****
****

CONSULTAS BASE DE DATOS

****
****
****
****

VER LAS COMPAÑIAS ULTIMO CURSO AGREGADO

```
SELECT
COMP.id_company, COMP.name, COMP.address, COMP.email, COMP.state, COMP.direct_users, COMP.indirect_users,
COMP.sector, COMP.city, COMP.phoneNumber,  COMP.id_course,
CORD.cubicle, CORD.staff_number,
PERSCORD.id_person, PERSCORD.name, PERSCORD.phoneNumber, PERSCORD.email,
COUR.NRC, COUR.Period, COUR.name, COUR.id_course
FROM Company AS COMP
INNER JOIN Coordinator AS CORD ON COMP.id_coordinator = CORD.id_person
INNER JOIN Person AS PERSCORD ON COMP.id_coordinator = PERSCORD.id_person
INNER JOIN Course AS COUR ON COMP.id_course = COMP.id_course and COUR.id_course = (SELECT max(id_course)
FROM Course);
```

****
****
****
****

PROYECTO ASIGNADO DE UN PRACTICANTE POR ID DE PRACICANTE

```
SELECT
PRAC.id_person, PROJ.id_project,
PROJ.name, PROJ.duration, PROJ.schedule, PROJ.general_purpose, PROJ.general_description,
PROJ.id_company, PROJ.charge_responsable, PROJ.name_responsable, PROJ.email_responsable,
COMP.id_company, COMP.name, COMP.address, COMP.email, COMP.state, COMP.direct_users,
COMP.indirect_users, COMP.sector, COMP.city, COMP.phoneNumber,
CORD.cubicle, CORD.staff_number,
PERSCORD.id_person, PERSCORD.name, PERSCORD.phoneNumber, PERSCORD.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM (SELECT id_person, id_project FROM Practitioner WHERE id_person = ?) AS PRAC
LEFT JOIN Project AS PROJ ON PRAC.id_project = PROJ.id_project
LEFT JOIN Company AS COMP ON  PROJ.id_company = COMP.id_company
LEFT JOIN Coordinator AS CORD ON COMP.id_coordinator = CORD.id_person
LEFT JOIN Person AS PERSCORD ON COMP.id_coordinator = PERSCORD.id_person
LEFT JOIN Course AS COUR ON COMP.id_course = COUR.id_course;
```

****
****
****

PROYECTOS SELECCIONADOS POR PRACTICANTE POR ID PRACTICANTE

```
SELECT
PSP.id_person,
PROJ.id_project, PROJ.name, PROJ.duration, PROJ.schedule, PROJ.general_purpose,
PROJ.general_description, PROJ.id_company, PROJ.charge_responsable, PROJ.name_responsable,
PROJ.email_responsable,
COMP.id_company, COMP.name, COMP.address, COMP.email, COMP.state, COMP.direct_users,
COMP.indirect_users, COMP.sector, COMP.city, COMP.phoneNumber,
CORD.cubicle, CORD.staff_number,
PERSCORD.id_person, PERSCORD.name, PERSCORD.phoneNumber, PERSCORD.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM (SELECT selected_project, id_person FROM Practitioner_Selected_Projects WHERE id_person = 3) AS
PSP
INNER JOIN Project AS PROJ ON PROJ.id_project = PROJ.id_project AND PSP.selected_project =
PROJ.id_project
INNER JOIN Company AS COMP ON  PROJ.id_company = COMP.id_company
INNER JOIN Coordinator AS CORD ON COMP.id_coordinator = CORD.id_person
INNER JOIN Person AS PERSCORD ON COMP.id_coordinator = PERSCORD.id_person
INNER JOIN Course AS COUR ON COMP.id_course = COUR.id_course;
**********************************************************************************************************************
***************
**********************************************************************************************************************
***************
```

PROYECTO ASIGNADO A PRACTICANTE POR ID PRACTICANTE

```
SELECT
PROJ.id_project, PROJ.name, PROJ.duration, PROJ.schedule, PROJ.general_purpose,
PROJ.general_description, PROJ.id_company, PROJ.charge_responsable, PROJ.name_responsable,
PROJ.email_responsable,
COMP.id_company, COMP.name, COMP.address, COMP.email, COMP.state, COMP.direct_users,
COMP.indirect_users, COMP.sector, COMP.city, COMP.phoneNumber,
CORD.cubicle, CORD.staff_number,
PERSCORD.id_person, PERSCORD.name, PERSCORD.phoneNumber, PERSCORD.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM (SELECT id_person, id_project FROM Practitioner WHERE id_person = 5) AS PRAC
INNER JOIN Project AS PROJ ON PRAC.id_project = PROJ.id_project
INNER JOIN Company AS COMP ON  PROJ.id_company = COMP.id_company
INNER JOIN Coordinator AS CORD ON COMP.id_coordinator = CORD.id_person
INNER JOIN Person AS PERSCORD ON COMP.id_coordinator = PERSCORD.id_person
INNER JOIN Course AS COUR ON COMP.id_course = COUR.id_course;
**********************************************************************************************************************
***************
**********************************************************************************************************************
***************
```

