

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Базы данных»
Тема: Реализация базы данных в СУБД PostgreSQL

Студент гр. 1304

Стародубов М.В.

Преподаватель

Заславский М.М.

Санкт-Петербург

2023

Цель работы.

Изучить и применить на практике основы работы с языком управления данными в реляционных базах данных *SQL*. Изучить основы работы с СУБД *PostgreSQL*.

Задание.

Вариант — 21.

Необходимо развернуть PostgreSQL локально:

- Написать запросы для создания таблиц из предыдущей лабораторной работы
- Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
- Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
- Исходный код выложить на www.db-fiddle.com для проверки работоспособности
- Исходный код в виде .sql файла запустить в виде PR в репо
- В отчете описать:
 - Цель
 - Текст задания в соответствии с вариантом
 - Скриншоты работы с СУБД PostgreSQL ([psql](#) / [DBeaver](#) / [Datagrip](#), ...)
 - Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
 - Исходный код в приложении
 - Ссылку на исходный код www.db-fiddle.com в приложении
 - Ссылка на PR в приложении
 - Вывод

Выполнение работы.

Работа с СУБД осуществляется с помощью интерактивного терминала *psql*.

Создаем таблицу *Readers*. Содержимое файла *create_table_readers.sql*:

```
create table Readers
(
    phone_number varchar(255) not null primary key,
    email varchar(255) unique,
    date_of_birth date,
    address varchar(255),
    full_name varchar(255)
);
```

Результат выполнения изображен на рисунке 1.



```
lab_2_db=# \i /var/lib/postgres/quiery/create_table_readers.sql
CREATE TABLE
lab_2_db=# \d
               List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | readers | table | postgres
(1 row)

lab_2_db=#
```

Рисунок 1- Создание таблицы *Readers*.

Создаем таблицу *Stores*. Содержимое файла *create_table_stores.sql*:

```
create table Stores
(
    id int not null primary key
);
```

Результат выполнения изображен на рисунке 2.

```

lab_2_db=# \i /var/lib/postgres/quiery/create_table_stores.sql
CREATE TABLE
lab_2_db=# \d
               List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | readers | table | postgres
 public | stores  | table | postgres
(2 rows)

lab_2_db=# \d Stores
               Table "public.stores"
 Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id     | integer |           | not null |
Indexes:
    "stores_pkey" PRIMARY KEY, btree (id)
lab_2_db=# █

```

Рисунок 2- Создание таблицы *Stores*.

Создаем таблицу Departments. Содержимое файла *create_table_departments.sql*:

```

create table Departments
(
    id int not null primary key,
    store_id int references Stores(id),
    name varchar(255),
    capacity int
);

```

Результат выполнения изображен на рисунке 2.

```

lab_2_db=# \i /var/lib/postgres/quiery/create_table_departments.sql
CREATE TABLE
lab_2_db=# \d departments
               Table "public.departments"
 Column | Type           | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id     | integer        |           | not null |
 store_id | integer        |           |          |
 name   | character varying(255) |           |          |
 capacity | integer        |           |          |
Indexes:
    "departments_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "departments_store_id_fkey" FOREIGN KEY (store_id) REFERENCES stores(id)
lab_2_db=# █

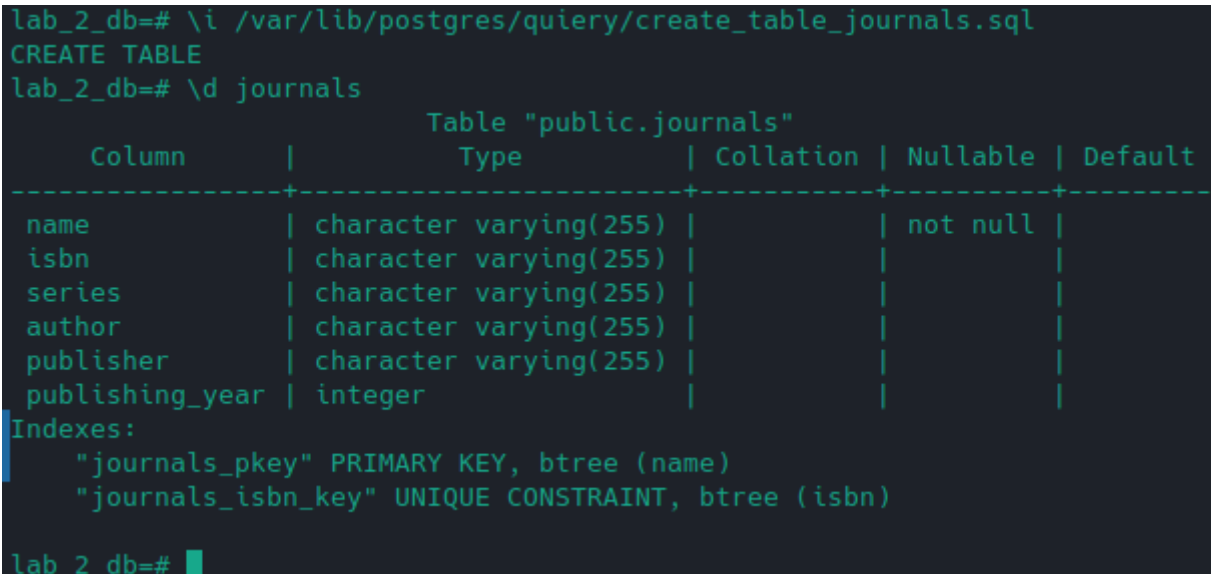
```

Рисунок 3- Создание таблицы *Departments*.

