

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по практической работе**  
**по дисциплине «Основы технологий хранения данных»**  
**Тема: Создание запросов MySQL**

Студентка гр. 6382

Охримук Е.С.

Преподаватель

Борисенко К. А.

Санкт-Петербург

2019

## Выполнение работы.

### 1. Создание базы данных с таблицами и заполнение их данными.

Запрос на создание БД:

```
CREATE DATABASE Shop CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Запросы на создание таблиц:

- Таблица «Clients» (id – уникальный идентификатор, name – имя клиента, surname – фамилия, birthdate – дата рождения, gender – пол, phone – номер телефона, city - город):

```
CREATE TABLE Students (  
    id INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    surname TEXT NOT NULL,  
    birthdate DATA NOT NULL,  
    gender TEXT NOT NULL,  
    phone INT NOT NULL,  
    city TEXT NOT NULL,  
    PRIMARY KEY (id)  
);
```

- Таблица «Employees» (id – уникальный идентификатор, name – имя сотрудника, surname – фамилия, birthdate – дата рождения, gender – пол, post - должность, salary – заработная плата):

```
CREATE TABLE Employees (  
    id INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    surname TEXT NOT NULL,  
    birthdate DATE NOT NULL,  
    gender TEXT NOT NULL,  
    post TEXT NOT NULL,  
    salary INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

- Таблица «Catalog» (number – серийный номер, modelId – id модели)

```
CREATE TABLE Catalog (  
    number INT NOT NULL AUTO_INCREMENT,  
    modelId INT NOT NULL,  
    PRIMARY KEY (number),  
    FOREIGN KEY (modelId) REFERENCES Models (model)
```

);

- Таблица «Models» (model – уникальный идентификатор модели, descr – описание модели, date – год выпуска):

```
CREATE TABLE Models (  
    model INT NOT NULL AUTO_INCREMENT,  
    descr TEXT NOT NULL,  
    date YEAR NOT NULL,  
    PRIMARY KEY (model)
```

);

- Таблица «Orders» (number – уникальный идентификатор заказа, ordate – дата заказа, empId – id сотрудника, clientId – id клиента, numberId – серийный номер, deldate – дата доставки)

```
CREATE TABLE Orders (  
    number INT NOT NULL AUTO_INCREMENT,  
    ordate DATE NOT NULL,  
    empId INT NOT NULL,  
    clientId INT NOT NULL,  
    numberId INT NOT NULL,  
    deldate DATE NOT NULL,  
    PRIMARY KEY (number),  
    FOREIGN KEY (empId) REFERENCES Employees (id),  
    FOREIGN KEY (clientId) REFERENCES Clients (id),  
    FOREIGN KEY (numberId) REFERENCES Catalog (number)
```

);

