

## ЛАБОРАТОРНАЯ РАБОТА 3

### РЕАЛИЗАЦИЯ СИСТЕМЫ ОБМЕНА ДАННЫМИ С БД

Цель работы: научиться разрабатывать систему обмена данными с БД.

Ход работы

#### 1. Концептуальное проектирование

На рисунке 1 представлена концептуальная модель информационной системы.

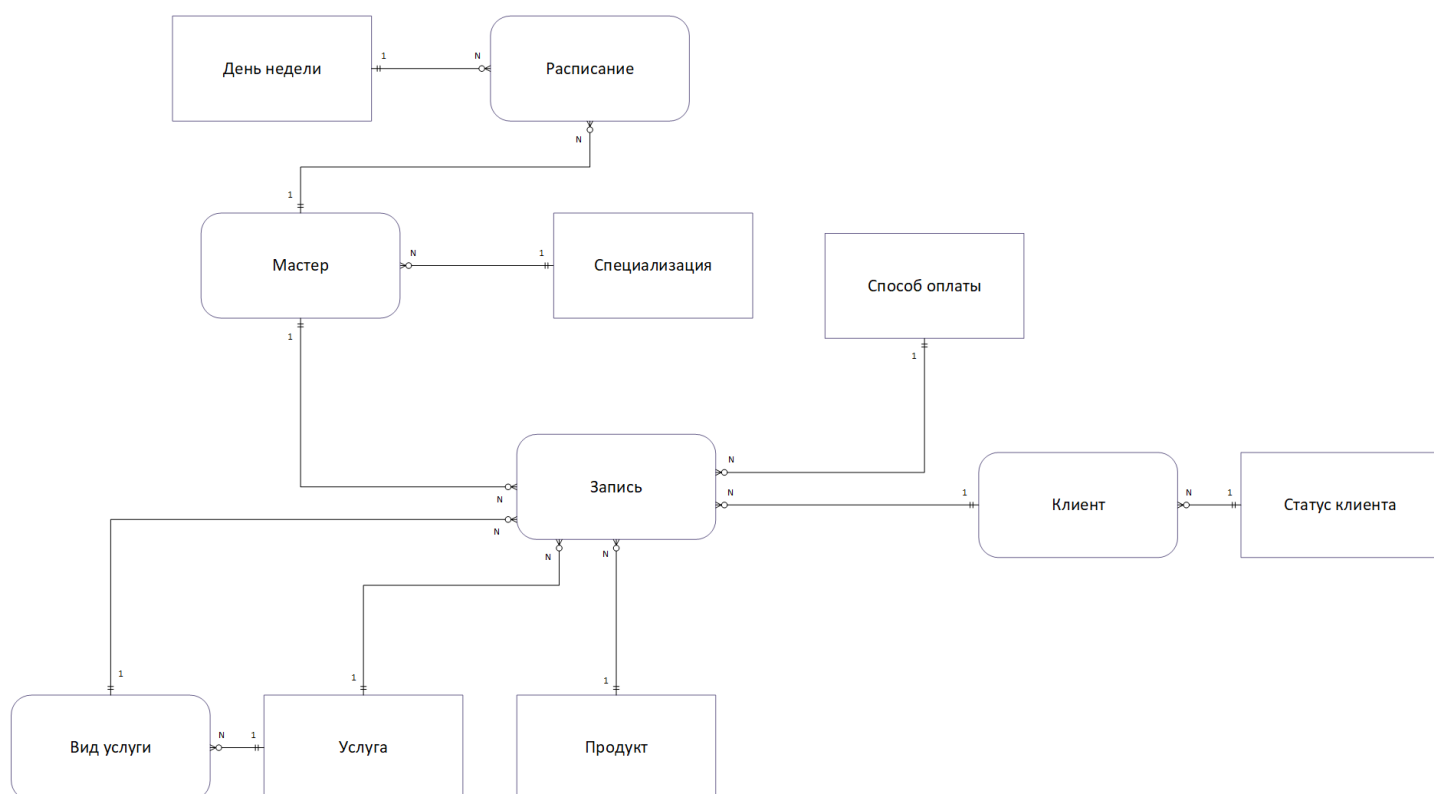


Рисунок 1 – Концептуальная модель

## 2. Проектирование логической модели БД

На рисунке 2 представлена логическая модель базы данных салона красоты.

