Task#1

Create Tables and INSERT data:

1) Students TABLE

```
CREATE TABLE students (
student_id SERIAL PRIMARY KEY,
first_name TEXT,
last_name TEXT,
email TEXT,
school_enrollement_date DATE
);
```

Fixed Typo

```
ALTER TABLE students
RENAME COLUMN school_enrollement_date TO school_enrollment_date;
```

```
INSERT INTO students (first_name, last_name, email, school_enrollment_date) VALUES
('Walid', 'Ali', 'walid.ali@example.com', '2024-04-20'),
('Sarah', 'William', 'sarah.william@example.com', '2024-02-07'),
('Jane', 'Dan', 'jane.dan@example.com', '2023-12-03'),
('Peter', 'John', 'peter.john@example.com', '2024-01-13'),
('David', 'Smith', 'david.smith@example.com', '2023-11-17');
```

SELECT * FROM sudents;

	student_id [PK] integer	first_name text	last_name text	email text	school_enrollment_date date
1	1	Walid	Ali	walid.ali@example.com	2024-04-20
2	2	Sarah	William	sarah.william@example.com	2024-02-07
3	3	Jane	Dan	jane.dan@example.com	2023-12-03
4	4	Peter	John	peter.john@example.com	2024-01-13
5	5	David	Smith	david.smith@example.com	2023-11-17

2) Professors Table

```
CREATE TABLE professors(
professor_id SERIAL PRIMARY KEY,
first_name TEXT,
last_name TEXT,
department TEXT
);
```

```
INSERT INTO professors (first_name, last_name, department) VALUES
    ('Matthew', 'Evan', 'Chemistry'),
    ('Noah', 'Smith', 'Physics'),
    ('Kalid', 'Ahmed', 'Computer Science'),
    ('Heather', 'Russel', 'Mathematics');
```

SELECT * FROM professors;

	professor_id [PK] integer	first_name text	last_name text	department text
1	1	Matthew	Evan	Chemistry
2	2	Noah	Smith	Physics
3	3	Kalid	Ahmed	Computer Science
4	4	Heather	Russel	Mathematics

3) Courses Table

```
CREATE TABLE courses(
course_id SERIAL PRIMARY KEY,
course_name TEXT,
course_description TEXT,
professor_id INT REFERENCES professors(professor_id)
);
```

```
INSERT INTO courses (course_name, course_description, professor_id) VALUES
('Chemistry 101', 'Introduction to Chemistry', 1),
('Physics 101', 'Introduction to Physics', 2),
('Math 201', 'Advanced Mathematics', (SELECT professor_id FROM professors WHERE first_name = 'Heather' AND last_name = 'Russel'));
```

SELECT * FROM courses;

	course_id [PK] integer	course_name text	course_description text	professor_id integer
1	1	Chemistry 101	Introduction to Chemistry	1
2	2	Physics 101	Introduction to Physics	2
3	3	Math 201	Advanced Mathematics	4

4) Enrollment Table

```sql
CREATE TABLE enrollments(
student_id INT REFERENCES students(student_id),
course_id INT REFERENCES courses(course_id),
enrollment_date DATE,
PRIMARY KEY (student_id,course_id)
);
```

```sql
INSERT INTO enrollments(student_id, course_id, enrollment_date)VALUES
((SELECT student_id FROM students WHERE first_name = 'Walid' AND last_name = 'Ali'),
(SELECT course_id FROM courses WHERE course_name = 'Chemistry 101'), '2024-05-15'),
(2, 2, '2024-05-14'),
(3, 1, '2025-05-06'),
(4, 2, '2023-05-14'),
((SELECT student_id FROM students WHERE first_name = 'David' AND last_name = 'Smith'),3, '2024-05-15');
```

	student_id [PK] integer	course_id [PK] integer	enrollment_date date
1	1	1	2024-05-15
2	2	2	2024-05-14
3	3	1	2025-05-06
4	4	2	2023-05-14
5	5	3	2024-05-15

There is another approach that the same student can take multiple courses since it is M:N relationship but I considered here, the course can be taken by multiple students

Query#1-1

```
SELECT first_name || ' ' || last_name AS full_name FROM students
JOIN enrollments ON students.student_id = enrollments.student_id
JOIN courses ON enrollments.course_id = courses.course_id
WHERE course_name = 'Physics 101';
```

	full_name text
1	Sarah William
2	Peter John

Query#1-2

```
SELECT course_name, first_name || ' ' || last_name AS professor_full_name FROM courses
JOIN professors ON courses.professor_id = professors.professor_id
```

	course_name text	professor_full_name text
1	Chemistry 101	Matthew Evan
2	Physics 101	Noah Smith
3	Math 201	Heather Russel