### Запросы на заполнение таблиц данными:

```
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Артём', 'Спас', '1998-08-02',  
'male', '9993456754', 'Сочи');  
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Екатерина', 'Охримук', '1998-  
04-15', 'female', '99908564754', 'Брест');  
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Альберт', 'Карамышев', '1996-  
12-04', 'male', '9815856754', 'Брест');  
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Артём', 'Азаревич', '1998-02-  
04', 'male', '9185856794', 'СПБ');  
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Илья', 'Быков', '1995-09-  
12', 'male', '9186756794', 'СПБ');  
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Кирилл', 'Никитин', '1998-02-  
06', 'male', '9999056794', 'СПБ');  
INSERT INTO Clients (name, surname, birthdate, gender, phone, city) VALUES ('Ирина', 'Кацер', '1998-08-  
24', 'female', '9189056794', 'Шклов');  
  
INSERT INTO Employees (name, surname, birthdate, gender, post, salary) VALUES ('Ирина', 'Керп',  
'1993-08-09', 'female', 'стажер', '30000');  
INSERT INTO Employees (name, surname, birthdate, gender, post, salary) VALUES ('Глеб', 'Медведев',  
'1991-11-10', 'male', 'секретарь', '80000');  
INSERT INTO Employees (name, surname, birthdate, gender, post, salary) VALUES ('Юрий', 'Михайлов',  
'1996-12-30', 'male', 'стажер', '30000');
```

```
INSERT INTO Employees (name, surname, birthdate, gender, post, salary) VALUES ('Андрей', 'Айгачев', '1986-10-12', 'male', 'продавец', '40000');
```

```
INSERT INTO Models (descr, date) VALUES ('ЖК 48', '2018');
INSERT INTO Models (descr, date) VALUES ('ЖК 40', '2008');
INSERT INTO Models (descr, date) VALUES ('телефон Samsung gs7', '2008');
INSERT INTO Models (descr, date) VALUES ('телефон Samsung gs9', '2016');
INSERT INTO Models (descr, date) VALUES ('телефон Iphone 7', '2015');
INSERT INTO Models (descr, date) VALUES ('телефон Iphone 6', '2014');
INSERT INTO Models (descr, date) VALUES ('телефон Iphone 6s', '2014');
INSERT INTO Models (descr, date) VALUES ('ЖК 26', '2010');
INSERT INTO Models (descr, date) VALUES ('ЖК 32', '2010');
INSERT INTO Models (descr, date) VALUES ('ЖК 37', '2011');
INSERT INTO Models (descr, date) VALUES ('холодильник lg 400л', '2011');
INSERT INTO Models (descr, date) VALUES ('холодильник lg 4500л', '2014');
INSERT INTO Models (descr, date) VALUES ('холодильник lg 900л', '2018');
INSERT INTO Models (descr, date) VALUES ('холодильник lg 600л', '2015');
INSERT INTO Models (descr, date) VALUES ('ЖК 46', '2017');
INSERT INTO Models (descr, date) VALUES ('телефон Iphone 10', '2099');
INSERT INTO Models (descr, date) VALUES ('телефон Iphone 5', '2009');
```

```
INSERT INTO Catalog (modelId) VALUES ('1');
INSERT INTO Catalog (modelId) VALUES ('2');
INSERT INTO Catalog (modelId) VALUES ('3');
INSERT INTO Catalog (modelId) VALUES ('3');
INSERT INTO Catalog (modelId) VALUES ('4');
INSERT INTO Catalog (modelId) VALUES ('4');
INSERT INTO Catalog (modelId) VALUES ('4');
INSERT INTO Catalog (modelId) VALUES ('5');
INSERT INTO Catalog (modelId) VALUES ('6');
INSERT INTO Catalog (modelId) VALUES ('7');
INSERT INTO Catalog (modelId) VALUES ('8');
INSERT INTO Catalog (modelId) VALUES ('9');
INSERT INTO Catalog (modelId) VALUES ('10');
INSERT INTO Catalog (modelId) VALUES ('11');
INSERT INTO Catalog (modelId) VALUES ('12');
INSERT INTO Catalog (modelId) VALUES ('12');
INSERT INTO Catalog (modelId) VALUES ('12');
INSERT INTO Catalog (modelId) VALUES ('12');
INSERT INTO Catalog (modelId) VALUES ('12');
INSERT INTO Catalog (modelId) VALUES ('13');
INSERT INTO Catalog (modelId) VALUES ('14');
INSERT INTO Catalog (modelId) VALUES ('15');
INSERT INTO Catalog (modelId) VALUES ('16');
INSERT INTO Catalog (modelId) VALUES ('17');
```

```
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2019-02-02', '1', '1', '1', '2019-02-20');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2019-01-08', '3', '7', '2', '2019-05-09');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2018-09-11', '3', '1', '3', '2019-08-19');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2018-10-01', '1', '4', '4', '2018-12-19');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2017-05-11', '4', '5', '5', '2017-12-09');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2017-05-11', '4', '6', '6', '2017-11-05');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2016-06-19', '3', '7', '7', '2017-01-03');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2016-03-09', '1', '4', '8', '2016-11-03');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2018-08-17', '1', '2', '9', '2018-11-11');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2018-01-17', '4', '3', '10', '2018-01-19');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2019-04-19', '3', '3', '11', '2019-07-19');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2019-05-19', '3', '6', '12', '2019-08-11');
INSERT INTO Orders (ordate, empId, clientId, numberId, deldate) VALUES ('2017-05-19', '1', '6', '13', '2018-03-11');
```

# Итоговые таблицы:

id	name	surname	birthdate	gender	post	salary
1	Ирина	Кепр	1993-08-09	female	стажер	30000
2	Глеб	Медведев	1991-11-10	male	секретарь	80000
3	Юрий	Михайлов	1996-12-30	male	стажер	30000
4	Андрей	Айгачев	1986-10-12	male	продавец	40000

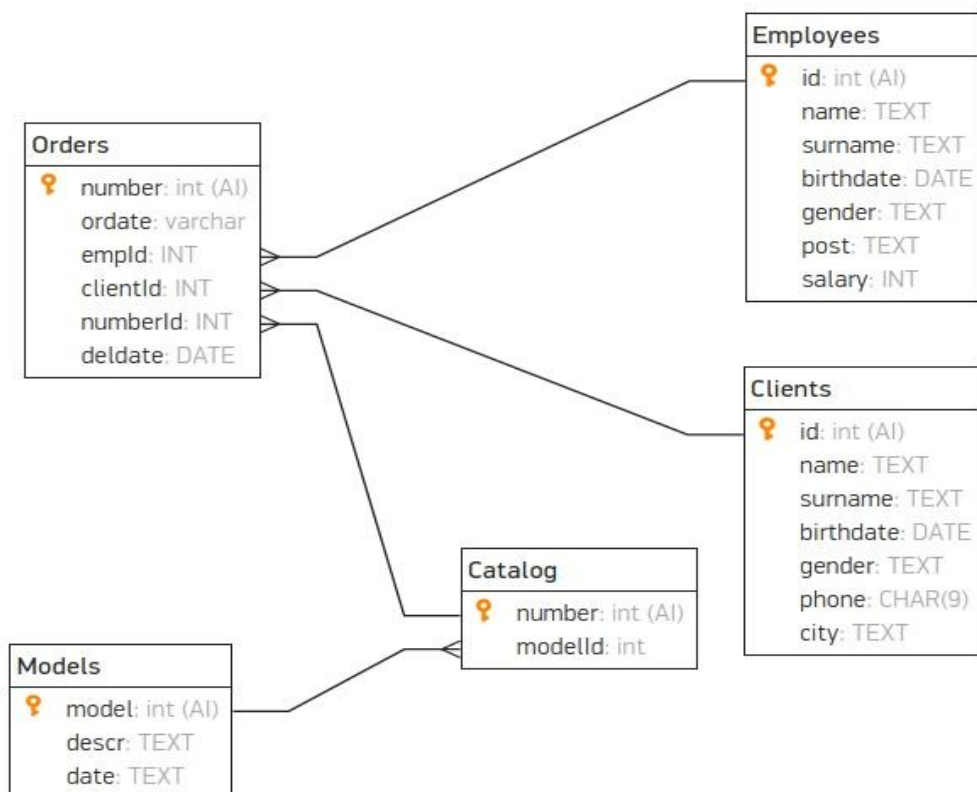
id	name	surname	birthdate	gender	phone	city
1	Артем	Спас	1998-08-02	male	9993456754	Сочи
2	Екатерина	Охримук	1998-04-15	female	9990856754	Брест
3	Альберт	Карамышев	1996-12-04	male	9815856754	Брест
4	Артем	Азаревич	1998-02-04	male	9185856794	СПБ
5	Илья	Быков	1995-09-12	male	9186756794	СПБ
6	Кирилл	Никитин	1998-02-06	male	9999056794	СПБ
7	Ирина	Кацер	1998-08-24	female	9189056794	Шклов

