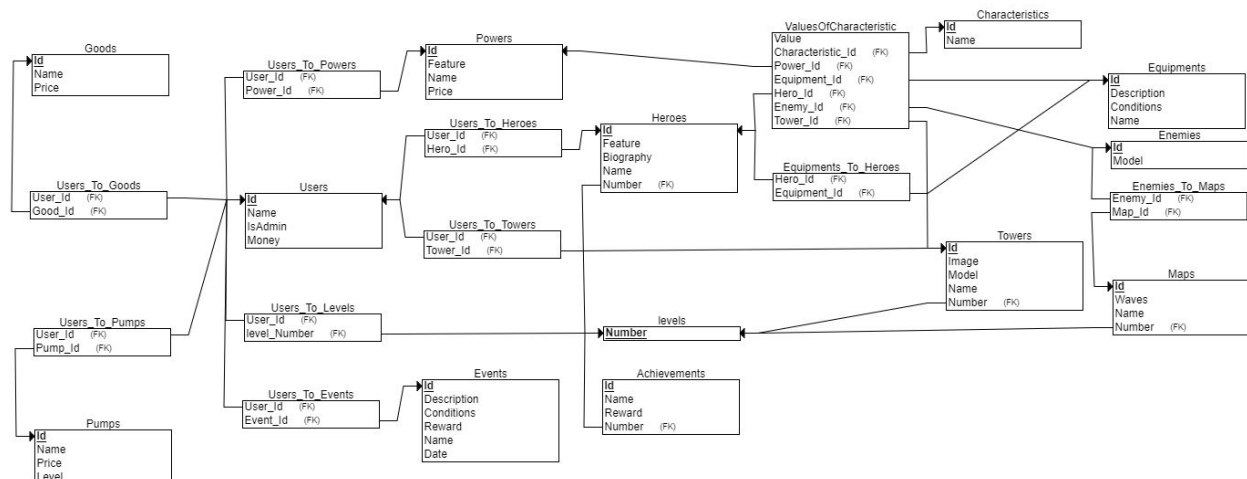


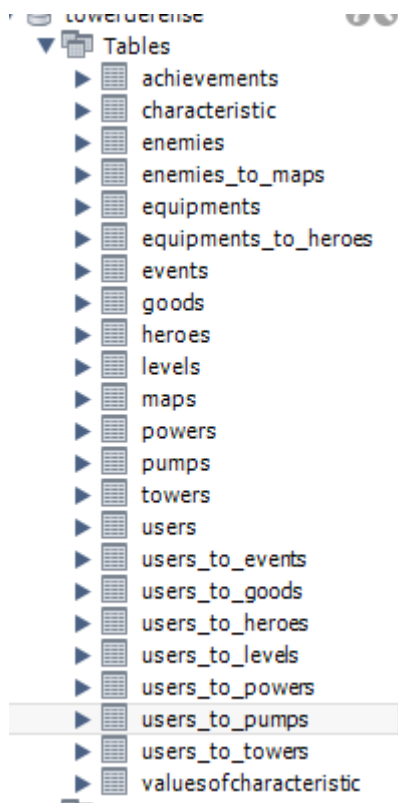
Лабораторная работа №3 Физическое проектирование баз данных	ФИО	Панасенко С.С.
	Группа	ИВТ-363
	Преподаватель	Соколов А.А.
	Дата сдачи	

1)Реляционная схема



2) Таблицы с структурой и наполнением данными

2.1)Список таблиц:



2.2)Таблицы

The screenshot shows the SQL Studio interface with two queries executed.

Query 1: `SELECT * FROM towerdefense.achievements;`

ID	Name	Reward	Number
0	Пройти все карты	10000	0
1	Открыть всех героев	3000	0
2	Пройти 3 эвента	1000	3
3	Пройти 5 эвентов	3000	3
4	Уничтожить 5000 врагов	5000	5
5	Экипировать одного героя	1000	0
6	Использовать все способности по 1 разу	3000	0
7	Повысить один из видов прокачки до 5 уровня	14000	0
8	Использовать одну из способностей 10 раз	10000	5
9	Прокачать любого из героев до уровня 30	10000	7
NULL	NULL	NULL	NULL