Query#1-3

DISTINCT: Filters out the duplicates // SQL SELECT DISTINCT Statement //W3Shcool

```
SELECT DISTINCT course_name FROM courses
JOIN enrollments ON courses.course_id = enrollments.course_id;
```

	course_name text
1	Physics 101
2	Chemistry 101
3	Math 201

2-Update Data

```
UPDATE students
SET email = 'walid.ali.newemail@example.com'
WHERE first_name = 'Walid' AND last_name = 'Ali';
```

```
SELECT * FROM students Where first_name = 'Walid' AND last_name = 'Ali';
```

	student_id [PK] integer	first_name text	last_name text	email text	school_enrollment_date date
1	1	Walid	Ali	walid.ali.newemail@example.com	2024-04-20

3-Delete Data

```
DELETE FROM enrollments
WHERE student_id = (SELECT student_id FROM students WHERE first_name = 'Sarah' AND last_name = 'William')
AND course_id = (SELECT course_id FROM courses WHERE course_name = 'Physics 101');
```

After Removing

SELECT * FROM enrollments;

	student_id [PK] integer	course_id [PK] integer	enrollment_date date
1	1	1	2024-05-15
2	3	1	2025-05-06
3	4	2	2023-05-14
4	5	3	2024-05-15

Task#2

After Creating the tables with inserting the Data

1) Products Table

	product_id [PK] integer	product_name text	price numeric (10,2)	stock_quantity integer
1	1	T-Shirt	19.99	50
2	2	Jeans	49.99	30
3	3	Jacket	89.99	20
4	4	Sweater	39.99	25
5	5	Sneakers	59.99	40

2) Customers Table

	customer_id [PK] integer	first_name text	last_name text	email text
1	1	Mohamed	Salah	mosalah@example.com
2	2	Roberto	Carlos	roberto.carl@example.com
3	3	Antonio	Modest	anton_modest@example.com
4	4	Amanda	Winfrey	amanda_win@example.com

3) Orders Table

	order_id [PK] integer	customer_id integer	order_date date
1	1	1	2023-09-01
2	2	2	2023-08-29
3	3	3	2023-09-05
4	4	4	2023-09-04
5	5	1	2023-08-23

4) Order_Items Table

	order_id [PK] integer	product_id [PK] integer	quantity integer
1	1	1	2
2	1	3	1
3	2	2	1
4	2	5	1
5	3	4	2
6	3	1	1
7	4	3	1
8	4	2	1
9	5	5	2
10	5	4	1

Query #1-1

```
Retrieve the names and stock quantities of all products
SELECT product_name, stock_quantity FROM products;
```

	product_name text	stock_quantity integer
1	T-Shirt	50
2	Jeans	30
3	Jacket	20
4	Sweater	25
5	Sneakers	40

Query #1-2 : Retrieve the product names and quantities for one of the orders placed

```
SELECT product_name, quantity FROM products
JOIN order_items ON products.product_id = order_items.product_id
JOIN orders ON order_items.order_id = orders.order_id
WHERE orders.order_id = 1;
```

	product_name text	quantity integer
1	T-Shirt	2
2	Jacket	1

Query#1-3

```
 Retrieve all orders placed by a specific customer (including ID's of what was
ordered and quantities)
SELECT
    orders.order_id, product_name, quantity,
    first_name || ' ' || last_name AS customer_full_name
FROM orders
JOIN customers ON orders.customer_id = customers.customer_id
JOIN order_items ON orders.order_id = order_items.order_id
JOIN products ON order_items.product_id = products.product_id
WHERE customers.first_name = 'Mohamed' AND customers.last_name = 'Salah';
```

	order_id integer	product_name text	quantity integer	customer_full_name text
1	1	T-Shirt	2	Mohamed Salah
2	1	Jacket	1	Mohamed Salah
3	5	Sweater	1	Mohamed Salah
4	5	Sneakers	2	Mohamed Salah

2) Update Data

	product_id [PK] integer	product_name text	stock_quantity integer
1	1	T-Shirt	48
2	3	Jacket	19

3) Delete data

a- Before Deleting

	order_id [PK] integer	product_id [PK] integer	quantity integer
1	4	3	1
2	4	2	1

b- After Deleting

	order_id [PK] integer	product_id [PK] integer	quantity integer