model	descr	date
1	ЖК 48	2018
2	ЖК 40	2008
3	телефон Samsung gs7	2008
4	телефон Samsung gs9	2016
5	телефон Iphone 7	2015
6	телефон Iphone 6	2014
7	телефон Iphone 6s	2014
8	ЖК 26	2010
9	ЖК 32	2010
10	ЖК 37	2011
11	холодильник lg 400л	2011
12	холодильник lg 4500л	2014
13	холодильник lg 900л	2018
14	холодильник lg 600л	2015
15	ЖК 46	2017
16	телефон Iphone 10	2019
17	телефон Iphone 5	2009

number	ordate	empId	clientId	numberId	deldate
1	2019-02-02	1	1	1	2019-02-20
2	2019-01-08	3	7	2	2019-05-09
3	2018-09-11	3	1	3	2019-08-19
4	2018-10-01	1	4	4	2018-12-19
5	2017-05-11	4	5	5	2017-12-09
6	2017-05-11	4	6	6	2017-11-05
7	2016-06-19	3	7	7	2017-01-03
8	2016-03-09	1	4	8	2016-11-03
9	2018-08-17	1	2	9	2018-11-11
10	2018-01-17	4	3	10	2018-01-19
11	2019-04-19	3	3	11	2019-07-19
12	2019-05-19	3	6	12	2019-08-11
13	2017-05-19	1	6	13	2018-03-11

number	modelId
1	1
2	2
3	3
4	3
5	4
6	4
7	4
8	5
9	6
10	7
11	8
12	9
13	10
14	11
15	12
16	12
17	12
18	12
19	12
20	13
21	14
22	15
23	16
24	17

Схема данных представлена на рисунке:



## 2. Создание запросов к данной базе данных:

### 2.1. Вывести имя, фамилию клиентов, которые заказывали товары в 2018 году, данный список отсортировать по дате.

```
SELECT name, surname, ordate
FROM Orders
JOIN Clients
ON Orders.clientId = Clients.id
WHERE ordate BETWEEN '2018-01-01' AND '2019-01-01'
ORDER BY ordate ASC;
```

name	surname	ordate
Альберт	Карамышев	2018-01-17
Екатерина	Охримук	2018-08-17
Артем	Спас	2018-09-11
Артем	Азаревич	2018-10-01

### 2.2. Вывести имя и фамилию сотрудника, который оформил больше всего заказов за 2018 год.

```
SELECT name, surname, Number
FROM (
    SELECT name, surname, empId, count(*) AS "Number"
    FROM Orders
    JOIN Employees
    ON Orders.empId = Employees.id
    WHERE ordate BETWEEN '2018-01-01' AND '2019-01-01'
    GROUP BY empId
    LIMIT 1) AS Result;
```

name	surname	Number
Ирина	Кепр	2

### 2.3. Вывести имена и фамилии клиентов и имена и фамилии сотрудников которые с ними работали, а также описание товара который они им продали.

```
SELECT Clients.name, Clients.surname, Employees.name, Employees.surname, descr
FROM Orders
```

JOIN Clients

ON Orders.clientId = Clients.id

JOIN Employees

ON Orders.empId = Employees.id

JOIN (

SEIECT \* FROM Catalog

JOIN Models

ON Catalog.modelId = Models.model)

AS Result

ON Orders.numberId = Result.number

name	surname	name	surname	descr
Артём	Спас	Ирина	Кепр	ЖК 48
Артём	Азаревич	Ирина	Кепр	телефон Samsung gs7
Артём	Азаревич	Ирина	Кепр	телефон Iphone 7
Екатерина	Охримук	Ирина	Кепр	телефон Iphone 6
Кирилл	Никитин	Ирина	Кепр	ЖК 37
Ирина	Кацер	Юрий	Михайлов	ЖК 40
Артём	Спас	Юрий	Михайлов	телефон Samsung gs7
Ирина	Кацер	Юрий	Михайлов	телефон Samsung gs9
Альберт	Карамышев	Юрий	Михайлов	ЖК 26
Кирилл	Никитин	Юрий	Михайлов	ЖК 32
Илья	Быков	Андрей	Айгачев	телефон Samsung gs9
Кирилл	Никитин	Андрей	Айгачев	телефон Samsung gs9
Альберт	Карамышев	Андрей	Айгачев	телефон Iphone 6s

## 2.4. Вывести имена и фамилии клиентов в алфавитном порядке, у которых в описании купленного товара встречается слово “телефон”.

