```
CREATE TABLE Course(id_course int AUTO_INCREMENT, NRC varchar(15), period
varchar(25), name varchar(75), PRIMARY KEY (id_course));
CREATE TABLE Person(id_person int AUTO_INCREMENT, name varchar(75),
phoneNumber varchar(15), email varchar(65), id_course int, PRIMARY
KEY(id_person), FOREIGN KEY(id_course) REFERENCES Course(id_course));
CREATE TABLE Coordinator(id_person int, cubicle int, staff_number varchar(25),
PRIMARY KEY(id_person), FOREIGN KEY(id_person) REFERENCES
Person(id_person));
CREATE TABLE Professor(id_person int, cubicle int, staff_number varchar(25),
PRIMARY KEY(id_person), FOREIGN KEY(id_person) REFERENCES
Person(id_person));
CREATE TABLE Company(id_company int AUTO_INCREMENT, name varchar(75),
address varchar(75), email varchar(65), state varchar(45), phoneNumber
varchar(15),direct_users int, indirect_users int, sector
enum('Primary','Secondary','Tertiary'), city varchar(65), id_coordinator int,
id_course int, PRIMARY KEY(id_company), FOREIGN KEY(id_coordinator)
REFERENCES Coordinator(id_person), FOREIGN KEY(id_course) REFERENCES
Course(id_course));
CREATE TABLE Project(id_project int AUTO_INCREMENT, name
varchar(75),duration float, schedule varchar(75), general_purpose text,
general_description text, id_company int, charge_Responsable varchar(35),
name_Responsable varchar(75), email_Responsable varchar(65),PRIMARY
KEY(id_project), FOREIGN KEY(id_company) REFERENCES
Company(id_company));
CREATE TABLE Project_Mediate_Objetives(objetive varchar(65), id_project int,
FOREIGN KEY(id_project) REFERENCES Project(id_project) ON DELETE
CASCADE);
CREATE TABLE Project_Methodologies(methodology varchar(65), id_project int,
FOREIGN KEY(id_project) REFERENCES Project(id_project) ON DELETE
CASCADE);
CREATE TABLE Project_Resources(resource varchar(65), id_project int, FOREIGN
KEY(id_project) REFERENCES Project(id_project) ON DELETE CASCADE);
CREATE TABLE Project_Responsabilities(responsability varchar(65), id_project int,
FOREIGN KEY(id_project) REFERENCES Project(id_project) ON DELETE
CASCADE);
CREATE TABLE Project_Activities(activity varchar(65), id_project int, FOREIGN
KEY(id_project) REFERENCES Project(id_project) ON DELETE CASCADE);
CREATE TABLE Project_Immediate_Objetives(objetive varchar(65), id_project int,
FOREIGN KEY(id_project) REFERENCES Project(id_project) ON DELETE
CASCADE);
CREATE TABLE Practitioner(id_person int, enrollment varchar(35), id_project int,
id_professor int, PRIMARY KEY(id_person), FOREIGN KEY(id_person) REFERENCES
Person(id_person),FOREIGN KEY(id_project) REFERENCES Project(id_project) ,
FOREIGN KEY(id_professor) REFERENCES Professor(id_person));
CREATE TABLE  Practitioner_Selected_Projects(selected_project int ,id_person int,
FOREIGN KEY(selected_project) REFERENCES Project(id_project), FOREIGN
KEY(id_person) REFERENCES Practitioner(id_person));
CREATE TABLE Activity(id_activity int AUTO_INCREMENT, name varchar(75),
description text, deadline datetime, id_professor int, PRIMARY KEY(id_activity),
FOREIGN KEY(id_professor) REFERENCES Professor(id_person));
CREATE TABLE Delivery(id_activity int , id_practitioner int, observation text ,score
float, file longblob, filename
enum('Partial_Report','Monthly_Report','Schedule','Assignment_Office','Acceptance_
Office','Self_Appraisal','Company_Evaluation'), FOREIGN KEY(id_activity)
REFERENCES Activity(id_activity), FOREIGN KEY(id_practitioner) REFERENCES
Practitioner(id_person) ON DELETE CASCADE);


CREATE VIEW view_project AS SELECT id_project, name ,duration, schedule,
id_company, charge_responsable, name_responsable, email_responsable from
Project;


DELIMITER $$
CREATE PROCEDURE addDelivery(activity int, practitioner int, file_to longblob,
filename_to varchar(65))
BEGIN
DECLARE name_act INT;
DECLARE EXIT HANDLER FOR SQLEXCEPTION
 BEGIN
 SHOW ERRORS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
 DECLARE EXIT HANDLER FOR SQLWARNING
 BEGIN
 SHOW WARNINGS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
SELECT NOW() INTO @now;