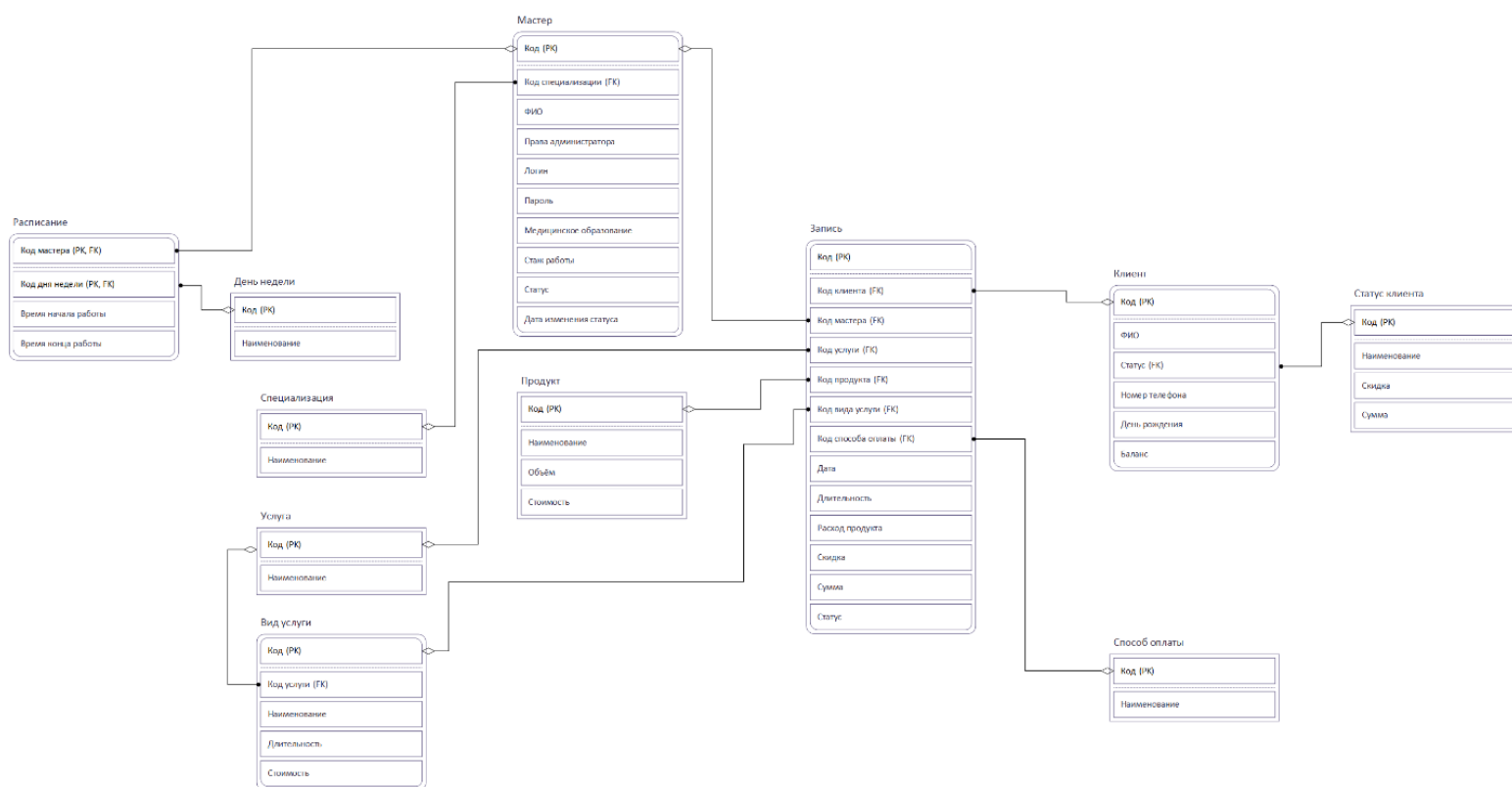


Рисунок 2 – Логическая модель БД

### Описание мощности связей между сущностями

Максимальная мощность связи между сущностями «День недели» и «Расписание» –  $1:N$  – один экземпляр сущности «День недели» может относиться ко многим экземплярам сущности «Расписание», и один экземпляр сущности «Расписание» может относиться только к одному экземпляру сущности «День недели».