PROYECTOS DISPONIBLES CURSO ACTUAL O ULTIMO CURSO

```
SELECT
PROJ.id_project, PROJ.name, PROJ.duration, PROJ.schedule, PROJ.general_purpose,
PROJ.general_description, PROJ.id_company, PROJ.charge_responsable, PROJ.name_responsable,
PROJ.email_responsable,
COMP.id_company, COMP.name, COMP.address, COMP.email, COMP.state, COMP.direct_users,
COMP.indirect_users, COMP.sector, COMP.city, COMP.phoneNumber,
CORD.cubicle, CORD.staff_number,
PERSCORD.id_person, PERSCORD.name, PERSCORD.phoneNumber, PERSCORD.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM Project AS PROJ
INNER JOIN Company AS COMP ON  PROJ.id_company = COMP.id_company
INNER JOIN Coordinator AS CORD ON COMP.id_coordinator = CORD.id_person
INNER JOIN Person AS PERSCORD ON COMP.id_coordinator = PERSCORD.id_person
INNER JOIN Course AS COUR ON COMP.id_course = COUR.id_course and COUR.id_course = (SELECT
max(id_course) FROM Course)
```

```
WHERE PROJ.id_project NOT IN (SELECT id_project FROM (SELECT count(id_project) AS counter, id_project
FROM Practitioner GROUP BY id_project HAVING counter = 3) AS Count);
```

****

****

LISTA DE TODOS LOS PROYECTOS DEL CURSO ACTUAL O ULTIMO CURSO

```
SELECT
PROJ.id_project, PROJ.name, PROJ.duration, PROJ.schedule, PROJ.general_purpose,
PROJ.general_description, PROJ.id_company, PROJ.charge_responsable, PROJ.name_responsable,
PROJ.email_responsable,
COMP.id_company, COMP.name, COMP.address, COMP.email, COMP.state, COMP.direct_users,
COMP.indirect_users, COMP.sector, COMP.city, COMP.phoneNumber,
CORD.cubicle, CORD.staff_number,
PERSCORD.id_person, PERSCORD.name, PERSCORD.phoneNumber, PERSCORD.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM Project AS PROJ
INNER JOIN Company AS COMP ON  PROJ.id_company = COMP.id_company
INNER JOIN Coordinator AS CORD ON COMP.id_coordinator = CORD.id_person
INNER JOIN Person AS PERSCORD ON CORD.id_person = PERSCORD.id_person
INNER JOIN Course AS COUR ON COMP.id_course = COUR.id_course and COUR.id_course = (SELECT
max(id_course) FROM Course);
```

****

****

LISTA DE LOS PRACTICANTES SIN PROYECTO ASIGNADO NI SELECCIONADO NI PROFESOR ASIGNADO

```
SELECT
PRAC.id_person, PRAC.enrollment,
PERS.name, PERS.phoneNumber, PERS.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM Practitioner AS PRAC
INNER JOIN Person AS PERS ON PRAC.id_person = PERS.id_person
INNER JOIN Course AS COUR ON PERS.id_course = COUR.id_course AND COUR.id_course = (SELECT
max(id_course) FROM Course) ;
```

****

****

PROFESOR ASIGNADO DE UN PRACTICANTE

```
SELECT
PROF.id_person, PROF.cubicle, PROF.staff_number,
PERSPROF.name, PERSPROF.phoneNumber, PERSPROF.email,
COUR.id_course, COUR.NRC, COUR.period, COUR.name
FROM Professor AS PROF
INNER JOIN Practitioner AS PRAC ON PROF.id_person = PRAC.id_professor AND PRAC.id_person = 4
INNER JOIN PERSON AS PERSPROF ON PROF.id_person = PERSPROF.id_person
INNER JOIN COURSE AS COUR ON PERSPROF.id_course = COUR.id_course;
```

****

****

ACTIVIDADES ENTREGADAS POR PRACTICANTE

```
SELECT
DEL.observation, DEL.score, DEL.file, DEL.filename,
```

```
ACT.id_activity, ACT.name, ACT.description, ACT.deadline,
COUR.id_course, COUR.NRC, COUR.name, COUR.PERIOD,
PERSPROF.name, PERSPROF.phoneNumber, PERSPROF.email,
PROF.cubicle, PROF.staff_number
FROM delivery AS DEL
INNER JOIN Activity AS ACT ON DEL.id_activity = ACT.id_activity AND DEL.id_practitioner = 3
INNER JOIN Professor AS PROF ON ACT.id_professor = PROF.id_person
INNER JOIN Person AS PERSPROF ON PROF.id_person = PERSPROF.id_person
INNER JOIN Course AS COUR ON PERSPROF.id_course = COUR.id_course;
```