SELECT Clients.name, Clients.surname, descr

FROM Orders

JOIN Clients

ON Orders.clientId = Clients.id

JOIN (

Select \* FROM Catalog

JOIN Models

ON Catalog.modelId = Models.model)

AS Result ON Orders.numberId = Result.number

WHERE descr LIKE "%телефон%"

ORDER BY surname ASC;



name	surname	descr
Арте́м	Аза́ревич	телефон Samsung gs7
Арте́м	Аза́ревич	телефон Iphone 7
Илья́	Быков	телефон Samsung gs9
Альбе́рт	Кара́мышев	телефон Iphone 6s
Ири́на	Каце́р	телефон Samsung gs9
Кири́лл	Ники́тин	телефон Samsung gs9
Екатери́на	Охри́мук	телефон Iphone 6
Арте́м	Спа́с	телефон Samsung gs7

## 2.5. Вывести кол-во заказов для жителей города СПб оформленные сотрудником Михайловым.

```

SELECT concat(Employees.name, " ", Employees.surname) Name, COUNT(*) AS Number
FROM Orders JOIN Clients ON Orders.clientId = Clients.id
JOIN Employees
ON Orders.empId = Employees.id
WHERE Employees.surname = 'Михайлов'
AND Clients.city = 'СПБ'
GROUP BY Name;

```

Name	Number
Юрий Михайлов	1

## 2.6. Вывести 3 товара, которые доставлялись быстрее всего и куда.

```

SELECT city, descr, datediff(deldate,ordate) AS "Diff"
FROM Orders
JOIN Clients
ON Orders.clientId = Clients.id
JOIN (
    Select * FROM Catalog
    JOIN Models
    ON Catalog.modelId = Models.model)
AS Result
ON Orders.numberId = Result.number
ORDER BY Diff LIMIT 3;

```

city	descr	Diff
Брест	телефон Iphone 6s	2
Сочи	ЖК 48	18
СПБ	телефон Samsung gs7	79

2.7. Вывести имена и фамилии клиентов мужского пола, которые покупали Samsung.

```

SELECT descr, Clients.name, Clients.surname
FROM Orders
JOIN Clients
ON Orders.clientId = Clients.id
JOIN (
    Select * FROM Catalog
    JOIN Models ON Catalog.modelId = Models.model)
AS Result
ON Orders.numberId = Result.number
WHERE descr LIKE "%Samsung%"
AND gender = 'male';

```

descr	name	surname
телефон Samsung gs7	Артем	Спас
телефон Samsung gs7	Артем	Азаревич
телефон Samsung gs9	Илья	Быков
телефон Samsung gs9	Кирилл	Никитин

2.8. Вывести имена и фамилию клиентов и описание товаров 2008-2012 годов выпуска, которые купили за 2019 год женщины.

```

SELECT name, surname, descr
FROM (
    SELECT Clients.name, Clients.surname, descr, gender
    From Orders
    JOIN Clients
    ON Orders.clientId = Clients.id
    JOIN (
        SELECT * FROM Catalog
        JOIN Models

```

```

        ON Catalog.modelId = Models.model) AS Result
    ON Orders.numberId = Result.number
    WHERE date BETWEEN 2008 AND 2012
    AND (ordate BETWEEN "2019-01-01" AND "2019-12-31")) AS New
    WHERE New.gender = 'female';

```

name	surname	descr
Ирина	Кацер	ЖК 40

2.9. Вывести имена и фамилии клиентов, у которых номер начинается с 999, и годы выпуска товаров которые они купили.

```

SELECT name, surname, date
FROM Orders
JOIN Clients
ON Orders.clientId = Clients.id
JOIN (
    SELECT * FROM Catalog
    JOIN Models
    ON Catalog.modelId = Models.model)
AS Result
ON Orders.numberId = Result.number
WHERE phone LIKE '999%';

```

name	surname	date
Арте́м	Спас	2018
Арте́м	Спас	2008
Екатерина	Охримук	2014
Кирилл	Никитин	2016
Кирилл	Никитин	2010
Кирилл	Никитин	2011

2.10. Вывести средний год выпуска распроданных товаров за апрель-август 2018 года.

```

SELECT AVG(date) AS Average
FROM Catalog
JOIN Models
ON Catalog.modelId = Models.model

```

JOIN Orders

ON Catalog.modelId = Orders.numberId

WHERE orddate BETWEEN "2018-04-01" AND "2018-09-01";

Average
2010.0000

2.11. Вывести кол-во купленных товаров женщинами в 2018 году.

SELECT gender, COUNT(\*) AS Result

FROM Orders

JOIN Clients

ON Orders.clientId = Clients.id

GROUP BY gender

having gender = 'female';

gender	Result
female	3

2.12. Вывести имя, фамилию и описание купленного товара по номеру телефона покупателя который начинается с 918.

