Laboratory work 2

Please write your answers to the pdf file for defence:

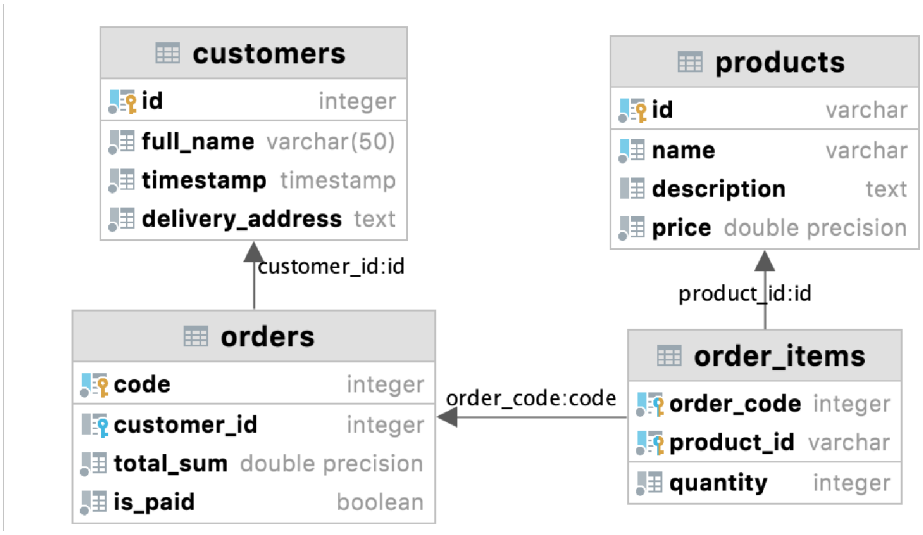
1. Explain the difference between DDL and DML, give the following examples:
   a. at least 3 DDL commands;

```sql
CREATE DATABASE uni_database;

DROP DATABASE IF EXISTS uni_database;

CREATE TABLE lessons
(
    id varchar NOT NULL UNIQUE PRIMARY KEY,
    name varchar(20) NOT NULL UNIQUE
);

DROP TABLE IF EXISTS lessons;

ALTER TABLE lessons
    DROP COLUMN name;

ALTER TABLE lessons
    ADD COLUMN name varchar(20) NOT NULL UNIQUE;
```

   b. at least 4 DML commands.

```sql
SELECT *
FROM lessons;

INSERT INTO lessons
VALUES ('DB1', 'Database');

INSERT INTO lessons
VALUES ('DB2', 'Basedata Adv');

UPDATE lessons
SET name = 'Database Adv'
WHERE id = 'DB2';

DELETE
FROM lessons
WHERE id = 'DB1';
```

2. Write SQL statements to create tables in the figure below:



grey circle - not null, b lue column - unique; quantity, total_sum, price > 0

```sql
CREATE TABLE customers
(
    id               integer     NOT NULL UNIQUE,
    full_name        varchar(50) NOT NULL,
    timestamp        timestamp   NOT NULL,
    delivery_address text        NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE orders
(
    code        integer          NOT NULL UNIQUE,
    customer_id integer REFERENCES customers (id),
    total_sum   double precision NOT NULL CHECK (total_sum > 0),
    is_paid     boolean          NOT NULL,
    PRIMARY KEY (code)
);

CREATE TABLE products
(
    id          varchar          NOT NULL,
    name        varchar          NOT NULL,
    description text,
    price       double precision NOT NULL CHECK (price > 0),
    UNIQUE (id, name),
    PRIMARY KEY (id)
);

CREATE TABLE order_items
(
    product_id varchar NOT NULL REFERENCES products (id),
    order_code integer NOT NULL REFERENCES orders (code),
    quantity   integer NOT NULL CHECK (quantity > 0),
    UNIQUE (product_id, order_code),
    PRIMARY KEY (product_id, order_code)
);
```

3. Write SQL statements describing tables with appropriate ***data types*** and ***constraints*** satisfying the following conditions*(maybe you need additional tables to store data **atomically** and **efficiently**)*:
   a. a students table storing data such as full name, age, birth date, gender, average grade, information about yourself, the need for a dormitory, additional info.
   b. an instructors table storing data such as full name, speaking languages, work experience, the possibility of having remote lessons.
   c. a lesson participants table storing data such as lesson title, teaching instructor, studying students, room number.

```sql
--a
CREATE TABLE students
(
    id             integer     NOT NULL UNIQUE,
    full_name      varchar(50) NOT NULL,
    age            integer     NOT NULL CHECK (age > 0),
    birth_date     date        NOT NULL,
    gender         varchar(20) NOT NULL,
    average_grade  double precision,
    self_info      varchar,
    dormitory_need boolean     NOT NULL,
    add_info       varchar,
    PRIMARY KEY (id)
);

--b
CREATE TABLE languages
(
    language_name varchar(50) NOT NULL UNIQUE,
    PRIMARY KEY (language_name)
);

CREATE TABLE instructors
(
    id                 integer          NOT NULL UNIQUE,
    full_name          varchar(50)      NOT NULL,
    speaking_languages varchar(50)      NOT NULL REFERENCES languages (language_name),
    work_experience    double precision NOT NULL CHECK (work_experience > 0),
    remote_possibility boolean          NOT NULL,
    PRIMARY KEY (id)
);

--c
CREATE TABLE lesson_participants
(
    lesson_title  varchar(20) NOT NULL,
    instructor_id integer     NOT NULL REFERENCES instructors (id),
    student_id    integer     NOT NULL REFERENCES students (id),
    room_number   integer     NOT NULL,
```

```
    UNIQUE (instructor_id, student_id),
    PRIMARY KEY (instructor_id, student_id)
);
```

4. Give examples of insertion, update and deletion of data on tables from exercise 2.

```
INSERT INTO products(id, name, description, price)
VALUES ('E100', 'colorant', 'to make colorful', 25.5);

INSERT INTO products
VALUES ('E630', 'amplifiers', 'boosts', 254.5);

UPDATE products
SET name = 'colourant'
WHERE id = 'E100';

DELETE
FROM products
WHERE id = 'E100'
  AND price = 25.5 OR id = 'E630';
```

Note: you can test your queries in datagrip