Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехника и комплексная автоматизация*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по курсу: «Базы данных»

Студент	Новокшанов Евгений Андреевич
Группа	РК6-46Б
Тип задания	Лабораторная работа
Тема лабораторной работы	Простые запросы

Студент _____ **Новокшанов Е.А.**
подпись, дата *фио.*

Преподаватель _____ **Пивоварова Н.В.**
подпись, дата *фио.*

Оценка _____

Москва, 2022 г.

Задание

Вариант 9. Ресторан.

В ресторане официанты принимают заказы от посетителей. Об официантах в БД хранятся следующие данные: уникальный номер, паспортные данные, дата приема на работу. Предусмотрена также дата увольнения, которая для работающих официантов не заполняется. Пространство ресторана поделено на зоны (количество столиков, vip, для курящих/нет, со специальными детскими местами и пр.). В каждой зоне располагаются столы. Посетителям за столами предлагается меню. В меню указывается шифр блюда, название блюда, цена, вес в граммах. Любой свободный официант может принять заказ. В заказе указывается номер стола, дата и время заказа. Сохраняется также общая стоимость заказа, которая вычисляется при окончательном расчете с посетителем. Заказ состоит из строк заказа. Каждая строка заказа ссылается на один из пунктов меню и показывает количество заказанных блюд.

Простые запросы:

1. Показать максимальную стоимость заказа из всех принятых в марте 2020 года.
2. Показать все сведения об официантах, принятых на работу в сентябре 2020 года.
3. Показать все сведения о заказах, принятых за последние 2 дня.
4. Найти сумму всех заказов, принятых каждым официантом за дату XXX.
5. Найти дату рождения самого молодого официанта.
6. Найти всех официантов, фамилии которых начинаются с М.

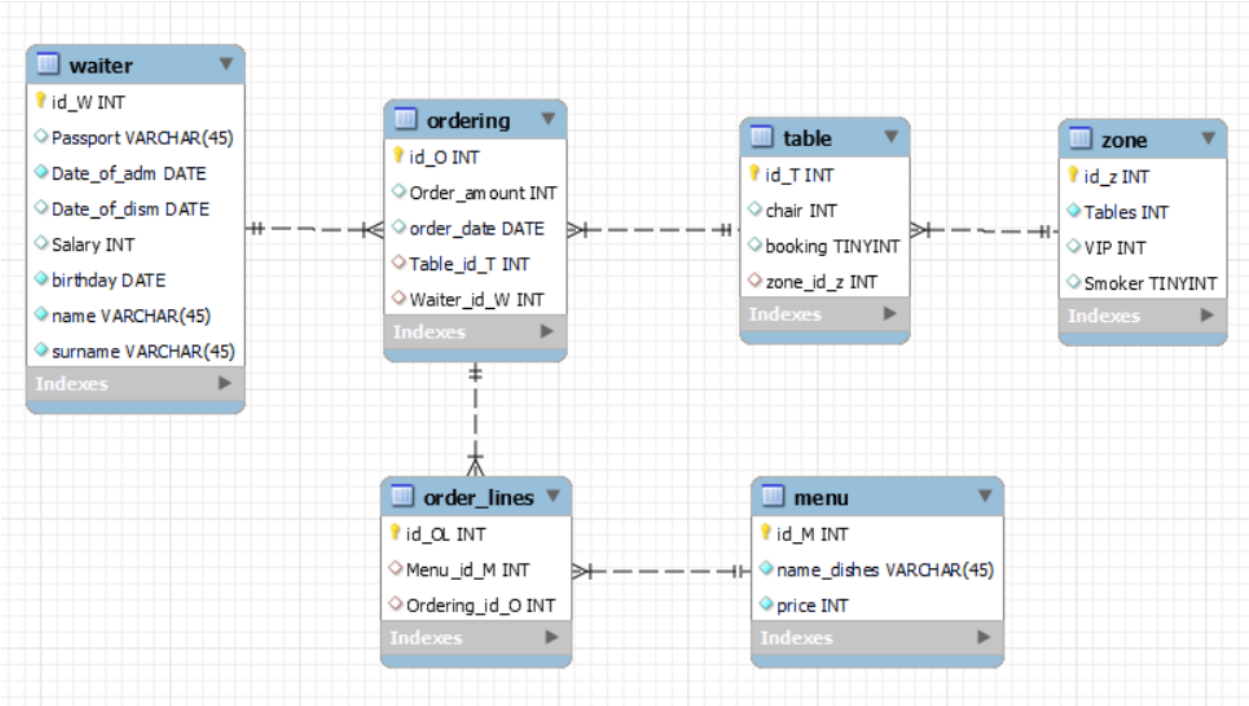
Сложные запросы:

