	<p>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)</p>
---	--

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине: «Базы данных»

Студент	Долженко Анастасия Тимофеевна
Группа	РК6-42Б
Тип задания	Лабораторная работа 1,2,3,4
Тема лабораторной работы	Интернет-провайдер

Студент	_____	<u>Долженко А.Т.</u> фамилия, и.о.
Преподаватель	_____	<u>Пивоварова Н.В.</u> фамилия, и.о.

Москва, 2024 г.

Задание

Родименко

Вариант 6. Интернет-провайдер.

Интернет-провайдер предоставляет своим клиентам интернет-услуги. С каждым клиентом заключается договор на предоставление услуг. В договоре указываются паспортные данные клиента, дата заключения договора. При расторжении договора проставляется дата прекращения оказания услуг.

За каждую подключенную услугу с клиента ежедневно списывается с баланса стоимость услуги. При этом фиксируется дата и время изменения баланса. Если баланс после списания стоимости услуги становится отрицательным, то клиент блокируется. Активный клиент (с положительным балансом) имеет статус 1. Заблокированный – статус 2.

Стоимость услуг время от времени меняется. При изменении стоимости для истории сохраняется старое значение, новое значение и дата изменения.

Каждый клиент может подключить/отключить любое количество услуг в любое время. Даты включения/отключения услуг у клиента сохраняются.

Простые запросы.

1. Показать все сведения о клиентах, родившихся в марте 2000 года.
2. Показать все сведения о клиентах, заключивших договора за последние 60 дней.
3. Показать общую величину положительного баланса всех клиентов на текущий момент.
4. Показать, у скольких клиентов подключена каждая из услуг.
5. Определить максимальную цену для услуг, уникальный код которых начинается с XXX
6. Показать, сколько раз изменялся баланс каждого из клиентов в марте 2020 года.

Сложные запросы.

1. Создать отчет об услугах за 2020 год по форме: месяц, код услуги, название услуги, количество подключений за месяц, количество отключений за месяц.
2. Создать отчет об изменениях баланса клиентов, номера которых находятся в диапазоне от XX1 до XX2 за январь 2020 года по форме: Номер клиента, Ф.И.О. клиента, Общая сумма пополнений баланса.
3. Покажите все сведения о самом молодом клиенте компании.
4. Показать все сведения о клиентах, которые ни разу не изменяли баланс (с помощью левостороннего соединения)
5. Показать все сведения о клиентах, ни разу не изменявших баланс в марте 2020 года.
6. Показать все сведения о клиенте/ах, наиболее активно расходовавших баланс в марте 2020 года (с помощью view).

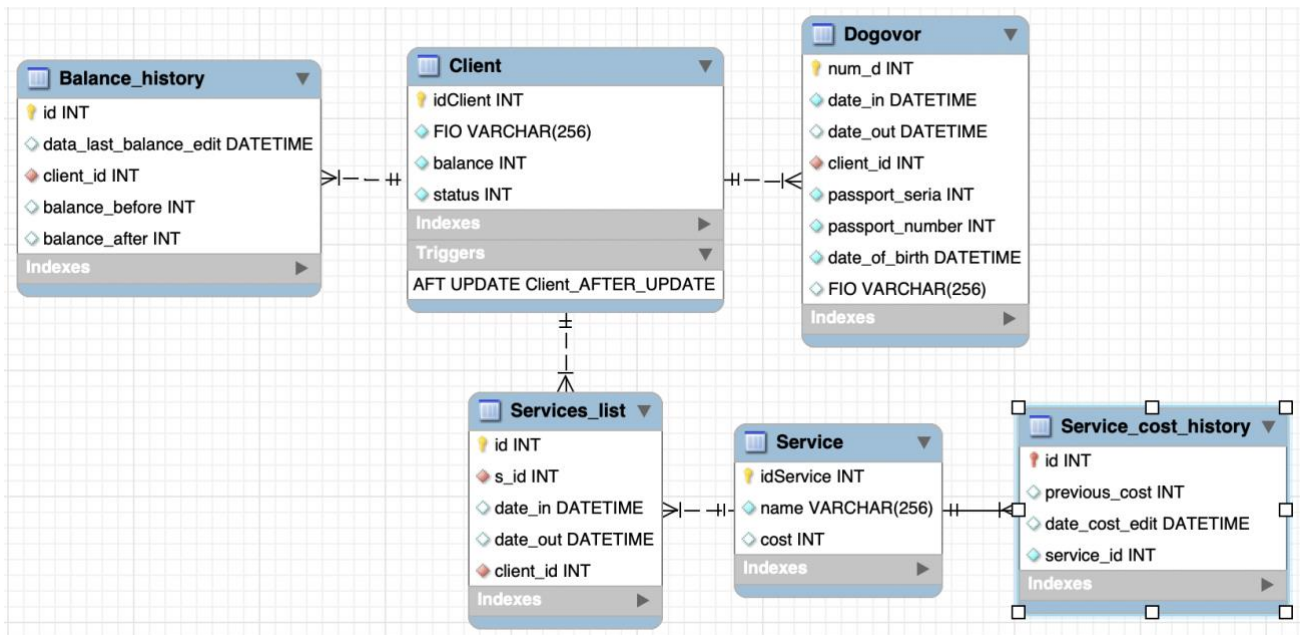
Триггеры и хранимые процедуры.

