1 task

DDL	DML
Data definition language	Data manipulation language
These are used to define data structure	It is used to manipulate the existing databases
It works on whole table	It works on one or more rows
CREATE ALTER DROP	SELECT , INSERT, DELETE, UPDATE

2 task

```
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,
   total_sum float(53) NOT NULL CHECK (total_sum > 0) ,
   is_paid boolean NOT NULL,
   PRIMARY KEY (code),
   FOREIGN KEY (costumer_id) REFERENCES customers(id)
);

CREATE TABLE products(
        id varchar NOT NULL UNIQUE,
   name varchar NOT NULL UNIQUE,
   description text,
   price float(53) NOT NULL CHECK (price > 0),
   PRIMARY KEY (id)
);
```

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

3 task

```
CREATE TABLE students(
        full_name varchar(255) NOT NULL ,
   age integer,
   birth_date date,
   gender ENUM('M','F'),
   average_grade float(5),
   info text(70),
   dormitory boolean,
   add_info text(70),
   PRIMARY KEY(full_name)
);
CREATE TABLE instructors(
        full_name_instructors varchar(255) NOT NULL,
   lang text(20),
   w_experience integer(2),
   rem_lessons boolean,
   PRIMARY KEY(full_name_instructors)
);
CREATE TABLE participants (
        title varchar,
   full_name_instructors varchar(255),
```

```
    full_name varchar(255),
    room integer(3),
    FOREIGN KEY (full_name) REFERENCES students(full_name),
    FOREIGN KEY (full_name_instructors) REFERENCES instructors(full_name_insctructors)

);
```

To make normalized DB we need to write full name of students not like an list, row by row. Also we can divide the full name by surname and name, divide them into different rows. There is not transitive connections between some tables also tables connected by primary keys and foreign keys. I made for gender a special type with a format as M for MAN and W for WOMEN.

4 task

```
INSERT INTO customers VALUES (123, "AKIM ALISHER", 2001-06-28 12:38:08,"ALMATY CITY, GOGOL ST. 12, FLAT 8");

UPDATE customers SET full_name = "AKIM ALUA" WHERE id = 123;

DELETE FROM customers WHERE id = 123;
```