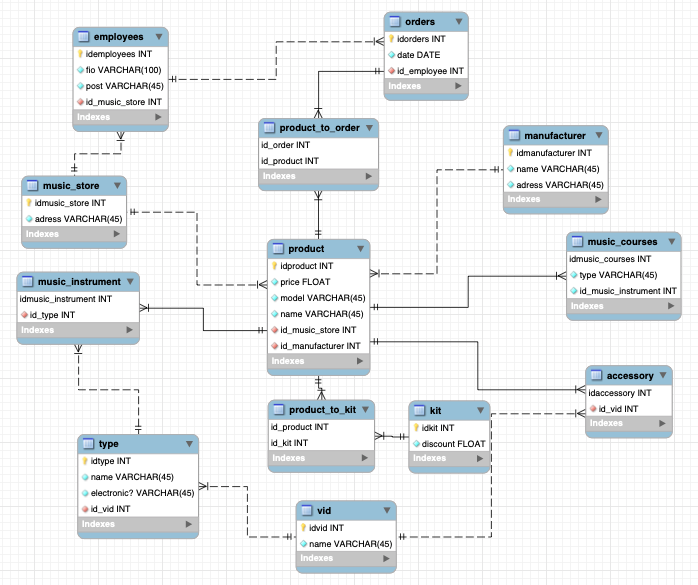
|  |  |  |
| --- | --- | --- |
| Лабораторная работа №4  Реализация SQL-запросов базы данных | Ф.И.О. | Авдеев И. А. |
| Группа | ИВТ-263 |
| Преподаватель | Соколов А.А. |
| Дата сдачи |  |

**Ход выполнения работы:**

1. Запросы, которые вы указали в **функциональных**требованиях (в заголовке указать, что за требование) (**15 шт. +**)
2. **UPDATE**в **разных**таблицах, с **WHERE**, можно условно, например, изменить заранее созданные некорректные данные (**5 шт.**)
3. **DELETE**в **разных**таблицах, с **WHERE**, можно условно, например, удалить заранее созданные некорректные данные (**5 шт.**)
4. **SELECT, DISTINCT, WHERE, AND/OR/NOT, IN, BETWEEN**, различная работа с **датами**и **числами**, преобразование данных, **IS NULL**, **AS**для таблиц и столбцов и др. в различных вариациях (**20 шт. +**)
5. **LIKE**и другая работа со строками (**5-7 шт.+**)
6. **SELECT INTO**или **INSERT SELECT**, что поддерживается СУБД (**2-3 шт.**). Для использования запроса **INSERT SELECT** вначале можно создать новую тестовую таблицу или несколько, в которые будут скопированы данные из существующих таблиц с помощью данного запроса. Код создания таблиц также приложить в лабораторную работу
7. **JOIN: INNER, OUTER (LEFT, RIGHT, FULL), CROSS, NATURAL**, разных, в различных вариациях, несколько запросов с более, чем одним **JOIN**(**20 шт.+**)
8. **GROUP BY**(некоторые с **HAVING**), с **LIMIT, ORDER BY**(**ASC**|**DESC**) вместе с **COUNT**, **MAX**, **MIN, SUM, AVG** в различных вариациях, можно по отдельности (**20 шт.+**)
9. **UNION**, **EXCEPT**, **INTERSECT**, что поддерживается СУБД (**3-5 шт.**)
10. Вложенные **SELECT**с **ALL, ANY, EXISTS**(**3-5 шт.**)
11. **GROUP\_CONCAT** и другие разнообразные функции SQL (**2-3 шт.**)
12. Сложные запросы, входящие в большинство групп выше, т.е. **SELECT** ... **JOIN**... **JOIN** ... **WHERE**... **GROUP BY** ... **ORDER BY** ... **LIMIT** ...; (**5-7 шт. +**), можно написать больше вместо простых.

