALUES

```
('anastasis9090', 'EN'),  
('anastasis9090', 'FR'),  
('tzinaaa', 'FR'),  
('michalis123', 'SP'),  
('lily_wilson78', 'EN'),  
('james_mitchell71', 'FR'),  
('james_mitchell71', 'SP'),  
('daniel_parker88', 'GE'),  
('user8', 'EN'),  
('user12', 'FR'),  
('user9', 'SP'),  
('user7', 'EN');
```

INSERT INTO project (candid, num, descr, url)

VALUES

```
('anastasis9090', 0, 'E-commerce Website', 'https://www.ecomweb123.com'),  
('tzinaaa', 0, 'Blog Platform', 'https://www.bigblog1.com'),  
('michalis123', 0, 'Data Analysis Tool', 'https://www.dataanalysisnow.com'),  
('manos1978', 0, 'Project Management System', 'https://www.proj.man.com'),  
('tzinaaa', 0, 'Brand Identity Redesign', 'https://www.edesign.name.com'),  
('emma_ross', 0, 'Network Infrastructure Upgrade', 'https://www.infranet12.com'),  
('anastasis9090', 0, 'Digital Marketing Campaign', 'https://www.marketing123.com'),  
('lily_wilson78', 0, 'OptiLink: Enhancing Network Efficiency and Scalability', 'https://www.optilink.com'),  
('noah_thompson55', 0, 'DataSift: Unraveling Insights to Shape Telecom Trends', 'https://www.datasift.com'),  
('ava_harrison86', 0, 'ClientConnect: Elevating Customer Interaction Experience', 'https://www.clientconnect.com'),  
('james_mitchell71', 0, 'CyberShield: Fortifying Telecom Networks Against Threats', 'https://www.cybershield.com'),  
('sophia_baker79', 0, 'NetArchitect: Designing Resilient Network Infrastructures', 'https://www.infranet.com'),  
('daniel_parker88', 0, 'TechVista: Charting Future Telecom Strategies', 'https://www.telstrategist.com'),  
('lily_wilson78', 0, 'DataFlow Nexus: Optimizing Network Bandwidth and Stability', 'https://www.dataflownexus.com'),  
('user7', 0, '2s Numbers Multiplier', 'https://www.arduino/toge.com'),  
('user8', 0, '2s Numbers Adder', 'https://www.arduino/egot.com'),  
('user9', 0, 'DFA to NFA', 'https://www.doityourselfNfaDfa.com'),  
('user10', 0, 'CSV reader', 'https://www.CsvREAD/toge.com'),  
('user11', 0, 'Sound Redesign', 'https://www.flstudio/toge.com'),  
('user12', 0, 'CAD simulator', 'https://www.VerilogLover/toge.com');
```

INSERT INTO job (id, start_date, salary, position, edra, evaluator, announce_date, submission_date)

VALUES

```
(NULL, '2023-01-15', 80000, 'Software Engineer', 'Metropolis', 'maria123', '2023-01-01 08:30:00', '2023-01-31'),  
(NULL, '2023-02-20', 90000, 'Data Analyst', 'Cityville', 'maria123', '2023-02-01 10:45:00', '2023-02-28'),  
(NULL, '2023-03-25', 100000, 'Project Manager', 'Innovation City', 'giorgospet', '2023-03-01 12:15:00', '2023-03-31'),  
(NULL, '2023-04-30', 75000, 'Graphic Designer', 'Techland', 'giorgospet', '2023-04-01 14:00:00', '2023-04-30'),  
(NULL, '2023-05-15', 85000, 'Network Administrator', 'Global City', 'peter_wilson', '2023-05-01 16:20:00', '2023-05-31'),  
(NULL, '2023-06-20', 95000, 'Marketing Specialist', 'Data City', 'mark_ion', '2023-06-01 18:00:00', '2023-06-30'),  
(NULL, '2023-07-25', 90000, 'Software Developer', 'Inno Lane', 'sara84', '2023-07-01 20:30:00', '2023-07-31'),  
(NULL, '2023-08-30', 110000, 'IT Manager', 'Globe Avenue', 'egwdean', '2023-08-01 22:45:00', '2023-08-31'),  
(NULL, '2023-01-10', 100000, 'Senior Network Engineer', 'Silicon Valley', 'emily_johnson84', '2023-01-01 08:30:00', '2023-01-31'),  
(NULL, '2023-02-10', 70000, 'Data Analyst', 'London', 'emily_johnson84', '2023-02-01 10:45:00', '2023-02-28');
```

```

(NULL, '2023-03-15', 60000, 'Customer Support Manager', 'Singapore', 'alex_smithson22', '2023-03-01 12:15:00', '2023-03-31'),
(NULL, '2023-04-15', 75000, 'Project Manager', 'Dubai', 'alex_smithson22', '2023-04-01 14:00:00', '2023-04-30'),
(NULL, '2023-05-20', 90000, 'Telecommunications Systems Architect', 'Tokyo', 'olivia_brown92', '2023-05-01 16:20:00', '2023-05-31'),
(NULL, '2023-06-20', 70000, 'Security Analyst', 'Tokyo', 'jack_robinson77', '2023-06-01 18:00:00', '2023-06-30'),
(NULL, '2023-07-25', 80000, 'Technology Strategist', 'Sydney', 'chloe_davis89', '2023-07-01 20:30:00', '2023-07-31'),
(NULL, '2023-08-30', 110000, 'Software Engineer', 'Toronto', 'ethan_miller64', '2023-08-01 22:45:00', '2023-08-31'),
(NULL, '2023-02-17', 70000, 'Sound Designer', 'Oakland', 'user1', '2023-08-02 09:30:00', '2023-11-20'),
(NULL, '2023-03-17', 90000, 'Software Engineer', 'Windows', 'user2', '2023-09-02 10:30:00', '2023-11-21'),
(NULL, '2023-02-21', 70500, 'Designer', 'Oakland', 'user3', '2023-10-02 09:40:00', '2023-01-20'),
(NULL, '2023-11-17', 170000, 'Tester', 'New Transistors', 'user4', '2023-10-02 09:30:00', '2023-11-20'),
(NULL, '2023-02-17', 100000, 'Data Analyst', 'Analyst', 'user5', '2023-08-02 09:30:00', '2023-09-20'),
(NULL, '2023-02-17', 75000, 'Sound Designer', 'Oakland', 'user6', '2023-08-02 19:30:00', '2023-03-20');

```

```
INSERT INTO applies (cand_usrname, job_id, application_date)
```

```
VALUES
```

```

('anastasis9090', 1, '2023-01-05'),
('tzinaaa', 2, '2023-02-15'),
('michalis123', 3, '2023-03-25'),
('nicnic', 4, '2023-04-10'),
('manos1978', 5, '2023-05-12'),
('emma_ross', 6, '2023-06-20'),
('lily_wilson78', 1, '2023-07-02'),
('noah_thompson55', 2, '2023-08-18'),
('ava_harrison86', 3, '2023-09-05'),
('james_mitchell71', 4, '2023-10-14'),
('sophia_baker79', 5, '2023-11-22'),
('daniel_parker88', 6, '2023-12-01'),
('user7', 1, '2023-01-27'),
('user8', 2, '2023-02-08'),
('user9', 3, '2023-03-11'),
('user10', 4, '2023-04-30'),
('user11', 5, '2023-05-19'),
('user12', 6, '2023-06-07');

```

```
INSERT INTO degree (titlos, idryma, bathmida)
```

```
VALUES
```

```

('Bachelor of Computer Engineering', 'University of Patras', 'BSc'),
('Master of Data Science', 'University of Peloponnesse', 'MSc'),
('Master of Business Administration', 'University of Patras', 'MSc'),
('Bachelor of Fine Arts', 'Aristotle University of Thessaloniki', 'BSc'),
('Bachelor of Information Technology', 'University of Patras', 'BSc'),
('Bachelor of Marketing', 'University of Pireaus', 'PhD'),
('Bachelor of Computer Science', 'University of Patras', 'BSc'),
('Master of Data Science', 'University of Patras', 'MSc'),
('Bachelor of Business Administration', 'University of Thessaloniki', 'BSc'),
('Master of Network Security', 'Aristotle University of Thessaloniki', 'MSc'),
('Bachelor of Telecommunications Engineering', 'University of Piraeus', 'BSc'),
('Bachelor of Technology Innovation and Strategy', 'Kapodistrian University of Athens', 'BSc'),
('Bachelor of Physics', 'Harvard University', 'BSc'),
('Bachelor of Mathematics', 'Oxford University', 'BSc'),

```

```

('Master of Computer Engineering', 'University of Patras', 'MSc'),
('Bachelor of Physics', 'Warwick University', 'BSc'),
('Bachelor of Computer Science', 'Athens University', 'BSc'),
('PhD of Computer Architecture', 'Harvard University', 'PhD');

INSERT INTO has_degree (degr_title, degr_idryma, cand_usrname, etos, grade)
VALUES
('Bachelor of Computer Science', 'University of Patras', 'anastasis9090', 2019, 5.5),
('Master of Data Science', 'University of Peloponnesse', 'tzinaaa', 2021, 9.0),
('Master of Business Administration', 'University of Patras', 'michalis123', 2020, 7.5),
('Bachelor of Fine Arts', 'Aristotle University of Thessaloniki', 'emma_ross', 2018, 6.5),
('Bachelor of Computer Science', 'University of Patras', 'lily_wilson78', 2020, 8),
('Master of Data Science', 'University of Patras', 'noah_thompson55', 2021, 7),
('Bachelor of Business Administration', 'University of Thessaloniki', 'ava_harrison86', 2019, 8.5),
('Master of Network Security', 'Aristotle University of Thessaloniki', 'james_mitchell71', 2017, 6.5),
('Bachelor of Physics', 'Harvard University', 'user7', 2020, 8.5),
('Bachelor of Mathematics', 'Oxford University', 'user8', 2018, 7),
('Master of Computer Engineering', 'University of Patras', 'user9', 2020, 7.5),
('PhD of Computer Architecture', 'Harvard University', 'user12', 2017, 9);

INSERT INTO subject (title, descr, belongs_to)
VALUES
('Programming Fundamentals', 'Introduction to programming concepts and problem-solving.', NULL),
('Data Analysis', 'Exploratory data analysis and statistical techniques.', NULL),
('Project Management', 'Principles and practices of project management.', NULL),
('Graphic Design Principles', 'Fundamental principles of graphic design.', NULL),
('Computer Networks', 'Fundamentals of computer networking.', 'Programming Fundamentals'),
('Digital Marketing Strategies', 'Strategies for effective digital marketing campaigns.', 'Programming Fundamentals'),
('Advanced Data Structures', 'In-depth study of advanced data structures.', 'Data Analysis'),
('Business Analytics', 'Application of statistical analysis to business data for decision-making.', 'Data Analysis'),
('Advanced Algorithms', 'Exploring complex algorithms and advanced data structures used in optimizing computational processes and problem-solving.', NULL),
('Machine Learning', 'Investigating the principles and applications of machine learning and AI techniques in developing intelligent systems and algorithms.', NULL),
('Network Security', 'Understanding cryptographic principles and network security protocols to protect information and ensure secure communication over networks.', NULL),
('Wireless Networking', 'Studying wireless communication technologies and protocols alongside mobile networking architectures and their applications in modern communication systems.', NULL),
('Ethical Hacking', 'Exploring the methods and tools used by ethical hackers to identify vulnerabilities and conduct penetration tests to secure networks and systems.', NULL),
('Incident Response', 'Investigating procedures for incident handling, digital evidence collection, and forensic analysis to respond effectively to cybersecurity incidents.', NULL),
('Digital Marketing', 'Analyzing digital marketing strategies, tools, and analytics methods to optimize campaigns and drive effective online marketing initiatives.', NULL),
('Vulnerability Assessment', 'Assessing vulnerabilities in systems, networks, and applications while managing and mitigating risks associated with potential security threats.', 'Ethical Hacking'),
('Micro Electronics', 'Introduction to the study and manufacture of very small electronic designs and components', NULL),
('Linear Algebra', 'Introduction to vector spaces and linear transformations', NULL),
('Computer Architecture', 'Hardware', NULL),
('Logic Design', 'Basic organization of the circuitry of a digital computer', 'Computer Architecture'),
('Signal and Systems', 'Basic description of signals and systems', NULL),
('Matrices', 'Basic tool of Linear Algebra', 'Linear Algebra'),
('Programming', 'Basic analysis of higher level languages', NULL),