SELECT deadline INTO @deadline_activity FROM Activity WHERE id_activity =
activity;
IF @now < @deadline_activity THEN
```

```
        CASE filename_to
        WHEN 'PARTIAL_REPORT' THEN SET name_act = 1;
        WHEN 'MONTHLY_REPORT' THEN SET name_act = 2;
        WHEN 'SCHEDULE' THEN SET name_act = 3;
        WHEN 'ASSIGNMENT_OFFICE' THEN SET name_act = 4;
        WHEN 'ACCEPTANCE_OFFICE' THEN SET name_act = 5;
        WHEN 'SELF_APRAISSAL' THEN SET name_act = 6;
        WHEN 'COMPANY_EVALUATION' THEN SET name_act = 7;
ELSE SET name_act = 0;
END CASE;
START TRANSACTION;
INSERT INTO Delivery(id_activity, id_practitioner, observation, score, file, filename)
VALUES(activity, practitioner, null, null, file_to, filename);
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Activitys deadline is over';
END IF;
COMMIT;
END $$
DELIMITER ;



DELIMITER $$
CREATE PROCEDURE assignProject(person INT, project INT)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
 BEGIN
 SHOW ERRORS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
 DECLARE EXIT HANDLER FOR SQLWARNING
 BEGIN
 SHOW WARNINGS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
START TRANSACTION;
SELECT COUNT(id_project) INTO @count FROM Practitioner WHERE id_project =
project;
IF @count < 3 THEN
        UPDATE Practitioner SET id_project = project WHERE id_person = person;
```

```
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Table size limit reached';
END IF;
COMMIT;
END $$
DELIMITER ;




DELIMITER $$
CREATE PROCEDURE selectProject(person INT, project INT)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
 BEGIN
 SHOW ERRORS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
 DECLARE EXIT HANDLER FOR SQLWARNING
 BEGIN
 SHOW WARNINGS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
START TRANSACTION;
SELECT COUNT(id_person) INTO @countSelected FROM
Practitioner_selected_projects WHERE id_person = person;
IF @countSelected < 3 THEN
        INSERT INTO Practitioner_Selected_Projects(selected_project, id_person)
VALUES(project, person);
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Table size limit reached';
END IF;
COMMIT;
END $$
DELIMITER ;
DELIMITER $$
CREATE PROCEDURE removeProject(id_remove INT)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
 BEGIN
 SHOW ERRORS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
 DECLARE EXIT HANDLER FOR SQLWARNING
 BEGIN
 SHOW WARNINGS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
START TRANSACTION;
UPDATE Practitioner SET id_project = NULL WHERE id_project = id_remove;
DELETE FROM Project WHERE id_project = id_remove;
COMMIT;
END $$
DELIMITER ;


DELIMITER $$
CREATE PROCEDURE addPractitioner(name_p varchar(75), phone_p varchar(15),
email_p varchar(65), id_cs INT, enrollment_p varchar(35))
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
 BEGIN
 SHOW ERRORS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
 DECLARE EXIT HANDLER FOR SQLWARNING
 BEGIN
 SHOW WARNINGS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
START TRANSACTION;
INSERT INTO Person(name, phoneNumber, email, id_course) VALUES(name_p,
phone_p, email_p, id_cs);
SELECT LAST_INSERT_ID() INTO @id_p;
```

```
INSERT INTO Practitioner(id_person, enrollment, id_project, id_professor)
VALUES(@id_p, enrollment_p, null, null);
COMMIT;
END $$
DELIMITER ;


DELIMITER $$
CREATE PROCEDURE removeMultivaluedAttributesProject(id INT)
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
 BEGIN
 SHOW ERRORS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
 DECLARE EXIT HANDLER FOR SQLWARNING
 BEGIN
 SHOW WARNINGS LIMIT 1;
RESIGNAL;
 ROLLBACK;
 END;
START TRANSACTION;
DELETE FROM Project_Activities WHERE id_project = id;
DELETE FROM Project_Responsabilities WHERE id_project = id;
DELETE FROM Project_Mediate_Objetives WHERE id_project = id;
DELETE FROM Project_Methodologies WHERE id_project = id;
DELETE FROM Project_Resources WHERE id_project = id;
DELETE FROM Project_Immediate_Objetives WHERE id_project = id;
COMMIT;
END $$
DELIMITER ;
```