SELECT concat(Clients.name, " ", Clients.surname) Name, descr

FROM Orders

JOIN Clients

ON Orders.clientId = Clients.id

JOIN Employees

ON Orders.empId = Employees.id

JOIN (

SELECT \* FROM Catalog

JOIN Models

ON Catalog.modelId = Models.model)

AS Result

ON Orders.numberId = Result.number

WHERE phone LIKE '918%';

Name	descr
Артём Азаревич	телефон Samsung gs7
Артём Азаревич	телефон Iphone 7
Илья Быков	телефон Samsung gs9
Ирина Кацер	ЖК 40
Ирина Кацер	телефон Samsung gs9

2.13. Вывести зарплату сотрудника который был вторым по продажам до 2019 года.

```
SELECT name, surname, salary
FROM (
    SELECT name, surname, empId, salary, count(*) AS "Number"
    FROM Orders
    JOIN Employees
    ON Orders.empId = Employees.id
    WHERE orddate <= '2019-01-01'
    GROUP BY empId
    LIMIT 1 OFFSET 1)
AS Result;
```

name	surname	salary
Юрий	Михайлов	30000

2.14. Вывести товары и их ожидаемые даты доставки в порядке ближайшей доставки, которые уже купили, но еще не доставили.

```
SELECT descr, deldate
FROM Catalog
JOIN Models
ON Catalog.modelId = Models.model
JOIN Orders
ON Catalog.number = Orders.numberId
WHERE CURDATE( ) < deldate
ORDER BY deldate asc;
```

descr	deldate
ЖК 26	2019-07-19
ЖК 32	2019-08-11
телефон Samsung gs7	2019-08-19

2.15. Вывести средний возраст покупателей и средний возраст сотрудников.

```

SELECT AVG(ABS(YEAR(Clients.birthdate)- YEAR(CURDATE()))))
AS "AVG_AGE_Clients",
AVG(ABS(YEAR(Employees.birthdate)- YEAR(CURDATE()))))
AS "AVG_AGE_EMP"
FROM Orders
JOIN Clients
ON Orders.clientId = Clients.id
JOIN Employees
ON Orders.empId = Employees.id;

```

AVG_AGE_Clients	AVG_AGE_EMP
21.5385	26.4615

## Выполнение работы.

### 1. Создание базы данных с таблицами и заполнение их данными.

Запрос на создание БД:

```
CREATE DATABASE Library CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Запросы на создание таблиц:

- Таблица «Студенты» (id – уникальный идентификатор, name – имя студента, surname – фамилия, birthdate – день рождение, gender – пол, class - группа):

```
CREATE TABLE Students (  
    id INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    surname TEXT NOT NULL,  
    birthdate DATA NOT NULL DEFAULT CURDATE(),  
    gender TEXT NOT NULL,  
    class INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

- Таблица «Автор» (authorID – уникальный идентификатор автора, name – имя автора, surname – фамилия автора)

```
CREATE TABLE Authors (  
    authorID INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    surname TEXT NOT NULL,  
    PRIMARY KEY (authorID)  
);
```

- Таблица «Типы» (typeID – уникальный идентификатор типа, name – название типа):

```
CREATE TABLE Types (  
    typeID INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    PRIMARY KEY (typeID)  
);
```

- Таблица «Книги» (bookID – уникальный идентификатор книги, name – имя автора, pagecount – количество страниц, authorID – уникальный идентификатор автора, typeID – уникальный идентификатор типа)

```
CREATE TABLE Books (  
    bookID INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    pagecount INT NOT NULL,  
    authorID INT NOT NULL,  
    typeID INT NOT NULL,  
    PRIMARY KEY (bookID),  
    FOREIGN KEY (authorID) REFERENCES Authors (authorID),  
    FOREIGN KEY (typeID) REFERENCES Types (typeID)
```

```

        bookID INT NOT NULL AUTO_INCREMENT,
        name TEXT NOT NULL,
        pagecount INT NOT NULL,
        authorID INT NOT NULL,
        typeId INT NOT NULL,
        PRIMARY KEY (bookID),
        FOREIGN KEY (authorID) REFERENCES Authors (authorID),
        FOREIGN KEY (typeID) REFERENCES Types (typeID)
    );

```

- Таблица «Взятые книги» (borrowID – уникальный идентификатор взятой книги, studentID – уникальный идентификатор, bookID – уникальный идентификатор книги, takenDate – дата взятия, broughtDate – дата возврата)

```

CREATE TABLE Borrows (
    borrowID INT NOT NULL AUTO_INCREMENT,
    studentID INT NOT NULL,
    bookID INT NOT NULL,
    takenDate DATE NOT NULL,
    broughtDate DATE NULL,
    PRIMARY KEY (borrowID),
    FOREIGN KEY (studentID) REFERENCES Authors (id),
    FOREIGN KEY (bookID) REFERENCES Types (typeID)
);

