

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Отчет по Лабораторной работе №3  
по предмету “Базы данных”

Студент

Группа ПИ-21-1

\_\_\_\_\_  
подпись, дата

Морозов Д.С.

Руководитель

\_\_\_\_\_  
подпись, дата

Алексеев В.А.

Липецк 2023 г.

## **Цель работы**

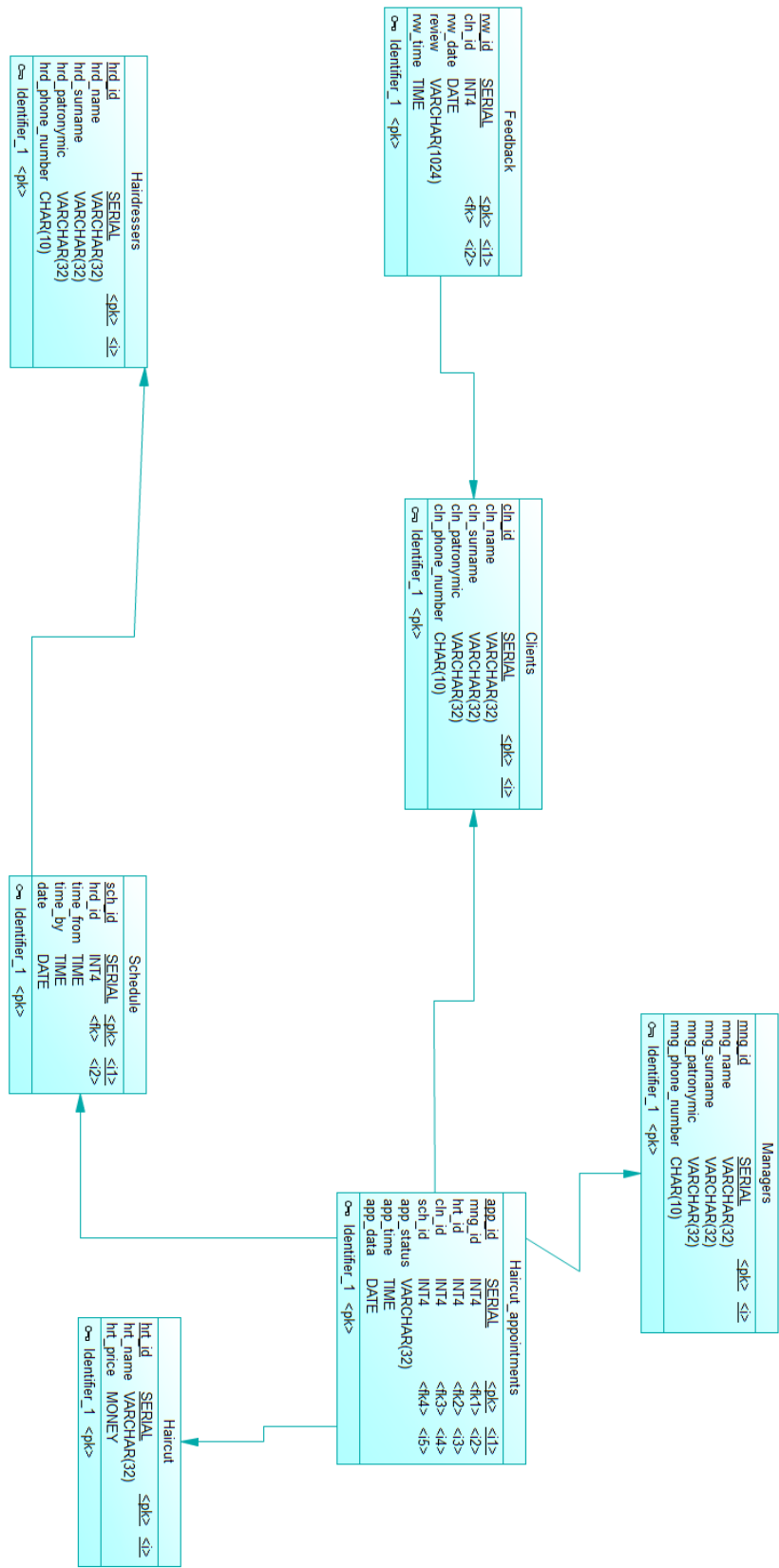
Изучить задачи физического проектирования базы данных. Освоить на практике разработку физической модели данных в среде Sybase PowerDesigner и модификацию схемы данных с использованием SQL-запросов.