Максимальная мощность связи между сущностями «Мастер» и «Расписание» –  $1:N$  – один экземпляр сущности «Мастер» может относиться ко многим экземплярам сущности «Расписание», и один экземпляр сущности «Расписание» может относиться только к одному экземпляру сущности «Мастер».

Максимальная мощность связи между сущностями «Специализация» и «Мастер»  $1:N$  – один экземпляр сущности «Специализация» может относиться ко многим экземплярам сущности «Мастер», и один экземпляр сущности «Мастер» может относиться только к одному экземпляру сущности «Специализация».

Максимальная мощность связи между сущностями «Статус клиента» и «Клиент»  $1:N$  – один экземпляр сущности «Статус клиента» может относиться ко многим экземплярам сущности «Клиент», и один экземпляр сущности «Клиент» может относиться только к одному экземпляру сущности «Статус клиента».

Максимальная мощность связи между сущностями «Услуга» и «Вид услуги»  $1:N$  – один экземпляр сущности «Услуга» может относиться ко многим экземплярам сущности «Вид услуги», и один экземпляр сущности «Вид услуги» может относиться только к одному экземпляру сущности «Услуга».

Максимальная мощность связи между сущностями «Продукт» и «Запись»  $1:N$  – один экземпляр сущности «Продукт» может относиться ко многим экземплярам сущности «Запись», и один экземпляр сущности «Запись» может относиться только к одному экземпляру сущности «Продукт».

Максимальная мощность связи между сущностями «Услуга» и «Запись»  $1:N$  – один экземпляр сущности «Услуга» может относиться ко многим экземплярам сущности «Запись», и один экземпляр сущности «Запись» может относиться только к одному экземпляру сущности «Услуга».

Максимальная мощность связи между сущностями «Вид услуги» и «Запись»  $1:N$  – один экземпляр сущности «Вид услуги» может относиться ко многим экземплярам сущности «Запись», и один экземпляр сущности «Запись» может относиться только к одному экземпляру сущности «Вид услуги».

Максимальная мощность связи между сущностями «Клиент» и «Запись» –  $1:N$  – один экземпляр сущности «Клиент» может относиться ко многим экземплярам сущности «Запись», и один экземпляр сущности «Запись» может относиться только к одному экземпляру сущности «Клиент».

Максимальная мощность связи между сущностями «Мастер» и «Запись» –  $1:N$  – один экземпляр сущности «Мастер» может относиться ко многим экземплярам сущности «Запись», и один экземпляр сущности «Запись» может относиться только к одному экземпляру сущности «Мастер».

Максимальная мощность связи между сущностями «Способ оплаты» и «Запись» –  $1:N$  – один экземпляр сущности «Способ оплаты» может относиться ко многим экземплярам сущности «Запись», и один экземпляр сущности «Запись» может относиться только к одному экземпляру сущности «Способ оплаты».

#### Описание связей модели

Сущности модели имеют следующие связи:

- каждый КЛИЕНТ <оформляет> ЗАПИСЬ;
- каждый МАСТЕР <ведёт> ЗАПИСЬ;
- каждый МАСТЕР <имеет> СПЕЦИАЛИЗАЦИЮ;
- каждый МАСТЕР <работает по> РАСПИСАНИЮ;
- каждое РАСПИСАНИЕ <составляется по> ДНЯМ НЕДЕЛИ;
- каждый КЛИЕНТ <имеет> СТАТУС;
- каждая УСЛУГА <имеет> ВИДЫ УСЛУГИ;
- каждая ЗАПИСЬ <предполагает расход> ПРЕПАРАТА;
- каждая ЗАПИСЬ <предполагает оказание> УСЛУГИ;
- каждая ЗАПИСЬ <предполагает выполнение> ВИДА УСЛУГИ;
- каждая ЗАПИСЬ <предполагает выбор> СПОСОБА ОПЛАТЫ.