```

```

('C Programming', 'C higher level language', 'Programming');

INSERT INTO requires (job_id, subject_title)
VALUES
(1, 'Programming Fundamentals'),
(2, 'Data Analysis'),
(3, 'Project Management'),
(4, 'Graphic Design Principles'),
(5, 'Computer Networks'),
(6, 'Digital Marketing Strategies'),
(7, 'Advanced Data Structures'),
(8, 'Business Analytics'),
(1, 'Advanced Algorithms'),
(2, 'Machine Learning'),
(3, 'Network Security'),
(4, 'Wireless Networking'),
(5, 'Ethical Hacking'),
(6, 'Incident Response'),
(7, 'Digital Marketing'),
(8, 'Vulnerability Assessment'),
(1, 'Micro Electronics'),
(2, 'Linear Algebra'),
(3, 'Computer Architecture'),
(4, 'Logic Design'),
(5, 'Signal and Systems'),
(6, 'Matrices'),
(7, 'Programming'),
(8, 'C Programming');

INSERT INTO evaluation (evaluator1, evaluator2, evaluated_user, grade1, grade2)
VALUES
('maria123', 'giorgospet', 'anastasis9090', 15, 8),
('mark_ion', NULL, 'tzinaaa', 18, 1),
('sara84', 'peter_wilson', 'michalis123', 12, 16),
('egwdean', NULL, 'nicnic', 10, 1),
('emily_johnson84', 'alex_smithson22', 'manos1978', 14, 5),
('olivia_brown92', NULL, 'emma_ross', 17, 1),
('user1', NULL, 'lily_wilson78', 8, 19),
('user2', NULL, 'noah_thompson55', 13, 1),
('jack_robinson77', 'chloe_davis89', 'ava_harrison86', 20, 2),
('ethan_miller64', NULL, 'james_mitchell71', 16, 1),
('user3', NULL, 'sophia_baker79', 9, 14),
('user4', NULL, 'daniel_parker88', 11, 1),
('user5', 'maria123', 'user7', 7, 18),
('user6', NULL, 'user8', 1, 1),
('user1', 'mark_ion', 'user9', 6, 20),
('user2', NULL, 'user10', 1, 1),
('user3', 'sara84', 'user11', 5, 11),
(NULL, NULL, 'user12', 1, 1);

```

```

INSERT INTO user (username, password, name, lastname, reg_date, email)
VALUES
('jimmynew', 'wifundsc', 'Dimitris', 'Andreou', '2023-01-15 10:20:00', 'jimmynew@gmail.com'),
('maria9090', '93ofveijfnv', 'Maria', 'Alexiou', '2023-02-20 10:45:00', 'maria9090@gmail.com');

INSERT INTO administrator (admin_name, start_date, end_date)
VALUES
('jimmynew', '2023-01-15', NULL),
('maria9090', '2023-02-20', '2023-10-20');

DELIMITER $
DROP PROCEDURE IF EXISTS generateRandomRecords$
CREATE PROCEDURE generateRandomRecords()
BEGIN
DECLARE i INT;
SET i = 0;

WHILE i < 60001 DO
    INSERT INTO application_log
    VALUES (
        CONCAT('user_', i),
        CONCAT('evaluator1_', i),
        CONCAT('evaluator2_', i),
        i MOD 500 + 1,
        'completed',
        (i MOD 20) + 1
    );
    SET i = i + 1;
END WHILE;
END $
DELIMITER ;
#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600)
CALL generateRandomRecords();

```

Νέες απαιτήσεις

Ακολουθεί περιγραφή όλων των νέων πινάκων, των τροποποιήσεων που κάναμε σε υφιστάμενους πινάκες, καθώς και σχολιασμός του κώδικα και της προσέγγισης μας για την υλοποίηση των ζητουμένων. Για τη διευκόλυνση του αναγνώστη, με πράσινα και μπλε γράμματα παρουσιάζονται οι απαιτήσεις και, για κάθε απαιτηση, ακολουθεί ο τρόπος που αυτή ικανοποιείται. Τέλος, με κόκκινα γράμματα επισημαίνονται γραμμές κώδικα που είναι κρίσιμες στην κατανόηση των παραπάνω.

3.1.2.1 Απαιτηση:

Να δημιουργηθεί μηχανισμός για την εισαγωγή και την διαχείριση των αιτήσεων προαγωγής.

- Οι αιτήσεις θα πρέπει να έχουν μία από τρεις καταστάσεις: ενεργή, ολοκληρωμένη και ακυρωμένη. →

Δημιουργία ενός νέου πίνακα applies ο οποίος θα περιέχει μία έξτρα στήλη (state). Εισάγω τον περιορισμό η στήλη αυτή να δέχεται μία από τις τιμές: 'active', 'completed', 'canceled' κάνοντας την declare ως ENUM string object με predefined values τις τιμές που αναφέρθηκαν.

```
CREATE TABLE IF NOT EXISTS applies(
    cand_username VARCHAR(30) NOT NULL,
    job_id INT(11) NOT NULL,
    application_date DATE DEFAULT '1900-01-01' NOT NULL,
    state ENUM('active', 'completed', 'canceled') DEFAULT 'active' NOT NULL,
    PRIMARY KEY(cand_username, job_id),
    CONSTRAINT APPEMPL FOREIGN KEY(cand_username) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT APPLJOB FOREIGN KEY(job_id) REFERENCES job(id)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

- Νέες αιτήσεις για μια θέση εργασίας μπορούν να καταχωρίζονται έως και 15 μέρες πριν την ημερομηνία έναρξης της θέσης (start_date). →

Δημιουργία ενός Trigger (τον ονομάζω validateApplicationDate) ο οποίος, ΠΡΙΝ γίνει Insert μίας νέας αιτησης στον πίνακα applies, θα ελέγχει εάν πληρείται η παραπάνω προϋπόθεση. Εάν όχι, τότε θα στέλνει μήνυμα που θα ενημερώνει τον χρήστη για την μη έγκυρη αιτησή του.

```
USE etaireia_aksiologisis;
DELIMITER $
DROP TRIGGER IF EXISTS validateApplicationDate$;
CREATE TRIGGER validateApplicationDate
BEFORE INSERT ON applies
FOR EACH ROW
BEGIN
    DECLARE applicationDate DATE;
    DECLARE startDate DATE;
    DECLARE diff INT;

    SELECT start_date INTO startDate
    FROM job
    WHERE id = NEW.job_id;
```

```

SELECT application_date INTO applicationDate
FROM applies
WHERE applies.application_date=NEW.application_date;

SET diff = DATEDIFF(startDate, application_date);

IF diff < 15 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Invalid application date! Must be more than 15 days before the start date of the job';
END IF;
END$
```

DELIMITER ;

•Δεν θα μπορεί να καταχωριστεί νέα αίτηση ή να ενεργοποιηθεί ακυρωμένη αίτηση για εργαζόμενο ο οποίος έχει ήδη τρεις ενεργές αιτήσεις.

→

Δημιουργία δύο Triggers, ενός για την εισαγωγή νέας αίτησης: **validateApplicationInsert** και ενός για την ενεργοποίηση(update) μιας ακυρωμένης αίτησης: **validateApplicationUpdate**.

Παρακάτω παρατίθεται ο κώδικας και των δύο Triggers:

```

USE etaireia_aksiologisis;
DELIMITER $
DROP TRIGGER IF EXISTS validateApplicationInsert$
CREATE TRIGGER validateApplicationInsert
BEFORE INSERT ON applies
FOR EACH ROW
BEGIN
DECLARE numOfActive INT;
SELECT COUNT(*) INTO numOfActive
FROM applies
WHERE applies.cand_usrname=NEW.cand_usrname
AND state='active';
IF numOfActive>=3 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Invalid application. The employee already has at least 3 active applications';
END IF;
END$
```

DELIMITER ;

```

USE etaireia_aksiologisis;
DELIMITER $
DROP TRIGGER IF EXISTS validateApplicationUpdate$
CREATE TRIGGER validateApplicationUpdate
BEFORE UPDATE ON applies
FOR EACH ROW
BEGIN
DECLARE numOfActive INT;
SELECT COUNT(*) INTO numOfActive
FROM applies
```

```

WHERE applies.cand_usrname=NEW.cand_usrname
AND state='active';
IF numOfActive>=3 THEN
  SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'Invalid application. The employee already has at least 3 active applications';
END IF;
END$
```

DELIMITER ;

- Η αίτηση ενός εργαζομένου για μια θέση εργασίας, θα μπορεί να ακυρωθεί μέχρι 10 μέρες πριν την ημερομηνία έναρξης της θέσης (start_date). →

Δημιουργία ενός Trigger **validateApplicationDateCancel** ο οποίος θα ελέγχει πριν την ακύρωση μιας αίτησης (δηλαδή πριν γίνει update της κατάστασης(state) της αίτησης σε 'canceled') εάν τηρείται η παραπάνω προϋπόθεση. Ακολουθεί ο κώδικας του Trigger:

```

USE etaireia_aksiologisis;
DELIMITER $
DROP TRIGGER IF EXISTS validateApplicationDateCancel$
CREATE TRIGGER validateApplicationDateCancel
BEFORE UPDATE ON applies
FOR EACH ROW
BEGIN
  DECLARE applicationDate DATE;
  DECLARE startDate DATE;
  DECLARE diff INT;

  SELECT start_date INTO startDate
  FROM job
  WHERE id = NEW.job_id;

  SELECT application_date INTO applicationDate
  FROM applies
  WHERE applies.application_date=NEW.application_date;

  SET diff = DATEDIFF(startDate, application_date);

  IF diff < 10 AND NEW.state='canceled' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Invalid application cancel! Must be within 10 days before the start date of the job';
  END IF;
END$
```

DELIMITER ;

3.1.2.2 Απαίτηση:

Να δημιουργηθεί μηχανισμός για την αξιολόγηση και την εξαγωγή αποτελεσμάτων των αιτήσεων προαγωγής.

Δημιουργία 3 νέων πινάκων:

- Πίνακας ***evaluation*** ο οποίος θα περιέχει τους users, τους δύο αξιολογητές που αξιολογούν τον κάθε user, τους βαθμούς του κάθε αξιολογητή και τον τελικό βαθμό.
- Πίνακας ***has_position*** που θα περιέχει τις θέσεις εργασίας (το id τους) και τον user ο οποίος επιλέχθηκε τελικά για αυτή τη θέση.
- Πίνακας ***application_log***(ιστορικό αιτήσεων).

```
CREATE TABLE IF NOT EXISTS evaluation(
    evaluator1 VARCHAR(30),
    evaluator2 VARCHAR(30),
    evaluated_user VARCHAR(30) NOT NULL,
    grade1 INT DEFAULT '1',
    grade2 INT DEFAULT '1',
    final_grade INT DEFAULT '1',
    PRIMARY KEY(evaluated_user),
    CONSTRAINT EVALEMPL FOREIGN KEY(evaluated_user) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT EVALEVAL1 FOREIGN KEY(evaluator1) REFERENCES evaluator(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT EVALEVAL2 FOREIGN KEY(evaluator2) REFERENCES evaluator(username)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS has_position(
    emp_username VARCHAR(30) NOT NULL,
    jobid INT(11) NOT NULL,
    PRIMARY KEY(jobid),
    CONSTRAINT POSEMPL FOREIGN KEY(emp_username) REFERENCES employee(username)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT POSJOB FOREIGN KEY(jobid) REFERENCES job(id)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS application_log(
    e_username VARCHAR(30) NOT NULL,
    e_evaluator1 VARCHAR(30),
    e_evaluator2 VARCHAR(30),
    positionID INT NOT NULL,
    a_state ENUM('active', 'completed', 'canceled') DEFAULT 'completed' NOT NULL,
    finalGrade INT NOT NULL,
    PRIMARY KEY(e_username, positionID)
    #CONSTRAINT LOGEMPL FOREIGN KEY(e_username) REFERENCES employee(username)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGJOB FOREIGN KEY(positionID) REFERENCES job(id)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGSTA FOREIGN KEY(a_state) REFERENCES applies(state)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGGRA FOREIGN KEY(finalGrade) REFERENCES evaluation(final_grade)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGEVAL1 FOREIGN KEY(e_evaluator1) REFERENCES evaluation(evaluator1)
    #ON UPDATE CASCADE ON DELETE CASCADE,
    #CONSTRAINT LOGEVAL2 FOREIGN KEY(e_evaluator2) REFERENCES evaluation(evaluator2)
    #ON UPDATE CASCADE ON DELETE CASCADE
);
```

• **Κάθε αίτηση θα αξιολογείται από 2 αξιολογητές με ένα βαθμό από το 1 έως το 20, από τον κάθε ένα →**

Δημιουργία ενός Trigger **evaluationGrade** ο οποίος ΠΡΙΝ την εισαγωγή εγγραφής στον πίνακα evaluation ελέγχει την εγκυρότητα των βαθμών grade1 και grade2 όπως ορίζεται παραπάνω.