## **Задание**

Сформировать в среде PowerDesigner физическую модель данных на основе логической модели, разработанной в лабораторной работе №2. Создать базу данных средствами выбранной СУБД и отработать сценарий модификации схемы данных.

# Физическая модель

## Диаграмма физической модели



## Спецификация таблиц

### Обратная связь (Feedback)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД отзыва	rvw_id	SERIAL		TRUE	TRUE	FALSE
ИД клиента	cln_id	INT4		TRUE	FALSE	TRUE
Дата отзыва	rvw_date	DATE		TRUE	FALSE	FALSE
Отзыв	review	VARCHAR (1024)	1024	TRUE	FALSE	FALSE
Время отзыва	rvw_time	TIME		TRUE	FALSE	FALSE

### Клиенты (Clients)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД клиента	cln_id	SERIAL		TRUE	TRUE	FALSE
Имя клиента	cln_name	VARCHAR (32)	32	TRUE	FALSE	FALSE
Фамилия клиента	cln_surname	VARCHAR (32)	32	TRUE	FALSE	FALSE
Отчество клиента	cln_patronymic	VARCHAR (32)	32	FALSE	FALSE	FALSE
Номер телефона клиента	cln_phone_number	CHAR (10)	10	TRUE	FALSE	FALSE

### Менеджеры (Managers)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД менеджера	mng_id	Serial		TRUE	TRUE	FALSE
Имя менеджера	mng_name	VARCHAR (32)	32	TRUE	FALSE	FALSE
Фамилия менеджера	mng_surname	VARCHAR (32)	32	TRUE	FALSE	FALSE
Отчество менеджера	mng_patronymic	VARCHAR (32)	32	FALSE	FALSE	FALSE

Номер телефона менеджера	mng_phone_number	CHAR (10)	10	TRUE	FALSE	FALSE
--------------------------	------------------	-----------	----	------	-------	-------

### Записи на стрижку (Haircut appointments)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД записи	app_id	SERIAL		TRUE	TRUE	FALSE
ИД менеджера	mng_id	INT4		TRUE	FALSE	TRUE
ИД стрижки	hrt_id	INT4		TRUE	FALSE	TRUE
ИД клиента	cln_id	INT4		TRUE	FALSE	TRUE
ИД смены	sch_id	INT4		TRUE	FALSE	TRUE
Статус записи	app_status	VARCHAR (32)	32	TRUE	FALSE	FALSE
Время записи	app_time	TIME		TRUE	FALSE	FALSE
Дата записи	app_date	DATE		TRUE	FALSE	FALSE

### Парикмахеры (Hairdressers)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД парикмахера	hrd_id	SERIAL		TRUE	TRUE	FALSE
Имя парикмахера	hrd_name	VARCHAR (32)	32	TRUE	FALSE	FALSE
Фамилия парикмахера	hrd_surname	VARCHAR (32)	32	TRUE	FALSE	FALSE
Отчество парикмахера	hrd_patronymic	VARCHAR (32)	32	FALSE	FALSE	FALSE
Номер телефона парикмахера	hrd_phone_number	CHAR (10)	10	TRUE	FALSE	FALSE

### График работы (Schedule)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД смены	sch_id	SERIAL		TRUE	TRUE	FALSE
ИД парикмахера	hrd_id	INT4		TRUE	FALSE	TRUE
Время с	time_from	TIME		TRUE	FALSE	FALSE
Время по	time_by	TIME		TRUE	FALSE	FALSE
Дата	date	DATE		TRUE	FALSE	FALSE

## Стрижка (Haircut)

Имя	Код	Тип данных	Длина	Обязательность	Первичный ключ	Внешний ключ
ИД стрижки	hrt_id	SERIAL		TRUE	TRUE	FALSE
Название стрижки	hrt_name	VARCHAR(32)	32	TRUE	FALSE	FALSE
Цена стрижки	hrt_price	MONEY		TRUE	FALSE	FALSE

