

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №4 «Запросы на выборку и модификацию данных. Представления. Работа с индексами»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Морозов Артём

Факультет: ИКТ

Группа: K3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

| | |
|---|----|
| Цель работы | 6 |
| Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов. | 6 |
| Практическое задание | 6 |
| 1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3). | 6 |
| 3. Изучить графическое представление запросов и просмотреть историю запросов. | 6 |
| Выполнение практического задания | 7 |
| Запросы: | 7 |
| Задание 3. Создать представление: | 14 |
| 1) Сведения о должниках | 14 |

The screenshot displays a SQL IDE interface. At the top, there are tabs for multiple SQL files: Library.sql, 1.sql, 2.sql, 3.sql, 4.sql, 5.sql, and 6.sql. The active file is 6.sql. Below the tabs is a toolbar with icons for running queries, saving, and other functions. The main editor area contains the following SQL query:

```
1 select
2     reader.id_reader,
3     reader.name,
4     reader.surname
5 from
6     reader,
7     book_circulation
8 where
9     reader.id_reader = book_circulation.id_reader and
10    book_circulation.fine = True;
```

Below the editor, the 'Output' tab is selected, showing the results of the query. The results are displayed in a table with 4 columns: id_reader, name, surname, and an unlabeled column. The table contains 11 rows of data:

| | id_reader | name | surname | |
|----|-----------|---------------|----------|--|
| 1 | 71 | Margalo | Delph | |
| 2 | 530 | Silvanus | Ticksall | |
| 3 | 892 | Robert | Seeds | |
| 4 | 300 | Renaldo | Kasting | |
| 5 | 491 | Jeremiah | Lapree | |
| 6 | 888 | Christophorus | Blest | |
| 7 | 662 | Cora | Henfre | |
| 8 | 294 | Verine | Noell | |
| 9 | 478 | Jenna | Hathwood | |
| 10 | 537 | Lucien | Geeve | |
| 11 | 167 | Ernst | Semmens | |

Запрос

История запросов

```

1  explain analyze
2  select
3      reader.id_reader,
4      reader.name,
5      reader.surname,
6      reader.count_of_books
7  from
8      reader,
9      book_circulation
10 where
11     reader.id_reader = book_circulation.id_reader and
12     book_circulation.fine = True and
13     reader.count_of_books > 9;

```

Data Output

Сообщения

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

| | QUERY PLAN | |
|----|--|---|
| | text | 🔒 |
| 1 | Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=1.839..2.179 rows=56 loops=1) | |
| 2 | Hash Cond: (book_circulation.id_reader = reader.id_reader) | |
| 3 | -> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.046..0.334 rows=489 loop... | |
| 4 | Filter: fine | |
| 5 | Rows Removed by Filter: 511 | |
| 6 | -> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=1.782..1.783 rows=103 loops=1) | |
| 7 | Buckets: 1024 Batches: 1 Memory Usage: 14kB | |
| 8 | -> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.006..1.747 rows=103 loops=1) | |
| 9 | Filter: (count_of_books > 9) | |
| 10 | Rows Removed by Filter: 897 | |
| 11 | Planning Time: 5.381 ms | |
| 12 | Execution Time: 2.208 ms | |

.....17

Простые индексы17

```

1 drop index idx_reader_circulation;
2
3 CREATE INDEX idx_reader_circulation ON book_circulation (id_reader);
4 CREATE INDEX idx_reader_circulation1 ON book_circulation (fine);
5
6 explain analyze
7 select
8     reader.id_reader,
9     reader.name,
10    reader.surname,
11    reader.count_of_books
12 from
13     reader,
14     book_circulation
15 where
16     reader.id_reader = book_circulation.id_reader and
17     book_circulation.fine = True and
18     reader.count_of_books > 9;