```

## Запросы на заполнение таблиц данными:

```

INSERT INTO Authors (name, surname) VALUES ('Александр', 'Пушкин');
INSERT INTO Authors (name, surname) VALUES ('Иван', 'Тургенев');
INSERT INTO Authors (name, surname) VALUES ('Сергей', 'Довлатов');
INSERT INTO Authors (name, surname) VALUES ('Мария', 'Цветаева');
INSERT INTO Authors (name, surname) VALUES ('Николай', 'Некрасов');
INSERT INTO Authors (name, surname) VALUES ('Эдгар', 'По');
INSERT INTO Authors (name, surname) VALUES ('Стивен', 'Кинг');
INSERT INTO Authors (name, surname) VALUES ('Даниэль', 'Дефо');

INSERT INTO Types (name) VALUES ('Роман');
INSERT INTO Types (name) VALUES ('Приключения');
INSERT INTO Types (name) VALUES ('Трагедия');
INSERT INTO Types (name) VALUES ('Сказка');
INSERT INTO Types (name) VALUES ('Биография');
INSERT INTO Types (name) VALUES ('Ужасы');
INSERT INTO Types (name) VALUES ('Стихотворение');
INSERT INTO Types (name) VALUES ('Поэзия');

INSERT INTO Books (name, pagecount, authorID, typeId) VALUES ('Сказка о рыбаке и рыбке', '8', '1', '5',);
INSERT INTO Books (name, pagecount, authorID, typeId) VALUES ('Евгений Онегин', '278', '1', '2',);
INSERT INTO Books (name, pagecount, authorID, typeId) VALUES ('Борис Годунов', '187', '1', '4',);
INSERT INTO Books (name, pagecount, authorID, typeId) VALUES ('Ворон', '11', '6', '8',);
INSERT INTO Books (name, pagecount, authorID, typeId) VALUES ('Черный кот', '9', '6', '7',);
INSERT INTO Books (name, pagecount, authorID, typeId) VALUES ('Кому на Руси жить хорошо', '305', '5', '9',);

```



```

INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Крестинские дети', '14', '5', '9',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Отцы и дети', '288', '2', '2',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Дворянское гнездо', '218', '2', '2',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Компромисс', '201', '3', '2',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Приключения Робина Круза', '138', '8', '3',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Кому на Руси жить хорошо', '305', '5', '9',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Молль Флендерс', '416', '8', '2',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Пленный дух', '336', '4', '6',);
INSERT INTO Books (name, pagecount, authorID, typeID) VALUES ('Под куполом', '1968', '7', '1',);

INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS) VALUES('НИКИТА','ПОПОВ','1998-05-17','MALE', '6381');
INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS) VALUES('ЕКАТЕРИНА','ОХРИМУК','1999-04-15','FEMALE', '6382');
INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS), VALUES('СПАС','АРТЕМ','1998-08-02','MALE', '6383');
INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS) VALUES('КИРИЛЛ','НИКИТИН','1998-08-06','MALE', '6383');
INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS) VALUES('АЛЬБЕРТ','КАРАМЫШЕВ','1995-12-04','MALE', '6383');
INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS) VALUES('ИРИНА','КАЦЕР','1999-09-24','FEMALE', '6382');
INSERT INTO STUDENTS (NAME, SURNAME, BIRTHDATE, GENDER, CLASS) VALUES('МИХАИЛ','ШОРОХОВ','1999-04-24','MALE', '6103');

INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('1','2','2019-04-24','2019-04-30');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('2','1','2018-04-24','2018-04-30');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE) VALUES('3','3','2019-05-14');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('4','8','2019-02-14','2019-03-1');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('4','4','2019-02-15','2019-03-1');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('5','5','2018-09-15','2018-10-31');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('6','6','2018-10-15','2018-10-31');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('7','7','2018-11-11','2018-11-20');
INSERT INTO BORROWS (STUDENTID, BOOKID, TAKENDATE, BROUGHTDATE) VALUES('7','9','2018-11-11','2018-11-20');

```

## Итоговые таблицы:

```
mysql> select * from Authors
-> ;
```

authorId	name	surname
1	Александр	Пушкин
2	Иван	Тургенев
3	Сергей	Довлатов
4	Мария	Цветаева
5	Николай	Некрасов
6	Эдгар	По
7	Стивен	Кинг
8	Даниэль	Дефо

8 rows in set (0.00 sec)

```
mysql> select * from Books;
```

bookId	name	pagecount	authorId	typeId
1	Сказка о рыбаке и рыбке	8	1	5
2	Евгений Онегин	278	1	2
3	Борис Годунов	187	1	4
4	Ворон	11	6	8
5	Черный кот	9	6	7
6	Кому на Руси жить хорошо	305	5	9
7	Крестьянские дети	14	5	9
8	Отцы и дети	288	2	2
9	Дворянское гнездо	218	2	2
10	Компромисс	201	3	2
11	Приключения Робинзона Крузо	138	8	3
12	Молль Флендерс	416	8	2
13	Пленный дух	336	4	6
14	Под куполом	1968	7	1

14 rows in set (0.00 sec)

```
mysql> select * from Borrows;
```

borrowId	studentId	bookId	takenDate	broughtDate
1	1	2	2019-04-24	2019-04-30
2	2	1	2018-04-24	2018-04-30
3	3	3	2019-05-14	NULL
4	4	8	2019-02-14	2019-03-01
5	4	4	2019-02-15	2019-03-01
6	5	5	2018-09-15	2018-10-31
7	6	6	2018-10-15	2018-10-31
8	7	7	2018-11-11	2018-11-20
9	7	9	2018-11-11	2018-11-20
10	5	10	2018-11-11	2018-11-20
11	2	11	2019-04-11	2019-03-20

11 rows in set (0.00 sec)

```
mysql> select * from Students;
```

id	name	surname	birthdate	gender	class
1	Никита	Попов	1998-05-17	male	6381
2	Екатерина	Охримук	1999-04-15	female	6382
3	Спас	Артем	1998-08-02	male	6383
4	Кирилл	Никитин	1998-08-06	male	6383
5	Альберт	Карамышев	1995-12-04	male	6383
6	Ирина	Кацер	1999-09-24	female	6382
7	Михаил	Шорохов	1999-04-24	male	6103