Создаем таблицу *Journals*. Содержимое файла *create_table_journals.sql*:

```
create table Journals
(
    name varchar(255) not null primary key,
    isbn varchar(255) unique,
    series varchar(255),
    author varchar(255),
    publisher varchar(255),
    publishing_year int
);
```

Результат выполнения изображен на рисунке 4.



```
lab_2_db=# \i /var/lib/postgres/quiery/create_table_journals.sql
CREATE TABLE
lab_2_db=# \d journals
```

Column	Type	Collation	Nullable	Default
name	character varying(255)		not null	
isbn	character varying(255)			
series	character varying(255)			
author	character varying(255)			
publisher	character varying(255)			
publishing_year	integer			

```
Indexes:
    "journals_pkey" PRIMARY KEY, btree (name)
    "journals_isbn_key" UNIQUE CONSTRAINT, btree (isbn)
lab_2_db=#
```

Рисунок 4- Создание таблицы *Journals*.

Создаем таблицу *DepartmentStorage*. Содержимое файла *create_table_departmentstorage.sql*:

```
create table DepartmentStorage
(
    department_id int not null references Departments(id),
    journal_name varchar(255) not null references Journals(name),
    amount int,
    primary key(department_id, journal_name)
);
```

Результат выполнения изображен на рисунке 5.

```
lab_2_db=# \i /var/lib/postgres/quiery/create_table_departmentstorage.sql
CREATE TABLE
lab_2_db=# \d departmentstorage
               Table "public.departmentstorage"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
department_id | integer         |           | not null |
journal_name  | character varying(255) |         | not null |
amount        | integer         |           |          |
Indexes:
    "departmentstorage_pkey" PRIMARY KEY, btree (department_id, journal_name)
Foreign-key constraints:
    "departmentstorage_department_id_fkey" FOREIGN KEY (department_id) REFERENCES departments(id)
    "departmentstorage_journal_name_fkey" FOREIGN KEY (journal_name) REFERENCES journals(name)
lab_2_db=#
```

Рисунок 5- Создание таблицы *DepartmentStorage*.

Создаем таблицу *Orders*. Содержимое файла *create_table_orders.sql*:

```
create table Orders
(
    id int not null primary key,
    reader_phone varchar(255) references Readers(phone_number),
    journal_name varchar(255) references Journals(name),
    store_id int references Stores(id),
    sale_date date
);
```

Результат выполнения изображен на рисунке 6.

```
lab_2_db=# \i /var/lib/postgres/quiery/create_table_orders.sql
CREATE TABLE
lab_2_db=# \d orders
               Table "public.orders"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
id            | integer         |           | not null |
reader_phone  | character varying(255) |         |          |
journal_name  | character varying(255) |         |          |
store_id      | integer         |           |          |
sale_date     | date            |           |          |
Indexes:
    "orders_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "orders_journal_name_fkey" FOREIGN KEY (journal_name) REFERENCES journals(name)
    "orders_reader_phone_fkey" FOREIGN KEY (reader_phone) REFERENCES readers(phone_number)
    "orders_store_id_fkey" FOREIGN KEY (store_id) REFERENCES stores(id)
lab_2_db=#
```

Рисунок 6- Создание таблицы *Orders*.

Заполняем таблицу *Readers*. Результат заполнения таблицы изображен на рисунке 7.

```
lab_2_db=# select * from Readers
lab_2_db=# ;
```

phone_number	email	date_of_birth	address	full_name
198-126-9699	csante0@bloglines.com	2005-01-10	082 Melby Point	Clemmy Sante
128-532-3683	cbolin1@plala.or.jp	1998-11-22	924 Melody Way	Cory Bolin
954-276-8852	dbathurst2@e-recht24.de	2002-03-28	37 Evergreen Street	Dennie Bathurst
599-544-0742	gbrainsby3@shareasale.com	1987-07-18	9948 New Castle Hill	Gert Brainsby
329-653-3443	cdullaghan4@squidoo.com	2009-03-10	1946 Ludington Drive	Charisse Dullaghan
945-847-2724	mgrishkov5@discuz.net	1983-08-12	94 Fallview Junction	Moreen Grishkov
807-342-7573	ulewnden6@upenn.edu	1983-04-17	52934 Erie Court	Umberto Lewnden
948-508-8083	efylan7@dedecms.com	2004-04-04	2666 Hoard Trail	Elisha Fylan
650-997-2228	eniess8@hugedomains.com	2001-05-27	058 Hoffman Terrace	Emelen Niess
781-124-6234	hsarson9@stanford.edu	2008-01-09	02 Lukken Court	Haleigh Sarson

```
(10 rows)
lab_2_db=#
```

Рисунок 7- Заполнение таблицы *Readers*.

