Отчет

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

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

Автор: Красюк К.А.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

ИТМО

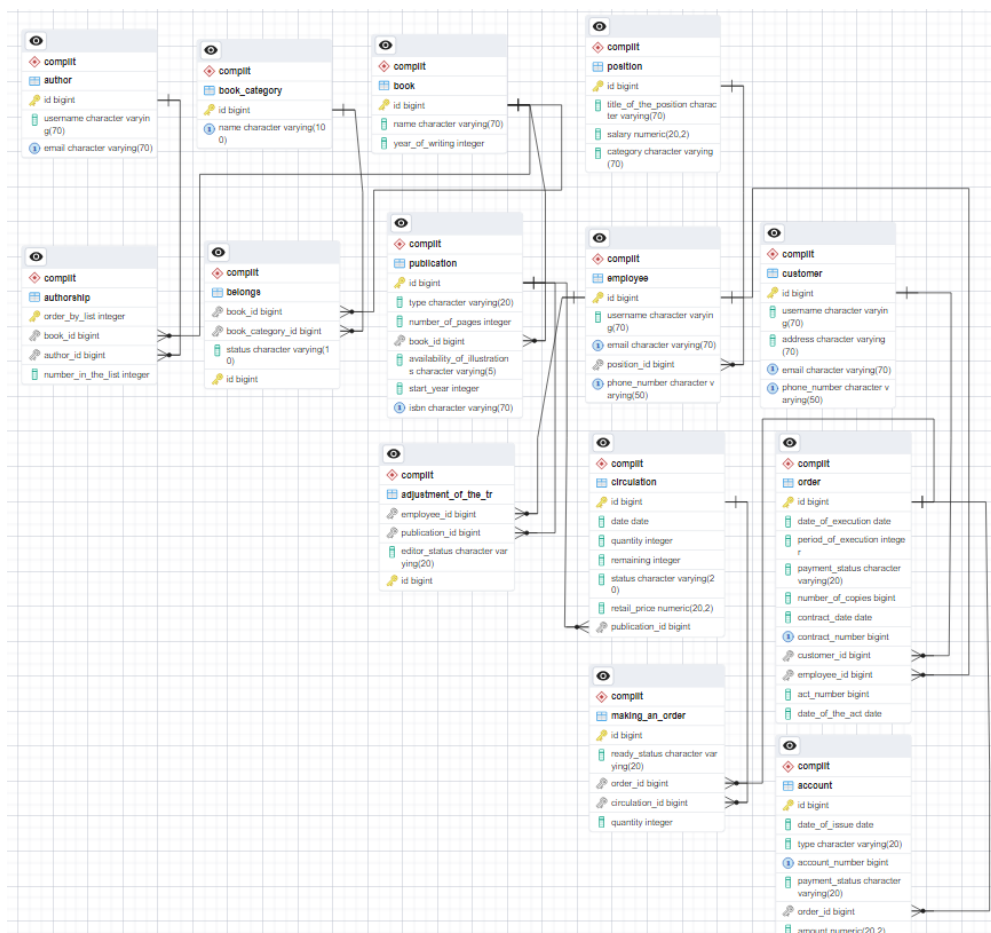Санкт-Петербург 2023

Оглавление

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.

Схема базы данных

Выполнение

1. Запросы к базе данных

- Список книг, изданных в текущем году и относящихся к категории «Базы данных».

SELECT book.id, book.name, book.year_of_writing, book_category.name

FROM complit.book

JOIN complit.belongs ON book.id = belongs.book_id

JOIN complit.book_category ON belongs.book_category_id = book_category.id

WHERE book.year_of_writing = EXTRACT(YEAR FROM CURRENT_DATE)

AND book_category.name = 'Databases';

```
Query    Query History
1  SELECT book.id, book.name, book.year_of_writing, book_category.name
2  FROM complit.book
3  JOIN complit.belongs ON book.id = belongs.book_id
4  JOIN complit.book_category ON belongs.book_category_id = book_category.id
5  WHERE book.year_of_writing = EXTRACT(YEAR FROM CURRENT_DATE)
6  AND book_category.name = 'Databases';
```

Data Output   Messages   Notifications

	id bigint	name character varying (70)	year_of_writing integer	name character varying (100)
1	974	Databases in our life	2023	Databases

- Список покупателей, заказавших книг на сумму, превышающую среднюю сумму заказа за год.

