|  |  |  |
| --- | --- | --- |
| Лабораторная работа №3  Физическое проектирование базы данных | Ф.И.О. | Авдеев И. А. |
| Группа | ИВТ-263 |
| Преподаватель | Соколов А.А. |
| Дата сдачи |  |

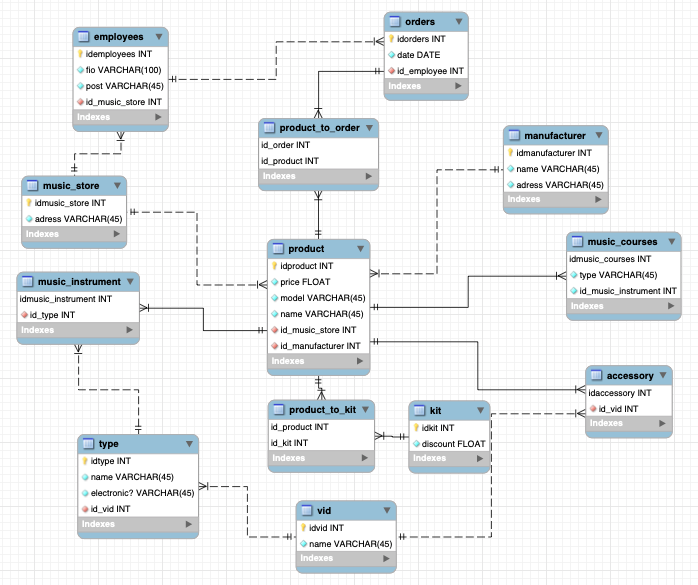
Реализация собственной БД

1. Выбрать СУБД (**MySQL, PostgreSQL, MS SQL Server, Oracle Database**)
2. Выбрать вместе с СУБД графическую оболочку. Как правило, они идут в комплекте с СУБД, но отдельные ссылки тоже приложены: [MySQL Workbench](https://dev.mysql.com/downloads/) (поставляется с MySQL), [pgAdmin](https://www.pgadmin.org/) (поставляется с PostgreSQL), [SQL Server Management Studio](https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15) (поставляется с MS SQL Server) или универсальную: [DBeaver](https://dbeaver.io/) (подключается к любым СУБД)
3. Скачать, установить, настроить, запустить, проверить, что всё работает
4. **Создать собственную БД** средствами графической оболочки или с помощью SQL-скриптов (кто уже разобрался)
5. **Сгенерировать**средствами графической оболочки **реляционную схему**, убедиться, что она совпадает с той, что вы рисовали на 2 лабораторной работе (разница может быть, если вы что-то улучшали при создании БД в СУБД)
6. **Наполнить БД данными** (10+ записей в каждой таблице, но если столько данных возможно придумать, в крайнем случае можно меньше при обсуждении с преподавателем), в интерфейсе или c помощью SQL-скриптов
7. В протокол поместить **изображение реляционной схемы**, которую вы сделали на 2 лабораторной работе, изображение **реляционной схемы, сгенерированную** на основе созданной БД, **скриншоты наполненных таблиц**