Query 2: `SELECT * FROM towerdefense.characteristic;`

ID	Name
9	Атака
4	Дальность атаки
0	Защита
3	Здоровье
1	Количество
7	Ловкость
2	Сила
8	Скорость
6	Скорость атаки
5	Урон
NULL	NULL

enemies

Limit to 1000 rows

1 • `SELECT * FROM towerdefense.enemies;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	ID	ModelUrl
▶	0	/models/enemy0.png
	1	/models/enemy1.png
	2	/models/enemy2.png
	3	/models/enemy3.png
	4	/models/enemy4.png
	5	/models/enemy5.png
	6	/models/enemy6.png
	7	/models/enemy7.png
	8	/models/enemy8.png
	9	/models/enemy9.png
*	NULL	NULL

enemies_to_maps

Limit to 1000 rows

1 • `SELECT * FROM towerdefense.enemies_to_maps;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	Enemy_ID	Map_ID
▶	1	0	0
	2	1	1
	3	1	2
	4	3	0
	5	3	1
	6	4	7
	7	5	2
	8	5	5
	9	5	8
	10	6	1
*	NULL	NULL	NULL

equipments

1 • SELECT * FROM towerdefense.equipments;

ID	Description	Conditions	Name
0	Обычный деревянный лук для...	Уровень: 5; Только для Адоры	Деревянный лук
1	Намного сильнее первоначаль...	Уровень: 5; Сила: 10; Только для Грема	Железный топор
2	Средняя броня для любого ге...	Уровень: 5; Сила: 5; Ловкость: 5	Кожаная броня
3	Продвинутый посох мага, уве...	Уровень: 12; Интеллект: 10; Только для Анг...	Посох огня
4	Повышает силу на 2 у любого ...	Уровень: 10; Интеллект: 5	Амулет силы
5	Повышает защиту от физичес...	Уровень: 20; Сила: 20	Стальная броня
6	Увеличивает стандартный ур...	Уровень: 10; Интеллект: 5; Сила: 10; Тольк...	Стальной посох
7	Повышает все характеристик...	Уровень: 30; Интеллект: 5; Только для Якса	Лиственный плащ
8	Бонус к урону, также ненадол...	Уровень: 30; Сила: 30; Только для Эмдала	Ледяной молот
9	Бонус к урону +50% от текущ...	Уровень: 30	Амулет атаки
NULL	NULL	NULL	NULL

equipments_to_heroes

1 • SELECT * FROM towerdefense.equipments_to_heroes;

id	Equipment_ID	Hero_ID
1	0	0
2	2	0
3	4	1
4	3	4
5	2	4
6	1	5
7	3	6
8	5	7
9	1	8
10	9	8
NULL	NULL	NULL

maps

Limit to 1000 rows

1 • SELECT * FROM towerdefense.maps;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	ID	Waves	Name	LevelNumber
▶	0	20	Лесная опушка	0
	1	50	Лес	0
	2	80	Река Элебра	3
	3	30	Старейший Энт	6
	4	50	Деревня	10
	5	0	Тропа в лес	10
	6	100	Храм Дэла	12
	7	120	Холм Агарот	15
	8	50	Мост Элебры	20
	9	0	Поле	25
*	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

powers

Limit to 1000 rows

1 • SELECT * FROM towerdefense.powers;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	ID	Feature	Name	Price
▶	0	Дает щит, увеличивающий общее количест...	Щит	150
	1	Один раз дает валюту в размере 20% от те...	Пополнение счета	500
	2	Наносит один раз урон всем противникам	Ударная волна	250
	3	На 10% повышает все характеристики баше...	Усиление	400
	4	Замедляет и отравляет всех врагов в текущ...	Проклятье	150
	5	Дает +50% валюты, исходя из текущей	Мешок золота	500
	6	На 15% повышает скорость передвижения г...	Сапоги-скороходы	300
	7	Наносит повреждения вышедшим врагам, в ...	Колючая лоза	500
	8	Все враги наносят на 10% меньше урона при...	Дорога слабости	400
	9	Останавливает всех врагов на 3 сек	Волна заморозки	300
*	NULL	NULL	NULL	NULL

pumps

Limit to 1000 rows

1 • `SELECT * FROM towerdefense.pumps;`

Result Grid

ID	Name	Price	Level
0	Увеличенный урон башен	1500	2
1	Увеличенная дальность башен	1000	2
2	Все товары дешевле	3285	3
3	Урон всех героев выше	11250	5
4	Щит в начале раунда	850	2
5	Больше денег с врагов	10000	5
6	Больше бонус с товаров	1200	2
7	Больше жизней в каждой игре	200	1
8	Герои сильнее	8200	4
9	Все награды выше	6000	4
NULL	NULL	NULL	NULL