USUARIOS BASE DE DATOS

****

```
CREATE USER 'anonimo'@'localhost' IDENTIFIED BY '12345';
GRANT INSERT, DELETE, SELECT, UPDATE, EXECUTE ON *.* TO 'anonimo'@'localhost';
FLUSH PRIVILEGES;
```

****

VISTAS

****

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

****

PROCEDIMIENTOS

****

Asignar proyecto: Coordinador
3 practicantes por proyecto:

****

```
DELIMITER $$
CREATE PROCEDURE assignProject(person INT, project INT)
BEGIN
SELECT COUNT(id_project) INTO @count FROM Practitioner WHERE id_project = project;
IF @count < 3 THEN
        UPDATE Practitioner SET id_project = project WHERE id_person = person;
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Table size limit reached';
END IF;
END $$
DELIMITER ;
```

****

Eliminar un proyecto: Coordinador

****
```
DELIMITER $$
CREATE PROCEDURE removeProject(id_remove INT)
BEGIN
UPDATE Practitioner SET id_project = NULL WHERE id_project = id_remove;
DELETE FROM Project WHERE id_project = id_remove;
END $$
DELIMITER ;
```

****

Solicitar proyecto: Practicante

3 proyectos seleccionados:

****
```
DELIMITER $$
CREATE PROCEDURE selectProject(person INT, project INT)
BEGIN
SELECT COUNT(id_person) INTO @countSelected FROM Practitioner_selected_projects WHERE id_person = person;
IF @countSelected < 3 THEN
	INSERT INTO Practitioner_Selected_Projects(selected_project, id_person) VALUES(project, person);
ELSE
	SIGNAL SQLSTATE '45000'
	SET MESSAGE_TEXT = 'Table size limit reached';
END IF;
END $$
DELIMITER ;
```
****

****

Agregar practicante: Coordinador ********

****
```
DELIMITER $$
CREATE PROCEDURE addPractitioner(name_p varchar(75), phone_p varchar(15), email_p varchar(65), id_cs INT,
enrollment_p varchar(35))
BEGIN
INSERT INTO Person(name, phoneNumber, email, id_course) VALUES(name_p, phone_p, email_p, id_cs);
SELECT LAST_INSERT_ID() INTO @id_p;
INSERT INTO Practitioner(id_person, enrollment, id_project, id_professor) VALUES(@id_p, enrollment_p, null, null);
END $$
DELIMITER ;
```
****

****

Eliminar atributos multivaluados de proyecto: Coordinador

****

```
DELIMITER $$
CREATE PROCEDURE removeMultivaluedAttributesProject(id INT)
BEGIN
DELETE FROM Project_Activities WHERE id_project = id;
DELETE FROM Project_Responsabilities WHERE id_project = id;
DELETE FROM Project_Mediate_Objetives WHERE id_project = id;
DELETE FROM Project_Methodologies WHERE id_project = id;
DELETE FROM Project_Resources WHERE id_project = id;
DELETE FROM Project_Immediate_Objetives WHERE id_project = id;
END $$
DELIMITER ;
```

****

Agregar archivo a actividad: Practicante

****

```
DELIMITER $$
CREATE PROCEDURE addDelivery(activity int, practitioner int, file_to longblob, filename_to varchar(65))
BEGIN
DECLARE name_act INT;
SELECT NOW() INTO @now;
SELECT deadline INTO @deadline_activity FROM Activity WHERE id_activity = activity;
IF @now < @deadline_activity THEN
CASE filename_to
WHEN 'PARTIAL_REPORT' THEN SET name_act = 1;
WHEN 'MONTHLY_REPORT' THEN SET name_act = 2;
WHEN 'SCHEDULE' THEN SET name_act = 3;
WHEN 'ASSIGNMENT_OFFICE' THEN SET name_act = 4;
WHEN 'ACCEPTANCE_OFFICE' THEN SET name_act = 5;
WHEN 'SELF_APRAISSAL' THEN SET name_act = 6;
WHEN 'COMPANY_EVALUATION' THEN SET name_act = 7;
ELSE SET name_act = 0;
END CASE;
INSERT INTO Delivery(id_activity, id_practitioner, observation, score, file, filename) VALUES(activity, practitioner, null,
null, file_to, filename);
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Activitys deadline is over';
END IF;
END $$
DELIMITER ;
```

```
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);
SELECT id_activity, deadline FROM Activity WHERE deadline > '2020-04-21 23:55:49';
DELETED
CREATE TABLE Practitioner_Record(document longblob, name enum('Partial Report','Monthly
Report','Schedule','Assignment Office','Acceptance Office','Self Appraisal','Company Evaluation'), id_person int,
```