Заполняем таблицу *Stores*. Результат заполнения таблицы изображен на рисунке 8.

```
lab_2_db=# select * from Stores;
lab_2_db=# ;
```

id
1
2
3
4
5
6
7
8
9
10

```
(10 rows)
lab_2_db=#
```

Рисунок 8- Заполнение
таблицы *Stores*.

Заполняем таблицу *Departments*. Результат заполнения таблицы изображен на рисунке 9.

```
lab_2_db=# select * from departments
lab_2_db=# ;
```

id	store_id	name	capacity
1	10	Comedy	147
2	2	Horror	181
3	9	Comedy	63
4	5	Comedy	124
5	6	Crime	71
6	8	Drama	83
7	7	Drama	116
8	2	Comedy	197
9	9	Drama	191
10	6	Comedy	101
11	10	Crime	63
12	5	Drama	168
13	3	Horror	74
14	9	Comedy	188
15	1	Comedy	111
16	6	Western	108
17	4	Horror	57
18	1	Thriller	75
19	3	Comedy	139
20	7	Horror	118

```
(20 rows)
lab_2_db=#
```

Рисунок 9- Заполнение таблицы
Departments.

Заполняем таблицу *Journals*. Результат заполнения таблицы изображен на рисунке 10.

```
lab_2_db=# select * from journals;
```

name	isbn	series	author	publisher	publishing_year
Hustler White	616320645-2	magnis	Isidro Scrinage	Bayer LLC	1985
Mental	150340734-9	consectetur	Fons Girogetti	Witting Group	2003
Northern Soul	307154553-3	ornare	Ronny Troup	Herzog and Sons	1993
Mr. Vampire (Geung si sin sang)	086960233-0	condimentum	Jackelyn Ovendale	Buckridge, Dach and Upton	1996
Freeway	008738412-4	consequat	Hy Brazil	Bailey, Crist and Maley	2009
8 (8, the Play)	211769445-4	vestibulum	Emlen Harvard	Fay, Hermann and Christiansen	1997
The Boy in the Mirror	178385427-8	dapibus	Charita Laise	Beier, Parker and Friesen	2009
Severed Arm, The	853192858-3	at	Ulysses Coulsen	Langworth Inc	1996
A Walk in the Old City of Warsaw	533848955-4	porttitor	Beth Cornewall	Kiehn Group	1992
Life for Sale (Kotirauha)	362373170-6	orci	Amity Frizell	Nolan, Leuschke and Friesen	1997

```
(10 rows)
lab_2_db=#
```

Рисунок 10- Заполнение таблицы *Journals*.

Заполняем таблицу *DepartmentStorage*. Результат заполнения таблицы изображен на рисунке 11.


```

lab_2_db=# select * from departmentstorage
lab_2_db=# ;
 department_id |          journal_name          | amount
-----+-----+-----
              1 | Severed Arm, The               |      20
              1 | Hustler White                  |        1
              1 | The Boy in the Mirror          |       11
              2 | Freeway                        |       20
              2 | Northern Soul                  |       20
              3 | 8 (8, the Play)                |       12
              3 | Severed Arm, The               |        9
              3 | The Boy in the Mirror          |        6
              3 | Mr. Vampire (Geung si sin sang)|       15
              4 | Severed Arm, The               |        5
              5 | Mental                         |        9
              5 | Northern Soul                  |       13
              6 | A Walk in the Old City of Warsaw |       10
              6 | Hustler White                  |        5
              6 | Mr. Vampire (Geung si sin sang)|       12
              7 | Northern Soul                  |       17
              8 | Life for Sale (Kotirauha)      |       17
              8 | A Walk in the Old City of Warsaw |        6
              8 | The Boy in the Mirror          |       12
              8 | Severed Arm, The               |       12
              9 | Life for Sale (Kotirauha)      |        9
             10 | 8 (8, the Play)                |       15
             10 | The Boy in the Mirror          |       17
             11 | Severed Arm, The               |        5
             12 | A Walk in the Old City of Warsaw |       19
             12 | 8 (8, the Play)                |       16
             13 | The Boy in the Mirror          |        6
             13 | Hustler White                  |       19
             14 | Life for Sale (Kotirauha)      |       20
             14 | Mr. Vampire (Geung si sin sang)|        5
             15 | Mr. Vampire (Geung si sin sang)|        9
             15 | Severed Arm, The               |       15
             16 | The Boy in the Mirror          |        8
             16 | Northern Soul                  |       13
             17 | A Walk in the Old City of Warsaw |       18
             18 | Hustler White                  |       17
             19 | Mr. Vampire (Geung si sin sang)|        5
             19 | Life for Sale (Kotirauha)      |       13
             20 | Mr. Vampire (Geung si sin sang)|       17
             20 | Northern Soul                  |       12
(40 rows)

lab_2_db=#

```

Рисунок 11- Заполнение таблицы *DepartmentStorage*.