## Создание БД

### SQL – запрос создания схемы данных БД

```
/*=====*/
/* Table: Clients */
/*=====*/
create table Clients (
    cln_id          SERIAL          not null,
    cln_name        VARCHAR(32)     not null,
    cln_surname     VARCHAR(32)     not null,
    cln_patronymic  VARCHAR(32)     null,
    cln_phone_number CHAR(10)       not null,
    constraint PK_CLIENTS primary key (cln_id)
);

/*=====*/
/* Index: Clients_PK */
/*=====*/
create unique index Clients_PK on Clients (
    cln_id
);

/*=====*/
/* Table: Feedback */
/*=====*/
create table Feedback (
    rvw_id          SERIAL          not null,
    cln_id          INT4            not null,
    rvw_date        DATE            not null,
    review          VARCHAR(1024)   not null,
    rvw_time        TIME            not null,
    constraint PK_FEEDBACK primary key (rvw_id)
);

/*=====*/
/* Index: Feedback_PK */
/*=====*/
create unique index Feedback_PK on Feedback (
```

```

rvw_id
);

/*=====*/
/* Index: "Write a review_FK" */
/*=====*/
create index "Write a review_FK" on Feedback (
cln_id
);

/*=====*/
/* Table: Haircut */
/*=====*/
create table Haircut (
    hrt_id          SERIAL          not null,
    hrt_name        VARCHAR(32)     not null,
    hrt_price       MONEY           not null,
    constraint PK_HAIRCUT primary key (hrt_id)
);

/*=====*/
/* Index: Haircut_PK */
/*=====*/
create unique index Haircut_PK on Haircut (
hrt_id
);

/*=====*/
/* Table: Haircut_appointments */
/*=====*/
create table Haircut_appointments (
    app_id          SERIAL          not null,
    mng_id          INT4            null,
    hrt_id          INT4            not null,
    cln_id          INT4            not null,
    sch_id          INT4            not null,
    app_status      VARCHAR(32)     not null,
    app_time        TIME            not null,
    app_data        DATE            not null,
    constraint PK_HAIRCUT_APPOINTMENTS primary key (app_id)
);

/*=====*/
/* Index: Haircut_appointments_PK */
/*=====*/
create unique index Haircut_appointments_PK on Haircut_appointments (
app_id
);

/*=====*/

```

```

/* Index: Confirm_FK */
/*=====*/
create index Confirm_FK on Haircut_appointments (
mng_id
);

/*=====*/
/* Index: Choose_FK */
/*=====*/
create index Choose_FK on Haircut_appointments (
hrt_id
);

/*=====*/
/* Index: "Sign up and pay_FK" */
/*=====*/
create index "Sign up and pay_FK" on Haircut_appointments (
cln_id
);

/*=====*/
/* Index: "Check entries_FK" */
/*=====*/
create index "Check entries_FK" on Haircut_appointments (
sch_id
);

/*=====*/
/* Table: Hairdressers */
/*=====*/
create table Hairdressers (
    hrd_id          SERIAL          not null,
    hrd_name        VARCHAR(32)     not null,
    hrd_surname     VARCHAR(32)     not null,
    hrd_patronymic  VARCHAR(32)     null,
    hrd_phone_number CHAR(10)       not null,
    constraint PK_HAIRDRESSERS primary key (hrd_id)
);

/*=====*/
/* Index: Hairdressers_PK */
/*=====*/
create unique index Hairdressers_PK on Hairdressers (
hrd_id
);

/*=====*/
/* Table: Managers */
/*=====*/
create table Managers (