towers

Limit to 1000 rows

1 • `SELECT * FROM towerdefense.towers;`

Result Grid

ID	ImageUrl	ModelUrl	Name	LevelNumber
0	/images/tower0.png	/models/tower0.png	Лесничий	1
1	/images/tower1.png	/models/tower1.png	Грибник	2
2	/images/tower2.png	/models/tower2.png	Древень	5
3	/images/tower3.png	/models/tower3.png	Маг	10
4	/images/tower4.png	/models/tower4.png	Алхимик	15
5	/images/tower5.png	/models/tower5.png	Друид	15
6	/images/tower6.png	/models/tower6.png	Эмп	20
7	/images/tower7.png	/models/tower7.png	Дриада	25
8	/images/tower8.png	/models/tower8.png	Энт	25
9	/images/tower9.png	/models/tower9.png	Эльф	30
NULL	NULL	NULL	NULL	NULL

users

Limit to 1000 rows

```
1 • SELECT * FROM towerdefense.users;
```

Result Grid

	ID	Name	IsAdmin	Money
▶	0	Gravet	1	9999
	1	Yarglek	0	1500
	2	Vanya	1	12300
	3	SuperLord	0	2500
	4	qwerty112345	0	1234
	5	Kraken	0	1200
	6	Petya	1	5000
	7	Geroy	0	17584
	8	Irttek	0	0
	9	Ray	0	12005
*	NULL	NULL	NULL	NULL

users_to_events

Limit to 1000 rows

```
1 • SELECT * FROM towerdefense.users_to_events;
```

Result Grid

	id	User_ID	Event_ID
▶	1	0	0
	2	1	2
	3	3	3
	4	3	1
	5	4	4
	6	5	1
	7	6	6
	8	6	8
	9	7	3
	10	7	7
*	NULL	NULL	NULL

users_to_goods x

Limit to 1000 rows

1 • `SELECT * FROM towerdefense.users_to_goods;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	User_ID	Good_ID
▶	1	0	4
	2	1	1
	3	3	1
	4	3	0
	5	4	3
	6	1	1
	7	3	1
	8	3	0
	9	4	3
	10	5	7
	11	5	9
	12	7	5
	13	9	1
	14	9	5
	NULL	NULL	NULL

users_to_heroes x

Limit to 1000 rows

1 • `SELECT * FROM towerdefense.users_to_heroes;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	User_ID	Hero_ID
▶	1	0	0
	2	2	1
	3	2	4
	4	3	2
	5	4	0
	6	5	0
	7	6	1
	8	6	7
	9	7	1
	10	8	3
*	NULL	NULL	NULL

users_to_levels x



Limit to 1000 rows

1 • `SELECT * FROM towerdefense.users_to_levels;`

<

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	User_ID	Level_ID
▶	1	0	13
	2	1	45
	3	2	23
	4	3	87
	5	4	12
	6	5	14
	7	6	56
	8	7	18
	9	8	20
	10	9	10
*	NULL	NULL	NULL

users_to_powers x



Limit to 1000 rows

1 • `SELECT * FROM towerdefense.users_to_powers;`

<

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	User_ID	Power_ID
▶	1	0	2
	2	1	3
	3	1	4
	4	3	4
	5	4	1
	6	5	1
	7	6	2
	8	6	5
	9	7	6
	10	7	8
*	NULL	NULL	NULL

users_to_pumps x

Limit to 1000 rows

```
1 • SELECT * FROM towerdefense.users_to_pumps;
```

Result Grid

	id	User_ID	Pump_ID
▶	1	0	2
	2	1	3
	3	2	4
	4	2	3
	5	4	4
	6	5	2
	7	5	4
	8	5	5
	9	6	1
	10	7	1
*	NULL	NULL	NULL