### 3. Проектирование физической модели

На рисунке 3 представлена физическая модель информационной системы в виде схемы, построенной в *SQL Developer Data Modeler*.

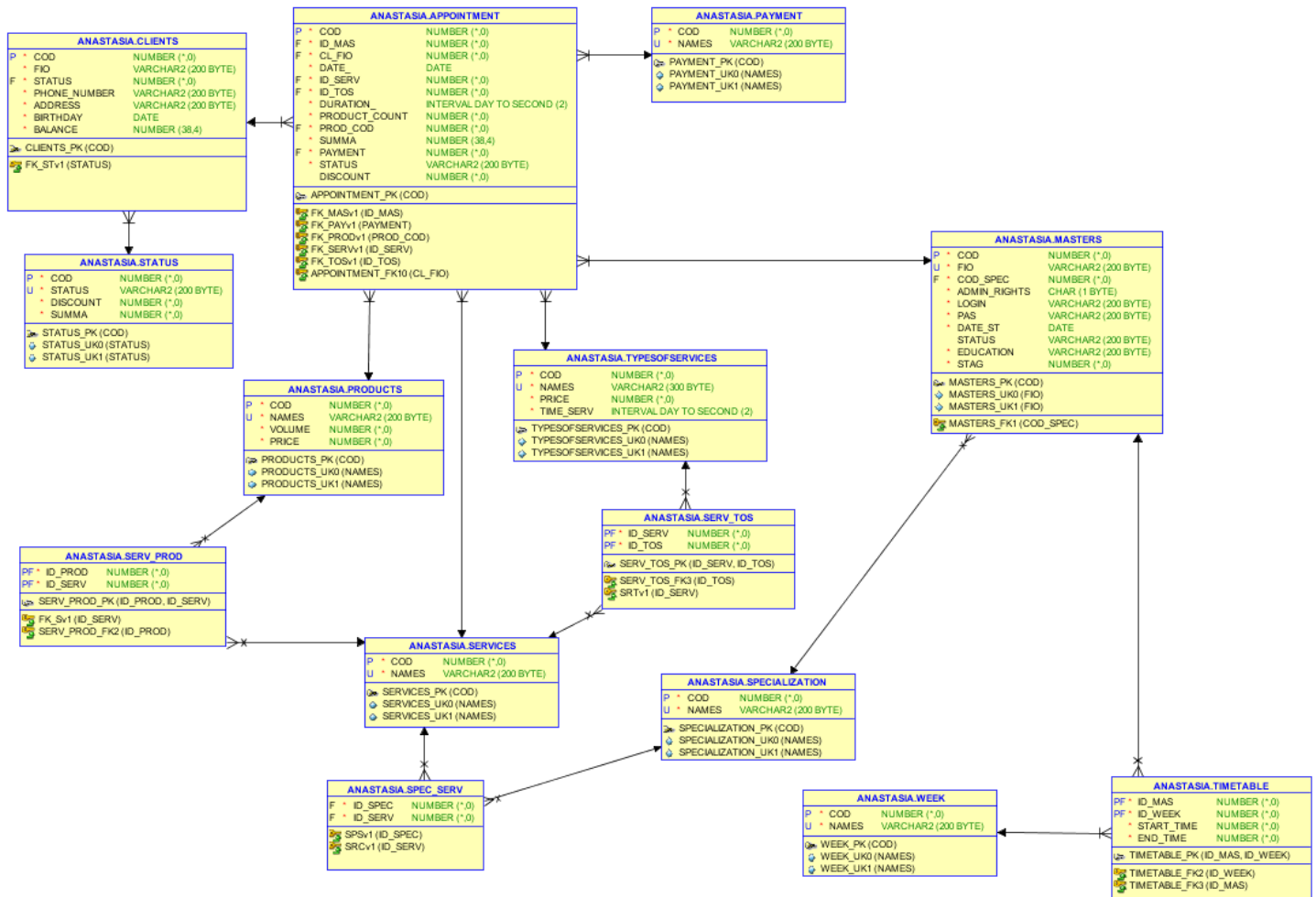


Рисунок 3 – Физическая модель

#### Описание таблиц БД

Таблица *clients* хранит информацию о клиентах, а именно: код (атрибут *cod* типа *integer*, первичный ключ), ФИО (атрибут *fio* типа *varchar*), код статуса (атрибут *status* типа *integer*, внешний ключ), номер телефона (атрибут *phone\_number* типа *varchar*), адрес (атрибут *address* типа *varchar*), дату дня рождения (атрибут *birthday* типа *date*), остаток денежных средств на личном счете (атрибут *balance* типа *integer*).

Таблица *masters* хранит информацию о мастерах салона красоты, а именно: код (атрибут *cod* типа *integer*, первичный ключ), ФИО (атрибут *fio* типа *varchar*), код специальности (атрибут *cod\_spec* типа *integer*, внешний ключ), наличие прав администратора (атрибут *admin\_rights* типа *char*), логин для входа в систему (атрибут *login* типа *varchar*), пароль для входа в систему (атрибут *pas* типа *varchar*), статус (атрибут *status* типа *varchar*), дату изменения статуса (атрибут *date\_st* типа *date*), наличие медицинского образования (атрибут *education* типа *char*), стаж работы (атрибут *stag* типа *integer*).

Таблица *appointment* хранит информацию о записях на прием, а именно: код (атрибут *cod* типа *integer*, первичный ключ), код мастера (атрибут *id\_mas* типа *integer*, внешний ключ), код клиента (атрибут *cl\_fio* типа *integer*, внешний ключ), дату записи (атрибут *date\_* типа *date*), код категории услуг (атрибут *id\_serv* типа *integer*, внешний ключ), код вида услуги (атрибут *id\_tos* типа *integer*, внешний ключ), длительность процедуры (атрибут *duration\_* типа *duration*), предполагаемый расход препарата (атрибут *product\_count* типа *integer*), код расходуемого препарата (атрибут *prod\_cod* типа *integer*, внешний ключ), итоговую стоимость услуги (атрибут *summa* типа *number(38,4)*), код способа оплаты (атрибут *payment* типа *integer*, внешний ключ), статус (атрибут *status* типа *varchar*), учтенная скидка (атрибут *discount* типа *integer*).

Таблица *payment* хранит информацию о способах оплаты, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *names* типа *varchar*).

Таблица *products* хранит информацию об используемых препаратах, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *names* типа *varchar*), объем (атрибут *volume* типа *integer*), стоимость за указанный объем (атрибут *price* типа *integer*).

Таблица *services* хранит информацию о категориях услуг, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *names* типа *varchar*).

Таблица *serv\_prod* предназначена для связи используемых препаратов с категориями услуг, при выполнении которых они используются, и содержит код препарата (атрибут *id\_prod* типа *integer*, внешний ключ) и код услуги (атрибут *id\_serv* типа *integer*, внешний ключ). Атрибуты таблицы входят в составной первичный ключ.

Таблица *typesofservices* содержит информацию о видах услуг, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *names* типа *varchar*), стоимость (атрибут *price* типа *integer*), длительность (атрибут *time\_serv* типа *duration*).

Таблица *serv\_tos* предназначена для связи категорий услуг с их видами и содержит код категории услуг (атрибут *id\_serv* типа *integer*, внешний ключ) и код вида услуги (атрибут *id\_tos* типа *integer*, внешний ключ). Атрибуты таблицы входят в составной первичный ключ.

Таблица *specialization* хранит информацию о специализациях мастеров салона красоты, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *names* типа *varchar*).