```

Data Output Сообщения Notifications



QUERY PLAN

text



| | |
|----|--|
| 1 | Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=0.125..0.228 rows=56 loops=1) |
| 2 | Hash Cond: (book_circulation.id_reader = reader.id_reader) |
| 3 | -> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.010..0.086 rows=489 loop... |
| 4 | Filter: fine |
| 5 | Rows Removed by Filter: 511 |
| 6 | -> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=0.108..0.108 rows=103 loops=1) |
| 7 | Buckets: 1024 Batches: 1 Memory Usage: 14kB |
| 8 | -> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.005..0.095 rows=103 loops=1) |
| 9 | Filter: (count_of_books > 9) |
| 10 | Rows Removed by Filter: 897 |
| 11 | Planning Time: 1.574 ms |
| 12 | Execution Time: 0.249 ms |

| | |
|------------------------|----|
| | 18 |
| Составные индексы..... | 18 |

Цель работы

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Вариант 3. БД «Библиотека»

Описание предметной области: Каждая книга может храниться в нескольких экземплярах. Для каждого экземпляра известно место его хранения (комната, стеллаж, полка). Читателю не может быть выдано более 3-х книг одновременно. Книги выдаются читателям на срок не более 10 дней. В случае просрочки читателю назначается денежный штраф.

Все издания, поступающие в библиотеку ставятся на библиотечный учет, согласно существующим требованиям. Необходимо хранить информацию, кто из сотрудников поставил экземпляр на учет.

Книги принимаются к учету на основании первичных учетных документов (накладной от поставщика, акта о приеме документов). Если документы поступают на безвозмездной основе (в результате передачи обязательных экземпляров и т. п.), оформляется акт о приеме документов. Документы, поступающие от читателей взамен утерянных и признанные равноценными утраченным, оформляются актом о приеме документов взамен утерянных.

Выбытие документов из библиотеки отражается в учете в связи с физической утратой либо утратой потребительских свойств (по причине ветхости, дефектности, устарелости по содержанию, непрофильности). Непрофильность издания определяется на основании профиля комплектования фонда или иного документа, утверждаемого руководителем библиотеки. При выбытии документов из библиотеки оформляется акт о списании исключенных объектов библиотечного фонда (далее – акт о списании), к которому прилагается список исключаемых объектов библиотечного фонда. В акте о списании отражаются сведения о количестве и общей стоимости исключаемых документов, а также причина списания и направление изданий после выбытия с учета. В прилагаемом к акту списке указываются:

- регистрационный номер и шифр хранения издания;
- краткое библиографическое описание;
- стоимость, зафиксированная в регистре индивидуального учета издания;
- коэффициент переоценки, стоимость после переоценки;
- общая стоимость исключаемых документов.

БД должна содержать следующий минимальный набор сведений: · Автор (фамилия и имя (инициалы) или псевдоним автора издания). · Название (заглавие) издания. · Номер тома (части, книги, выпуска). · Составитель (фамилия и имена (инициалы) каждого из составителей издания). · Язык, с которого выполнен перевод издания. · Вид издания (сборник, справочник, монография ...). · Область знания. · Переводчик (фамилия и инициалы переводчика). · Место издания (город). · Издательство (название издательства). · Год выпуска издания. · Библиотечный шифр (например, ББК 32.973). · Номер (инвентарный номер) экземпляра. · Номер комнаты (помещения для хранения экземпляров). · Номер стеллажа в комнате. · Номер полки на стеллаже. · Цена конкретного экземпляра. · Дата изъятия

экземпляра с установленного места. · Номер читательского билета (формуляра). · Фамилия читателя. · Имя читателя. · Отчество читателя. · Паспортные данные. Адрес читателя (фактический). Телефон читателя. Электронная почта читателя.

Дополнить исходные данные информацией о читательском абонементе (выдаче книг).

Задание 1.1 (ЛР 1 БД). Выполните инфологическое моделирование базы данных системы. (Ограничения задать самостоятельно.)

Задание 1.2. Создайте логическую модель БД, используя ИЛМ (задание 1.1). Используйте необходимые средства поддержки целостности данных в СУБД.