```
use etaireia_aksiologisis;
DELIMITER $
DROP TRIGGER IF EXISTS evaluationGrade$ 
CREATE TRIGGER evaluationGrade
BEFORE INSERT ON evaluation
FOR EACH ROW
BEGIN
    DECLARE Grade1 INT;
    DECLARE Grade2 INT;

    SELECT evaluation.grade1 INTO Grade1
    FROM evaluation
    WHERE evaluation.evaluated_user = NEW.evaluated_user;

    SELECT evaluation.grade2 INTO Grade2
    FROM evaluation
    WHERE evaluation.evaluated_user = NEW.evaluated_user;

    IF (Grade1 < 1 OR Grade1 > 20) OR (Grade2 < 1 OR Grade2 > 20) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'The grade has to be between 1 and 20.';
    END IF;
END$ 
DELIMITER ;
```

• Αν, κατά την εξαγωγή αποτελεσμάτων για μια θέση εργασίας, μια αίτηση εργαζομένου δεν έχει και τους δύο βαθμούς αξιολόγησης, τότε ο ένας βαθμός ή και οι δύο βαθμοί που λείπουν θα συμπληρώνονται ανάλογα με τα προσόντα του εργαζομένου και με βάση την βαθμολογία του ανά κριτήριο.

Δημιουργία μίας Stored Procedure **gradeCorrection** η οποία δέχεται ως παράμετρο εισόδου το username του υποψηφίου και υλοποιεί τα παραπάνω ως εξής:

Αρχικά, αποθηκεύουμε σε μεταβλητές το πλήθος από το κάθε είδος των προσόντων που διαθέτει ο υποψήφιος, όπως αυτά παρατίθενται στον Πίνακα 2 και θέτουμε corrected_grade = numLang+numProj+numBSc+2*numMSc+3*numPhD

Στη συνέχεια ελέγχουμε με συνθήκη IF εάν είναι κενά(NULL) τα πεδία των evaluator1 και evaluator2 στον πίνακα evaluation. Αν είναι, τότε θέτω τις μεταβλητές grade1 ή grade2(αντίστοιχα) ίσες με τον corrected_grade.

Τέλος, κάνω update τον πίνακα evaluation αναθέτοντας την τιμή (grade1+grade2)/2 στο πεδίο final_grade.

• **Αιτήσεις σε κατάσταση ακυρωμένη δεν συμμετέχουν στην επεξεργασία.**

• **Την θέση εργασίας θα παίρνει ο εργαζόμενος με τον υψηλότερο μέσο όρο των δυο αξιολογήσεων. Σε περίπτωση ισοβαθμίας, επιλέγεται αυτός που έκανε νωρίτερα αίτηση**

• **Ο αιτήσεις για θέση εργασίας για τις οποία έχει βγει αποτέλεσμα διαγράφονται από τον πίνακα που περιέχει τις αιτήσεις προς επεξεργασία και μεταφέρονται στο ιστορικό αιτήσεων (Απαίτηση 3.1.2.3). Η κατάστασή τους στο ιστορικό θα πρέπει να μετατραπεί από ενεργή ή ακυρωμένη σε ολοκληρωμένη. Αν η αίτηση ήταν ακυρωμένη θα καταχωρίζεται βαθμός 0, αλλιώς θα καταχωρίζεται ο βαθμός της αξιολόγησης.**

Δημιουργία μίας Stored Procedure **positionEvaluation** η οποία δέχεται ως όρισμα το id μιας θέσης εργασίας και ικανοποιεί όλα τα παραπάνω ως εξής:

Αρχικά, δημιουργούμε έναν CURSOR ο οποίος θα διατρέξει τον πίνακα applies και θα συλλέξει όλες τις ενεργές(active) αιτήσεις που αντιστοιχούν στην θέση εργασίας που εισάγω ως παράμετρο στην Procedure. Κάθε entry του CURSOR αποθηκεύεται στη μεταβλητή candidate_username και

καλείται η Procedure gradeCorrection(candidate_username), ώστε να διασφαλιστεί ότι κάθε υποψήφιος για τη θέση έχει και τους δύο βαθμούς αξιολόγησης.

Στη συνέχεια, κάνω SELECT από τον πίνακα applies την ενεργή εκείνη αίτηση(που αντιστοιχεί στη θέση εργασίας), ο υποψήφιος της οποίας έχει τον μεγαλύτερο μέσο όρο, ή έχει κάνει νωρίτερα την αίτηση. Για να το πετύχω αυτό κάνω INNER JOIN του πίνακα applies με τον πίνακα evaluation και ταξινομώ τους μέσους όρους με φθίνουσα σειρά και τις ημερομηνίες υποβολής αίτησης με αύξουσα σειρά. Θέτω LIMIT 1 ώστε να πάρω μόνο την κορυφαία αίτηση που επιλέχθηκε με βάση τα παραπάνω κριτήρια.

Προς το τέλος του κώδικα μου κάνω UPDATE τον πίνακα has_position με το candidate_username της κορυφαίας αίτησης(εάν η θέση εργασίας υπάρχει) ή κάνω INSERT τη θέση εργασίας με το candidate_username(εάν δεν υπάρχει ήδη η θέση).

Ακόμη, διαγράφω το entry που αντιστοιχεί στην αίτηση αυτή από τον πίνακα applies και το μεταφέρω(INSERT) στον πίνακα application_log θέτοντας την κατάσταση σε completed(το τελευταίο το πετυχαίνω με τη χρήση μίας «SELECT εντός INSERT» η οποία προβάλλει κάθε στήλη του entry σε μία νέα εγγραφή στον πίνακα application_log, προσθέτοντας την τιμή 'completed' στη θέση του field a_state). Φροντίζω επίσης για κάθε ακυρωμένη αίτηση στον πίνακα applies να προβάλλω τα στοιχεία της στον πίνακα application_log με την προσθήκη των τιμών 'completed' και 0 στα αντίστοιχα πεδία, εργαζόμενος με τον ίδιο τρόπο.

3.1.2.3 Απαίτηση:

Να δημιουργηθεί ιστορικό αιτήσεων. Θα δημιουργήσετε ένα πίνακα στον οποίον θα καταχωρίζετε τις ολοκληρωμένες αιτήσεις. Για κάθε εγγραφή του πίνακα θα διατηρείτε: τα username των δυο αξιολογητών (evaluator), τα username του εργαζόμενου (employee), το id της θέσης εργασίας (job), την κατάσταση (ολοκληρωμένη) και το βαθμό της αξιολόγησης. Σε αυτόν τον πίνακα θα πρέπει να καταχωρήσετε πάνω από 60.000 εγγραφές. Οι τιμές στο πεδίο που αφορά τον βαθμό αξιολόγησης θα είναι τυχαίες ακέραιες τιμές από 1-20. Προκειμένου να διευκολυνθείτε σε αυτήν την διαδικασία, μπορείτε να κάνετε την σχεδιαστική παραδοχή ότι ο πίνακας του ιστορικού δεν χρειάζεται να έχει περιορισμούς αναφορικής ακεραιότητας με άλλους πίνακες της ΒΔ.

Τον πίνακα application_log τον έχω από το προηγούμενο ερώτημα.

Προχωρώ στη δημιουργία ενός Trigger **application_logGrade** ο οποίος ΠΡΙΝ γίνει INSERT μίας νέας εγγραφής στον πίνακα application_log ελέγχει εάν η τιμή του πεδίου final_grade είναι μεταξύ 1 και 20, αλλιώς ειδοποιεί με μήνυμα λάθους.

Τέλος, δημιουργώ μία Stored Procedure **generateRandomRecords** η οποία εισάγει 60 χιλιάδες τυχαίες εγγραφές στον πίνακα application_log.

3.1.2.4 Απαίτηση:

Για τη βάση δεδομένων είναι υπεύθυνοι οι Διαχειριστές Βάσης Δεδομένων (DBA), οι οποίοι θα είναι μια νέα κατηγορία user, αντίστοιχα με τους αξιολογητές και τους εργαζόμενους. Για τους Διαχειριστές Βάσης Δεδομένων (DBA) τηρείται η ημερομηνία που ανέλαβαν το ρόλο (start_date), η οποία δεν μπορεί να είναι null. Επίσης υπάρχει end_date ημερομηνία για όσους DBA έχουν φύγει από τη θέση. Μπορεί να υπάρχουν περισσότεροι από ένας Διαχειριστές Βάσης Δεδομένων την ίδια χρονική περίοδο. Οι ενέργειες των Διαχειριστών Βάσης Δεδομένων καταγράφονται σε ένα πίνακα log όπως περιγράφεται στο Ερώτημα 3.1.4.1.

Δημιουργία ενός νέου πίνακα **administrator** και ενός πίνακα **administrator_log**.

```
CREATE TABLE IF NOT EXISTS administrator(
    admin_name VARCHAR(30) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE DEFAULT NULL,
    PRIMARY KEY(admin_name),
    CONSTRAINT ADMINUSER FOREIGN KEY(admin_name) REFERENCES user(username)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS administrator_log(
    administrator VARCHAR(30) NOT NULL,
    excecution_time DATETIME DEFAULT '2024-01-01 00:00:00',
```

```
event_type TEXT(256) NOT NULL,  
PRIMARY KEY (event_type(256))  
);
```


ΚΕΦΑΛΑΙΟ 2

Δημιουργία Stored Procedures

Παρακάτω παρουσιάζονται ο κώδικας για την κάθε Procedure καθώς και screenshots από παραδείγματα κλήσης τους.

3.1.3.1)

Ακολουθεί ο κώδικας:

```
DELIMITER $  
DROP PROCEDURE IF EXISTS checkIfEvaluated$  
CREATE PROCEDURE checkIfEvaluated(IN eval_username VARCHAR(30), IN empl_username VARCHAR(30), IN job_id INT(11), OUT result INT(11))  
BEGIN  
  
DECLARE grade1 INT(11);  
DECLARE grade2 INT(11);  
DECLARE eval1 VARCHAR(11);  
DECLARE eval2 VARCHAR(11);  
  
IF NOT EXISTS(  
    SELECT evaluator.username, employee.username, job.id  
    FROM evaluator  
    INNER JOIN job ON evaluator.username=job.evaluator  
    INNER JOIN applies ON job.id=applies.job_id  
    INNER JOIN employee ON applies.cand_username=employee.username  
    WHERE evaluator.username=eval_username  
    AND employee.username=empl_username  
    AND job.id=job_id)  
THEN  
    SET result=0;  
  
ELSE  
  
    SELECT evaluator1 INTO eval1 FROM evaluation WHERE evaluator1 = eval_username;  
    IF (eval1 IS NOT NULL)  
    THEN  
        SELECT evaluation.grade1 INTO grade1  
        FROM evaluation  
        INNER JOIN employee ON evaluation.evaluated_user=employee.username  
        WHERE evaluation.evaluator1=eval_username  
        AND evaluation.evaluated_user=empl_username;  
        IF grade1 IS NOT NULL
```

```

THEN
SET result=grade1;
ELSE
CALL gradeCorrection(empl_username);
END IF;
END IF;

SELECT evaluator2 INTO eval2 FROM evaluation WHERE evaluator2 = eval_username;
IF (eval2 IS NOT NULL)
THEN
SELECT evaluation.grade2 INTO grade2
FROM evaluation
INNER JOIN employee ON evaluation.evaluated_user=employee.username
WHERE evaluation.evaluator2=eval_username
AND evaluation.evaluated_user=empl_username;
IF grade2 IS NOT NULL
THEN
SET result=grade2;
ELSE
CALL gradeCorrection(empl_username);
END IF;
END IF;

END IF;
END$


DELIMITER ;

```

Παράδειγμα 3.1.3.1

The figure consists of three side-by-side screenshots of the MySQL Workbench interface, all connected to a "Local instance MYSQL57".

- Left Screenshot (job table):** Shows the results of the query `SELECT * FROM etaireia_aksiologisis.job;`. The table has columns: id, start_date, salary, position, edra, evaluator, announce_date, and submission_date. Data includes rows for Software Engineer, Data Analyst, Project Manager, Graphic Designer, Network Administrator, Marketing Specialist, Software Developer, IT Manager, and Senior Network Engineer.
- Middle Screenshot (evaluation table):** Shows the results of the query `SELECT * FROM etaireia_aksiologisis.evaluation;`. The table has columns: evaluator1, evaluator2, evaluated_user, grade1, grade2, and final_grade. Data includes rows for various users like maria123, giorgospet, and emma_ross.
- Right Screenshot (applies table):** Shows the results of the query `SELECT * FROM etaireia_aksiologisis.applies;`. The table has columns: cand_username, job_id, application_date, and state. Data includes rows for users like anastasis090, ava_harrison86, daniel_parker88, emma_ross, james_mitchell71, lily_wilson78, manos1978, michalis123, and nicnic.

Screenshots με την κατάσταση των πινάκων *applies*, *job* και *evaluation* πριν την κλήση της *checkIfEvaluated*.

MySQL Workbench

Local instance MYSQL57

File Edit View Query Database Server Tools Scripting Help

Navigator: create_DB_and_tables (1) stored_procedures (1) initialize_DB (2) test* evaluator job applies employee evaluation

SCHEMAS: etaireia_aksiologisis (Tables, Views, Stored Procedures, Functions), girafe, sys, test

```

1 • USE etaireia_aksiologisis;
2 • SET @RESULT=1;
3 • CALL checkIfEvaluated('maria123', 'anastasis9090', 1, @RESULT);
4 • SELECT @RESULT AS result;
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | result | 15 |

Administration Schemas

Information: No object selected

Action Output:

#	Time	Action	Message
299	16:53:23	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL);	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
300	16:53:23	DROP PROCEDURE IF EXISTS generateRandomRecords;	0 row(s) affected
301	16:53:23	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ... 0 row(s) affected	0 row(s) affected
302	16:53:23	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom... 1 row(s) affected	1 row(s) affected
303	16:57:25	USE etaireia_aksiologisis	0 row(s) affected
304	16:57:25	SET @RESULT=1	0 row(s) affected
305	16:57:25	CALL checkIfEvaluated('maria123', 'anastasis9090', 1, @RESULT)	1 row(s) affected
306	16:57:25	SELECT @RESULT AS result LIMIT 0, 1000	1 row(s) returned

Object Info Session

MySQL Workbench

Local instance MYSQL57

File Edit View Query Database Server Tools Scripting Help

Navigator: create_DB_and_tables (1) stored_procedures (1) initialize_DB (2) test* evaluator job applies employee evaluation