Заполняем таблицу *Orders*. Результат заполнения таблицы изображен на рисунке 12.

```
lab_2_db=# select * from orders;
 id | reader_phone |          journal_name          | store_id | sale_date
-----+-----+-----+-----+-----
  1 | 128-532-3683 | 8 (8, the Play)                |        2 | 2023-10-08
  2 | 948-508-8083 | The Boy in the Mirror          |        9 | 2023-09-07
  3 | 945-847-2724 | A Walk in the Old City of Warsaw |        9 | 2023-10-18
  4 | 948-508-8083 | Mental                        |        2 | 2023-10-16
  5 | 128-532-3683 | Hustler White                  |        7 | 2023-10-16
  6 | 781-124-6234 | Severed Arm, The               |       10 | 2023-10-17
  7 | 807-342-7573 | Northern Soul                  |        4 | 2023-09-04
  8 | 807-342-7573 | Freeway                       |        4 | 2023-09-25
  9 | 329-653-3443 | 8 (8, the Play)                |        9 | 2023-09-05
 10 | 945-847-2724 | Northern Soul                  |        7 | 2023-09-18
 11 | 948-508-8083 | Life for Sale (Kotirauha)      |        1 | 2023-09-08
 12 | 781-124-6234 | The Boy in the Mirror          |        4 | 2023-09-13
 13 | 945-847-2724 | Freeway                       |        4 | 2023-09-16
 14 | 945-847-2724 | Freeway                       |        1 | 2023-09-21
 15 | 650-997-2228 | Life for Sale (Kotirauha)      |       10 | 2023-10-21
(15 rows)

lab_2_db=#
```

Рисунок 12- Заполнение таблицы *Orders*.

Произведено выполнение запроса на поиск журналов, купленных покупателем с номером телефона 948-508-8083. Текст запроса:

```
SELECT journal_name FROM Orders WHERE reader_phone = '948-508-8083';
```

Результат выполнения запроса изображен на рисунке 13.

```
lab_2_db=# SELECT journal_name FROM Orders WHERE reader_phone = '948-508-8083';
 journal_name
-----
The Boy in the Mirror
Mental
Life for Sale (Kotirauha)
(3 rows)

lab_2_db=#
```

Рисунок 13- Выполнение запроса на поиск журналов, купленных определенным покупателем.

Произведено выполнение запроса на поиск журнала с *ISBN* 362373170-6. Текст запроса:

```
SELECT name FROM Journals WHERE isbn = '362373170-6';
```

Результат выполнения запроса изображен на рисунке 14.

```
lab_2_db=# SELECT name FROM Journals WHERE isbn = '362373170-6';
          name
-----
Life for Sale (Kotirauha)
(1 row)

lab_2_db=#
```

Рисунок 14- Выполнение запроса на название журнала с заданным *ISBN*.

Произведено выполнение запроса на поиск *ISBN* журнала с названием *Freeway*. Текст запроса:

```
SELECT isbn FROM Journals WHERE name = 'Freeway';
```

Результат выполнения запроса изображен на рисунке 15.

```
lab_2_db=# SELECT isbn FROM Journals WHERE name = 'Freeway';
          isbn
-----
008738412-4
(1 row)

lab_2_db=#
```

Рисунок 15- Результат выполнение запроса на поиск *ISBN* журнала с заданным названием.

Произведено выполнение запроса на поиск даты покупки журнала с названием *Life for Sale (Kotirauha)*. Текст запроса:

```
SELECT sale_date FROM Orders WHERE journal_name = 'Life for Sale (Kotirauha)';
```