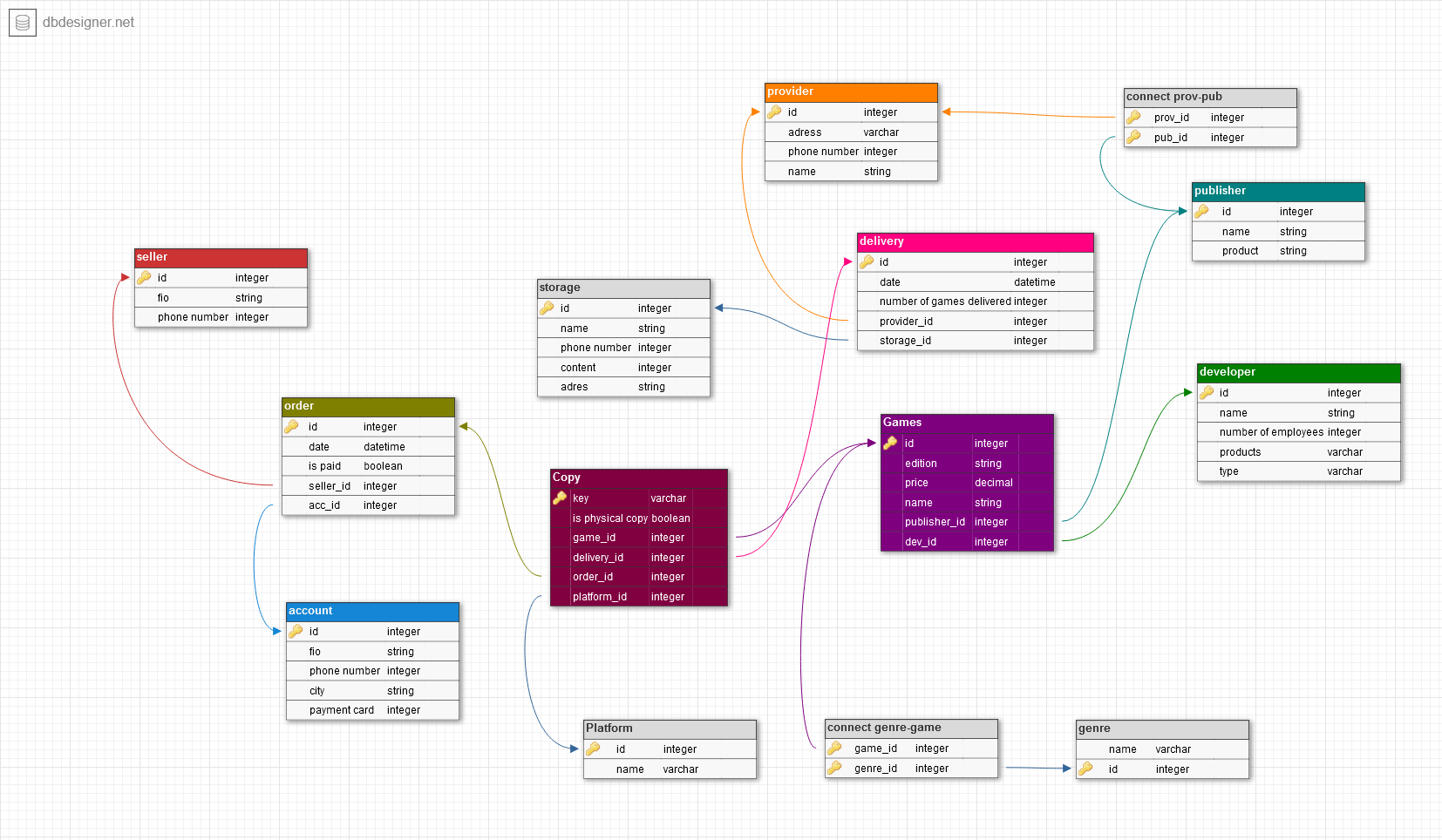
Реализация БД одногруппника

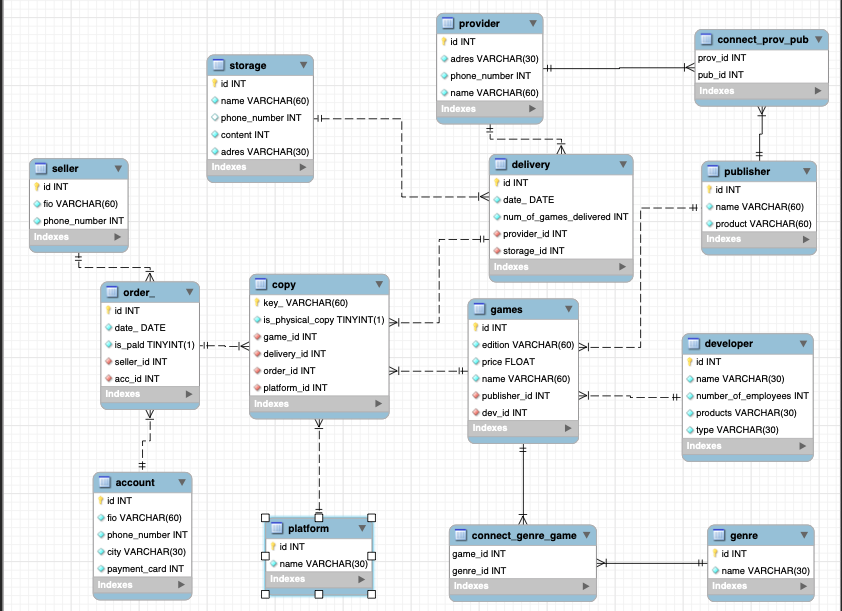
1. В выбранной СУБД реализовать БД вашего одногруппника, идущего**следующем (ниже)** за вами в списке ведомости (последний в списке берёт БД первого в списке), **получить от него реляционную схему или ER-диаграмму**(что есть). Ссылка на ведомость есть в описании курса. Уже не учащихся студентов пропускаем в списке
2. Если одногруппник до сих пор не смог создать свою реляционную схему или ER-диаграмму (обсуждается с преподавателем), необходимо **самому создать одну из следующих БД** (от 5 до 10 таблиц) на выбор: БД учёта потребления энергоресурсов ЖКХ, БД садовода (фермера), БД катка, БД сайта психологических тестов, БД музея, БД завода, БД турбазы
3. Если вы не согласны с тем, как одногруппник создал БД, можно обговорить это с ним или внести свои небольшие изменения, т.е. реализовывать БД точно по схеме не обязательно
4. БД должна быть создана **обязательно**с помощью написанных самостоятельно вручную **SQL-скриптов**создания БД (**create database, create table, alter table**)
5. Необходимо написать**10 доп. запросов на изменение таблиц** (помимо задания внешних ключей). Например, можно заведомо создать таблицы с некорректными наборами столбцов или связями, а потом исправить это) - **alter table**
6. Необходимо написать **10 запросов на добавление данных** (всего 10 на всю БД одногруппника, в **разные**таблицы) - **insert**
7. Необходимо написать 2-3 запроса **RENAME**для таблиц или их атрибутов
8. **Сгенерировать визуализацию реляционной модели** полученной БД и убедиться, что она совпадает с тем, что вам изначально дал одногруппник
9. Написать скрипты удаления таблиц и БД (отдельно **drop table и drop database**). Не забудьте предварительно сделать бэкап, чтобы случайно не потерять БД
10. В протокол поместить изображение **ER-диаграммы, которую вам дал одногруппник**(или реляционной схемы), изображение **реляционной схемы, сгенерированной** на основе созданной БД, текст **всех**написанных **SQL-скриптов**и скриншоты результатов работы скриптов