SCHEMAS: etaireia_aksiologisis (Tables, Views, Stored Procedures, Functions), girafe, sys, test

```

1 • USE etaireia_aksiologisis;
2 • SET @RESULT=1;
3 • CALL checkIfEvaluated('maria123', 'anastasis9090', 5, @RESULT);
4 • SELECT @RESULT AS result;
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | result | 0 |

Administration Schemas

Information: No object selected

Action Output:

#	Time	Action	Message
303	16:57:25	USE etaireia_aksiologisis	0 row(s) affected
304	16:57:25	SET @RESULT=1	0 row(s) affected
305	16:57:25	CALL checkIfEvaluated('maria123', 'anastasis9090', 1, @RESULT)	1 row(s) affected
306	16:57:25	SELECT @RESULT AS result LIMIT 0, 1000	1 row(s) returned
307	17:00:15	USE etaireia_aksiologisis	0 row(s) affected
308	17:00:15	SET @RESULT=1	0 row(s) affected
309	17:00:15	CALL checkIfEvaluated('maria123', 'anastasis9090', 5, @RESULT)	0 row(s) affected
310	17:00:15	SELECT @RESULT AS result LIMIT 0, 1000	1 row(s) returned

Object Info Session

Μετά την κλήση της `checkIfEvaluated` (περίπτωση όπου `result=0` και `result=grade1`).

3.1.3.2.)

Ακολουθεί ο κώδικας:

```
DELIMITER $  
DROP PROCEDURE IF EXISTS applicationManagement$  
CREATE PROCEDURE applicationManagement (empl_usrname VARCHAR(30), jobId INT(11), identifier ENUM ('i', 'c', 'a'))  
BEGIN  
DECLARE applDate DATE;  
DECLARE eva1 VARCHAR(30);  
DECLARE eva2 VARCHAR (30);  
DECLARE empl VARCHAR(30);  
DECLARE job INT(11);  
DECLARE sameFirm CHAR(9);  
  
SET applDate = CURDATE();  
  
IF (identifier = 'i') THEN  
  
    IF (NOT EXISTS(SELECT employee.username FROM employee WHERE employee.username=empl_usrname) OR NOT EXISTS (SELECT job.id FROM job  
WHERE job.id=jobId))  
        THEN  
            SIGNAL SQLSTATE '45000'  
            SET MESSAGE_TEXT = 'There is no such employee or job';  
        END IF;  
  
    SELECT evaluation.evaluator1 INTO eva1 FROM evaluation  
    INNER JOIN employee ON evaluation.evaluated_user = employee.username  
    INNER JOIN applies ON employee.username = applies.cand_usrname  
    WHERE evaluation.evaluated_user = empl_usrname AND applies.job_id = jobId;  
  
    SELECT evaluation.evaluator2 INTO eva2 FROM evaluation  
    INNER JOIN employee ON evaluation.evaluated_user = employee.username  
    INNER JOIN applies ON applies.cand_usrname = employee.username  
    WHERE evaluation.evaluated_user = empl_usrname AND applies.job_id = jobId;  
  
    IF (eva1 IS NULL) THEN  
        SELECT evaluator.username INTO eva1 FROM evaluator  
        INNER JOIN job ON evaluator.username=job.evaluator  
        INNER JOIN applies ON job.id=applies.job_id  
        INNER JOIN employee ON applies.cand_usrname=employee.username  
        WHERE employee.username = empl_usrname AND applies.job_id = jobId;  
        UPDATE evaluation  
        SET evaluation.evaluator1=eva1  
        WHERE evaluation.evaluated_user=empl_usrname;  
    END IF;  
  
    IF(eva2 IS NULL) THEN  
  
        SELECT evaluator.firm INTO sameFirm FROM evaluator  
        INNER JOIN evaluation ON evaluator.username=evaluation.evaluator1  
        WHERE evaluation.evaluator1 = eva1;  
  
        SELECT evaluator.username INTO eva2 FROM evaluator
```

```

WHERE evaluator.firm=sameFirm
LIMIT 1;

UPDATE evaluation
SET evaluation.evaluator2=eva2
WHERE evaluation.evaluated_user=empl_usrname;
END IF;

INSERT INTO applies VALUES (empl_usrname, jobId, applDate, 'active');

ELSEIF (identifier = 'c') THEN
IF EXISTS ( SELECT * FROM applies WHERE empl_usrname = cand_usrname AND jobId = job_id) THEN

UPDATE applies
SET applies.state = 'canceled'
WHERE applies.cand_usrname = empl_usrname AND applies.job_id = jobId;
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'The application has been canceled';
ELSE
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'There is not any application or the application has already been canceled';
END IF;

ELSEIF (identifier = 'a') THEN
IF EXISTS ( SELECT * FROM applies WHERE applies.cand_usrname = empl_usrname AND applies.job_id = jobId AND
applies.state='canceled' ) THEN

UPDATE applies
SET applies.state = 'active'
WHERE applies.cand_usrname = empl_usrname AND applies.job_id = jobId;
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'The application has been activated';
ELSE
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'There is not any application or the application has already been activated';
END IF;

END IF;

END$
```

DELIMITER ;

Παραδείγματα 3.1.3.2

Operation 'i':

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 •   SELECT * FROM etaireia_aksiologisis.applies;
```

The result grid displays the following data:

	cand_username	job_id	application_date	state
▶	anastasis9090	1	2023-01-05	active
	ava_harrison86	3	2023-09-05	active
	daniel_parker88	6	2023-12-01	active
	emma_ross	6	2023-06-20	active
	james_mitchell71	4	2023-10-14	active
	lily_wilson78	1	2023-07-02	active
	manos1978	5	2023-05-12	active
	michalis123	3	2023-03-25	active
	nicnic	4	2023-04-10	active

Πίνακας *applies* πριν την κλήση της *applicationManagement*.

The screenshot shows the MySQL Workbench interface with the SQL Editor pane containing the following code:

```
1 •   USE etaireia_aksiologisis;
2
3 •   CALL applicationManagement ('anastasis9090', 2, 'i');
```

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
169	20:22:32	INSERT INTO user (username, password, name, lastname, reg_date, email) VALUES ('jimmynew', 'wifundsc...', 'jimmy', 'new', '2023-01-15', 'jimmynew@gmail.com')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
170	20:22:32	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
171	20:22:32	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected	0.000 sec
172	20:22:32	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ...	0 row(s) affected	0.000 sec
173	20:22:32	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandomRecords();	1 row(s) affected	189.281 sec
174	20:25:50	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
175	20:25:50	CALL applicationManagement ('anastasis9090', 2, 'i')	1 row(s) affected	0.047 sec
176	20:26:04	SELECT * FROM etaireia_aksiologisis.applies LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec

Κλήση της *applicationManagement*.

The screenshot shows the MySQL Workbench interface with the 'Local instance MYSQL57' selected. In the Navigator pane, the 'Schemas' section is expanded, showing the 'etaireia_aksiologisis' schema. Under this schema, the 'Tables' section is expanded, and the 'applies' table is selected. The main query editor window displays the following SQL query:

```
1 • SELECT * FROM etaireia_aksiologisis.applies;
```

The Result Grid shows the following data:

cand_username	job_id	application_date	state
anastasis9090	1	2023-01-05	active
anastasis9090	2	2024-01-11	active
ava_harrison86	3	2023-09-05	active
daniel_parker88	6	2023-12-01	active
emma_ross	6	2023-06-20	active
james_mitchell71	4	2023-10-14	active
lily_wilson78	1	2023-07-02	active
manos1978	5	2023-05-12	active
michalis123	3	2023-03-25	active
nicnic	4	2023-04-10	active

Πίνακας *applies* μετά την κλήση της *applicationManagement*.

Operation 'c':

The screenshot shows the MySQL Workbench interface with the 'Local instance MYSQL57' selected. In the Navigator pane, the 'Schemas' section is expanded, showing the 'etaireia_aksiologisis' schema. Under this schema, the 'Tables' section is expanded, and the 'applies' table is selected. The main query editor window displays the following SQL query:

```
1 • SELECT * FROM etaireia_aksiologisis.applies;
```

The Result Grid shows the following data:

cand_username	job_id	application_date	state
anastasis9090	1	2023-01-05	active
ava_harrison86	3	2023-09-05	active
daniel_parker88	6	2023-12-01	active
emma_ross	6	2023-06-20	active
james_mitchell71	4	2023-10-14	active
lily_wilson78	1	2023-07-02	active
manos1978	5	2023-05-12	active
michalis123	3	2023-03-25	active
nicnic	4	2023-04-10	active

Πίνακας *applies* πριν την κλήση της *applicationManagement*.

```

USE etaireia_aksiologisis;
CALL applicationManagement ('anastasis9090', 1 , 'c');

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

#	Time	Action	Message	Duration / Fetch
361	20:55:23	INSERT INTO subject (title, descr, belongs_to) VALUES ('Programming Fundamentals', 'Introduction to prog...', 24 row(s) affected Records: 24 Duplicates: 0 Warnings: 0)		0.000 sec
362	20:55:23	INSERT INTO requires (job_id, subject_title) VALUES (1, 'Programming Fundamentals'), (2, 'Data Analysis'), ... 24 row(s) affected Records: 24 Duplicates: 0 Warnings: 0		0.015 sec
363	20:55:23	INSERT INTO evaluation (evaluator1, evaluator2, evaluated_user, grade1, grade2) VALUES ('maria123', 'gi... 18 row(s) affected Records: 18 Duplicates: 0 Warnings: 0		0.032 sec
364	20:55:23	INSERT INTO user (username, password, name, lastname, reg_date, email) VALUES ('jimmynew', 'wifunds...', 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0		0.000 sec
365	20:55:23	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL), ... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0		0.000 sec
366	20:55:23	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected	0.000 sec
367	20:55:23	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ... 0 row(s) affected		0.000 sec
368	20:55:23	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom... 1 row(s) affected		224.641 sec

Κλήση της applicationManagement.

```

SELECT * FROM etaireia_aksiologisis.applies;

```

cand_username	job_id	application_date	state
anastasis9090	1	2023-01-05	canceled
ava_harrison86	3	2023-09-05	active
daniel_parker88	6	2023-12-01	active
emma_ross	6	2023-06-20	active
james_mitchell71	4	2023-10-14	active
lily_wilson78	1	2023-07-02	active
manos1978	5	2023-05-12	active
michalis123	3	2023-03-25	active
nicnic	4	2023-04-10	active

Πίνακας applies μετά την κλήση της applicationManagement.

Operation 'a':

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the 'SCHEMAS' section with 'giraffe', 'sys', and 'test' listed.
- Query Editor:** Displays the SQL query: `SELECT * FROM etaireia_aksiologisis.applies;`
- Result Grid:** Shows the data from the 'applies' table:

cand_username	job_id	application_date	state
anastasis9090	1	2023-01-05	canceled
ava_harrison86	3	2023-09-05	active
daniel_parker88	6	2023-12-01	active
emma_ross	6	2023-06-20	active
james_mitchell71	4	2023-10-14	active
lily_wilson78	1	2023-07-02	active
manos1978	5	2023-05-12	active
michalis123	3	2023-03-25	active
nicnic	4	2023-04-10	active

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the 'SCHEMAS' section with 'etaireia_aksiologisis' expanded, showing tables like 'active_admin', 'administrator', 'administrator_log', 'application_log', and 'applies'.
- Query Editor:** Displays the SQL code for the stored procedure:


```

1 USE etaireia_aksiologisis;
2
3 CALL applicationManagement ('anastasis9090', 1 , 'a');
4
5
6
7
8

```
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
501	21:05:48	INSERT INTO user (username, password, name, lastname, reg_date, email) VALUES ('jimmynew', 'wfundsc...', 'jimmynew', 'jimmynew', '2023-01-15', 'NULL')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
502	21:05:48	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', 'NULL')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.015 sec
503	21:05:49	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected	0.000 sec
504	21:05:49	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ...	0 row(s) affected	0.000 sec
505	21:05:49	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom...	1 row(s) affected	201.687 sec
506	21:09:27	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
507	21:09:27	CALL applicationManagement ('anastasis9090', 1 , 'c')	Error Code: 1644. The application has been canceled	0.031 sec
508	21:09:41	SELECT * FROM etaireia_aksiologisis.applies LIMIT 0, 1000	18 row(s) returned	0.000 sec / 0.000 sec

Επάνω: πίνακας applies πριν την κλήση της applicationManagement και Κάτω: στιγμιότυπο κλήσης της applicationManagement.

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the 'SCHEMAS' section with 'etaireia_aksiologisis' expanded, showing tables like 'active_admin', 'administrator', 'administrator_log', 'application_log', and 'applies'.
- Query Editor:** Displays the SQL query: `SELECT * FROM etaireia_aksiologisis.applies;`
- Result Grid:** Shows the data from the 'applies' table, identical to the previous screenshot:

cand_username	job_id	application_date	state
anastasis9090	1	2023-01-05	active
ava_harrison86	3	2023-09-05	active
daniel_parker88	6	2023-12-01	active
emma_ross	6	2023-06-20	active
james_mitchell71	4	2023-10-14	active
lily_wilson78	1	2023-07-02	active
manos1978	5	2023-05-12	active
michalis123	3	2023-03-25	active
nicnic	4	2023-04-10	active

Πίνακας applies μετά την κλήση της applicationManagement.

3.1.3.3.)

Ακολουθεί ο κώδικας:

```
DELIMITER $  
DROP PROCEDURE IF EXISTS checkPositionOccupied$  
CREATE PROCEDURE checkPositionOccupied(IN job_id VARCHAR(30))  
BEGIN  
CALL positionEvaluation(job_id);  
SELECT emp_username, jobid  
FROM has_position  
WHERE jobid=job_id;  
END$  
DELIMITER ;
```

Παράδειγμα 3.1.3.3

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, with 'etaireia_aksiologisis' selected. Under 'Tables', the 'has_position' table is listed. The main pane shows a query editor with the following SQL statement:

```
1 • SELECT * FROM etaireia_aksiologisis.has_position;
```

The result grid displays the following data:

emp_username	jobid
NULL	NULL

Κενός πίνακας `has_position` πριν την κλήση της `checkPositionOccupied`.