Таблица *spec\_serv* предназначена для связи специализаций с категориями услуг, которые могут оказывать мастера в зависимости от их специализации, и содержит код категории услуг (атрибут *id\_serv* типа *integer*, внешний ключ) и код специализации (атрибут *id\_spec* типа *integer*, внешний ключ). Атрибуты таблицы входят в составной первичный ключ.

Таблица *status* содержит информацию о возможных статусах клиентов, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *status* типа *varchar*); скидку, которая будет учитываться при оформлении записи на прием (атрибут *discount* типа *integer*); минимальную

сумму, на которую клиенту должны быть оказаны услуги, чтобы ему был присвоен статус (атрибут *summa* типа *integer*).

Таблица *week* содержит информацию о днях недели, а именно: код (атрибут *cod* типа *integer*, первичный ключ), наименование (атрибут *names* типа *varchar*).

Таблица *timetable* содержит информацию о расписаниях мастеров, а именно: код мастера (атрибут *id\_mas* типа *integer*, внешний ключ), код дня недели (атрибут *id\_week* типа *integer*, внешний ключ), время начала смены (атрибут *start\_time* типа *integer*), время конца смены (атрибут *end\_time* типа *integer*). Атрибуты *id\_mas* и *id\_week* входят в составной первичный ключ.

## Запросы к БД

### Функциональные запросы

«Добавить мастера в БД»:

```
INSERT INTO
masters(fio,cod_spec,admin_rights,login,pas,date_
st,status,education,stag) VALUES (@fio,(SELECT cod
FROM specialization WHERE names =
@specialization),@admin_rights,@login,
@password, @date, 'Свободен', @education, @stag);
```

, где *fio* – имя поля таблицы БД, хранящего ФИО мастера;

*cod\_spec*, *cod* – имена полей таблиц БД, хранящих код специализации мастера;

*names* – имя поля таблицы БД, хранящего наименование специализации;

*admin\_rights* – имя поля таблицы БД, хранящего информацию о наличии прав администратора у мастера;

*login* – имя поля таблицы БД, хранящего логин мастера для входа в систему;



*pas* – имя поля таблицы БД, хранящего пароль мастера для входа в систему;

*status* – имя поля таблицы БД, хранящего статус мастера (болен, уволен, свободен);

*date\_st* – имя поля таблицы БД, хранящего дату изменения статуса;

*education* – имя поля таблицы БД, хранящего информацию о наличии медицинского образования у мастера;

*stag* – имя поля таблицы БД, хранящего количество лет работы мастера в сфере эстетических услуг;

*@fio*, *@specialization*, *@admin\_rights*, *@login*, *@password*, *@date*, *@education*, *@stag* – условные обозначения подставляемых значений из формы приложения.

«Добавить клиента в БД»:

```
INSERT INTO clients  
(fio, phone_number, address, birthday, balance)  
VALUES (@fio, @phone_number, @address, @birthday, 0);
```

, где *fio* – имя поля таблицы БД, хранящего ФИО клиента;

*phone\_number* – имя поля таблицы БД, хранящего номер телефона клиента;

*address* – имя поля таблицы БД, хранящего адрес проживания клиента;

*birthday* – имя поля таблицы БД, хранящего дату дня рождения клиента;

*balance* – имя поля таблицы БД, хранящего сумму баланса клиента;

*@ fio*, *@phone\_number*, *@address*, *@birthday* – условные обозначения подставляемых значений из формы приложения.

«Изменить информацию о мастере»:

```
UPDATE masters SET fio = @fio, cod_spec = (SELECT  
cod FROM specialization WHERE names = @names),  
education = @education, stag = @stag WHERE fio=@fio;
```

, где *fio* – имя поля таблицы БД, хранящего ФИО мастера;

*cod\_spec*, *cod* – имена полей таблиц БД, хранящих код специализации мастера;

*names* – имя поля таблицы БД, хранящего наименование специализации;

*education* – имя поля таблицы БД, хранящего информацию о наличии медицинского образования у мастера;

*stag* – имя поля таблицы БД, хранящего количество лет работы мастера в сфере эстетических услуг;

*@fio*, *@names*, *@education*, *@stag* – условные обозначения подставляемых значений из формы приложения.