Результат выполнения запроса изображен на рисунке 16.

```
lab_2_db=# SELECT sale_date FROM Orders WHERE journal_name = 'Life for Sale (Kotirauha)';
          sale_date
-----
2023-09-08
2023-10-21
(2 rows)

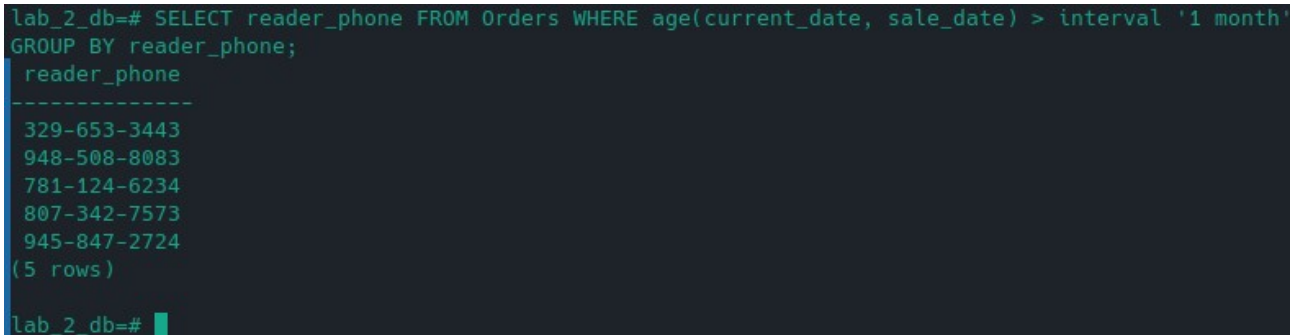
lab_2_db=#
```

Рисунок 16- Результат выполнение запроса на поиск даты покупки журнала с заданным названием.

Произведено выполнение запроса на поиск покупателей, совершивших покупку журнала более месяца тому назад. Текст запроса:

```
SELECT reader_phone FROM Orders WHERE age(current_date, sale_date)
> interval '1 month'
GROUP BY reader_phone;
```

Результат выполнения запроса изображен на рисунке 17.



```
lab_2_db=# SELECT reader_phone FROM Orders WHERE age(current_date, sale_date) > interval '1 month'
GROUP BY reader_phone;
 reader_phone
-----
329-653-3443
948-508-8083
781-124-6234
807-342-7573
945-847-2724
(5 rows)
lab_2_db=#
```

Рисунок 17- Результат выполнения запроса на поиск покупателей, купивших журнал более месяца таму назад.

Произведено выполнение запроса на поиск покупателей самых редких по наличию журналов в магазине с идентификатором 10. Текст запроса:

```
WITH JournalAmountInStore AS
(
    SELECT
        DepartmentStorage.journal_name,
SUM(DepartmentStorage.amount) FROM Departments
    JOIN    DepartmentStorage    ON    Departments.id    =
DepartmentStorage.department_id WHERE Departments.store_id = 10
    GROUP BY DepartmentStorage.journal_name
), RarestJournals AS
(
    SELECT journal_name FROM JournalAmountInStore WHERE sum = (SELECT
MIN(sum) FROM JournalAmountInStore)
)

SELECT Orders.reader_phone FROM Orders
JOIN    RarestJournals    ON    Orders.journal_name    =
RarestJournals.journal_name;
```

Результат выполнения запроса изображен на рисунке 18.

```

lab_2_db=# WITH JournalAmountInStore AS
(
  SELECT DepartmentStorage.journal_name, SUM(DepartmentStorage.amount) FROM Departments
  JOIN DepartmentStorage ON Departments.id = DepartmentStorage.department_id WHERE Departments.store_id = 10
  GROUP BY DepartmentStorage.journal_name
), RarestJournals AS
(
  SELECT journal_name FROM JournalAmountInStore WHERE sum = (SELECT MIN(sum) FROM JournalAmountInStore)
)
SELECT Orders.reader_phone FROM Orders
JOIN RarestJournals ON Orders.journal_name = RarestJournals.journal_name;
reader_phone
-----
128-532-3683
(1 row)

lab_2_db=#

```

Рисунок 18- Результат выполнения запроса на поиск покупателя самых редких журналов по наличию в магазине.

Произведено выполнение запроса на определение числа покупателей, пользующимися магазином с идентификатором 4. Текст запроса:

```
SELECT COUNT(DISTINCT reader_phone) FROM Orders WHERE store_id = 4;
```

Результат выполнения запроса изображен на рисунке 19.

```

lab_2_db=# SELECT COUNT(DISTINCT reader_phone) FROM Orders WHERE store_id = 4;
count
-----
3
(1 row)

lab_2_db=#

```

Рисунок 19- Результат выполнения запроса на определение количества покупателей, пользующихся определенным магазином.

Произведено выполнение запроса на определение количества покупателей младше 20 лет. Текст запроса:

```
SELECT COUNT(*) FROM Readers WHERE age(date_of_birth) < interval
'20 years';
```

Результат выполнения запроса изображен на рисунке 20.

```
lab_2_db=# SELECT COUNT(*) FROM Readers WHERE age(date_of_birth) < interval '20 years';
 count
-----
      4
(1 row)

lab_2_db=#
```

Рисунок 20- Результат выполнения запроса на определение количества покупателей младше 20 лет.

Выводы.