```

USE etaireia_aksiologisis;
CALL positionEvaluation(1);
CALL checkPositionOccupied(1);

```

emp_username	jobid
user7	1

Action Output

#	Time	Action	Message	Duration / Fetch
129	22:04:38	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL);	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.016 sec
130	22:04:38	DROP PROCEDURE IF EXISTS generateRandomRecords;	0 row(s) affected	0.000 sec
131	22:04:38	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ... 0 row(s) affected	0.000 sec	
132	22:04:38	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom... 1 row(s) affected	199.485 sec	
133	22:08:08	SELECT * FROM etaireia_aksiologisis.has_position LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
134	22:08:28	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
135	22:08:28	CALL positionEvaluation(1)	1 row(s) affected	0.063 sec
136	22:08:29	CALL checkPositionOccupied(1)	1 row(s) returned	0.015 sec / 0.000 sec

Κλήση της `checkPositionOccupied` και εμφάνιση αποτελέσματος πλήρωσης θέσης.

```

SELECT * FROM etaireia_aksiologisis.has_position;

```

emp_username	jobid
user7	1
NULL	NULL

Πίνακας `has_position` μετά την κλήση της `checkPositionOccupied`.

3.1.3.4α.)

Ακολουθεί ο κώδικας:

```

DELIMITER $
CREATE INDEX idx_evaluator ON application_log(e_evaluator1, e_evaluator2)$

DROP PROCEDURE IF EXISTS searchByEval$

```

```

CREATE PROCEDURE searchByEval(IN eval_username VARCHAR(30))
BEGIN
    SELECT e_username, positionID
    FROM application_log
    WHERE e_evaluator1=eval_username OR e_evaluator2=eval_username;
END$
```

DELIMITER ;

Παράδειγμα 3.1.3.4α

Χωρίς INDEX: Duration=0.016 sec

The screenshot shows the MySQL Workbench interface. In the top-left, there's a 'Navigator' pane showing databases: 'giraffe', 'sys', and 'test'. The main area has a 'Query' tab open with the following SQL code:

```

USE etaireia_aksiologisis;
CALL searchByGradeRange(7, 10);
```

The results grid below shows 12 rows of data:

e_username	positionID	finalGrade
user_10006	7	7
user_10007	8	8
user_10008	9	9
user_10009	10	10
user_10026	27	7
user_10027	28	8
user_10028	29	9
user_10029	30	10
user_10046	47	7

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
445	22:48:17	INSERT INTO evaluation (evaluator1, evaluator2, evaluated_user, grade1, grade2) VALUES ('maria123', 'gi...', '18 row(s) affected Records: 18 Duplicates: 0 Warnings: 0')	0.016 sec	
446	22:48:17	INSERT INTO user (username, password, name, lastname, reg_date, email) VALUES ('jimmynew', 'wfundsc...', '2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0')	0.000 sec	
447	22:48:17	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL)...	0.016 sec	
448	22:48:17	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected	
449	22:48:17	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ... 0 row(s) affected	0.000 sec	
450	22:48:17	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom... 1 row(s) affected	178.921 sec	
451	22:51:27	USE etaireia_aksiologisis 0 row(s) affected	0.000 sec	
452	22:51:27	CALL searchByGradeRange(7, 10) 12000 row(s) returned	0.016 sec / 0.156 sec	

Με INDEX: Duration=0.000 sec

MySQL Workbench - Local instance MYSQL57

Schemas: etaireia_aksiologisis (Tables, Views, Stored Procedures, Functions), giraffe, sys, test

Query Editor:

```
1 USE etaireia_aksiologisis;
2
3 • CALL searchByGradeRange(7, 10);
```

Result Grid:

e_username	positionID	finalGrade
user_10006	7	7
user_10026	27	7
user_10046	47	7
user_1006	7	7
user_10066	67	7
user_10086	87	7
user_10106	107	7
user_10126	127	7
user_10146	147	7

Output:

#	Time	Action	Message	Duration / Fetch
511	22:56:59	INSERT INTO evaluation (evaluator1, evaluator2, evaluated_user, grade1, grade2) VALUES ('maria123', 'gi...', 18 row(s) affected Records: 18 Duplicates: 0 Warnings: 0)		0.016 sec
512	22:56:59	INSERT INTO user (username, password, name, lastname, reg_date, email) VALUES ('jimmynew', 'wfundsc...', 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0)		0.000 sec
513	22:56:59	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL)...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.015 sec
514	22:56:59	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected	0.000 sec
515	22:56:59	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ... 0 row(s) affected	1 row(s) affected	0.000 sec
516	22:56:59	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom...	1 row(s) affected	183.140 sec
517	23:00:07	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
518	23:00:07	CALL searchByGradeRange(7, 10)	12000 row(s) returned	0.000 sec / 0.016 sec

3.1.3.4β.)

Ακολουθεί ο κώδικας:

```
DELIMITER $

CREATE INDEX idx_empljob ON application_log(finalGrade)$

DROP PROCEDURE IF EXISTS searchByGradeRange$
CREATE PROCEDURE searchByGradeRange(IN low INT(11), IN high INT(11))
BEGIN
SELECT application_log.e_username, application_log.positionID, application_log.finalGrade
FROM application_log
WHERE application_log.finalGrade BETWEEN low AND high;
END$

DELIMITER ;
```

Παράδειγμα 3.1.3.4β

Χωρίς INDEX: Duration=0.157 sec

MySQL Workbench

Local instance MYSQL57

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Tables

application_log

Columns:

- e_username varchar(30) PK
- e_evaluator1 varchar(30)
- e_evaluator2 varchar(30)
- positionID int(11) PK
- a_state enum('active','com')
- finalGrade int(11)

Stored Procedures

Administration Schemas

Information

Table: application_log

Columns:

- e_username varchar(30) PK
- e_evaluator1 varchar(30)
- e_evaluator2 varchar(30)
- positionID int(11) PK
- a_state enum('active','com')
- finalGrade int(11)

Result Grid

e_username	positionID
user_10897	398

Result 14 x

Action Output

#	Time	Action	Message	Duration / Fetch
448	22:48:17	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected	0.000 sec
449	22:48:17	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ...	0 row(s) affected	0.000 sec
450	22:48:17	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom...	1 row(s) affected	178.921 sec
451	22:51:27	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
452	22:51:27	CALL searchByGradeRange(7, 10)	12000 row(s) returned	0.016 sec / 0.156 sec
453	22:52:13	SELECT * FROM etaireia_aksiologisis.application_log LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
454	22:52:42	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
455	22:52:42	CALL searchByEval('evaluator1_10897')	1 row(s) returned	0.157 sec / 0.000 sec

Result Only Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Mε INDEX: Duration=0.063 sec

MySQL Workbench

Local instance MYSQL57

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Tables

application_log

Columns:

- e_username varchar(30) PK
- e_evaluator1 varchar(30)
- e_evaluator2 varchar(30)
- positionID int(11) PK
- a_state enum('active','com')
- finalGrade int(11)

Indexes

Foreign Keys

Triggers

Columns

- e_username
- e_evaluator1
- e_evaluator2
- positionID
- a_state
- finalGrade

application_log

Result Grid

e_username	positionID
user_10897	398

Result 11 x

Action Output

#	Time	Action	Message	Duration / Fetch
382	22:43:28	CALL searchByEval('user4')	0 row(s) returned	0.078 sec / 0.000 sec
383	22:44:39	SELECT * FROM etaireia_aksiologisis.application_log LIMIT 0, 1000	1000 row(s) returned	0.016 sec / 0.000 sec
384	22:45:09	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
385	22:45:09	CALL searchByEval('evaluator1_0')	0 row(s) returned	0.078 sec / 0.000 sec
386	22:45:31	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
387	22:45:31	CALL searchByEval('evaluator1_0')	1 row(s) returned	0.062 sec / 0.000 sec
388	22:46:31	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
389	22:46:31	CALL searchByEval('evaluator1_10897')	1 row(s) returned	0.063 sec / 0.000 sec

Result Only Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

ΚΕΦΑΛΑΙΟ 3

Δημιουργία Trigger

Παρακάτω παρουσιάζονται ο κώδικας για το κάθε Trigger καθώς και screenshots από παραδείγματα εκτέλεσής τους.

3.1.4.1)

Ακολουθεί ο κώδικας:

#Δημιουργία νέου πίνακα active_admin για την αποθήκευση του ενεργού administrator μετά από Login στη θάση

```
CREATE TABLE active_admin(
    username VARCHAR(30) NOT NULL,
    PRIMARY KEY(username)
);
```

#Κώδικας Trigger

```
DELIMITER $  
DROP TRIGGER IF EXISTS job_insert$  
CREATE TRIGGER job_insert AFTER INSERT ON job  
FOR EACH ROW  
BEGIN  
DECLARE job_id INT(11);  
DECLARE current_datetime DATETIME;  
DECLARE active_admin VARCHAR(30);  
DECLARE event TEXT(256);  
SELECT MAX(job.id) INTO job_id FROM job;  
SET current_datetime = NOW();  
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;  
SET event = CONCAT('The job with id ', job_id, ' has just been inserted by the administrator ', active_admin);  
INSERT INTO administrator_log VALUES (active_admin, current_datetime, event);  
END$  
DELIMITER;
```

```
DELIMITER $  
DROP TRIGGER IF EXISTS job_delete$  
CREATE TRIGGER job_delete BEFORE DELETE ON job  
FOR EACH ROW  
BEGIN  
DECLARE job_id INT(11);  
DECLARE current_datetime DATETIME ;  
DECLARE active_admin VARCHAR(30);  
DECLARE event TEXT(256);
```

```

SET current_datetime = NOW();
SELECT id INTO job_id FROM job
WHERE id = OLD.id;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The job with id ', job_id, ' has just been deleted by the administrator ', active_admin);
INSERT INTO administrator_log VALUES (active_admin, current_datetime, event);
END$  

DELIMITER;  
  

DELIMITER $  

DROP TRIGGER IF EXISTS job_update$  

CREATE TRIGGER job_update BEFORE UPDATE ON job
FOR EACH ROW
BEGIN
DECLARE job_id INT(11);
DECLARE current_datetime DATETIME ;
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);
SELECT id INTO job_id FROM job
WHERE id = OLD.id;
SET current_datetime = NOW();
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The job with id ', job_id, ' has just been updated by the administrator ', active_admin);
INSERT INTO administrator_log VALUES (active_admin, current_datetime, event);
END$  

DELIMITER;  
  

DELIMITER $  

DROP TRIGGER IF EXISTS degree_insert$  

CREATE TRIGGER degree_insert AFTER INSERT ON degree
FOR EACH ROW
BEGIN
DECLARE d_titlos VARCHAR(150);
DECLARE d_idryma VARCHAR (140);
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);
SELECT titlos, idryma INTO d_titlos, d_idryma FROM degree
WHERE titlos = NEW.titlos AND idryma = NEW.idryma;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The degree with title ', d_titlos, 'and idryma ', d_idryma, ' has just been inserted by the administrator ', active_admin);
INSERT INTO administrator_log VALUES (active_admin, NOW(), event);
END$  

DELIMITER;  
  

DELIMITER $  

DROP TRIGGER IF EXISTS degree_update$  

CREATE TRIGGER degree_update BEFORE UPDATE ON degree
FOR EACH ROW

```

```

BEGIN

DECLARE d_titlos VARCHAR(150);
DECLARE d_idryma VARCHAR (140);
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);

SELECT titlos, idryma INTO d_titlos, d_idryma FROM degree
WHERE titlos = OLD.titlos AND idryma = OLD.idryma;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The degree with title ', d_titlos, 'and idryma ', d_idryma, ' has just been updated by the administrator ',
active_admin);
INSERT INTO administrator_log VALUES (active_admin, NOW(), event);
END$

DELIMITER;

DELIMITER $

DROP TRIGGER IF EXISTS degree_delete$
CREATE TRIGGER degree_delete BEFORE DELETE ON degree
FOR EACH ROW
BEGIN
DECLARE d_titlos VARCHAR(150);
DECLARE d_idryma VARCHAR (140);
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);

SELECT titlos, idryma INTO d_titlos, d_idryma FROM degree
WHERE titlos = OLD.titlos AND idryma = OLD.idryma;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The degree with title ', d_titlos, 'and idryma ', d_idryma, ' has just been deleted by the administrator ',
active_admin);
INSERT INTO administrator_log VALUES (active_admin, NOW(), event);
END$

DELIMITER;

DELIMITER $

DROP TRIGGER IF EXISTS user_insert$
CREATE TRIGGER user_insert AFTER INSERT ON user
FOR EACH ROW
BEGIN
DECLARE t_username VARCHAR(30);
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);

SELECT username INTO t_username FROM user
WHERE username = NEW.username;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The user with username ', t_username, ' has just been inserted by the administrator ', active_admin);
INSERT INTO administrator_log VALUES (active_admin, NOW(), event);
END$

DELIMITER;

DELIMITER $

DROP TRIGGER IF EXISTS user_update$
CREATE TRIGGER user_update BEFORE UPDATE ON user

```

```

FOR EACH ROW
BEGIN
DECLARE t_username VARCHAR(30);
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);
SELECT username INTO t_username FROM user
WHERE username = OLD.username;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The user with username ', t_username, ' has just been updated by the administrator ', active_admin);
INSERT INTO administrator_log VALUES (active_admin, NOW(), event);
END$