```



```

    mng_id          SERIAL          not null,
    mng_name        VARCHAR(32)     not null,
    mng_surname     VARCHAR(32)     not null,
    mng_patronymic  VARCHAR(32)     null,
    mng_phone_number CHAR(10)       not null,
    constraint PK_MANAGERS primary key (mng_id)
);

/*=====*/
/* Index: Managers_PK */
/*=====*/
create unique index Managers_PK on Managers (
mng_id
);

/*=====*/
/* Table: Schedule */
/*=====*/
create table Schedule (
    sch_id          SERIAL          not null,
    hrd_id          INT4           not null,
    time_from       TIME           not null,
    time_by         TIME           not null,
    date            DATE           not null,
    constraint PK_SCHEDULE primary key (sch_id)
);

/*=====*/
/* Index: Schedule_PK */
/*=====*/
create unique index Schedule_PK on Schedule (
sch_id
);

/*=====*/
/* Index: "Check work schedule_FK" */
/*=====*/
create index "Check work schedule_FK" on Schedule (
hrd_id
);

alter table Feedback
    add constraint "FK_FEEDBACK_WRITE A R_CLIENTS" foreign key (cln_id)
        references Clients (cln_id)
        on delete restrict on update restrict;

alter table Haircut_appointments
    add constraint "FK_HAIRCUT__CHECK ENT_SCHEDULE" foreign key (sch_id)
        references Schedule (sch_id)
        on delete restrict on update restrict;

```

```

alter table Haircut_appointments
  add constraint FK_HAIRCUT__CHOOSE_HAIRCUT foreign key (hrt_id)
    references Haircut (hrt_id)
    on delete restrict on update restrict;

alter table Haircut_appointments
  add constraint FK_HAIRCUT__CONFIRM_MANAGERS foreign key (mng_id)
    references Managers (mng_id)
    on delete restrict on update restrict;

alter table Haircut_appointments
  add constraint "FK_HAIRCUT__SIGN UP A_CLIENTS" foreign key (cln_id)
    references Clients (cln_id)
    on delete restrict on update restrict;

alter table Schedule
  add constraint "FK_SCHEDULE__CHECK WOR_HAIRDRES" foreign key (hrd_id)
    references Hairdressers (hrd_id)
    on delete restrict on update restrict;

```

## Скриншот консоли управления СУБД с представлением созданной структуры БД

- > Aa Анализаторы FTS
- > Домены
- > Конфигурации FTS
- > Материализованные представления
- > 1.3 Последовательности
- > Правила сортировки
- > Представления
- > Процедуры
- > Словари FTS
- > Сторонние таблицы
- > Таблицы (7)
  - > clients
  - > feedback
  - > haircut
  - > haircut\_appointments
  - > hairdressers
  - > managers
  - > schedule
- > Типы
- > Триггерные функции
- > Функции
- > Шаблоны FTS
- > Языки
- postares

Data Output
Сообщения
Notifications

+ 📄 ▼ 📋 ▼ 🗑️ 🔄 📥 📡

	table_name name	column_name name	data_type character varying
1	clients	cln_patronymic	character varying
2	clients	cln_phone_number	character
3	clients	cln_name	character varying
4	clients	cln_id	integer
5	clients	cln_surname	character varying
6	feedback	review	character varying
7	feedback	rvw_time	time without time zone
8	feedback	rvw_date	date
9	feedback	cln_id	integer
10	feedback	rvw_id	integer
11	haircut	hrt_id	integer
12	haircut	hrt_name	character varying
13	haircut	hrt_price	money
14	haircut_appointments	app_id	integer
15	haircut_appointments	cln_id	integer
16	haircut_appointments	app_time	time without time zone
17	haircut_appointments	hrt_id	integer
18	haircut_appointments	app_data	date
19	haircut_appointments	app_status	character varying

## Пример создание индекса для столбцов:

Запрос История запросов

```
1 CREATE INDEX cln_fio ON clients (cln_name, cln_surname, cln_patronymic);  
2
```

Data Output Сообщения Notifications

CREATE INDEX

Запрос завершён успешно, время выполнения: 46 мсек.

## Сценарий модификации БД

### Описание сценария

Парикмахерской нужно вести учет расходов за каждый месяц. В таблице будут присутствовать расходы по сменам за свет, воду, расходные материалы(лаки, шампуни и т.д).