В ходе выполнения работы на практике была реализована спроектированная база данных с использованием СУБД *PostgreSQL*. Были реализованы заданные запросы с использованием языка *SQL*.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: *lab2.sql*

```
create table Readers
(
    phone_number varchar(255) not null primary key,
    email varchar(255) unique,
    date_of_birth date,
    address varchar(255),
    full_name varchar(255)
);

create table Stores
(
    id int not null primary key
);

create table Departments
(
    id int not null primary key,
    store_id int references Stores(id),
    name varchar(255),
    capacity int
);

create table Journals
(
    name varchar(255) not null primary key,
    isbn varchar(255) unique,
    series varchar(255),
    author varchar(255),
    publisher varchar(255),
    publishing_year int
);

create table DepartmentStorage
(
    department_id int not null references Departments(id),
    journal_name varchar(255) not null references Journals(name),
    amount int,
    primary key(department_id, journal_name)
);

create table Orders
(
    id int not null primary key,
    reader_phone varchar(255) references Readers(phone_number),
    journal_name varchar(255) references Journals(name),
    store_id int references Stores(id),
    sale_date date
);
```

```

insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('198-126-9699', 'csante0@bloglines.com', '2005-01-10',
'082 Melby Point', 'Clemmy Sante');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('128-532-3683', 'cbolin1@plala.or.jp', '1998-11-22',
'924 Melody Way', 'Cory Bolin');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('954-276-8852', 'dbathurst2@e-recht24.de', '2002-03-
28', '37 Evergreen Street', 'Dennie Bathurst');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('599-544-0742', 'gbrainsby3@shareasale.com', '1987-07-
18', '9948 New Castle Hill', 'Gert Brainsby');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('329-653-3443', 'cdullaghan4@squidoo.com', '2009-03-
10', '1946 Ludington Drive', 'Charisse Dullaghan');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('945-847-2724', 'mgrishkov5@discuz.net', '1983-08-12',
'94 Fallview Junction', 'Moreen Grishkov');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('807-342-7573', 'ulewnden6@upenn.edu', '1983-04-17',
'52934 Erie Court', 'Umberto Lewnden');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('948-508-8083', 'efylan7@dedecms.com', '2004-04-04',
'2666 Hoard Trail', 'Elisha Fylan');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('650-997-2228', 'eniess8@hugedomains.com', '2001-05-
27', '058 Hoffman Terrace', 'Emelen Niess');
insert into Readers (phone_number, email, date_of_birth, address,
full_name) values ('781-124-6234', 'hsarson9@stanford.edu', '2008-01-09',
'02 Lukken Court', 'Haleigh Sarson');

insert into Stores (id) values (1);
insert into Stores (id) values (2);
insert into Stores (id) values (3);
insert into Stores (id) values (4);
insert into Stores (id) values (5);
insert into Stores (id) values (6);
insert into Stores (id) values (7);
insert into Stores (id) values (8);
insert into Stores (id) values (9);
insert into Stores (id) values (10);

insert into Departments (id, store_id, name, capacity) values (1,
10, 'Comedy', 147);
insert into Departments (id, store_id, name, capacity) values (2,
2, 'Horror', 181);
insert into Departments (id, store_id, name, capacity) values (3,
9, 'Comedy', 63);
insert into Departments (id, store_id, name, capacity) values (4,
5, 'Comedy', 124);
insert into Departments (id, store_id, name, capacity) values (5,
6, 'Crime', 71);
insert into Departments (id, store_id, name, capacity) values (6,
8, 'Drama', 83);
insert into Departments (id, store_id, name, capacity) values (7,
7, 'Drama', 116);
insert into Departments (id, store_id, name, capacity) values (8,
2, 'Comedy', 197);

```



```

insert into Departments (id, store_id, name, capacity) values (9,
9, 'Drama', 191);
insert into Departments (id, store_id, name, capacity) values (10,
6, 'Comedy', 101);
insert into Departments (id, store_id, name, capacity) values (11,
10, 'Crime', 63);
insert into Departments (id, store_id, name, capacity) values (12,
5, 'Drama', 168);
insert into Departments (id, store_id, name, capacity) values (13,
3, 'Horror', 74);
insert into Departments (id, store_id, name, capacity) values (14,
9, 'Comedy', 188);
insert into Departments (id, store_id, name, capacity) values (15,
1, 'Comedy', 111);
insert into Departments (id, store_id, name, capacity) values (16,
6, 'Western', 108);
insert into Departments (id, store_id, name, capacity) values (17,
4, 'Horror', 57);
insert into Departments (id, store_id, name, capacity) values (18,
1, 'Thriller', 75);
insert into Departments (id, store_id, name, capacity) values (19,
3, 'Comedy', 139);
insert into Departments (id, store_id, name, capacity) values (20,
7, 'Horror', 118);

insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Hustler White', '616320645-2', 'magnis',
'Isidro Scrinage', 'Bayer LLC', 1985);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Mental', '150340734-9', 'consectetuer', 'Fons
Girogetti', 'Witting Group', 2003);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Northern Soul', '307154553-3', 'ornare', 'Ronny
Troup', 'Herzog and Sons', 1993);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Mr. Vampire (Geung si sin sang)', '086960233-
0', 'condimentum', 'Jackelyn Ovendale', 'Buckridge, Dach and Upton',
1996);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Freeway', '008738412-4', 'consequat', 'Hy
Brazil', 'Bailey, Crist and Haley', 2009);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('8 (8, the Play)', '211769445-4', 'vestibulum',
'Emlen Harvard', 'Fay, Hermann and Christiansen', 1997);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('The Boy in the Mirror', '178385427-8',
'dapibus', 'Charita Laise', 'Beier, Parker and Friesen', 2009);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Severed Arm, The', '853192858-3', 'at',
'Ulysses Coulsen', 'Langworth Inc', 1996);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('A Walk in the Old City of Warsaw', '533848955-
4', 'porttitor', 'Beth Cornewall', 'Kiehn Group', 1992);
insert into Journals (name, isbn, series, author, publisher,
publishing_year) values ('Life for Sale (Kotirauha)', '362373170-6',
'orci', 'Amity Frizell', 'Nolan, Leuschke and Friesen', 1997);