users_to_towers x

Limit to 1000 rows

```
1 • SELECT * FROM towerdefense.users_to_towers;
```

Result Grid

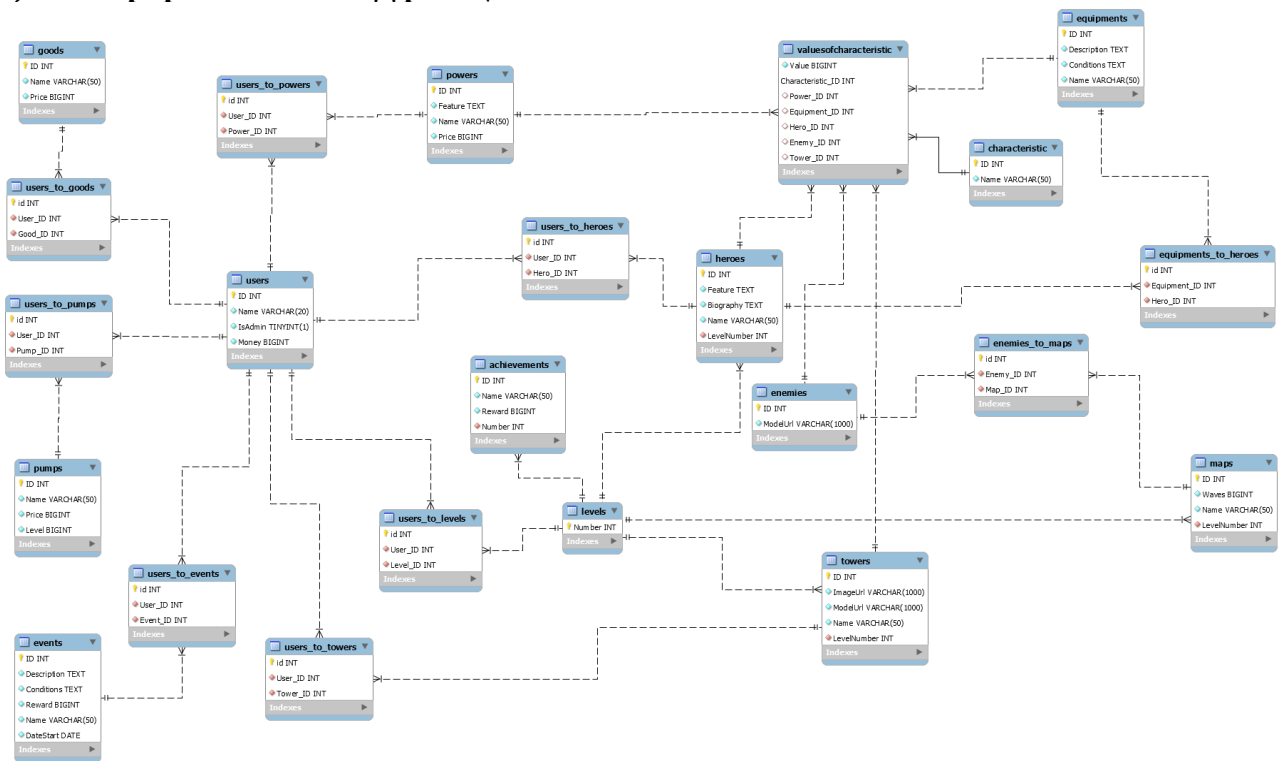
	id	User_ID	Tower_ID
▶	1	0	0
	2	1	4
	3	2	1
	4	2	0
	5	4	0
	6	5	0
	7	5	3
	8	6	2
	9	7	2
	10	8	7
*	NULL	NULL	NULL