FOREIGN KEY(id_person) REFERENCES Practitioner(id_person) ON DELETE CASCADE);

****

BASE DE DATOS 1.0

****

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_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) ON DELETE CASCADE);
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 $$
CREATE PROCEDURE addDelivery(activity int, practitioner int, file_to longblob, filename_to varchar(65))
BEGIN
DECLARE name_act INT;
SELECT NOW() INTO @now;
SELECT deadline INTO @deadline_activity FROM Activity WHERE id_activity = activity;
IF @now < @deadline_activity THEN
CASE filename_to
WHEN 'PARTIAL_REPORT' THEN SET name_act = 1;
WHEN 'MONTHLY_REPORT' THEN SET name_act = 2;
WHEN 'SCHEDULE' THEN SET name_act = 3;
WHEN 'ASSIGNMENT_OFFICE' THEN SET name_act = 4;
WHEN 'ACCEPTANCE_OFFICE' THEN SET name_act = 5;
WHEN 'SELF_APRAISSAL' THEN SET name_act = 6;
WHEN 'COMPANY_EVALUATION' THEN SET name_act = 7;
ELSE SET name_act = 0;
END CASE;
INSERT INTO Delivery(id_activity, id_practitioner, observation, score, file, filename) VALUES(activity, practitioner, null,
null, file_to, filename);
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Activitys deadline is over';
END IF;
END $$
DELIMITER ;



DELIMITER $$
CREATE PROCEDURE assignProject(person INT, project INT)
BEGIN
SELECT COUNT(id_project) INTO @count FROM Practitioner WHERE id_project = project;
IF @count < 3 THEN
        UPDATE Practitioner SET id_project = project WHERE id_person = person;
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Table size limit reached';
END IF;
END $$
DELIMITER ;


DELIMITER $$
CREATE PROCEDURE selectProject(person INT, project INT)
BEGIN
SELECT COUNT(id_person) INTO @countSelected FROM Practitioner_selected_projects WHERE id_person = person;
IF @countSelected < 3 THEN
        INSERT INTO Practitioner_Selected_Projects(selected_project, id_person) VALUES(project, person);
ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Table size limit reached';
END IF;
END $$
DELIMITER ;


DELIMITER $$
CREATE PROCEDURE removeProject(id_remove INT)
BEGIN
UPDATE Practitioner SET id_project = NULL WHERE id_project = id_remove;
DELETE FROM Project WHERE id_project = id_remove;
END $$
```

```
DELIMITER ;


DELIMITER $$
CREATE PROCEDURE addPractitioner(name_p varchar(75), phone_p varchar(15), email_p varchar(65), id_cs INT,
enrollment_p varchar(35))
BEGIN
INSERT INTO Person(name, phoneNumber, email, id_course) VALUES(name_p, phone_p, email_p, id_cs);
SELECT LAST_INSERT_ID() INTO @id_p;
INSERT INTO Practitioner(id_person, enrollment, id_project, id_professor) VALUES(@id_p, enrollment_p, null, null);
END $$
DELIMITER ;


DELIMITER $$
CREATE PROCEDURE removeMultivaluedAttributesProject(id INT)
BEGIN
DELETE FROM Project_Activities WHERE id_project = id;
DELETE FROM Project_Responsabilities WHERE id_project = id;
DELETE FROM Project_Mediate_Objetives WHERE id_project = id;
DELETE FROM Project_Methodologies WHERE id_project = id;
DELETE FROM Project_Resources WHERE id_project = id;
DELETE FROM Project_Immediate_Objetives WHERE id_project = id;
END $$
DELIMITER ;



INSERT INTO Course(NRC, period, name) values("SRC01","AGO 2020 DIC 2020", "Practicas Profesionales");
INSERT INTO Person (name, phoneNumber, email, id_course) values ('Howard Baylie', '2357245959',
'hbayliep@sfgate.com', 1);

INSERT INTO Coordinator(id_person, cubicle, staff_number) VALUES(1,1,"1");
INSERT INTO Professor (id_person, cubicle, staff_number) values (2, 2, '495753472-1');

INSERT INTO Company (name, address, email, state, direct_users, indirect_users, sector, city, id_coordinator,
id_course, phoneNumber) values ('Kazu', '9 Bluestem Circle', 'jmewis0@weather.com', 'Texas', 81, 94, 1, 'Waco', 1, 1,
'2543993996');

INSERT INTO Practitioner (id_person, enrollment, id_project, id_professor) values (17, 'S18052710', 1, 15);
```