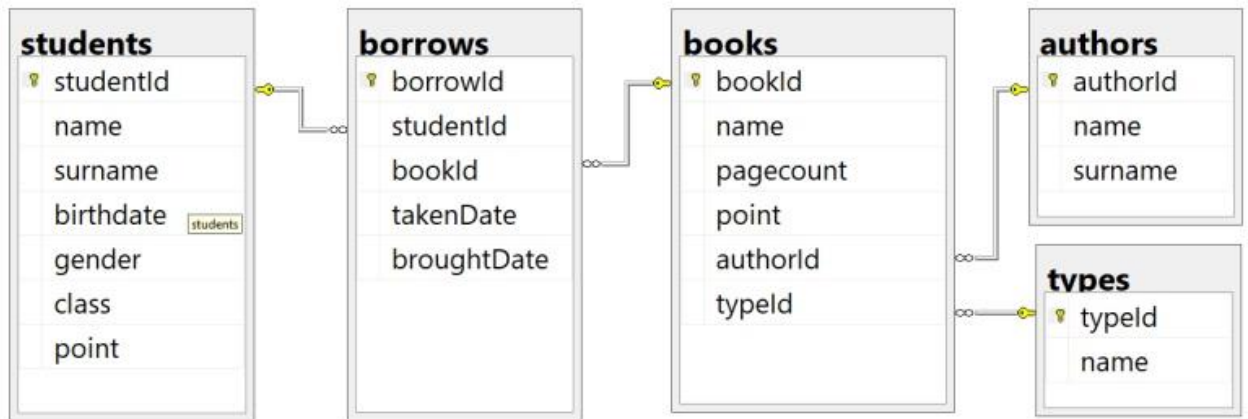
7 rows in set (0.00 sec)

```
mysql> select * from Types;
```

typeId	name
1	Фантастика
2	Роман
3	Приключения
4	Трагедия
5	Сказка
6	Биография
7	Ужасы
8	Стихотворение
9	Поэзия

9 rows in set (0.00 sec)

Схема данных представлена на рисунке:



## 2. Создание запросов к данной базе данных:

### 2.1. Вывести среднее количество страниц в книгах, которые брали студенты мужского пола.

```
SELECT AVG(pagecount) AS "Average pagecount"
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
JOIN Books
ON Borrows.bookId = Books.bookId
WHERE gender = 'male';
```

Average pagecount
150.7500

### 2.2. Вывести имена и фамилии студентов, бравших книгу с жанром Роман.

```
SELECT Students.name, Students.surname
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
WHERE Borrows.bookId IN
    (SELECT Books.bookid
     FROM Books
     JOIN Types
     ON Books.typeId = Types.typeId
     WHERE Types.name = 'Roman');
```

WHERE Types.name = 'Роман');

name	surname
Никита	Попов
Кирилл	Никитин
Михаил	Шорохов
Альберт	Карамышев

2.3. Вывести имена и фамилии студентов, мужского пола из группы 6381.

```
SELECT NAME, SURNAME
FROM STUDENTS
WHERE GENDER = 'MALE'
AND CLASS = 6381;
```

name	surname
Никита	Попов

2.4. Вывести по возрастанию даты имена и фамилии студентов, а также книги, которые они брали в 2019 году.

```
SELECT Students.name, Students.surname,
Books.name AS "Title", Borrows.takenDate
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
JOIN Books
ON Borrows.bookId = Books.bookId
WHERE takenDate > "2019-01-01"
ORDER BY takenDate ASC;
```

name	surname	Title	takenDate
Кирилл	Никитин	Отцы и дети	2019-02-14
Кирилл	Никитин	Ворон	2019-02-15
Екатерина	Охримук	Приключения Робинзона Крузо	2019-04-11
Никита	Попов	Евгений Онегин	2019-04-24
Спас	Артем	Борис Годунов	2019-05-14

2.5. Вывести имена и фамилии студентов, которые родились в августе 1998.

```

SELECT name, surname
FROM Students
WHERE gender = 'male'
AND class = 6383
AND birthdate BETWEEN "1998-08-01"
AND "1998-09-1";

```

name	surname
Спас	Артем
Кирилл	Никитин

2.6. Вывести имена и фамилии студентов, мужского пола, учащихся группы 6383, которые родились в августе 1998 и бравших книгу с жанром Трагедия.

```

SELECT name, surname
FROM Borrows JOIN
  (SELECT * FROM Students
   WHERE gender = 'male'
   AND class = 6383
   AND birthdate BETWEEN "1998-08-01"
   AND "1998-09-1")
AS Result ON Borrows.studentId = Result.id
WHERE Borrows.bookId IN
  (SELECT Books.bookid FROM Books
   JOIN Types
   ON Books.typeId = Types.typeId
   WHERE Types.name = 'Трагедия');

```

name	surname
Артем	Спас

2.7. Вывести в порядке убывания по кол-ву взятых книг номера групп и кол-во ЭТИХ КНИГ.

```

SELECT class FROM

```

```

(SELECT class, COUNT(*) AS "Number"
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
GROUP BY class
ORDER BY Number DESC)