**Реализация собственной БД**



**Реализация БД одногруппника**



****

CREATE DATABASE Avilov;

USE Avilov;

CREATE TABLE genre (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(30) NOT NULL

);

CREATE TABLE developer (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(30) NOT NULL,

number\_of\_employees INT NOT NULL,

products VARCHAR(30) NOT NULL,

type VARCHAR(30) NOT NULL,

extra\_line INT

);

CREATE TABLE publisher (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(60) NOT NULL,

product VARCHAR(60) NOT NULL,

extra\_line VARCHAR(30)

);

CREATE TABLE provider (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

adres VARCHAR(30) NOT NULL,

phone\_number INT NOT NULL,

name VARCHAR(60) NOT NULL,

extra\_line INT

);

CREATE TABLE storage (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(60) NOT NULL,

phone\_number INT not null,

content INT NOT NULL,

adres VARCHAR(30) NOT NULL

);

CREATE TABLE seller (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

fio VARCHAR(60) NOT NULL,

phone\_number INT NOT NULL

);

CREATE TABLE account (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

fio VARCHAR(60) NOT NULL,

phone\_number INT NOT NULL,

city VARCHAR(30) NOT NULL,

payment\_card INT NOT NULL

);

CREATE TABLE platform (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(30) NOT NULL

);

CREATE TABLE order\_ (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

date\_ DATE NOT NULL,

is\_pald BOOL NOT NULL,

seller\_id INT NOT NULL,

acc\_id INT NOT NULL,

FOREIGN KEY (seller\_id)

REFERENCES seller (id),

FOREIGN KEY (acc\_id)

REFERENCES account (id)

);

CREATE TABLE delivery (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

date\_ DATE NOT NULL,

num\_of\_games\_delivered INT NOT NULL,

provider\_id INT NOT NULL,

storage\_id INT NOT NULL,

FOREIGN KEY (provider\_id)