**Реализация собственной БД**



#Функциональные требоавания: добавить товар в заказ, изменить цену товара, оформить заказ, показать текущую цену товара

USE lab3;

#1

#добавление товара в заказ

INSERT product\_to\_order(id\_order, id\_product)

VALUES (1, 1);

INSERT product\_to\_order(id\_order, id\_product)

VALUES (2, 1);

INSERT orders( date, id\_employee)

VALUES ('2021-06-01', 20);

INSERT product\_to\_order(id\_order, id\_product)

VALUES (3, 4);

INSERT product\_to\_order(id\_order, id\_product)

VALUES (3, 5);

#изменение цены товара

UPDATE product set price = 5000 where idproduct = 11;

UPDATE product set price = 20000 where idproduct = 1;

UPDATE product set price = 1000 where idproduct = 3;

UPDATE product set price = 3000 where idproduct = 10;

UPDATE product set price = 15000 where idproduct = 2;

#оформление заказа

INSERT orders(idorders, date, id\_employee)

VALUES (8, '2021-03-01', 19);

INSERT orders(idorders, date, id\_employee)

VALUES (4, '2014-02-14', 13);

INSERT orders(idorders, date, id\_employee)

VALUES (5, '2021-01-01', 14);

INSERT orders(idorders, date, id\_employee)

VALUES (6, '2021-02-10', 15);

INSERT orders(idorders, date, id\_employee)

VALUES (7, '2021-04-22', 16);

#показ текущей цены товара

select price from lab3.product where idproduct = 1;

select price from lab3.product where idproduct = 2;

select price from lab3.product where idproduct = 3;

select price from lab3.product where idproduct = 4;

select price from lab3.product where idproduct = 10;

#2 UPDATE в разных таблицах, с WHERE

UPDATE product set price = 15000 where idproduct = 5;

UPDATE employees set fio = 'Бибиков А.Н.' where idemployees = 13;

UPDATE product set model = 'F310 super' where idproduct = 1;

UPDATE music\_store set adress = 'Михайловка' where idmusic\_store = 1;

UPDATE music\_courses set type = '20ти дневный курс' where idmusic\_courses = 11;

#3 DELETE в разных таблицах, с WHERE

DELETE from music\_store where idmusic\_store = 10;

DELETE from orders where idorders = 8;

INSERT manufacturer(idmanufacturer, name, adress)

VALUES (11, 'test', 'test region');

DELETE from manufacturer where idmanufacturer = 11;

DELETE from employees where idemployees = 18;

INSERT music\_courses(idmusic\_courses, type, id\_music\_instrument)

VALUES (12, 'test', 6);

DELETE from music\_courses where idmusic\_courses = 12;

#4 SELECT, DISTINCT, WHERE, AND/OR/NOT, IN, BETWEEN, различная работа с датами и числами, преобразование данных, IS NULL, AS

select fio from employees where idemployees = 11;

select type from music\_courses where idmusic\_courses = 11;

select price from lab3.product where idproduct = 3;

select distinct id\_music\_store from employees;

select distinct post from employees;

select distinct id\_kit from product\_to\_kit;

select distinct date from orders;

select \*from employees where post = 'Продавец' or post = 'Кассир';

select \*from product where name = 'Электрогитара' or name = 'Акустическая гитара';

select \*from employees where post = 'Продавец' and id\_music\_store = 5;

select \*from product where id\_music\_store = 1 and price<30000;

select \*from music\_store where adress in ('Волгоград', 'Москва', 'Михайловка');

select \*from product where id\_music\_store in (1, 2, 3);

select \*from music\_store where adress not in ('Волгоград', 'Москва', 'Михайловка');

select \*from product where id\_music\_store not in (5, 3, 8);

select \*from product where price between 1000 and 30000;

select \*from employees where id\_music\_store between 1 and 5;

select \*from orders where date between '2013-01-01' and '2021-06-01';

select \*from product where price<10000 and name is not null;

#5 LIKE

select \*from music\_courses where type like '%20%';

select \*from product where name like '%гитара%';

select \*from employees where idemployees like '1\_';

select \*from product where name like 'Электро%';

select \*from employees where fio like 'Б%';

#6 SELECT INTO или INSERT SELECT

CREATE TABLE orders\_copy (

idorders INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

date date NOT NULL,

id\_employee INT NOT NULL

);

insert INTO orders\_copy (idorders, date, id\_employee) select idorders, date, id\_employee from orders;

CREATE TABLE music\_store\_copy (

idmusic\_store INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

adress varchar(30) not null

);

insert into music\_store\_copy values(idmusic\_store, adress);

delete from music\_store\_copy where idmusic\_store = 1;

insert INTO music\_store\_copy (idmusic\_store, adress) select idmusic\_store, adress from music\_store where adress = 'Сочи' or adress = 'Михайловка';