1. Составить отчет о заказе блюд в марте 2020 года по форме: Номер блюда, название блюда, общее количество заказов блюда, общая выручка от заказов блюда.
2. Составить отчет о работе официантов в марте 2020 года по форме: Уникальный номер официанта, Фамилия официанта, общее количество принятых заказов, общая сумма принятых заказов.
3. Показать все сведения о самом молодом официанте.
4. Показать сведения об официантах, которые пока не приняли ни одного заказа (с помощью левостороннего соединения)

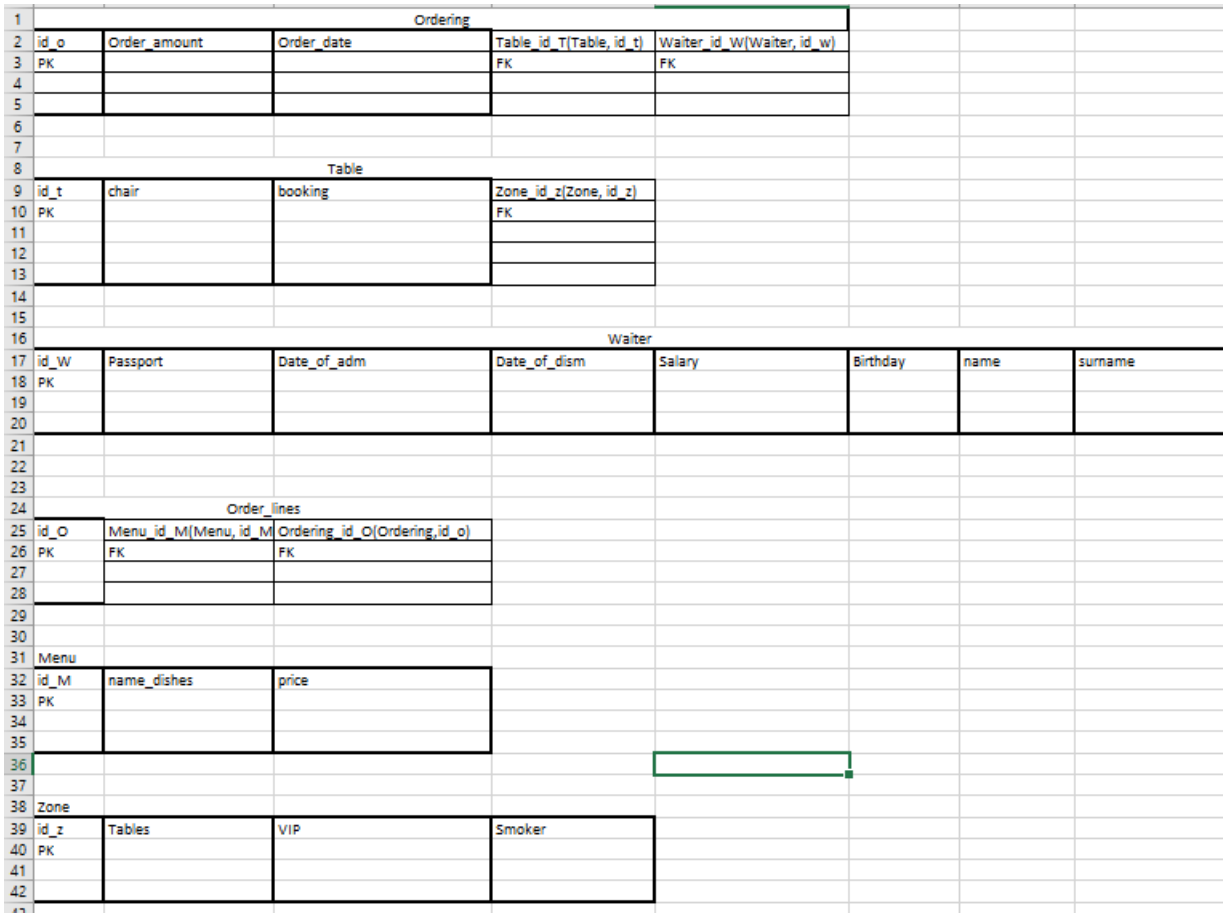
5. Показать сведения об официантах, не принявших ни одного заказа в марте 2020 года.

