

1. Explain the difference between DDL and DML, give the following examples:
  - a. at least 3 DDL commands;
  - b. at least 4 DML commands.

1.

DDL (Data Definition Language) used to define data structures

DML (Data Manipulation Language) used to manipulate the data itself

Also DML commands can be rolled back. (cancelled)

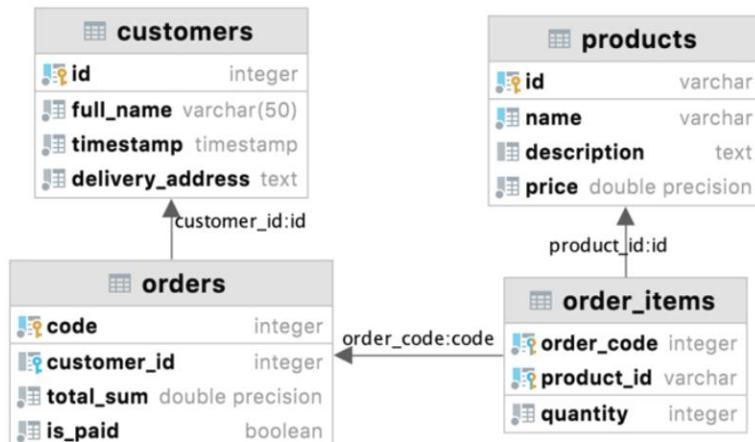
a.

- CREATE DATABASE university\_staff;
- DROP DATABASE lab2w3;
- CREATE TABLE studs( Stud\_Id int, Name VARCHAR(255) );
- ALTER TABLE studs ADD COLUMN gpa float(2);

b.

- INSERT INTO studs VALUES ( 151421, 'Arman', 3.88 );
- DELETE FROM studs WHERE name='Arman';
- UPDATE studs SET id\_stud = 45784565 WHERE name='Arman';
- SELECT \* FROM studs;

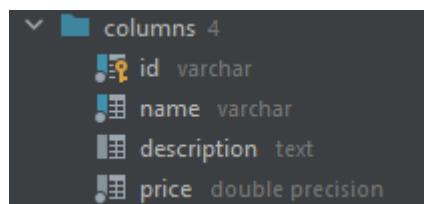
2. Write SQL statements to create tables in the figure below:



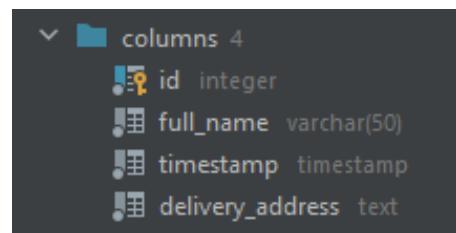
grey circle - not null, blue column - unique; quantity, total\_sum, price > 0

2.

```
CREATE TABLE products(
    id VARCHAR NOT NULL UNIQUE,
    name VARCHAR NOT NULL UNIQUE,
    description TEXT ,
    price double precision NOT NULL CHECK(price > 0),
    primary key(id) );
```



```
CREATE TABLE customers(
    id INT 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 INT NOT NULL UNIQUE,
    customer_id INT,
    total_sum DOUBLE PRECISION NOT NULL CHECK(total_sum > 0),
    is_paid BOOLEAN NOT NULL,
    PRIMARY KEY(code),
    FOREIGN KEY(customer_id) REFERENCES customers(id)
);

```

columns 4

- code integer
- customer\_id integer
- total\_sum double precision
- is\_paid boolean

```

CREATE TABLE order_items(
    order_code INT NOT NULL UNIQUE,
    product_id VARCHAR NOT NULL UNIQUE,
    quantity INT NOT NULL CHECK(quantity > 0),
    PRIMARY KEY(order_code, product_id) ,
    foreign key (order_code) references orders(code),
    foreign key (product_id) references products(id)
);

```

columns 3

- order\_code integer
- product\_id varchar
- quantity integer

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.

3.

```
CREATE TABLE students(
    id serial not null unique ,
    full_name varchar(50) NOT NULL unique ,
    age int not null check ( age > 0 ),
    birth_date DATE not null,
    gender varchar(20) not null,
    gpa float(2),
    info_about_yourself text,
    need_dormitory boolean,
    additional_info text,
    primary key(id)
);

create table instructors(
    id serial not null unique ,
    full_name varchar(50) NOT NULL unique ,
    language varchar ,
    experience text,
    possibility_remote boolean,
    primary key (id)
);

create table lesson_participants(
    title varchar(55) not null unique ,
    teaching_instructor varchar not null,
    studying_studs varchar,
    room int not null ,
    primary key (title),
    foreign key (teaching_instructor) references instructors(full_name),
    foreign key (studying_studs) references students(full_name)
);
```

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

4.

```
insert into products values ('face1' , 'AloeCream' , 'cream from aloe used for face' , 24.99 );
select id from products;
insert into products values ('hair1' , 'Pour' , 'hair powder' , 5.5 );
delete from products
where id='face1';
select * from products;
--
insert into customers values(1, 'James Charles' , '2004-10-19 10:23:54' , 'LA, Tempe');
select * from customers;
--

insert into orders values (1465, 1 , 11, false );
select * from orders;
--
insert into order_items values (1465, 'hair1' , 2 );
select * from order_items;
```