#7 JOIN: INNER, OUTER (LEFT, RIGHT, FULL), CROSS, NATURAL

select \*from accessory join vid on accessory.id\_vid = vid.idvid;

select idaccessory, name from accessory join vid on accessory.id\_vid = vid.idvid;

select \*from type join vid on type.id\_vid = vid.idvid;

select \*from music\_instrument join type on music\_instrument.id\_type = type.idtype;

select \*from music\_courses join music\_instrument on music\_courses.id\_music\_instrument = music\_instrument.idmusic\_instrument;

select \*from music\_instrument join type on music\_instrument.id\_type = type.idtype;

select \*from employees join music\_store on employees.id\_music\_store = music\_store.idmusic\_store;

select \*from orders join employees on orders.id\_employee = employees.idemployees;

select \*from accessory right join vid on accessory.id\_vid = vid.idvid;

select \*from type right join vid on type.id\_vid = vid.idvid;

select \*from music\_instrument right join type on music\_instrument.id\_type = type.idtype;

select \*from music\_courses right join music\_instrument on music\_courses.id\_music\_instrument = music\_instrument.idmusic\_instrument;

select \*from music\_instrument right join type on music\_instrument.id\_type = type.idtype;

select \*from employees right join music\_store on employees.id\_music\_store = music\_store.idmusic\_store;

select \*from orders right join employees on orders.id\_employee = employees.idemployees;

select \*from employees full join music\_store;

select \*from product cross join music\_store;

select \*from accessory cross join vid;

select \*from music\_instrument natural join type;

select \*from employees natural join music\_store;

select

\*from manufacturer

left join product on manufacturer.idmanufacturer = product.id\_manufacturer;

select

\*from music\_store

left join product on music\_store.idmusic\_store = product.id\_music\_store;

#8 GROUP BY (некоторые с HAVING), с LIMIT, ORDER BY (ASC|DESC) вместе с COUNT, MAX, MIN, SUM, AVG

select music\_store.adress, COUNT(\*) AS count

from product

join music\_store on product.id\_music\_store = music\_store.idmusic\_store

group by id\_music\_store; -- !!!!join + left

select music\_store.adress, COUNT(\*) AS count

from product

join music\_store on product.id\_music\_store = music\_store.idmusic\_store

group by id\_music\_store

having count>1;

select id\_music\_store, sum(price) as sum

from product

group by id\_music\_store;

select music\_store.adress, sum(price) as sum

from product

join music\_store on product.id\_music\_store = music\_store.idmusic\_store

group by id\_music\_store

limit 5;

select music\_store.adress, max(price) as max

from product

join music\_store on product.id\_music\_store = music\_store.idmusic\_store

group by id\_music\_store

limit 3;

select music\_store.adress, min(price) as min

from product

join music\_store on product.id\_music\_store = music\_store.idmusic\_store

group by id\_music\_store

limit 3;

select music\_store.adress, avg(price) as avg

from product

join music\_store on product.id\_music\_store = music\_store.idmusic\_store

group by id\_music\_store;

select price, idproduct from product order by price;

select \*from vid order by name;

select \*from vid order by name desc;

select \*from music\_store order by adress;

select price, id\_music\_store from product order by price desc;

select price, name from product order by price;

select id\_manufacturer, COUNT(\*) AS count

from product

group by id\_manufacturer;

select id\_manufacturer, sum(price) as sum

from product

group by id\_manufacturer;

select id\_manufacturer, max(price) as sum

from product

group by id\_manufacturer;

#9 UNION, EXCEPT, INTERSECT

select adress

from manufacturer

union

select adress

from music\_store;

#10 SELECT с ALL, ANY, EXISTS

select price from product where price>all(select id\_music\_store from product);

select id\_music\_store from product where id\_music\_store=any(select id\_manufacturer from product);

select price from product where price>exists(select id\_manufacturer from product);

#11 GROUP\_CONCAT

SELECT GROUP\_CONCAT(fio) FROM employees;

SELECT GROUP\_CONCAT(name SEPARATOR ' + ') as 'Наименования' FROM type;

SELECT GROUP\_CONCAT(price SEPARATOR ' + ') as 'Цены' FROM product;