6. Показать сведения о блюде, которое чаще всех заказывали в апреле 2020 года (с помощью view)

Инфологическая Модель



Логическая модель



Ordering				
id_o	Order_amount	Order_date	Table_id_T(Table, id_t)	Waiter_id_W(Waiter, id_w)
PK			FK	FK

Table			
id_t	chair	booking	Zone_id_z(Zone, id_z)
PK			FK

Waiter							
id_W	Passport	Date_of_adm	Date_of_dism	Salary	Birthday	name	surname
PK							

Order_lines		
id_O	Menu_id_M(Menu, id_M	Ordering_id_O(Ordering,id_o)
PK	FK	FK

Menu		
id_M	name_dishes	price
PK		

Zone			
id_z	Tables	VIP	Smoker
PK			

Простые запросы

1. Показать максимальную стоимость заказа из всех принятых в марте 2020 года

```
SELECT max(Order_amount) FROM restaurant.ordering
    where order_date like ('2020-03%')
```

2. Показать все сведения об официантах, принятых на работу в сентябре 2020 года.

```
SELECT * FROM restaurant.waiter
    where Date_of_adm like ('2020-09%')
```

3. Показать все сведения о заказах, принятых за последние 2 дня.

```
SELECT * FROM restaurant.ordering
    where order_date > date(curdate()-2)
```

4. Найти сумму всех заказов, принятых каждым официантом за дату XXX.

```
SELECT sum(Order_amount) FROM restaurant.ordering
    where order_date like ('2020-01-07')
```

5. Найти дату рождения самого молодого официанта.

```
SELECT min(birthday) FROM restaurant.waiter;
```

6. Найти всех официантов, фамилии которых начинаются с М.

```
SELECT name, surname FROM restaurant.waiter
    where surname like ('М%')
```

Сложные запросы

1. Составить отчет о заказе блюд в марте 2020 года по форме: Номер блюда, название блюда, общее количество заказов блюда, общая выручка от заказов блюда.

```
SELECT id_M, name_dishes, count(Menu_id_M) FROM restaurant.menu
    join order_lines ol on ol.Menu_id_M = menu.id_M
    join ordering ord on ol.Ordering_id_O = ord.id_O
    where order_date like ('2020-03%')
    group by id_M, name_dishes
```

2. Составить отчет о работе официантов в марте 2020 года по форме: Уникальный номер официанта, Фамилия официанта, общее количество принятых заказов, общая сумма принятых заказов.

```
SELECT id_W, surname, count(id_O), sum(Order_amount) FROM restaurant.waiter
    join ordering ord on waiter.id_W = ord.Waiter_id_W
    where order_date like ('2020-03%')
    group by id_W, surname
```

3. Показать все сведения о самом молодом официанте.

```
SELECT * FROM restaurant.waiter
where birthday = (select min(birthday) from waiter)
```

4. Показать сведения об официантах, которые пока не приняли ни одного заказа (с помощью левостороннего соединения).

```
SELECT waiter.* FROM restaurant.waiter
    left join ordering ord on waiter.id_W = ord.Waiter_id_W
    where Waiter_id_W is NULL
```