Разработать процедуру, которая будет принимать на вход дату. В процедуре реализовать списание с баланса клиентов стоимости всех подключенных на этот день услуг и добавление записи в историю изменения балансов.

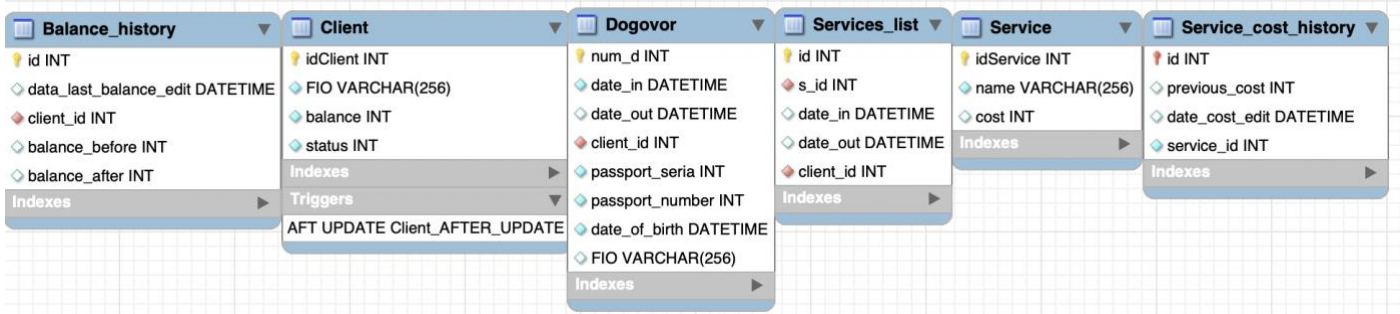
Разработать триггер, который будет срабатывать при изменении текущего баланса клиента. В триггере реализовать добавление соответствующей записи в историю балансов клиента.

Клиент подключает/отключает услуги

Инфологическая модель



Логическая модель



Простые запросы

1. Показать все сведения о клиентах, родившихся в марте 2000 года.

```
select * from Dogovor where year(date_of_birth) = 2000 and month(date_of_birth) = 3;
```

2. Показать все сведения о клиентах, заключивших договора за последние 60 дней.

```
select * from Dogovor where datediff(curdate(), date_in) <= 60;
```

3. Показать общую величину положительного баланса всех клиентов на текущий момент.

```
select sum(balance) from Client where status = 1;
```

4. Показать у скольких клиентов подключена каждая из услуг.

```
select s_id as service_id, count(*) from Services_list group by s_id;
```