Задание 2. Создайте запросы:

- Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.
- Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
- Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
- Вывести список книг, которые находятся в библиотеке в единственном экземпляре.
- Подсчитать количество читателей, которые не обращались в библиотеку в течение года.
- Подсчитать количество читателей библиотеки по уровню образования.
- Вывести список книг по программированию на C#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

Задание 3. Создать представления для администрации библиотеки, содержащие:

- сведения о должниках;
- сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

Выполнение практического задания

Запросы:

1. Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.

```
select
    reader.id_reader,
    reader.name,
    reader.surname
from
    reader,
    book_circulation,
    copy_of_book,
    publication,
    book
where
    reader.id_reader = book_circulation.id_reader and
    book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
    copy_of_book.id_publication = publication.library_code and
    publication.id_book = book.id_book and
    book.language = 'English' and
    publication.year_of_release > 2000;
```

Library.sql 1.sql x

Tx: Auto v ✓ Playground v

```

1 ✓ select
2     reader.id_reader,
3     reader.name,
4     reader.surname
5 from
6     reader,
7     book_circulation,
8     copy_of_book,
9     publication,
10    book
11 where
12     reader.id_reader = book_circulation.id_reader and
13     book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
14     copy_of_book.id_publication = publication.library_code and
15     publication.id_book = book.id_book and
16     book.language = 'English' and
17     publication.year_of_release > 2000;

```

Output Library.public.reader Tx: Auto v DDL

9 rows v

| | id_reader | name | surname |
|---|-----------|---------------|---------------------|
| 1 | 473 | Krystalle | Coleridge |
| 2 | 888 | Christophorus | Blest |
| 3 | 230 | Osbourne | Gratrex |
| 4 | 726 | Eryn | Peterson |
| 5 | 766 | Ches | Frackiewicz |
| 6 | 67 | Hyacinth | Grote |
| 7 | 615 | Cord | Butlin |
| 8 | 348 | Artair | Mac Giolla Pheadair |
| 9 | 741 | Jaclyn | Goulbourne |

2. Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.

```

select
    reader.id_reader,
    reader.name,
    reader.surname,
    reader.count_of_books
from
    reader,
    book_circulation
where
    reader.id_reader = book_circulation.id_reader and
    book_circulation.fine = True and
    reader.count_of_books > 9;
/* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```


Library.sql 1.sql 2.sql x 3.sql

Tx: Auto ✓ Playground

```

1 ✓ select
2     reader.id_reader,
3     reader.name,
4     reader.surname,
5     reader.count_of_books
6 from
7     reader,
8     book_circulation
9 where
10    reader.id_reader = book_circulation.id_reader and
11    book_circulation.fine = True and
12    reader.count_of_books > 9;
13 /* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

Output Library.public.reader x Tx ✓ DDL

56 rows

| | id_reader | name | surname | count_of_books |
|----|-----------|----------|-----------|----------------|
| 1 | 662 | Cora | Henfre | 10 |
| 2 | 499 | Bette | Kilmister | 10 |
| 3 | 266 | Claire | Robertson | 10 |
| 4 | 437 | Jarvis | Huzzey | 10 |
| 5 | 706 | Edgar | Leving | 10 |
| 6 | 353 | Noble | Terrazzo | 10 |
| 7 | 596 | Waite | Marcum | 10 |
| 8 | 344 | Robbie | Badgers | 10 |
| 9 | 121 | Beltran | Stockings | 10 |
| 10 | 230 | Osbourne | Gratrex | 10 |
| 11 | 179 | Lizzy | McTear | 10 |
| 12 | 596 | Waite | Marcum | 10 |
| 13 | 344 | Robbie | Badgers | 10 |
| 14 | 451 | Bengt | Hanbury | 10 |
| 15 | 94 | Raffarty | Sandbrook | 10 |

3. Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.

```

select
    reader.id_reader,
    reader.name,
    reader.surname,
    reader.count_of_books
from
    reader,
    book_circulation
where
    reader.id_reader = book_circulation.id_reader and
    book_circulation.fine = True and
    reader.count_of_books > 9;
/* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

Library.sql 1.sql 2.sql x 3.sql

Tx: Auto ✓ Playground

```

1 ✓ select
2     reader.id_reader,
3     reader.name,
4     reader.surname,
5     reader.count_of_books
6 from
7     reader,
8     book_circulation
9 where
10    reader.id_reader = book_circulation.id_reader and
11    book_circulation.fine = True and
12    reader.count_of_books > 9;
13 /* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

Output Library.public.reader x Tx ✓ DDL

56 rows

| | id_reader | name | surname | count_of_books |
|----|-----------|----------|-----------|----------------|
| 1 | 662 | Cora | Henfre | 10 |
| 2 | 499 | Bette | Kilmister | 10 |
| 3 | 266 | Claire | Robertson | 10 |
| 4 | 437 | Jarvis | Huzzey | 10 |
| 5 | 706 | Edgar | Leving | 10 |
| 6 | 353 | Noble | Terrazzo | 10 |
| 7 | 596 | Waite | Marcum | 10 |
| 8 | 344 | Robbie | Badgers | 10 |
| 9 | 121 | Beltran | Stockings | 10 |
| 10 | 230 | Osbourne | Gratrex | 10 |
| 11 | 179 | Lizzy | McTear | 10 |
| 12 | 596 | Waite | Marcum | 10 |
| 13 | 344 | Robbie | Badgers | 10 |
| 14 | 451 | Bengt | Hanbury | 10 |
| 15 | 94 | Raffarty | Sandbrook | 10 |

4. Вывести список книг, которые находятся в библиотеке в единственном экземпляре.

```

select
    distinct copy_of_book.id_publication
from
    copy_of_book
group by
    copy_of_book.id_publication
having count(id copy of book) = 1;

```

Library.sql 1.sql 2.sql 3.sql 4.sql x

Tx: Auto Playground

```

1 select
2   reader.id_reader,
3   reader.name,
4   reader.surname
5
6 from
7   book_circulation,
8   reader
9 where
10  reader.id_reader = book_circulation.id_reader and
    book_circulation.issue_date < date('2022-12-22');

```

Output Library.public.reader x Tx ✓

| | id_reader | name | surname |
|----|-----------|---------|----------------|
| 1 | 8 | Carolee | Cambridge |
| 2 | 8 | Carolee | Cambridge |
| 3 | 34 | Ellene | Mignot |
| 4 | 35 | Rena | Pochet |
| 5 | 64 | Nanete | Buckingham |
| 6 | 79 | Paige | Rheubottom |
| 7 | 111 | Dinny | Geare |
| 8 | 184 | Ailis | Jefferies |
| 9 | 188 | Allyce | Thowless |
| 10 | 243 | Josias | Klimkin |
| 11 | 290 | Meridel | Reiner |
| 12 | 301 | Terri | Von Der Empten |
| 13 | 325 | Redford | Fernez |
| 14 | 373 | Piper | Bus |
| 15 | 376 | Anette | Golley |

5. Подсчитать количество читателей, которые не обращались в библиотеку в течение года.

```

select
  DISTINCT reader.id_reader,
  reader.name,
  reader.surname
from
  book_circulation,
  reader
where
  reader.id_reader = book_circulation.id_reader and
  book_circulation.issue_date < date('2022-12-22');

```

The screenshot shows a SQL IDE with a query editor and an output window. The query editor contains the following SQL code:

```

1 select
2   copy_of_book.*
3 from
4   book_circulation,
5   copy_of_book,
6   publication,
7   book
8 where
9   book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
10  copy_of_book.id_publication = publication.library_code and
11  publication.id_book = book.id_book and
12  copy_of_book.availability = FALSE and
13  book_circulation.return_date_plan <= date('2023-12-25') and
14  book_circulation.return_date_plan >= date('2023-12-23') and
15  book.area_of_knowledge = 'Computer Science';

```

The output window shows the results of the query. The columns are: id_copy_of_book, availability, id_publication, and condition. The output is currently empty, showing 0 rows.