REFERENCES provider (id),

FOREIGN KEY (storage\_id)

REFERENCES storage (id)

);

CREATE TABLE games (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

edition VARCHAR(60) NOT NULL,

price FLOAT NOT NULL,

name VARCHAR(60) NOT NULL,

publisher\_id INT NOT NULL,

dev\_id INT NOT NULL,

FOREIGN KEY (publisher\_id)

REFERENCES publisher (id),

FOREIGN KEY (dev\_id)

REFERENCES developer (id)

);

CREATE TABLE copy (

key\_ VARCHAR(60) NOT NULL PRIMARY KEY,

is\_physical\_copy BOOL NOT NULL,

game\_id INT NOT NULL,

delivery\_id INT NOT NULL,

order\_id INT NOT NULL,

platform\_id INT NOT NULL,

FOREIGN KEY (game\_id)

REFERENCES games (id),

FOREIGN KEY (delivery\_id)

REFERENCES delivery (id),

FOREIGN KEY (order\_id)

REFERENCES order\_ (id),

FOREIGN KEY (platform\_id)

REFERENCES platform (id)

);

CREATE TABLE connect\_genre\_game (

game\_id INT NOT NULL,

genre\_id INT NOT NULL,

PRIMARY KEY (game\_id , genre\_id),

FOREIGN KEY (game\_id)

REFERENCES games (id),

FOREIGN KEY (genre\_id)

REFERENCES genre (id)

);

CREATE TABLE connect\_prov\_pub (

prov\_id INT NOT NULL,

pub\_id INT NOT NULL,

PRIMARY KEY (prov\_id , pub\_id),

FOREIGN KEY (prov\_id)

REFERENCES provider (id),

FOREIGN KEY (pub\_id)

REFERENCES publisher (id)

);

ALTER TABLE developer

MODIFY COLUMN extra\_line CHAR(100) NULL;

ALTER TABLE developer

DROP COLUMN extra\_line;

ALTER TABLE publisher

MODIFY COLUMN extra\_line int NULL;

ALTER TABLE publisher

DROP COLUMN extra\_line;

ALTER TABLE provider

MODIFY COLUMN extra\_line CHAR(100) NULL;

ALTER TABLE provider

DROP COLUMN extra\_line;

ALTER TABLE storage

ADD extra\_line INT NOT NULL;

ALTER TABLE storage

MODIFY COLUMN extra\_line CHAR(100) NULL;

ALTER TABLE storage

DROP COLUMN extra\_line;

ALTER TABLE seller

ADD extra\_line INT NOT NULL;

ALTER TABLE seller

DROP COLUMN extra\_line;

INSERT genre(id, name)

VALUES (1, 'rpg');

INSERT genre(name)

VALUES ('strategy');

INSERT platform(id, name)

VALUES (1, 'nintendo switch');

INSERT developer(id, name, number\_of\_employees, products, type)

VALUES (1, 'Limbic Entertainment', 80, 'games', 'private');

INSERT publisher(id, name, product)

VALUES (1, 'Kalypso Media', 'video games');

INSERT provider(id, adres, phone\_number, name)

VALUES (1, 'Russia, Volgograd, Kukuevo, 1', 88444210, 'Kukuevo');

INSERT storage(id, name, phone\_number, content, adres)

VALUES (1, 'Kolotushka', 55555, 10000, 'Volgograd, Kolotushkina, 1');

INSERT delivery(id, date\_, num\_of\_games\_delivered, provider\_id, storage\_id)

VALUES (1, '2021-03-11', 50, 1, 1);

INSERT games(id, edition, price, name, publisher\_id, dev\_id)

VALUES (1, 'standart', 750, 'disco elysium', 1, 1);

INSERT connect\_prov\_pub(prov\_id, pub\_id)

VALUES (1, 1);

INSERT publisher(name, product)