5. Показать сведения об официантах, не принявших ни одного заказа в марте 2020 года.

```
SELECT waiter.* FROM restaurant.waiter
    left join (select * from ordering where order_date like ('2020-03%')) ord on waiter.id_W = ord.Waiter_id_W
    where ord.Waiter_id_W is NULL
```

6. Показать сведения о блюде, которое чаще всех заказывали в апреле 2020 года (с помощью view).

```
create view v as
    SELECT id_M, name_dishes, count(Menu_id_M) count_M   FROM restaurant.menu
    join order_lines ol on ol.Menu_id_M = menu.id_M
    join ordering ord on ol.Ordering_id_O = ord.id_O
    where order_date like ('2020-03%')
    group by id_M, name_dishes;
```

```
select * from v
    where v.count_M = (select max(count_M) from v)
```

Лабораторная работа 3

Процедура:

Разработать процедуру, которая будет принимать на вход год и месяц. В процедуре реализовать формирование и занесение в БД отчетов о популярности блюд по установленной форме.

План:

1) Декларация курсора с выбором таблицы menu и присоединения к ней таблиц Order_lines и Ordering, из которых мы сможем извлечь необходимые данные даты(year и month):

```
declare C1 cursor for
    select Menu_id_M, sum(count_dishes) colvo, sum(price) summa from order_lines OL
    join menu on menu.id_M = OL.Menu_id_M
    join ordering ord on ord.id_O = OL.Ordering_id_O
    where year(order_date) = in_year and month(order_date) = in_month
    group by name_dishes;
```

2) Выполняем проверку на существование записи:

```
if((select count(id_dish) from report where `year` = in_year and `month` = in_month)=0) then
```

3) Открываем курсор:

```
open C1;
```

4) Открываем цикл:

```
while done = 0 do
```

- Извлекаем данные из курсора

```
fetch C1 into `id`, col, `sum`;
```

- Добавляем строку в отчет report

```
insert report
    values(NULL, in_year, in_month, `id`, col, `sum`);
```

5) Закрываем цикл:

```
end while;
```

6) Закрываем курсор:

```
close C1;
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `reported`(in_year INT, in_month INT)
BEGIN
    declare col, `sum`, done int default 0;
    declare `id` varchar(45) default 0;
    declare C1 cursor for
        select Menu_id_M, sum(count_dishes) colvo, sum(price) summa from order_lines OL
        join menu on menu.id_M = OL.Menu_id_M
        join ordering ord on ord.id_O = OL.Ordering_id_O
        where year(order_date) = in_year and month(order_date) = in_month
        group by name_dishes;
    declare Exit handler for sqlstate '02000'
        set done = 1;
    open C1;
    if((select count(id_dish) from report where `year` = in_year and `month` = in_month)=0) then
        while done = 0 do
            fetch C1 into `id`, col, `sum`;
            insert report
                values(NULL, in_year, in_month, `id`, col, `sum`);
            end while;
    end if;
    close C1;
END
```

Триггер:

Разработать триггер, который будет срабатывать при добавлении строки заказа. В триггере реализовать обновление соответствующей строки в отчете о популярности блюд.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `order_lines_AFTER_INSERT` AFTER INSERT ON `order_lines` FOR EACH ROW BEGIN
    declare col, `sum`,  year_out, month_out, done int default 0;
    declare `id` varchar(45) default 0;
    (select year(order_date), month(order_date), Menu_id_M, sum(count_dishes), price* count_dishes   from order_lines OL
                join ordering ord on ord.id_O = OL.Ordering_id_O
                join menu on menu.id_M = OL.Menu_id_M
                where id_OL = new.id_OL
                group by Menu_id_M
            ) into year_out, month_out, id, col, sum;
    if((select count(id_dish) from report where `year` = year_out and `month` = month_out and
                                                id_dish = new.Menu_id_M)=0) then
            insert report
                values(NULL, year_out, month_out, `id`, col, `sum`);
        else
            update report
                set month_amount = `sum`
                where id_dish = new.Menu_id_M
                and `year` = year_out and `month` = month_out;
    end if;
END
```