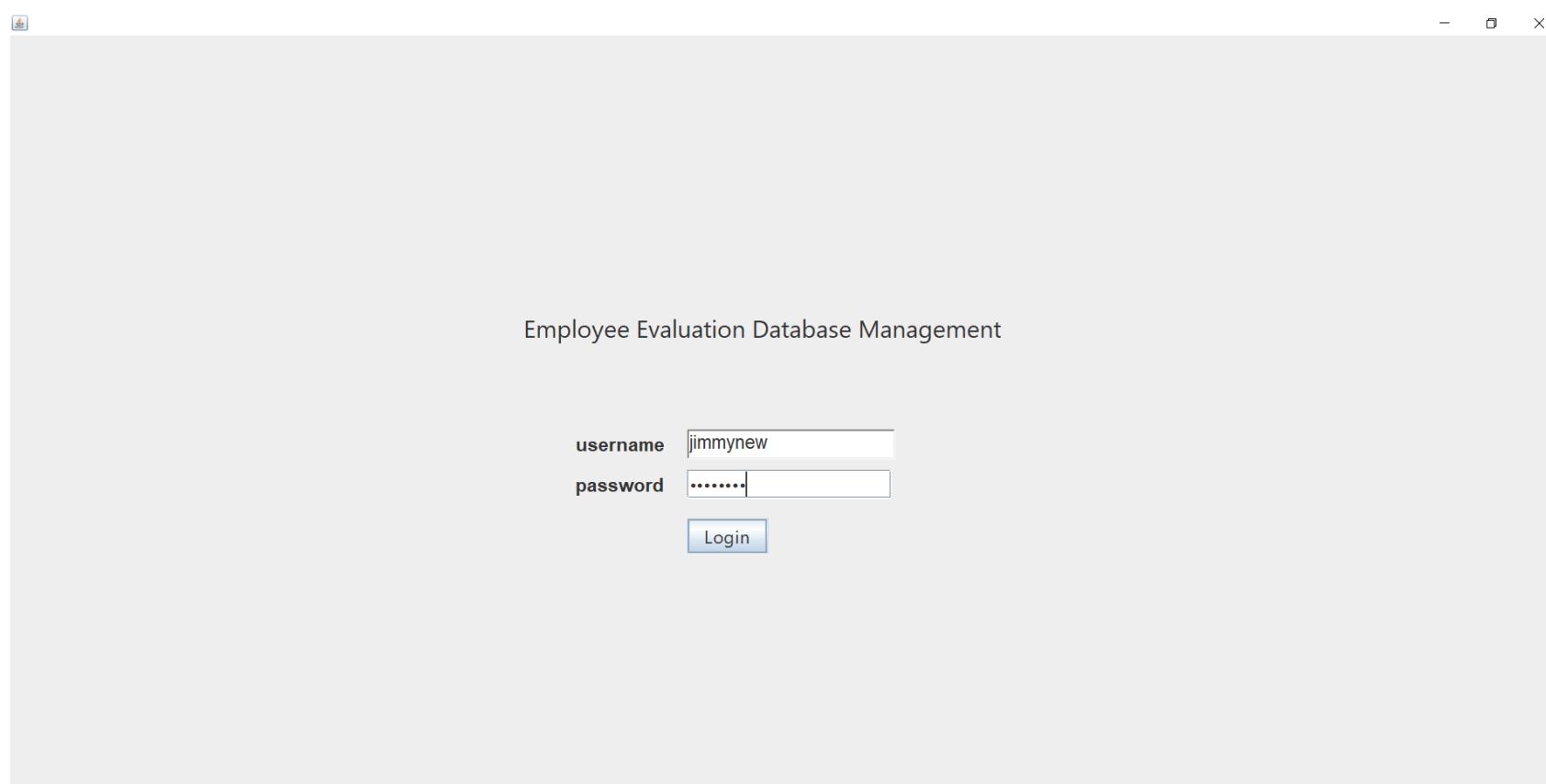
DELIMITER;

DELIMITER $
DROP TRIGGER IF EXISTS user_delete$
CREATE TRIGGER user_delete BEFORE DELETE ON user
FOR EACH ROW
BEGIN
DECLARE t_username VARCHAR(30);
DECLARE active_admin VARCHAR(30);
DECLARE event TEXT(256);
SELECT username INTO t_username FROM user
WHERE username = OLD.username;
SELECT active_admin.username INTO active_admin FROM active_admin LIMIT 1;
SET event = CONCAT('The user with username ', t_username, ' has just been deleted by the administrator ', active_admin);
INSERT INTO administrator_log VALUES (active_admin, NOW(), event);
END$

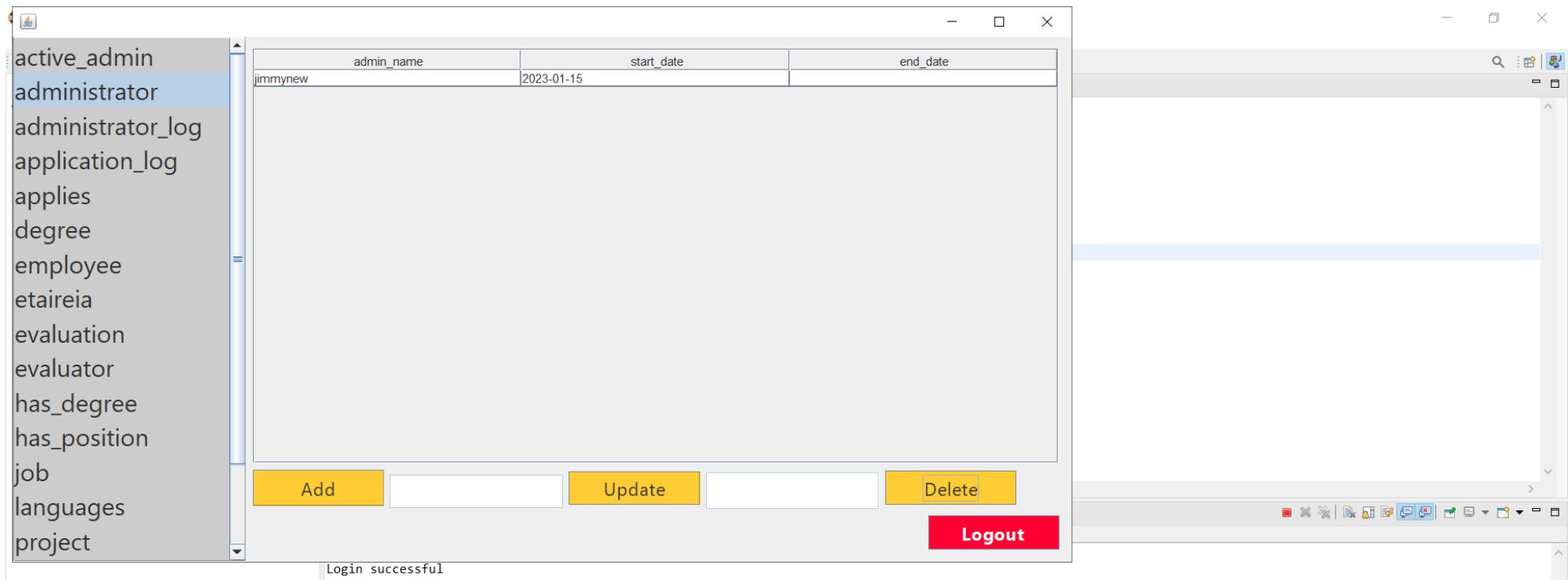
DELIMITER ;

```

Παράδειγμα 3.1.4.1



GUI Login.



Περιεχόμενα πίνακα administrator όπου φαίνεται ο ενεργός administrator.

```

USE etaireia_aksiologisis;
INSERT INTO degree VALUES ('Bachelor of Algorithm Engineering', 'Oxford University', 'BSc');
INSERT INTO job VALUES (37, '2023-01-16', 7500, 'Data Analyst', 'Athens', 'maria123', '2023-02-01 10:45:01', '2023-02-28');
INSERT INTO user VALUES ('johndaaag34374', 'ggtrg5', 'John', 'Daglas', '2023-07-07 09:20:12', 'johndaaaglas@gmail.com');
DELETE FROM degree WHERE idryma='University of Thessaloniki';
DELETE FROM job WHERE edra='Global City';
DELETE FROM user WHERE name='Ethan';
UPDATE degree SET idryma='Oxford University' WHERE titlos='PhD of Computer Architecture';
UPDATE job SET salary>100000 WHERE position='Project Manager';
UPDATE user SET password='bduib34u24' WHERE username='emma_ross';
    
```

Action	Time	Message	Duration / Fetch
781	14:23:02	INSERT INTO subject (title, descr, belongs_to) VALUES ('Programming Fundamentals', 'Introduction to prog...', 24 row(s) affected Records: 24 Duplicates: 0 Warnings: 0)	0.000 sec
782	14:23:02	INSERT INTO requires (job_id, subject_title) VALUES (1, 'Programming Fundamentals'), (2, 'Data Analysis'), ... 24 row(s) affected Records: 24 Duplicates: 0 Warnings: 0	0.000 sec
783	14:23:02	INSERT INTO evaluation (evaluator1, evaluator2, evaluated_user, grade1, grade2) VALUES ('maria123', 'gi...', 18 row(s) affected Records: 18 Duplicates: 0 Warnings: 0)	0.016 sec
784	14:23:02	INSERT INTO user (username, password, name, lastname, reg_date, email) VALUES ('jimmynew', 'wfundsc...', 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0)	0.000 sec
785	14:23:02	INSERT INTO administrator (admin_name, start_date, end_date) VALUES ('jimmynew', '2023-01-15', NULL), ... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
786	14:23:02	DROP PROCEDURE IF EXISTS generateRandomRecords	0 row(s) affected
787	14:23:02	CREATE PROCEDURE generateRandomRecords() BEGIN DECLARE i INT; SET i = 0; WHILE i < ... 0 row(s) affected	0.000 sec
788	14:23:02	#Edit->Preferences->SQL Editor (all the timers in MySQL session to be set to 3600) CALL generateRandom... Running...	?

Εντολές INSERT, DELETE, UPDATE για τους πίνακες degree, job και user.

MySQL Workbench

Local instance MYSQL57

File Edit View Query Database Server Tools Scripting Help

Navigator: create_DB_and_tables (1) stored_procedures (1) initialize_DB (2) triggers (1) test administrator_log (1)

SCHEMAS: etaireia_aksiologisis

Tables: active_admin, administrator, administrator_log, application_log, applies, degree, employee, etaireia, evaluation, evaluator, has_degree, has_position, job, languages, project, requires, subject, user

Views, Stored Procedures

Administrator Schemas

Information: administrator_log1

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

administrator	execucion_time	event_type
jimmynew	2024-01-12 15:39:36	The degree with title Bachelor of Algorithm Engineeringand idryma Oxford University has just been inserted by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The degree with title Bachelor of Business Administrationand idryma University of Thessaloniki has just been deleted by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The degree with title PhD of Computer Architectureand idryma Harvard University has just been updated by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The job with id 12 has just been updated by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The job with id 3 has just been updated by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The job with id 37 has just been inserted by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The job with id 5 has just been deleted by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The user with username emma_ross has just been updated by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The user with username ethan_miller64 has just been deleted by the administrator jimmynew
jimmynew	2024-01-12 15:39:36	The user with username johndaaag34374 has just been inserted by the administrator jimmynew
*	NULL	NULL

active_admin
administrator
administrator_log
application_log
applies
degree
employee
etaireia
evaluation
evaluator
has_degree
has_position
job
languages
project

Add Update Delete Logout

Πίνακας administrator_log μετά την εκτέλεση των εντολών στο MySQL Workbench (επάνω) και στο GUI (κάτω).

3.1.4.2)

Ακολουθεί ο κώδικας:

```
DELIMITER $  
  
DROP TRIGGER IF EXISTS validateApplicationDate$  
  
CREATE TRIGGER validateApplicationDate  
AFTER INSERT ON applies  
FOR EACH ROW  
BEGIN  
  
    DECLARE applicationDate DATE;  
    DECLARE startDate DATE;  
    DECLARE diff INT;  
  
    SELECT job.start_date INTO startDate  
    FROM job  
    WHERE job.id = NEW.job_id;  
  
    SELECT application_date INTO applicationDate  
    FROM applies  
    WHERE applies.application_date=NEW.application_date;  
  
    SET diff = DATEDIFF(startDate, applicationDate);  
  
    IF (diff<15) THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Invalid application date! Must be at least 15 days before the start date of the job';  
    END IF;  
END$  
  
DELIMITER ;
```

```
DELIMITER $  
  
DROP TRIGGER IF EXISTS validateApplicationInsert$  
  
CREATE TRIGGER validateApplicationInsert  
BEFORE INSERT ON applies  
FOR EACH ROW  
BEGIN  
  
    DECLARE numOfActive INT;  
    SELECT COUNT(*) INTO numOfActive  
    FROM applies  
    WHERE applies.cand_username=NEW.cand_username  
    AND state='active';  
    IF numOfActive>=3 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Invalid application. The employee already has at least 3 active applications';  
    END IF;  
END$  
  
DELIMITER ;
```

Παράδειγμα 3.1.4.2

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** etaireia_aksiologisis (selected), containing tables like active_admin, administrator, administrator_log, application_log, applies, degree, employee, etaireia, evaluation, evaluator, has_degree, has_position, job, languages, project, requires, subject, user.
- Query Editor:** A tab labeled "test" is selected. The SQL code is:


```
1 • USE etaireia_aksiologisis;
2
3 • INSERT INTO applies VALUES('anastasis9090', 2, '2023-02-20', 'active');
4
```
- Output Window:** Shows the execution log with the last entry being:

Action	Time	Message	Duration / Fetch
271 17:03:47	INSERT INTO applies VALUES('anastasis9090', 2, '2023-02-20', 'active')	Error Code: 1644. Invalid application date! Must be at least 15 days before the start date of the job	0.015 sec

Παράδειγμα εισαγωγής ημερομηνίας αίτησης η οποία δεν πληρεί τις προϋποθέσεις (εδώ ταυτίζεται η application_date με την start_date).

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** etaireia_aksiologisis (selected), containing tables like active_admin, administrator, administrator_log, application_log, applies, degree, employee, etaireia, evaluation, evaluator, has_degree, has_position, job, languages, project, requires, subject, user.
- Query Editor:** A tab labeled "test" is selected. The SQL code is:


```
1 • USE etaireia_aksiologisis;
2
3 • INSERT INTO applies VALUES('anastasis9090', 2, '2023-02-05', 'active');
4 • INSERT INTO applies VALUES('anastasis9090', 3, '2023-02-13', 'active');
5 • INSERT INTO applies VALUES('anastasis9090', 4, '2023-04-10', 'active');
```
- Output Window:** Shows the execution log with the last entry being:

Action	Time	Message	Duration / Fetch
389 17:10:27	INSERT INTO applies VALUES('anastasis9090', 4, '2023-04-10', 'active')	Error Code: 1644. Invalid application. The employee already has at least 3 active applications	0.000 sec

Παράδειγμα όπου ο user έχει ήδη 3 ενεργές αιτήσεις.

3.1.4.3)

Ακολουθεί ο κώδικας:

```
DELIMITER $  
DROP TRIGGER IF EXISTS validateApplicationDateCancel$  
CREATE TRIGGER validateApplicationDateCancel  
BEFORE UPDATE ON applies  
FOR EACH ROW  
BEGIN  
    DECLARE applicationDate DATE;  
    DECLARE startDate DATE;  
    DECLARE diff INT;  
  
    SELECT start_date INTO startDate  
    FROM job  
    WHERE id = NEW.job_id;  
  
    SELECT application_date INTO applicationDate  
    FROM applies  
    WHERE applies.application_date=NEW.application_date;  
  
    SET diff = DATEDIFF(startDate, applicationDate);  
  
    IF diff < 10 AND NEW.state='canceled' THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Invalid application cancel! Must be within 10 days before the start date of the job';  
    END IF;  
END$  
DELIMITER ;  
  
DELIMITER $  
DROP TRIGGER IF EXISTS validateApplicationUpdate$  
CREATE TRIGGER validateApplicationUpdate  
BEFORE UPDATE ON applies  
FOR EACH ROW  
BEGIN  
    DECLARE numOfActive INT;  
    SELECT COUNT(*) INTO numOfActive  
    FROM applies  
    WHERE applies.cand_username=NEW.cand_username  
    AND state='active';  
    IF numOfActive>=3 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Invalid application. The employee already has at least 3 active applications';  
    END IF;  
END$
```

DELIMITER ;

Παράδειγμα 3.1.4.3

```
USE etaireia_aksiologisis;
UPDATE applies SET applies.state = 'canceled' WHERE applies.cand_usename = 'daniel_parker88';
```

Table: job

Columns:

id	int(11)
start_date	date
salary	float
position	varchar(64)
edra	varchar(64)
evaluator	varchar(32)
announce_date	datetime
submission_date	date

Action Output

#	Time	Action	Message	Duration / Fetch
420	17:19:43	DROP TRIGGER IF EXISTS user_insert	0 row(s) affected	0.015 sec
421	17:19:43	CREATE TRIGGER user_insert AFTER INSERT ON user FOR EACH ROW BEGIN DECLARE t_usename...	0 row(s) affected	0.031 sec
422	17:19:43	DROP TRIGGER IF EXISTS user_update	0 row(s) affected	0.032 sec
423	17:19:43	CREATE TRIGGER user_update BEFORE UPDATE ON user FOR EACH ROW BEGIN DECLARE t_usename...	0 row(s) affected	0.031 sec
424	17:19:43	DROP TRIGGER IF EXISTS user_delete	0 row(s) affected	0.031 sec
425	17:19:43	CREATE TRIGGER user_delete BEFORE DELETE ON user FOR EACH ROW BEGIN DECLARE t_usename...	0 row(s) affected	0.031 sec
426	17:19:46	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
427	17:19:46	UPDATE applies SET applies.state = 'canceled' WHERE applies.cand_usename = 'daniel_parker88'	Error Code: 1644. Invalid application cancel! Must be within 10 days before the start date of the job	0.016 sec

Παράδειγμα λανθασμένης ακύρωσης αίτησης.

```
USE etaireia_aksiologisis;
INSERT INTO applies VALUES('anastasis9090', 2, '2023-02-05', 'active');
INSERT INTO applies VALUES('anastasis9090', 3, '2023-02-13', 'active');
UPDATE applies SET applies.state = 'canceled' WHERE applies.cand_usename = 'anastasis9090';
```

Table: applies

Columns:

cand_usename	varchar(30) PK
job_id	int(11) PK
application_date	date
state	enum('active','canceled')

Action Output

#	Time	Action	Message	Duration / Fetch
425	17:19:43	CREATE TRIGGER user_delete BEFORE DELETE ON user FOR EACH ROW BEGIN DECLARE t_usename...	0 row(s) affected	0.031 sec
426	17:19:46	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
427	17:19:46	UPDATE applies SET applies.state = 'canceled' WHERE applies.cand_usename = 'daniel_parker88'	Error Code: 1644. Invalid application cancel! Must be within 10 days before the start date of the job	0.016 sec
428	17:20:42	SELECT * FROM etaireia_aksiologisis.applies LIMIT 0, 1000	18 row(s) returned	0.000 sec / 0.000 sec
429	17:23:46	USE etaireia_aksiologisis	0 row(s) affected	0.000 sec
430	17:23:46	INSERT INTO applies VALUES('anastasis9090', 2, '2023-02-05', 'active')	1 row(s) affected	0.016 sec
431	17:23:46	INSERT INTO applies VALUES('anastasis9090', 3, '2023-02-13', 'active')	1 row(s) affected	0.000 sec
432	17:23:46	UPDATE applies SET applies.state = 'canceled' WHERE applies.cand_usename = 'anastasis9090'	Error Code: 1644. Invalid application. The employee already has at least 3 active applications	0.000 sec

Παράδειγμα όπου δίνεται εντολή για επαναφορά ακυρωμένης αίτησης, αλλά ο χρήστης έχει ήδη 3 ενεργές αιτήσεις.

ΚΕΦΑΛΑΙΟ 4

Τεκμηρίωση και κώδικας σε Java

Η εφαρμογή μας αποτελείται από 2 διεπαφές, την διεπαφή για το **Login screen**(JframeDB) και την **Κύρια Διεπαφή**(JframeDBMain).

Όσον αφορά τη δημιουργία του **Login screen** γίνεται χρήση:

- της κλάσης JFrame για τη δημιουργία ενός top-level window
- των κλάσεων Label και JTextField για τη δημιουργία των ετικετών "username" και "password" μαζί με τα αντίστοιχα πεδία εισαγωγής τιμών από τον user
- της κλάσης JLabel για τη δημιουργία του τίτλου του frame και ενός status_label που θα χρησιμοποιηθεί αργότερα
- της κλάσης JButton για τη δημιουργία του Login button.

- Ο constructor της κλάσης JframeDB αρχικοποεί την κλάση καλώντας την μέθοδο **initComponents()**, εντός της οποίας δημιουργούνται αντικείμενα των παραπάνω αναφερθέντων κλάσεων. Επομένως, με την δημιουργία ενός αντικειμένου της κλάσης JframeDB (αυτό γίνεται στην main συνάρτηση της κλάσης ProjectDB), δημιουργείται η διεπαφή του Login screen.

Για να προστεθεί λειτουργικότητα στο Login screen, ορίστηκαν εντός της κλάσης JframeDB οι μέθοδοι:

```
private void login_buttonActionPerformed(java.awt.event.ActionEvent evt)
private void insertIntoActiveAdmin(String username)
```

Η πρώτη μέθοδος καλείται με το πάτημα του Login button που έχω δημιουργήσει προηγουμένως εντός της initComponents.

Ελέγχει εάν το username και το password που εισήγαγε ο χρήστης υπάρχουν στον πίνακα user. Εάν δεν υπάρχουν τότε έχουμε failed login και γίνεται visible το status_label υποδεικνύοντας το μήνυμα "[Invalid username or password](#)".

Εάν υπάρχουν, ελέγχει εάν το username υπάρχει επίσης στον πίνακα administrator(ελέγχει δηλαδή εάν ο χρήστης είναι επίσης ένας administrator) και εάν η τιμή του πεδίου end_date είναι NULL. Εάν ο χρήστης είναι administrator και το end_date είναι NULL, τότε το username που εισήγαγε ο χρήστης γίνεται INSERT στον πίνακα active_admin με την κλήση της μεθόδου

insertIntoActiveAdmin(String username). Εάν όχι, τότε το status_label εμφανίζει το μήνυμα "[User is not an active administrator](#)". Τέλος, γίνεται launch η **Κύρια Διεπαφή** με τη δημιουργία ενός αντικειμένου της κλάσης JframeDBMain και dispose() το Login window.

Όσον αφορά τη δημιουργία της Κύριας Διεπαφής, γίνεται χρήση:

- των κλάσεων JButton, JTextField και JLabel για τη δημιουργία: των buttons Add, Update, Delete, Logout, δύο πεδίων εισαγωγής τιμών από τον χρήστη για τις λειτουργίες insert και update και, τέλος, ενός insert_label που θα χρησιμοποιηθεί αργότερα.
 - των κλάσεων JScrollPane, JList<> και Jtable για τη δημιουργία μίας λίστας με τα ονόματα των πινάκων της ΒΔ μας η οποία τοποθετείται εντός scrollable container στα αριστερά του παραθύρου, καθώς και τη δημιουργία ενός main πίνακα στα δεξιά του παραθύρου όπου θα προβάλλονται τα περιεχόμενα του κάθε πίνακα όταν αυτός θα επιλέγεται από την λίστα.
- Ο constructor της κλάσης JframeDBMain αρχικοποεί την κλάση καλώντας την μέθοδο **initComponents()**, εντός της οποίας δημιουργούνται αντικείμενα των παραπάνω αναφερθέντων κλάσεων, καθώς και τη μέθοδο **loadTableNames()**, η οποία φορτώνει τα ονόματα των πινάκων από τη ΒΔ πρώτα σε ένα list model και στη συνέχεια στη λίστα στα αριστερά του παραθύρου. Επιπλέον, εντός του constructor ορίζεται η default συμπεριφορά όταν πατιέται το κόκκινο X button του παραθύρου να είναι η διαγραφή του χρήστη από τον πίνακα active_admin και η αποσύνδεση από την ΒΔ. Αυτό επιτυγχάνεται με την κλήση της μεθόδου **cleanup()**, εντός της οποίας καλείται και η **deleteFromActiveAdmin()**.

Για να προστεθεί λειτουργικότητα στην Κύρια Διεπαφή, ορίστηκαν εντός της κλάσης JframeDBMain οι μέθοδοι:

```
private void logout_buttonActionPerformed(java.awt.event.ActionEvent evt)
private void table_listValueChanged(javax.swing.event.ListSelectionEvent evt)
private void main_tableMouseClicked(java.awt.event.MouseEvent evt)
private void insert_buttonActionPerformed(java.awt.event.ActionEvent evt)
private void text_dataActionPerformed(java.awt.event.ActionEvent evt)
private void update_buttonActionPerformed(java.awt.event.ActionEvent evt)
private void update_dataActionPerformed(java.awt.event.ActionEvent evt)
private void delete_buttonActionPerformed(java.awt.event.ActionEvent evt)

private List<String> getColumnsForTable(String tableName)
private List<List<Object>> getDataForTable(String tableName)
```

Καθώς και οι μεταβλητές:

```
private String tableName;                                // Table clicked on
private Object[] specificRowData;                      // Data for row clicked on
private Map<String, Object> selectedRowData = new HashMap<>(); // Mapping between column names and column values
```

Η μέθοδος **logout_buttonActionPerformed** καλείται με το πάτημα του button Logout. Καλεί την `deleteFromActiveAdmin()` και κάνει disconnect από την ΒΔ.

Η μέθοδος **table_listValueChanged()** ενεργοποιείται όταν ο χρήστης κάνει κλικ σε ένα πεδίο της λίστας με τα ονόματα των πινάκων στα αριστερά. Αρχικά αποθηκεύει το όνομα του πίνακα στον οποίο έγινε κλικ σε μια class variable `tableName`. Στη συνέχεια δημιουργεί ένα table model με στήλες τα ονόματα στηλών του πίνακα `tableName` (κλήση της μεθόδου `getColumnsForTable(String tableName)`) και αποθηκεύει σε αυτό το table model όλα τα δεδομένα του πίνακα (κλήση της μεθόδου `getDataForTable(String tableName)`). Τέλος, αφού καθαρίσει το main table στα δεξιά από όλα τα δεδομένα, φορτώνει σε αυτό το table model και ενεργοποιεί το Insert button.

Η μέθοδος **main_tableMouseClicked()** ενεργοποιείται αφού ο χρήστης έχει ήδη επιλέξει έναν πίνακα και σε αυτόν τον πίνακα έχει επιλέξει ένα συγκεκριμένο row. Ενεργοποιεί τα Update και Delete buttons, αποθηκεύει τα δεδομένα κάθε στήλης στο Object `specificRowData` και στη συνέχεια κάνει Mapping μεταξύ των ονομάτων των στηλών και των τιμών των στηλών και αποθηκεύει το αποτέλεσμα στο Object `selectedRowData`.

Η μέθοδος **insert_buttonActionPerformed()** ενεργοποιείται με το πάτημα του Add button και θέτει ενεργά το `insert_label` και το αντίστοιχο πεδίο εισαγωγής τιμών από τον χρήστη, ενώ θέτει ανενεργό το πεδίο εισαγωγής τιμών του Update button.

Η μέθοδος **text_dataActionPerformed()** ενεργοποιείται όταν ο χρήστης κάνει κλικ εντός του πεδίου εισαγωγής τιμών(Insert) και εισάγει τις τιμές που εισάγονται από τον χρήστη, στον αντίστοιχο επιλεγμένο πίνακα.

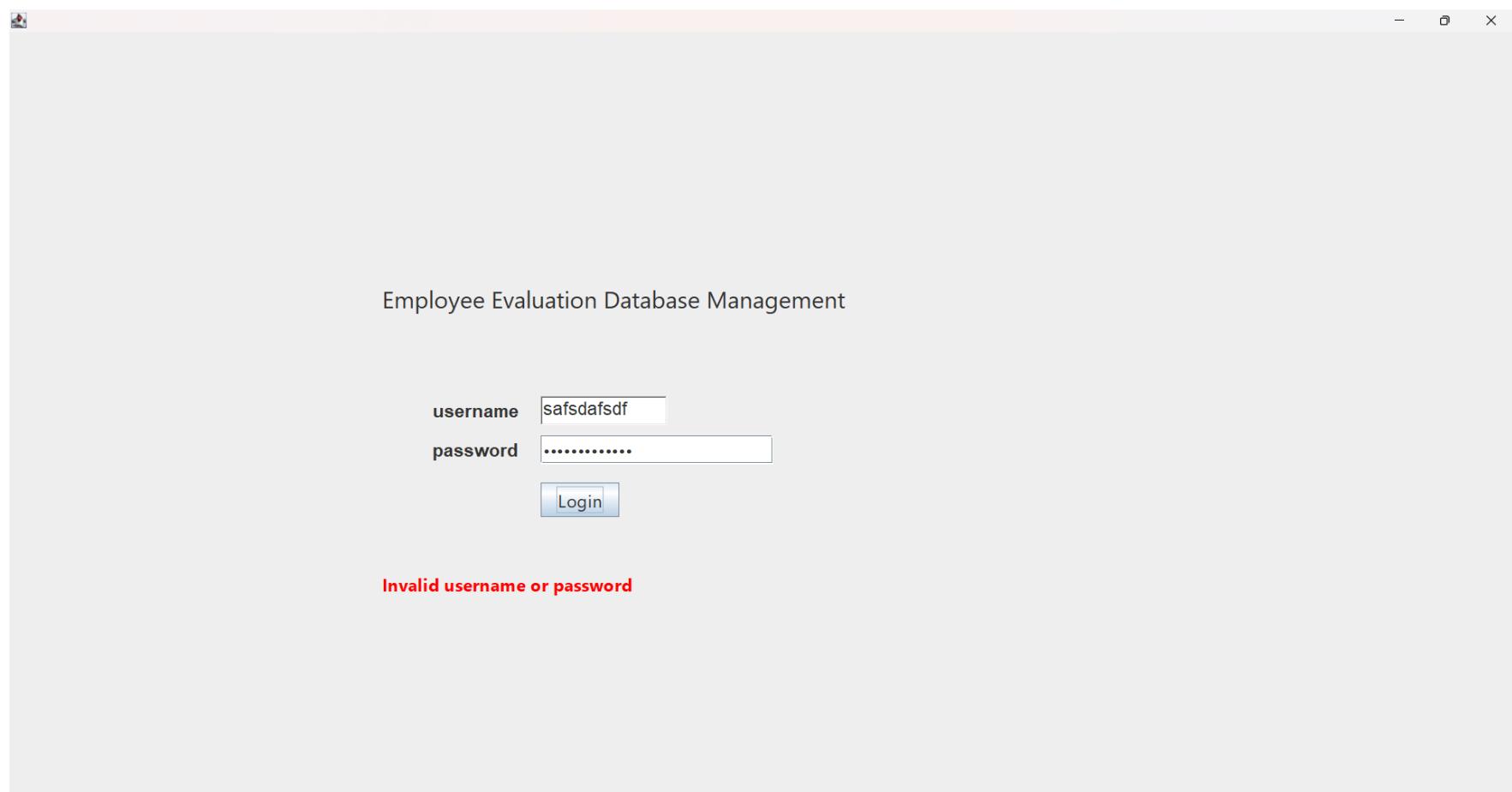
Η μέθοδος **update_buttonActionPerformed()** ενεργοποείται με το πάτημα του Update button και θέτει ενεργά το `insert_label` και το αντίστοιχο πεδίο εισαγωγής τιμών από τον χρήστη, ενώ θέτει ανενεργό το πεδίο εισαγωγής τιμών του Add button.

Η μέθοδος **update_dataActionPerformed()** ενεργοποιείται όταν ο χρήστης κάνει κλικ εντός του πεδίου εισαγωγής τιμών(Update) και εκτελεί update με βάση τις τιμές που εισάγονται από τον χρήστη, στον αντίστοιχο επιλεγμένο πίνακα.

Η μέθοδος **delete_buttonActionPerformed()** ελέγχει εάν υπάρχουν δεδομένα στο επιλεγμένο από τον χρήστη row του επίσης επιλεγμένου από τον χρήστη πίνακα και εάν ναι, συντάσσει εκτελεί την εντολή delete στη ΒΔ μας.

Σενάριο Χρήσης + Screenshots

Ακολουθούν screenshots στα οποία φαίνονται η σύνδεση του χρήστη στη ΒΔ και η εκτέλεση καθεμίας από τις πράξεις Insert, Update και Delete.



Failed Login example.

active_admin
administrator
administrator_log
application_log
applies
degree
employee
etaireia
evaluation
evaluator
has_degree
has_position
job
languages
project
requires
subject
user

titlos	idryma	bathmida
Bachelor of Business Administration	University of Thessaloniki	BSc
Bachelor of Computer Engineering	University of Patras	BSc
Bachelor of Computer Science	Athens University	BSc
Bachelor of Computer Science	University of Patras	BSc
Bachelor of Fine Arts	Aristotle University of Thessaloniki	BSc
Bachelor of Information Technology	University of Patras	BSc
Bachelor of Marketing	University of Piraeus	PhD
Bachelor of Mathematics	Oxford University	BSc
Bachelor of Physics	Harvard University	BSc
Bachelor of Physics	Warwick University	BSc
Bachelor of Technology Innovation and Strategy	Kapodistrian University of Athens	BSc
Bachelor of Telecommunications Engineering	University of Piraeus	BSc
Master of Business Administration	University of Patras	MSc
Master of Computer Engineering	University of Patras	MSc
Master of Data Science	University of Patras	MSc
Master of Data Science	University of Peloponnesse	MSc
Master of Network Security	Aristotle University of Thessaloniki	MSc
PhD of Computer Architecture	Harvard University	PhD

Add Update Delete Logout

Type data separated with commas and hit enter

Εισαγωγή τιμών από τον χρήστη μετά την επιλογή κάποιου table.

active_admin
administrator
administrator_log
application_log
applies
degree
employee
etaireia
evaluation
evaluator
has_degree
has_position
job
languages
project
requires
subject
user

titlos	idryma	bathmida
Bachelor of Business Administration	University of Thessaloniki	BSc
Bachelor of Computer Engineering	University of Patras	BSc
Bachelor of Computer Science	Athens University	BSc
Bachelor of Computer Science	University of Patras	BSc
Bachelor of Fine Arts	Aristotle University of Thessaloniki	BSc
Bachelor of Information Technology	University of Patras	BSc
Bachelor of Marketing	University of Piraeus	PhD
Bachelor of Mathematics	Oxford University	BSc
Bachelor of Physics	Harvard University	BSc
Bachelor of Physics	Warwick University	BSc
Bachelor of Technology Innovation and Strategy	Kapodistrian University of Athens	BSc
Bachelor of Telecommunications Engineering	University of Piraeus	BSc
Master of Business Administration	University of Patras	MSc
Master of Computer Engineering	University of Patras	MSc
Master of Data Science	University of Patras	MSc
Master of Data Science	University of Peloponnesse	MSc
Master of Network Security	Aristotle University of Thessaloniki	MSc
PhD of Computer Architecture	Harvard University	PhD
PhD of Computer Science	University of Patras	PhD

Add Update Delete Logout

Type data separated with commas and hit enter

Αποτέλεσμα μετά την εισαγωγή(κλικ στο πλήκτρο Add).

active_admin
administrator
administrator_log
application_log
applies
degree
employee
etaireia
evaluation
evaluator
has_degree
has_position
job
languages
project
requires
subject
user

titlos	idryma	bathmida
Bachelor of Business Administration	University of Thessaloniki	BSc
Bachelor of Computer Engineering	University of Patras	BSc
Bachelor of Computer Science	Athens University	BSc
Bachelor of Computer Science	University of Patras	BSc
Bachelor of Fine Arts	Aristotle University of Thessaloniki	BSc
Bachelor of Information Technology	University of Patras	BSc
Bachelor of Marketing	University of Piraeus	PhD
Bachelor of Mathematics	Oxford University	BSc
Bachelor of Physics	Harvard University	BSc
Bachelor of Physics	Warwick University	BSc
Bachelor of Technology Innovation and Strategy	Kapodistrian University of Athens	BSc
Bachelor of Telecommunications Engineering	University of Piraeus	BSc
Master of Business Administration	University of Patras	MSc
Master of Computer Engineering	University of Patras	MSc
Master of Data Science	University of Patras	MSc
Master of Data Science	University of Peloponnesse	MSc
Master of Network Security	Aristotle University of Thessaloniki	MSc
PhD of Computer Architecture	Harvard University	PhD
PhD of Computer Science	University of Patras	PhD

Add Update Delete

Type data separated with commas and hit enter Logout

Εισαγωγή τιμών από τον χρήστη για την ενημέρωση επιλεγμένου table.

active_admin
administrator
administrator_log
application_log
applies
degree
employee
etaireia
evaluation
evaluator
has_degree
has_position
job
languages
project
requires
subject
user

titlos	idryma	bathmida
Bachelor of Business Administration	University of Thessaloniki	BSc
Bachelor of Computer Engineering	University of Patras	BSc
Bachelor of Computer Science	Athens University	BSc
Bachelor of Computer Science	University of Patras	BSc
Bachelor of Fine Arts	Aristotle University of Thessaloniki	BSc
Bachelor of Information Technology	University of Patras	BSc
Bachelor of Marketing	University of Piraeus	PhD
Bachelor of Mathematics	Oxford University	BSc
Bachelor of Physics	Harvard University	BSc
Bachelor of Physics	Warwick University	BSc
Bachelor of Technology Innovation and Strategy	Kapodistrian University of Athens	BSc
Bachelor of Telecommunications Engineering	University of Piraeus	BSc
BSc of Data Structures	University of Peloponnesse	BSc
Master of Business Administration	University of Patras	MSc
Master of Computer Engineering	University of Patras	MSc
Master of Data Science	University of Patras	MSc
Master of Data Science	University of Peloponnesse	MSc
Master of Network Security	Aristotle University of Thessaloniki	MSc
PhD of Computer Architecture	Harvard University	PhD

Add **Update** Delete

Type data separated with commas and hit enter Logout

Αποτέλεσμα μετά την ενημέρωση(κλικ στο πλήκτρο Update).

The screenshot shows a Java Swing application window. On the left, a vertical menu bar lists several items: active_admin, administrator, administrator_log, application_log, applies, degree (which is highlighted in blue), employee, etaireia, evaluation, evaluator, has_degree, has_position, job, languages, project, requires, subject, and user. Below the menu is a table with three columns: titlos, idryma, and bathmida. The table contains numerous rows of degree information. At the bottom of the table are four buttons: Add, Update, Delete, and a text input field with the placeholder "Type data separated with commas and hit enter". To the right of the table is a red Logout button.

titlos	idryma	bathmida
Bachelor of Business Administration	University of Thessaloniki	BSc
Bachelor of Computer Engineering	University of Patras	BSc
Bachelor of Computer Science	Athens University	BSc
Bachelor of Computer Science	University of Patras	BSc
Bachelor of Fine Arts	Aristotle University of Thessaloniki	BSc
Bachelor of Information Technology	University of Patras	BSc
Bachelor of Marketing	University of Piraeus	PhD
Bachelor of Mathematics	Oxford University	BSc
Bachelor of Physics	Harvard University	BSc
Bachelor of Physics	Warwick University	BSc
Bachelor of Technology Innovation and Strategy	Kapodistrian University of Athens	BSc
Bachelor of Telecommunications Engineering	University of Piraeus	BSc
Master of Business Administration	University of Patras	MSc
Master of Computer Engineering	University of Patras	MSc
Master of Data Science	University of Patras	MSc
Master of Data Science	University of Peloponnesse	MSc
Master of Network Security	Aristotle University of Thessaloniki	MSc
PhD of Computer Architecture	Harvard University	PhD

Παράδειγμα διαγραφής entry από τον χρήστη(επιλογή του row και κλικ στο πλήκτρο Delete).