[illegible]

```
1 • SELECT * FROM towerdefense.valuesofcharacteristic;
```

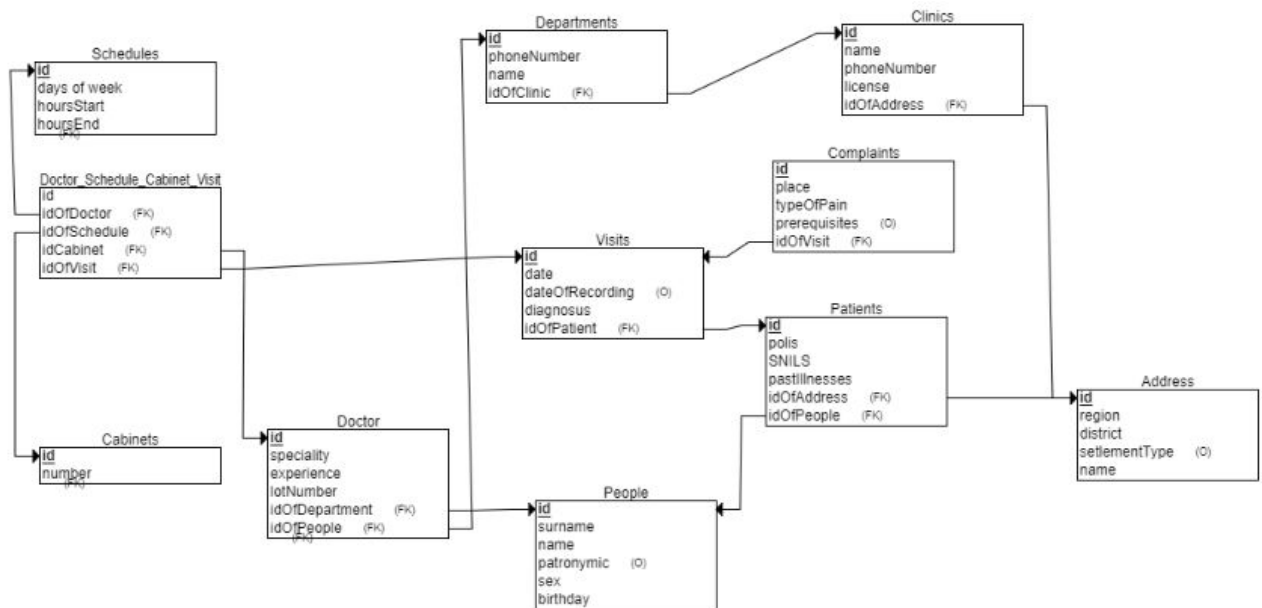
[illegible]

3) Сгенерированная СУБД реляционная схема



Часть 2:

1) Получил от одногруппника проверенную преподавателем схему



2) Написал скрипты: (createdb, createtables, adddata, changetables, droptbandbd)

2.1) createdb-создание базы данных

```
CREATE DATABASE Bdclinic;
```

2.2) createtables-создание таблиц базы данных

-- Создание таблиц

```
use bdclinic;
```

```
CREATE TABLE schedule(
id_of_schedule int primary key auto_increment,
days_of_week varchar(2),
hours_start time,
hours_end time);
```

```
CREATE TABLE address(
id_of_address int primary key auto_increment,
region varchar(50) not null,
settlement varchar(30) not null,
district varchar(30),
street varchar(30) not null,
house_number varchar(30) not null,
```

flat_number int

);

CREATE TABLE clinic(

id_of_clinic int primary key auto_increment,

firstname varchar(50) not null,

phone_number char(11) not null,

license varchar(30) not null,

id_of_address int,

foreign key(id_of_address) references address(id_of_address)

);

CREATE TABLE department(

id_of_department int primary key auto_increment,

phone_number char(11) not null,

firstname varchar(50) not null,

id_of_clinic int,

foreign key(id_of_clinic) references clinic(id_of_clinic)

);

CREATE TABLE people(

id_of_people int primary key auto_increment,

surname varchar(30) not null,

firstname varchar(30) not null,

patronymic varchar(30),

sex char(1) not null,

birthday date not null

);

CREATE TABLE patient(

id_of_patient int primary key auto_increment,

polis char(16) not null,

snils char(11) not null,

pastlinesses varchar(30),

id_of_address int,


```
id_of_people int,  
foreign key(id_of_address) references address(id_of_address),  
foreign key(id_of_people) references people(id_of_people)  
);
```

```
CREATE TABLE visit(  
id_of_visit int primary key auto_increment,  
date_visit datetime not null,  
date_of_recording date,  
diagnosus varchar(30),  
id_of_patient int not null auto_increment,  
foreign key(id_of_patient) references patient(id_of_patient)  
);
```

```
CREATE TABLE complaint(  
id_of_complaint int primary key auto_increment,  
place varchar(30) not null,  
type_of_pain varchar(30),  
prerequisites varchar(30),  
id_of_visit int,  
foreign key(id_of_visit) references visit(id_of_visit)  
);
```

```
CREATE TABLE doctor(  
id_of_doctor int primary key auto_increment,  
speciality varchar(30),  
experience int not null,  
lotNumber int,  
id_of_department int,  
id_of_people int,  
foreign key(id_of_department) references department(id_of_department),  
foreign key(id_of_people) references people(id_of_people)  
);
```

```
CREATE TABLE cabinet(  

```

```
id_of_cabinet int primary key auto_increment,  
number char(8)  
);
```

```
CREATE TABLE doc_sch_cab_vis(  
id_of_doc_sch_cab_vis int primary key auto_increment,  
id_of_doctor int not null,  
id_of_schedule int not null,  
id_of_cabinet int not null,  
id_of_visit int not null,  
foreign key(id_of_doctor) references doctor(id_of_doctor),  
foreign key(id_of_schedule) references schedule(id_of_schedule),  
foreign key(id_of_cabinet) references cabinet(id_of_cabinet),  
foreign key(id_of_visit) references visit(id_of_visit)  
);
```