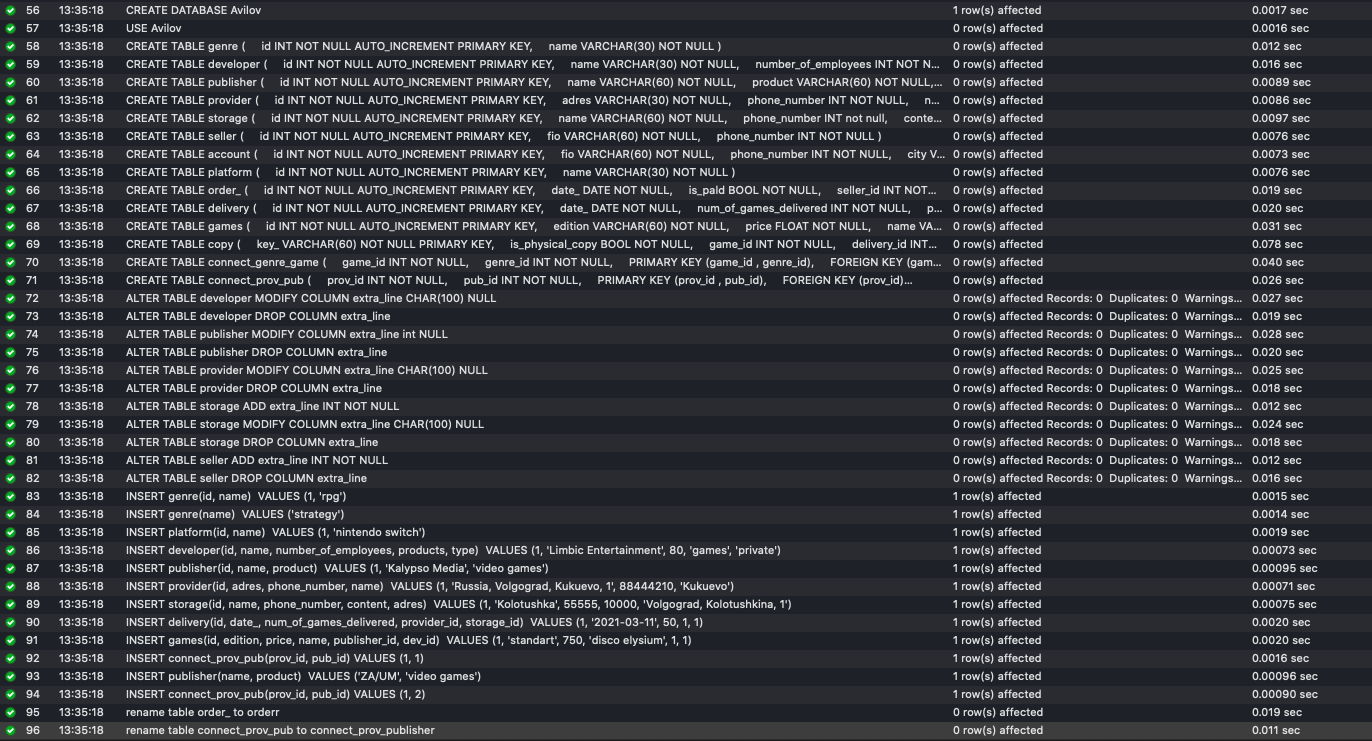
VALUES ('ZA/UM', 'video games');

INSERT connect\_prov\_pub(prov\_id, pub\_id)

VALUES (1, 2);

rename table order\_ to orderr;

rename table connect\_prov\_pub to connect\_prov\_publisher;



mod

#mod n1

drop table connect\_genre\_game;

alter table games

add genre\_id INT NOT NULL;

UPDATE games SET `genre\_id` = '1' WHERE (`id` = '1');

alter table games

ADD FOREIGN KEY(genre\_id) REFERENCES genre(id) ON UPDATE CASCADE ON DELETE CASCADE;

#mod n2

ALTER TABLE copy

DROP FOREIGN KEY copy\_ibfk\_4;

ALTER TABLE copy

DROP COLUMN platform\_id;

CREATE TABLE connect\_copy\_platform (

copy\_key VARCHAR(60) NOT NULL,

platform\_id INT NOT NULL,

PRIMARY KEY (copy\_key , platform\_id),

FOREIGN KEY (copy\_key)

REFERENCES copy (key\_),

FOREIGN KEY (platform\_id)

REFERENCES platform (id)

);

#mod n3

ALTER TABLE orderr

MODIFY date\_ DATETIME DEFAULT NOW();