6. Вывести список книг по программированию на C#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

```

select
  copy_of_book.*
from
  book_circulation,
  copy_of_book,
  publication,
  book
where
  book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
  copy_of_book.id_publication = publication.library_code and
  publication.id_book = book.id_book and
  copy_of_book.availability = FALSE and
  book_circulation.return_date_plan <= date('2023-12-25') and
  book_circulation.return_date_plan >= date('2023-12-23') and
  book.area_of_knowledge = 'Computer Science';

```

Library.sql 1.sql 2.sql 3.sql 4.sql 5.sql x

Tx: Auto v ✓ Playground v

```
1 select
2   copy_of_book.*
3 from
4   book_circulation,
5   copy_of_book,
6   publication,
7   book
8 where
9   book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
10  copy_of_book.id_publication = publication.library_code and
11  publication.id_book = book.id_book and
12  copy_of_book.availability = FALSE and
13  book_circulation.return_date_plan <= date('2023-12-25') and
14  book_circulation.return_date_plan >= date('2023-12-23') and
15  book.area_of_knowledge = 'Computer Science';
```

Output Library.public.copy_of_book x Tx: ✓

0 rows v Tx: Auto v DDL

| id_copy_of_book | availability | id_publication | condition |
|-----------------|--------------|----------------|-----------|
|-----------------|--------------|----------------|-----------|

Задание 3. Создать представление:

1) Сведения о должниках

```
select
    reader.id_reader,
    reader.name,
    reader.surname
from
    reader,
    book_circulation
where
    reader.id_reader = book_circulation.id_reader and
    book_circulation.fine = True;
```

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

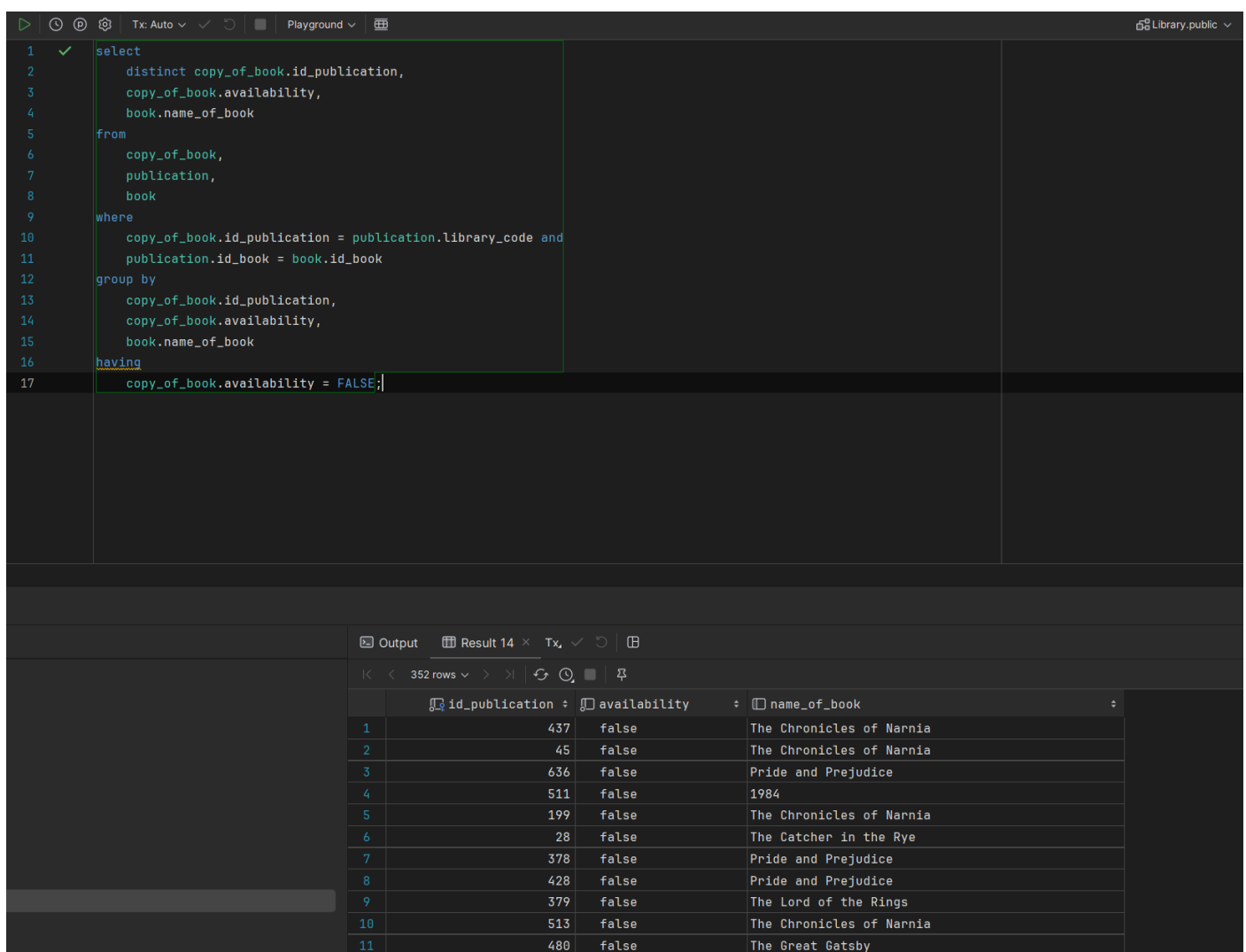
```
1 select
2     reader.id_reader,
3     reader.name,
4     reader.surname
5 from
6     reader,
7     book_circulation
8 where
9     reader.id_reader = book_circulation.id_reader and
10    book_circulation.fine = True;
```