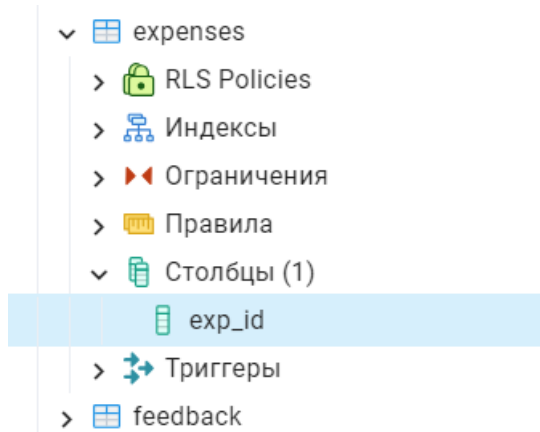
Создание таблицы Расходы с атрибутом ID расхода.

- 1) Добавление в таблицу атрибуты “Свет”, “Вода”, “Расходные материалы”.
- 2) Добавление в таблицу Расходы внешний ключ от таблицы График работы, ограничение целостность.
- 3) Модификация атрибута “ Расходные материалы ” изменение с INT4 на MONEY.
- 4) Удаление таблицы Расходы.

### Текст SQL – запросов модификации схемы данных

1) Создание таблицы Расходы с атрибутом ID расхода.

```
CREATE TABLE expenses (  
    exp_id SERIAL NOT NULL,  
    CONSTRAINT PK_INSURANCE PRIMARY KEY (exp_id)  
);
```

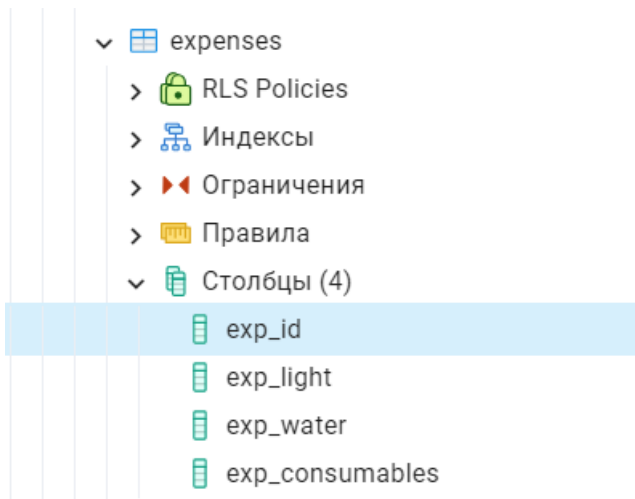


2) Добавление в таблицу атрибуты “Свет”, “Вода”, “Расходные материалы”.

```
ALTER TABLE expenses ADD COLUMN exp_light MONEY NOT NULL;
```

```
ALTER TABLE expenses ADD COLUMN exp_water MONEY NOT NULL;
```

```
ALTER TABLE expenses ADD COLUMN exp_consumables MONEY NOT NULL;
```



3) Добавление в таблицу Расходы внешний ключ от таблицы График работы, ограничение целостность.

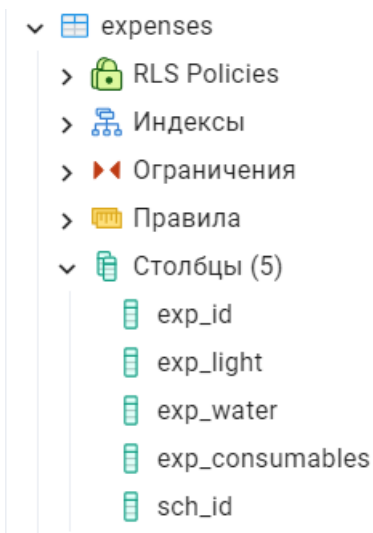
```
ALTER TABLE expenses ADD COLUMN sch_id INT4;
```

```
ALTER TABLE expenses
```

```
ADD CONSTRAINT FK_SCH_ID FOREIGN KEY (sch_id)
```

```
REFERENCES schedule (sch_id)
```

```
ON DELETE RESTRICT ON UPDATE RESTRICT;
```



5) Модификация атрибута “Вода” изменение с MONEY на NUMERIC.

```
ALTER TABLE expenses ALTER COLUMN exp_water type NUMERIC;
```

exp_id [PK] integer	exp_light money	exp_water numeric	exp_consumables money	sch_id integer
------------------------	--------------------	----------------------	--------------------------	-------------------

## 5) Удаление таблицы Расходы.

```
DROP TABLE expenses CASCADE;
```