5. Определить максимальную цену для услуг, уникальный код которых начинается с XXX.

```
select max(cost) from Service where name like 'интернет%';
```

6. Показать сколько раз изменялся баланс каждого из клиентов в марте 2020 года.

```
select client_id, count(data_last_balance_edit)
from Balance_history
where year(data_last_balance_edit) = 2020 and month(data_last_balance_edit) = 3
group by client_id;
```


Сложные запросы

1. Создать отчет об услугах за 2020 год по форме: месяц, код услуги, название услуги, количество подключений за месяц, количество отключений за месяц.

```
select month(date_in) as month_, s_id, name, count(date_in) as amount_in, count(date_out) as amount_out
from Services_list join Service on Services_list.s_id=Service.idService
where year(date_in)=2020
group by month_, s_id, name;
```

2. Создать отчет об изменениях баланса клиентов, номера которых находятся в диапазоне от XX1 до XX2 за январь 2020 года по форме: Номер клиента, ФИО, Общая сумма пополнений баланса.

```
select client_id as client_number, FIO, sum(balance_after - balance_before) as sum_of_balance_edit
from Balance_history join Client on Client.idClient=Balance_history.client_id
where idClient < 10 and idClient > 4
    and month(data_last_balance_edit)=1 and year(data_last_balance_edit)=2020
    and balance_after - balance_before > 0
group by client_id;
```

3. Показать все сведения о самом молодом клиенте компании.

```
select * from Dogovor where date_of_birth = (select max(date_of_birth) from Dogovor);
```

4. Показать все сведения о клиентах, которые ни разу не изменяли баланс (с помощью левостороннего соединения).

```
select num_d, date_in, date_out, client_id, passport_seria, passport_number, date_of_birth, FIO
from Dogovor left join Balance_history using(client_id)
where id is NULL;
```

5. Показать все сведения о клиентах, которые ни разу не изменяли баланс в марте 2020 года.

```
select num_d, date_in, date_out, client_id, passport_seria, passport_number, date_of_birth, FIO
from Dogovor left join (select * from Balance_history where month(data_last_balance_edit) = 3 and year(data_last_balance_edit) = 2020)table1
using(client_id)
where id is NULL;
```

6. Показать все сведения о клиенте/ах, наиболее активно расходовавших баланс в марте 2020 года.

```
create or replace view traty as
select client_id, count(data_last_balance_edit) as amount
from Balance_history join Dogovor using(client_id)
where month(data_last_balance_edit)=3 and year(data_last_balance_edit)=2020
group by client_id;
select * from traty where amount = (select max(amount) from traty);
```

Процедура

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `updateBalance`(input_date datetime)
2 BEGIN
3
4     declare done integer default 0;
5     declare id_client integer;
6     declare service_id integer;
7     declare service_cost integer;
8     declare balanceClient integer;
9
10    declare c1 cursor for
11        select client_id, s_id, cost
12        from Service join Services_list on Service.idService = Services_list.s_id
13        where datediff(input_date, date_in) > 0 and
14            (datediff(date_out, input_date) > 0 or date_out is null);
15
16
17    declare continue handler for sqlstate '02000' set done = 0;
18    open c1;
19
20    while done = 0 do
21        fetch c1 into id_client, service_id, service_cost;
22
23        set balanceClient := (select balance from Client where idClient = id_client);
24
25        insert into Balance_history(data_last_balance_edit, client_id, balance_before, balance_after)
26            values (curdate(), id_client, balanceClient, balanceClient-service_cost);
27
28        update Client set Client.`balance` = balanceClient-service_cost where idClient = id_client;
29
30    end while;
31
32    close c1;
33 END
```

Триггер

```
1 CREATE DEFINER = CURRENT_USER TRIGGER Client_AFTER_UPDATE AFTER UPDATE ON Client FOR EACH ROW
2 BEGIN
3
4     insert into Balance_history(data_last_balance_edit, client_id, balance_before, balance_after)
5         values(curdate(), NEW.idClient, OLD.balance, NEW.balance);
6
7     END
```