```

```

insert into DepartmentStorage (department_id, journal_name, amount)
values (1, 'Severed Arm, The', 20);
insert into DepartmentStorage (department_id, journal_name, amount)
values (1, 'Hustler White', 1);
insert into DepartmentStorage (department_id, journal_name, amount)
values (1, 'The Boy in the Mirror', 11);
insert into DepartmentStorage (department_id, journal_name, amount)
values (2, 'Freeway', 20);
insert into DepartmentStorage (department_id, journal_name, amount)
values (2, 'Northern Soul', 20);
insert into DepartmentStorage (department_id, journal_name, amount)
values (3, '8 (8, the Play)', 12);
insert into DepartmentStorage (department_id, journal_name, amount)
values (3, 'Severed Arm, The', 9);
insert into DepartmentStorage (department_id, journal_name, amount)
values (3, 'The Boy in the Mirror', 6);
insert into DepartmentStorage (department_id, journal_name, amount)
values (3, 'Mr. Vampire (Geung si sin sang)', 15);
insert into DepartmentStorage (department_id, journal_name, amount)
values (4, 'Severed Arm, The', 5);
insert into DepartmentStorage (department_id, journal_name, amount)
values (5, 'Mental', 9);
insert into DepartmentStorage (department_id, journal_name, amount)
values (5, 'Northern Soul', 13);
insert into DepartmentStorage (department_id, journal_name, amount)
values (6, 'A Walk in the Old City of Warsaw', 10);
insert into DepartmentStorage (department_id, journal_name, amount)
values (6, 'Hustler White', 5);
insert into DepartmentStorage (department_id, journal_name, amount)
values (6, 'Mr. Vampire (Geung si sin sang)', 12);
insert into DepartmentStorage (department_id, journal_name, amount)
values (7, 'Northern Soul', 17);
insert into DepartmentStorage (department_id, journal_name, amount)
values (8, 'Life for Sale (Kotirauha)', 17);
insert into DepartmentStorage (department_id, journal_name, amount)
values (8, 'A Walk in the Old City of Warsaw', 6);
insert into DepartmentStorage (department_id, journal_name, amount)
values (8, 'The Boy in the Mirror', 12);
insert into DepartmentStorage (department_id, journal_name, amount)
values (8, 'Severed Arm, The', 12);
insert into DepartmentStorage (department_id, journal_name, amount)
values (9, 'Life for Sale (Kotirauha)', 9);
insert into DepartmentStorage (department_id, journal_name, amount)
values (10, '8 (8, the Play)', 15);
insert into DepartmentStorage (department_id, journal_name, amount)
values (10, 'The Boy in the Mirror', 17);
insert into DepartmentStorage (department_id, journal_name, amount)
values (11, 'Severed Arm, The', 5);
insert into DepartmentStorage (department_id, journal_name, amount)
values (12, 'A Walk in the Old City of Warsaw', 19);
insert into DepartmentStorage (department_id, journal_name, amount)
values (12, '8 (8, the Play)', 16);
insert into DepartmentStorage (department_id, journal_name, amount)
values (13, 'The Boy in the Mirror', 6);
insert into DepartmentStorage (department_id, journal_name, amount)
values (13, 'Hustler White', 19);
insert into DepartmentStorage (department_id, journal_name, amount)
values (14, 'Life for Sale (Kotirauha)', 20);