#12 Сложные запросы

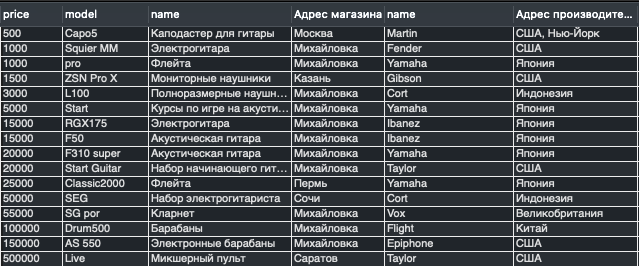
select p.price, p.model, p.name, m.adress as 'Адрес магазина', mn.name, mn.adress as 'Адрес производителя'

from product p

join music\_store m on p.id\_music\_store=m.idmusic\_store

join manufacturer mn on p.id\_manufacturer=mn.idmanufacturer

order by p.price;



select o.idorders, o.date, o.id\_employee, e.fio, e.post, e.id\_music\_store, m.adress

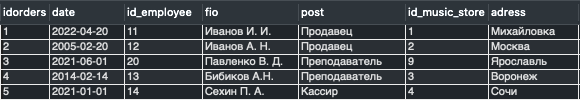
from orders o

join employees e on o.id\_employee=e.idemployees

join music\_store m on e.id\_music\_store=m.idmusic\_store

order by o.idorders

limit 5;



select p.idproduct, p.model, t.name, `t`.`electronic?`, v.name

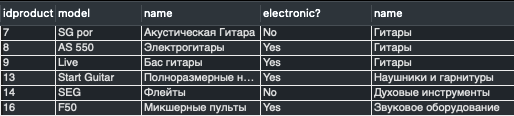
from product p

join type t on p.idproduct=t.idtype

join vid v on t.id\_vid=v.idvid

where p.price>5000

order by p.idproduct;



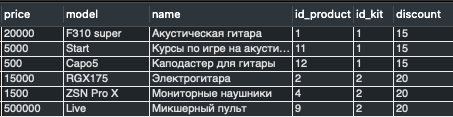
select p.price, p.model, p.name, ptk.id\_product, ptk.id\_kit, k.discount

from product\_to\_kit ptk

join product p on ptk.id\_product=p.idproduct

join kit k on ptk.id\_kit=k.idkit

order by k.discount;



select o.idorders, o.date, p.price, p.model, p.name, ms.adress, mn.name, mn.adress

from orders o

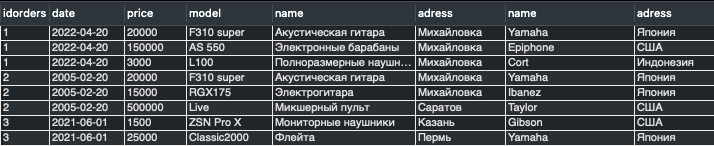
join product\_to\_order pto on o.idorders=pto.id\_order

join product p on pto.id\_product=p.idproduct

join music\_store ms on p.id\_music\_store=ms.idmusic\_store

join manufacturer mn on p.id\_manufacturer=mn.idmanufacturer

order by o.idorders;



**Модификация:**

#1) для сотрудника по фио за период с 5 по 10 вывести сумму, на которую он оформил

select e.fio, sum(p.price)

from employees e

join orders o on e.idemployees=o.id\_employee

join product\_to\_order pto on o.idorders=pto.id\_order

join product p on pto.id\_product=p.idproduct

where o.date between '2013-01-01' and '2022-06-01'

group by e.fio;



#2) для магазина топ 5 по количеству инструментов для каждого типа

select t.name, count(t.name) as count

from music\_store ms

join product p on ms.idmusic\_store=p.id\_music\_store

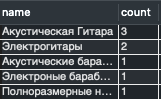
join music\_instrument mi on p.idproduct=mi.idmusic\_instrument

join type t on mi.id\_type=t.idtype

where ms.idmusic\_store=1

group by t.name

limit 5;



#3) вывести все аксессуары для конкретного музыкального инструмента

select mi.idmusic\_instrument, p.name, p.model, mi.id\_type, t.name, t.`electronic?`, t.id\_vid, v.name, a.idaccessory, pr.name as name\_accessory

from music\_instrument mi

join product p on mi.idmusic\_instrument=p.idproduct

join type t on mi.id\_type=t.idtype

join vid v on t.id\_vid=v.idvid

join accessory a on v.idvid=a.id\_vid

join product pr on a.idaccessory=pr.idproduct

where mi.idmusic\_instrument = 1;

