



\c iti

- + Add gender column for the student table[Enum]. It holds two value (male or female).
 - CREATE TYPE gender AS ENUM('male','female');
 - ALTER TABLE student ADD gender gender;

- + Add birth date column for the student table.
 - ALTER TABLE student ADD birthDate date;

- + Delete the name column and replace it with two columns first name and last name.
 - ALTER TABLE student DROP name;
 - ALTER TABLE student ADD firstname varchar(25);
 - ALTER TABLE student ADD lastname varchar(25);

- + Delete the address and email column and replace it with contact info (Address, email) as object/Composite Data type.
 - ALTER TABLE student DROP email;
 - ALTER TABLE student DROP address;
 - CREATE TYPE contactInfo AS (address varchar(25), email varchar(25));
 - ALTER TABLE student ADD contactInfo contactInfo;

Change any Serial Datatype at your tables to smallInt

- ALTER TABLE student ALTER id SET DATA TYPE smallint;
- ALTER TABLE student ALTER trackid SET DATA TYPE smallint;
- ALTER TABLE track ALTER id SET DATA TYPE smallint;
- ALTER TABLE course ALTER id SET DATA TYPE smallint;
- ALTER TABLE student_phone ALTER studentid SET DATA TYPE smallint;
- ALTER TABLE student_course ALTER studentid SET DATA TYPE smallint;
- ALTER TABLE student_course ALTER courseid SET DATA TYPE smallint;
- ALTER TABLE track_course ALTER courseid SET DATA TYPE smallint;
- ALTER TABLE track_course ALTER trackid SET DATA TYPE smallint;

Add/Alter foreign key constrains in Your Tables.

- ALTER TABLE student ADD CONSTRAINT fk1 FOREIGN KEY (trackid) REFERENCES track (id);
- ALTER TABLE student_phone ADD CONSTRAINT fk2 FOREIGN KEY (studentid) REFERENCES student (id);
- ALTER TABLE student_course ADD CONSTRAINT fk3 FOREIGN KEY (studentid) REFERENCES student (id);
- ALTER TABLE student_course ADD CONSTRAINT fk4 FOREIGN KEY (courseid) REFERENCES course (id);
- ALTER TABLE track_course ADD CONSTRAINT fk6 FOREIGN KEY (trackid) REFERENCES track (id);
- ALTER TABLE track_course ADD CONSTRAINT fk5 FOREIGN KEY (courseid) REFERENCES course (id);

Insert new data in all Tables.

- INSERT INTO track VALUES (6, 'Power BI');
- INSERT INTO student VALUES (6, 1, 'female', '9/8/2000', 'Sarah', 'Mohamed', ROW('Minyat Elnasr', 'sarah@gmail.com')), (7, 4, 'female', '6/8/2003', 'yara', 'Mohamed', ROW('Minyat Elnasr', 'yara@gmail.com')), (8, 5, 'male', '26/9/2007', 'Adham', 'Mohamed', ROW('Minyat Elnasr', 'adham@gmail.com')), (9, 3, 'male', '18/7/1972', 'Mohamed', 'Abdelrahman', ROW('Minyat Elnasr', 'mohamed@gmail.com')), (10, 2, 'female', '6/3/1976', 'Abeer', 'Mohamed', ROW('Minyat Elnasr', 'abeer@gmail.com'));
- INSERT INTO course VALUES (6, 'PostgreSQL', 'Database', 100);
- INSERT INTO student_phone VALUES (6, 0102174532);
- INSERT INTO student_course VALUES (6, 6, '8/7/2023', 93);
- INSERT INTO track_course VALUES (6, 6);

Display all students' information.

- SELECT * FROM student;

Display male students only.

- SELECT * FROM student WHERE gender = 'male';

Display the number of female students.

- SELECT COUNT(gender) AS count FROM student WHERE gender = 'female';

✚ Display the students who are born before 1992-10-01.

- `SELECT * FROM student WHERE birthdate < '1/10/1992';`

✚ Display male students who are born before 1991-10-01.

- `SELECT * FROM student WHERE birthdate < '1/10/1991' AND gender = 'male';`

✚ Display subjects and their max score sorted by max score.

- `SELECT name, maxScore FROM course ORDER BY maxScore desc;`

✚ Display the subject with highest max score

- `SELECT name, maxScore FROM course ORDER BY maxScore desc LIMIT 1;`

✚ Display students' names that begin with A.

- `SELECT CONCAT(firstname, ' ', lastname) AS "Full Name" FROM student WHERE firstname LIKE 'A%';`

✚ Display the number of students' their name is "Mohammed"

- `SELECT COUNT(firstname) FROM student WHERE firstname = 'Mohamed';`

✚ Display the number of males and females.

- `SELECT COUNT(gender), gender FROM student GROUP BY gender;`

✚ Display the repeated first names and their counts if higher than 2.

- `SELECT firstname, COUNT(firstname) AS Count FROM student GROUP BY firstname HAVING COUNT(firstname) > 2 ;`

✚ Display the all Students and track name that belong to it

- `SELECT CONCAT(s.firstname, ' ', s.lastname) AS "Full Name", t.name FROM student s , track t WHERE s.trackid = t.id;`

✚ (Bouns) Display students' names, their score and subject name.

- `SELECT CONCAT(s.firstname, ' ', s.lastname) AS "Full Name", c.name, sc.examscore FROM student s , course c , student_course sc WHERE s.id = sc.studentid AND c.id = sc.courseid;`