```

AS Result;

class
6383
6382
6103
6381

2.8. Вывести в порядке убывания по кол-ву взятых книг с жанром Роман номера групп.

```

SELECT class
FROM (
    SELECT class, COUNT(*) AS "Number"
    FROM Borrows
    JOIN Students
    ON Borrows.studentId = Students.id
    WHERE Borrows.bookId IN (
        SELECT Books.bookid
        FROM Books
        JOIN Types
        ON Books.typeId = Types.typeId
        WHERE Types.name = 'Роман')
    GROUP BY class
    ORDER BY Number DESC)

```

AS Result;

class
6383
6381
6103

2.9. Вывести название жанра и количество книг каждого жанра в порядке убывания.

```
SELECT Types.name, count(*) AS "Number"
FROM Types
JOIN Books
ON Types.typeId = Books.typeId
GROUP BY name
ORDER BY Number DESC ;
```

name	Number
Роман	5
Поэзия	2
Фантастика	1
Приключения	1
Трагедия	1
Сказка	1
Биография	1
Ужасы	1
Стихотворение	1

2.10. Вывести имя и фамилии студентов и книги, которые они брали.

```
SELECT Students.name, Students.surname, Books.name AS "Title"
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
JOIN Books
ON Borrows.bookId = Books.bookId;
```



name	surname	Title
Никита	Попов	Евгений Онегин
Екатерина	Охримук	Сказка о рыбаке и рыбке
Екатерина	Охримук	Приключения Робинзона Крузо
Артем	Спас	Борис Годунов
Кирилл	Никитин	Отцы и дети
Кирилл	Никитин	Ворон
Альберт	Карамышев	Черный кот
Альберт	Карамышев	Компромисс
Ирина	Кацер	Кому на Руси жить хорошо
Михаил	Шорохов	Крестьянские дети
Михаил	Шорохов	Дворянское гнездо

2.11. Вывести названия книг и имена и фамилии студентов, которые брали книги автора Пушкин.

```

SELECT Students.name, Students.surname, Books.name AS "Title"
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
JOIN Books
ON Borrows.bookId = Books.bookId
WHERE Borrows.bookId IN
    (SELECT Books.bookid
     FROM Books
     JOIN Authors
     ON Books.authorId = Authors.authorId
     WHERE Authors.surname = 'Пушкин');
```

name	surname	Title
Екатерина	Охримук	Сказка о рыбаке и рыбке
Никита	Попов	Евгений Онегин
Артем	Спас	Борис Годунов

2.12. Вывести кол-во студентов мальчиков, которые взяли книгу в 2018 году.

```

SELECT Students.gender, COUNT(*) AS Number
FROM Borrows
JOIN Students
ON Borrows.studentId = Students.id
WHERE takenDate BETWEEN "2018-01-01" AND "2019-01-01"
```



GROUP BY gender

HAVING gender = 'male';

gender	Number
male	4

2.13. Вывести список студентов, отсортированных в алфавитном порядке по фамилиям, с книгами, которые они брали и вернули за период апрель - ноябрь 2018 года.

```
SELECT Students.name, Students.surname, Books.name AS "Title"
```

```
FROM Borrows
```

```
JOIN Students
```

```
ON Borrows.studentId = Students.id
```

```
JOIN Books
```

```
ON Borrows.bookId = Books.bookId
```

```
WHERE takenDate >= "2018-04-01"
```

```
AND broughtDate <= "2018-12-01"
```

```
ORDER BY surname ASC;
```

name	surname	Title
Альберт	Карамышев	Черный кот
Альберт	Карамышев	Компромисс
Ирина	Кацер	Кому на Руси жить хорошо
Екатерина	Охримук	Сказка о рыбаке и рыбке
Михаил	Шорохов	Крестьянские дети
Михаил	Шорохов	Дворянское гнездо

2.14. Вывести книгу, которую прочли быстрее всего и имя и фамилию студента, который ее взял и вернул быстрее всего и количество затраченных на это дней.

```
SELECT Students.name, Students.surname,
```

```
Books.name AS "title", datediff(broughtDate, takenDate) AS "diff"
```

```
FROM Borrows
```

```
JOIN Books
```

```
ON Borrows.bookId = Books.bookId
```

```
JOIN Students
```

```
ON Borrows.studentId = Students.id
```

```
WHERE broughtDate is not null
```

```
ORDER BY diff ASC
```

LIMIT 1;

name	surname	title	diff
Никита	Попов	Евгений Онегин	6

2.15. Вывести список книг вместе с их жанрами, которые брали студенты с именами, в которых содержится буква «р», упорядочить их имена по алфавиту и вывести первые 5.

```
SELECT firstname, surname, title, Types.name AS "Type_name"
```

```
FROM (
```

```
    SELECT Students.name AS "firstname", Students.surname,
```

```
    Books.name AS "Title", typeId
```

```
FROM Borrows
```

```
JOIN Students
```

```
ON Borrows.studentId = Students.id
```

```
JOIN Books
```

```
ON Borrows.bookId = Books.bookId) AS Result
```

```
JOIN Types
```

```
ON Result.typeId = Types.typeId
```

```
WHERE firstname LIKE '%p%'
```

```
ORDER BY firstname ASC
```

```
LIMIT 5;
```

firstname	surname	Title	Type_name
Альберт	Карамышев	Черный кот	Ужасы
Альберт	Карамышев	Компромисс	Роман
Артем	Спас	Борис Годунов	Трагедия
Екатерина	Охримук	Сказка о рыбаке и рыбке	Сказка
Екатерина	Охримук	Приключения Робинзона Крузо	Приключения