The results pane displays the output of the query, showing 11 rows of data. The columns are labeled `id_reader`, `name`, and `surname`.

| | id_reader | name | surname |
|----|-----------|---------------|----------|
| 1 | 71 | Margalo | Delph |
| 2 | 530 | Silvanus | Ticksall |
| 3 | 892 | Robert | Seeds |
| 4 | 300 | Renaldo | Kasting |
| 5 | 491 | Jeremiah | Lapree |
| 6 | 888 | Christophorus | Blest |
| 7 | 662 | Cora | Henfre |
| 8 | 294 | Verine | Noell |
| 9 | 478 | Jenna | Hathwood |
| 10 | 537 | Lucien | Geeve |
| 11 | 167 | Ernst | Semmens |

2) Найти самую популярную зону звонков за истекший год.

```
• select
  distinct copy_of_book.id_publication,
  copy_of_book.availability,
  book.name_of_book
from
  copy_of_book,
  publication,
  book
where
  copy_of_book.id_publication = publication.library_code and
  publication.id_book = book.id_book
group by
  copy_of_book.id_publication,
  copy_of_book.availability,
  book.name_of_book
having
  copy_of_book.availability = FALSE;
```



The screenshot shows a SQL playground interface. The top section contains a SQL query that selects distinct publication IDs, availability status, and book names from the copy_of_book, publication, and book tables, filtered by availability = FALSE. The bottom section displays the results of the query as a table with 14 rows. The table has three columns: id_publication, availability, and name_of_book. The results show various book titles and their availability status.

| | id_publication | availability | name_of_book |
|----|----------------|--------------|--------------------------|
| 1 | 437 | false | The Chronicles of Narnia |
| 2 | 45 | false | The Chronicles of Narnia |
| 3 | 636 | false | Pride and Prejudice |
| 4 | 511 | false | 1984 |
| 5 | 199 | false | The Chronicles of Narnia |
| 6 | 28 | false | The Catcher in the Rye |
| 7 | 378 | false | Pride and Prejudice |
| 8 | 428 | false | Pride and Prejudice |
| 9 | 379 | false | The Lord of the Rings |
| 10 | 513 | false | The Chronicles of Narnia |
| 11 | 480 | false | The Great Gatsby |