«Изменить расписание мастера»:

```
DECLARE n INTEGER;  
c INTEGER;  
BEGIN  
SELECT COUNT(fio) INTO n FROM masters, timetable  
WHERE fio=@fio AND timetable.id_mas=masters.cod  
AND timetable.id_week=@week_id;  
SELECT cod INTO c FROM masters WHERE fio=@fio;  
IF n > 0 THEN UPDATE timetable SET  
start_time=@start_time, end_time=@end_time WHERE  
id_mas=c AND id_week=@week_id;  
ELSE INSERT INTO timetable VALUES(c, @week_id,  
@start_time, @end_time) END IF;  
END;
```

, где *fio* – имя поля таблицы БД, хранящего ФИО мастера;

*id\_mas*, *cod* – имена полей таблиц БД, хранящих уникальный идентификатор мастера;

*id\_week* – имя поля таблицы БД, хранящего уникальный идентификатор дня недели;

*start\_time* – имя поля таблицы БД, хранящего время начала смены мастера;

*end\_time* – имя поля таблицы БД, хранящего время конца смены мастера;

*@fio*, *@week\_id*, *@start\_time*, *@end\_time* – условные обозначения подставляемых значений из формы приложения.

«Записать на приём»:

```
INSERT INTO appointment (id_mas, cl_fio, date_,
id_serv, id_tos, product_count, prod_cod,
summa, payment, status, discount) VALUES (@id_mas,
@cl, @date_, @id_serv, @id_tos, @prod_count,
@prod_cod, @summa, @payment, 'Ожидается',
@discount);
```

, где *id\_mas* – имя поля таблицы БД, хранящего уникальный идентификатор мастера;

*cl\_fio* – имя поля таблицы БД, хранящего уникальный идентификатор клиента;

*date\_* – имя поля таблицы БД, хранящего дату записи на приём;

*id\_serv* – имя поля таблицы БД, хранящего уникальный идентификатор услуги;

*id\_tos* – имя поля таблицы БД, хранящего уникальный идентификатор вида услуги;

*product\_count* – имя поля таблицы БД, хранящего количество израсходованного продукта в мл.;

*prod\_cod* – имя поля таблицы БД, хранящего уникальный идентификатор продукта;

*summa* – имя поля таблицы БД, хранящего итоговую стоимость записи на приём;

*payment* – имя поля таблицы БД, хранящего уникальный идентификатор способа оплаты;

*status* – имя поля таблицы БД, хранящего статус записи на приём;

*discount* – имя поля таблицы БД, хранящего скидку клиента на момент совершения записи;

*@date\_*, *@prod\_count* – условные обозначения подставляемых значений из формы приложения;

*@summa*, *@id\_mas*, *@cl*, *@id\_serv*, *@id\_tos*, *@prod\_cod*, *@payment*, *@discount* – условные обозначения подставляемых значений из алгоритма процедуры.

«Пополнить счет клиента»:

```
UPDATE clients SET balance = balance+@summa;
```

, где *balance* – имя поля таблицы БД, хранящего сумму баланса клиента;

*@summa* – условное обозначение подставляемого значения из формы приложения.

«Заменить мастера»:

```
UPDATE appointment SET id_mas = (SELECT cod FROM  
masters WHERE fio=@fio_mas1), status = 'Ожидается'  
WHERE id_mas = (SELECT cod FROM masters WHERE fio  
=@fio_mas2) AND date_ = @date;
```

, где *id\_mas*, *cod* – имена полей таблиц БД, хранящих уникальный идентификатор мастера;

*fio* – имя поля таблицы БД, хранящего ФИО мастера;

*status* – имя поля таблицы БД, хранящего статус мастера;

*date\_* – имя поля таблицы БД, хранящего дату записи;

*@fio\_mas1*, *@fio\_mas2*, *@date* – условные обозначения подставляемых значений из формы приложения.