```

```

insert into DepartmentStorage (department_id, journal_name, amount)
values (14, 'Mr. Vampire (Geung si sin sang)', 5);
insert into DepartmentStorage (department_id, journal_name, amount)
values (15, 'Mr. Vampire (Geung si sin sang)', 9);
insert into DepartmentStorage (department_id, journal_name, amount)
values (15, 'Severed Arm, The', 15);
insert into DepartmentStorage (department_id, journal_name, amount)
values (16, 'The Boy in the Mirror', 8);
insert into DepartmentStorage (department_id, journal_name, amount)
values (16, 'Northern Soul', 13);
insert into DepartmentStorage (department_id, journal_name, amount)
values (17, 'A Walk in the Old City of Warsaw', 18);
insert into DepartmentStorage (department_id, journal_name, amount)
values (18, 'Hustler White', 17);
insert into DepartmentStorage (department_id, journal_name, amount)
values (19, 'Mr. Vampire (Geung si sin sang)', 5);
insert into DepartmentStorage (department_id, journal_name, amount)
values (19, 'Life for Sale (Kotirauha)', 13);
insert into DepartmentStorage (department_id, journal_name, amount)
values (20, 'Mr. Vampire (Geung si sin sang)', 17);
insert into DepartmentStorage (department_id, journal_name, amount)
values (20, 'Northern Soul', 12);

insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (1, '128-532-3683', '8 (8, the Play)', 2, '2023-10-08
11:52:04');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (2, '948-508-8083', 'The Boy in the Mirror', 9, '2023-
09-07 14:40:05');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (3, '945-847-2724', 'A Walk in the Old City of Warsaw',
9, '2023-10-18 13:06:30');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (4, '948-508-8083', 'Mental', 2, '2023-10-16
06:35:25');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (5, '128-532-3683', 'Hustler White', 7, '2023-10-16
06:08:00');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (6, '781-124-6234', 'Severed Arm, The', 10, '2023-10-17
17:00:33');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (7, '807-342-7573', 'Northern Soul', 4, '2023-09-04
03:21:59');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (8, '807-342-7573', 'Freeway', 4, '2023-09-25
00:34:09');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (9, '329-653-3443', '8 (8, the Play)', 9, '2023-09-05
19:20:28');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (10, '945-847-2724', 'Northern Soul', 7, '2023-09-18
13:21:55');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (11, '948-508-8083', 'Life for Sale (Kotirauha)', 1,
'2023-09-08 16:05:56');

```

```

insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (12, '781-124-6234', 'The Boy in the Mirror', 4, '2023-
09-13 13:43:45');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (13, '945-847-2724', 'Freeway', 4, '2023-09-16
17:16:32');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (14, '945-847-2724', 'Freeway', 1, '2023-09-21
05:26:01');
insert into Orders (id, reader_phone, journal_name, store_id,
sale_date) values (15, '650-997-2228', 'Life for Sale (Kotirauha)', 10,
'2023-10-21 06:02:25');

-- Какие журналы были куплены определенным покупателем?
SELECT journal_name FROM Orders WHERE reader_phone = '948-508-
8083';

-- Как называется журнал с заданным ISBN?
SELECT name FROM Journals WHERE isbn = '362373170-6';

-- Какой ISBN у журнала с заданным названием?
SELECT isbn FROM Journals WHERE name = 'Freeway';

-- Когда журнал был куплен?
SELECT sale_date FROM Orders WHERE journal_name = 'Life for Sale
(Kotirauha)';

-- Кто из покупателей купил журнал более месяца тому назад?
SELECT reader_phone FROM Orders WHERE age(current_date, sale_date)
> interval '1 month'
GROUP BY reader_phone;

-- Найти покупателя самых редких журналов (по наличию в магазине)?
WITH JournalAmountInStore AS
(
    SELECT
        DepartmentStorage.journal_name,
SUM(DepartmentStorage.amount) FROM Departments
    JOIN DepartmentStorage ON Departments.id =
DepartmentStorage.department_id WHERE Departments.store_id = 10
    GROUP BY DepartmentStorage.journal_name
), RarestJournals AS
(
    SELECT journal_name FROM JournalAmountInStore WHERE sum = (SELECT
MIN(sum) FROM JournalAmountInStore)
)

SELECT Orders.reader_phone FROM Orders
JOIN RarestJournals ON Orders.journal_name =
RarestJournals.journal_name;

-- Какое число покупателей пользуется определенным магазином?
SELECT COUNT(DISTINCT reader_phone) FROM Orders WHERE store_id = 4;

-- Сколько покупателей младше 20 лет?
SELECT COUNT(*) FROM Readers WHERE age(date_of_birth) < interval
'20 years';

```

ПРИЛОЖЕНИЕ Б

ССЫЛКА НА ИСХОДНЫЙ КОД WWW.DB-FIDDLE.COM

<https://www.db-fiddle.com/f/ejPKF6ba6Xv6mUuJiiZ6mi/0>

ПРИЛОЖЕНИЕ В

ССЫЛКА НА *PR*

<https://github.com/moevm/sql-2023-1304/pull/43>