Создание INSERT, UPDATE и DELETE запросов

UPDATE

```

update
  copy_of_book
set
  condition = 'Fair'
where
  copy_of_book.id_copy_of_book in
  (
    select
      copy_of_book.id_copy_of_book
    from
      copy_of_book, act_of_registration
    where
      copy_of_book.id_copy_of_book = act_of_registration.id_copy_of_book and
      act_of_registration.date_of_signing < date('2000-01-01')
  )

```

INSERT

```

INSERT INTO
  copy_of_book
VALUES
  (1000,
   true,
   (SELECT library_code FROM publication, publishing_house WHERE
publication.id_publishing_house = publishing_house.id_publishing_house and
publishing_house.name = 'Penguin Books' and
publishing_house.compilers = 'Manuel Buesden'),
   'New');

```

DELETE

```

delete from
  copy_of_book
where
  copy_of_book.id_copy_of_book in
  (
    select
      copy_of_book.id_copy_of_book
    from
      copy_of_book,
      act_of_debiting
    where
      copy_of_book.id_copy_of_book = act_of_debiting.id_copy_of_book
  )

```

Создание индексов

- 1) Оператор EXPLAIN ANALYZE в запросе без индексов:


```

1  explain analyze
2  select
3      reader.id_reader,
4      reader.name,
5      reader.surname,
6      reader.count_of_books
7  from
8      reader,
9      book_circulation
10 where
11     reader.id_reader = book_circulation.id_reader and
12     book_circulation.fine = True and
13     reader.count_of_books > 9;

```



| | QUERY PLAN | |
|----|--|--|
| | text | |
| 1 | Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=1.839..2.179 rows=56 loops=1) | |
| 2 | Hash Cond: (book_circulation.id_reader = reader.id_reader) | |
| 3 | -> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.046..0.334 rows=489 loop... | |
| 4 | Filter: fine | |
| 5 | Rows Removed by Filter: 511 | |
| 6 | -> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=1.782..1.783 rows=103 loops=1) | |
| 7 | Buckets: 1024 Batches: 1 Memory Usage: 14kB | |
| 8 | -> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.006..1.747 rows=103 loops=1) | |
| 9 | Filter: (count_of_books > 9) | |
| 10 | Rows Removed by Filter: 897 | |
| 11 | Planning Time: 5.381 ms | |
| 12 | Execution Time: 2.208 ms | |


```

1 drop index idx_reader_circulation;
2
3 CREATE INDEX idx_reader_circulation ON book_circulation (id_reader);
4 CREATE INDEX idx_reader_circulation1 ON book_circulation (fine);
5
6 explain analyze
7 select
8     reader.id_reader,
9     reader.name,
10    reader.surname,
11    reader.count_of_books
12 from
13     reader,
14     book_circulation
15 where
16     reader.id_reader = book_circulation.id_reader and
17     book_circulation.fine = True and
18     reader.count_of_books > 9;

```

Data Output Сообщения Notifications



QUERY PLAN

text



| | |
|----|--|
| 1 | Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=0.125..0.228 rows=56 loops=1) |
| 2 | Hash Cond: (book_circulation.id_reader = reader.id_reader) |
| 3 | -> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.010..0.086 rows=489 loop... |
| 4 | Filter: fine |
| 5 | Rows Removed by Filter: 511 |
| 6 | -> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=0.108..0.108 rows=103 loops=1) |
| 7 | Buckets: 1024 Batches: 1 Memory Usage: 14kB |
| 8 | -> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.005..0.095 rows=103 loops=1) |
| 9 | Filter: (count_of_books > 9) |
| 10 | Rows Removed by Filter: 897 |
| 11 | Planning Time: 1.574 ms |
| 12 | Execution Time: 0.249 ms |