«Изменить статус мастера»:

```
UPDATE masters SET status = @status, date_st =  
@date WHERE fio=@fio;
```

, где *status* – имя поля таблицы БД, хранящего статус мастера;

*date\_st* – имя поля таблицы БД, хранящего дату изменения статуса мастера;

*fio* – имя поля таблицы БД, хранящего ФИО мастера;

*@status*, *@date*, *@fio* – условные обозначения подставляемых значений из формы приложения.

### Информационно-поисковые запросы

«Найти препарат по параметрам»:

```
SELECT products.names FROM products, services,  
serv_prod WHERE products.cod=serv_prod.id_prod  
AND services.cod=serv_prod.id_serv AND  
services.names = @name;
```

, где *products.names* – имя поля таблицы БД, хранящего наименование препарата;

*products.cod*, *id\_prod* – имена полей таблиц БД, хранящих код препарата;

*services.cod*, *id\_serv* – имена полей таблиц БД, хранящих код услуги;

*services.names* – имя поля таблицы БД, хранящего наименование услуги;

*@name* – условное обозначение подставляемого значения из формы приложения.

«Найти вид услуги по параметрам»:

```
SELECT typesofservices.names FROM typesofservices,  
services, serv_tos WHERE serv_tos.id_serv=  
services.cod AND serv_tos.id_tos =  
typesofservices.cod AND services.names = @name;
```

, где *typesofservices.names* – имя поля таблицы БД, хранящего наименование вида услуги;

*id\_serv*, *services.cod* – имена полей таблиц БД, хранящих код услуги;

*id\_tos*, *typesofservices.cod* – имена полей таблиц БД, хранящих код вида услуги;

*services.names* – имя поля таблицы БД, хранящего наименование услуги;

*@name* – условное обозначение подставляемого значения из формы приложения.

«Найти мастера по параметрам»:

```
SELECT distinct fio FROM
spec_serv,masters,specialization,services,timetab
le WHERE spec_serv.id_serv = services.cod AND
spec_serv.id_spec = specialization.cod AND
masters.cod_spec = spec_serv.id_spec and
services.names = @name AND masters.status =
'Свободен' and timetable.id_mas = masters.cod";
```

, где *typesofservices.names* – имя поля таблицы БД, хранящего наименование вида услуги;

*fio* – имя поля таблицы БД, хранящего ФИО мастера;

*id\_serv, services.cod* – имена полей таблиц БД, хранящих код услуги;

*id\_spec, specialization.cod, cod\_spec* – имена полей таблиц БД, хранящих код специализации;

*services.names* – имя поля таблицы БД, хранящего наименование услуги;

*status* – имя поля таблицы БД, хранящего статус мастера;

*id\_mas, masters.cod* – имена полей таблиц БД, хранящих статус мастера;

*@name* – условное обозначение подставляемого значения из формы приложения.

### Информационно-агрегативные запросы

«Вычислить сумму, на которую клиенту были оказаны услуги»:

```
SELECT SUM(summa) FROM appointment WHERE cl_fio =
@cl;
```

, где *summa* – имя поля таблицы БД, хранящего сумму, выплаченную клиентом за оказание услуги;

*@cl* – условное обозначение подставляемого значения из формы приложения.

«Узнать сколько мастеров работают в указанный день недели»:

```
select count(fio) into n from masters,
timetable where fio = @fio and timetable.id_mas
= masters.cod and timetable.id_week= @id_week;
```

, где *fio* – имя поля таблицы БД, хранящего ФИО мастера;

*id\_mas, cod* – имена полей таблиц БД, хранящих уникальный идентификатор мастера;

*id\_week* – имя поля таблицы БД, хранящего уникальный идентификатор дня недели;

*@fio, @week\_id* – условные обозначения подставляемых значений из формы приложения.

«Узнать сколько мастеров имеют указанный логин»:

```
SELECT COUNT(fio) FROM masters WHERE  
login=@login;
```

, где *fio* – имя поля таблицы БД, хранящего ФИО мастера;

*login* – имя поля таблицы БД, хранящего логин мастера

*@login* – условное обозначение подставляемого значения из формы приложения.

Вывод: в ходе выполнения работы были получены практические навыки разработки системы обмена данными с БД в рамках предметной области «Салон красоты».