2.3) adddata-Добавление данных в таблицу+rename таблицы

-- Заполнение таблиц данными
use bdclinic;

```
insert into address(region, settlement, district, street, house_number, flat_number) values  
("Волгоградская область", "Волгоград", "Ворошиловский", "улица Читинская", 56, null),  
("Краснодарский край", "Краснодар", "Центральный", "улица Быкова", 5, 7);
```

```
insert into people(surname, firstname, patronymic, sex, birthday) values  
("Петров", "Иван", "Петрович", 'М', '2000-10-05');
```

```
insert into doctor(speciality, experience, lotNumber) values  
("Хирург", 5, 3), ("Терапевт", 1, 5);
```

```
insert into cabinet(number) values  
('123'), ('53');
```

```
insert into clinic(firstname, phone_number, license) values  
("Клиника имени Иванова", '345-12-56-4', "QWER License..."),  
("Центральная государственная больница", '567-32-81-9', "CENTR License...");
```

```
insert into schedule(days_of_week, hours_start, hours_end) values  
('01', '12:00:00', '16:00:00'), ('05', '16:00:00', '21:30:00');
```

```
rename table doctor to doctors;
rename table visit to visitors;
```

2.4) changetables-запросы на изменение таблиц

```
use bdclinic;
```

```
-- Будем работать с таблицей complaint
-- Сначала удалим ее.
drop table complaint;
```

```
-- Таблица которая должна быть
/*
```

```
CREATE TABLE complaint(
id_of_complaint int primary key auto_increment,
place varchar(30) not null,
type_of_pain varchar(30),
prerequisites varchar(30),
id_of_visit int,
foreign key(id_of_visit) references visit(id_of_visit)
);
*/
```

```
-- Создадим таблицу с ошибками
```

```
CREATE TABLE complaint_error( -- Ошибочное имя таблицы
id_of_complaintqwe int primary key auto_increment, -- Ошибочный столбец
placeque varchar(30) not null, -- Ошибочное имя столбца
-- Отсутствует столбец
prerequisites varchar(30),
id_of_visit int,
foreign key(id_of_visit) references patient(id_of_patient) -- Ошибочный внешний ключ
);
-- 1) Переименуем Ошибочное имя таблицы в правильное
alter table complaint_error
rename to complaint;
```

```
-- 2) Исправим Ошибочный столбец путем удаления столбца и последующим созданием
правильного
alter table complaint
drop column id_of_complaintqwe,
add id_of_complaint int primary key auto_increment first;
```

```
-- 3) Переименуем Ошибочное имя столбца в правильное
alter table complaint
change placeque place varchar(30) not null;
```

```

-- 4) Добавим колонку с неправильным типом
alter table complaint
add type_of_pain int not null;

-- Изменим колонку на правильный тип
alter table complaint
modify type_of_pain varchar(30);

-- 5) Удалим ошибочный внешний ключ (FK)
alter table complaint
drop CONSTRAINT complaint_ibfk_1;

-- 6) Добавим внешний ключ(FK)
alter table complaint
add foreign key(id_of_visit) references visit(id_of_visit);

-- Работа с другой таблицей
-- 7)Создадим несколько колонок в другой таблице(patient)
alter table patient
add(age int not null, work char(50), medical_card char(10) not null);

-- 8)Изменим несколько колонок в другой таблице(patient)
alter table patient
modify age bigint,
modify work varchar(50),
modify medical_card char(20);

-- 9) Поставим колонку age после work
alter table patient
modify column age bigint after work;

-- 10)Удалим несколько колонок в другой таблице(patient)
alter table patient
drop column age,
drop column work,
drop column medical_card;

```

2.5)dropdbandbd-Удаление таблиц и базы данных

```

use bdclinic;
-- Удаление таблиц
drop table doc_sch_cab_vis;
drop table cabinet;

```

```
drop table doctors;  
drop table complaint;  
drop table visitors;  
drop table patient;  
drop table people;  
drop table department;  
drop table clinic;  
drop table address;  
drop table schedule;
```

```
-- Удаление базы данных  
drop database bdclinic;
```

3) Сгенерировал СУБД базу данных одnogруппника