SELECT customer.id, customer.username, SUM(account.amount) AS ac_amount

FROM complit.customer

JOIN complit."order" ON customer.id = "order".customer_id

JOIN complit.account ON "order".id = account.order_id

GROUP BY customer.id, customer.username

HAVING SUM(account.amount) > (

SELECT AVG(amount)

FROM complit.account

JOIN complit."order" ON account.order_id = "order".id

WHERE ("order".date_of_execution - NOW()) < interval '1 year');

```
Query   Query History
 1  SELECT customer.id, customer.username, SUM(account.amount) AS ac_amount
 2  FROM complit.customer
 3  JOIN complit."order" ON customer.id = "order".customer_id
 4  JOIN complit.account ON "order".id = account.order_id
 5  GROUP BY customer.id, customer.username
 6  HAVING SUM(account.amount) > (
 7  SELECT AVG(amount)
 8  FROM complit.account
 9  JOIN complit."order" ON account.order_id = "order".id
10  WHERE ("order".date_of_execution - NOW()) < interval '1 year'
11  );
```

Data Output   Messages   Notifications

id [PK] bigint	username character varying (70)	ac_amount numeric
868	Rhonda Hughes	98454.00
559	Jonathan Smith	68762.00
303	Alan Evans	74954.00
290	Jason Montgomery	76848.00
810	Kelli Key	105533.00

- Список книг, которые не заказывались в течение последних двух кварталов.

SELECT book.id, book.name

FROM complit.book

WHERE book.id NOT IN (

SELECT DISTINCT book.id

FROM complit.book b

JOIN complit.publication ON book.id = "publication".book_id

JOIN complit.circulation ON "publication".id = circulation.publication_id

JOIN complit.making_an_order ON circulation.id = making_an_order.circulation_id

JOIN complit."order" ON making_an_order.order_id = "order".id

WHERE "order".date_of_execution > (NOW() - INTERVAL '6 months')

);

```
Query   Query History

 1   SELECT book.id, book.name
 2   FROM complit.book
 3   WHERE book.id NOT IN (
 4   SELECT DISTINCT book.id
 5   FROM complit.book b
 6   JOIN complit.publication ON book.id = "publication".book_id
 7   JOIN complit.circulation ON "publication".id = circulation.publication_id
 8   JOIN complit.making_an_order ON circulation.id = making_an_order.circulation_id
 9   JOIN complit."order" ON making_an_order.order_id = "order".id
10   WHERE "order".date_of_execution > (NOW() - INTERVAL '6 months')
11   );
```

Data Output   Messages   Notifications

	id [PK] bigint	name character varying (70)
1	969	Virtual asymmetric toolset
2	35	Multi-channeled neutral neural-net
3	458	Realigned value-added open system
4	366	Optional maximized middleware
5	877	Synergistic dynamic Internet solution
6	298	Balanced real-time software
7	407	Optional asynchronous collaboration
8	941	Universal upward-trending toolset
9	161	Polarized coherent time-frame

- Список авторов, не написавших ни одной книги, относящейся к категории "Языки программирования".

SELECT author.id, author.username

FROM complit.author

WHERE author.id NOT IN (

SELECT DISTINCT author.id

FROM complit.author

JOIN complit.authorship ON author.id = authorship.author_id

JOIN complit.book ON authorship.book_id = book.id

JOIN complit.belongs ON book.id = belongs.book_id

JOIN complit.book_category ON belongs.book_category_id = book_category.id

WHERE book_category.name LIKE 'Programming languagess');

Query   Query History

```
 1    SELECT author.id, author.username
 2    FROM complit.author
 3    WHERE author.id NOT IN (
 4    SELECT DISTINCT author.id
 5    FROM complit.author
 6    JOIN complit.authorship ON author.id = authorship.author_id
 7    JOIN complit.book ON authorship.book_id = book.id
 8    JOIN complit.belongs ON book.id = belongs.book_id
 9    JOIN complit.book_category ON belongs.book_category_id = book_category.id
10    WHERE book_category.name LIKE 'Programming languagess');
11
```

Data Output   Messages   Notifications

	id [PK] bigint	username character varying (70)
1	508	Michael Taylor
2	592	John Johnson
3	984	Michael Clark
4	222	Patricia Simpson
5	377	Isaac Collins
6	89	Ricky Santos
7	422	Michelle Richardson
8	778	Seth Craig
9	163	Steven Tyler

- Список книг, в названиях которых содержится слово "проектирование" и которые присутствуют на базе в количестве, превышающем 50 экземпляров.

SELECT book.id, book.name, circulation.remaining

FROM complit.book

JOIN complit.publication ON book.id = publication.book_id

JOIN complit.circulation ON publication.id = circulation.publication_id

WHERE book.name LIKE '%проектирование%'

GROUP BY book.id, book.name, circulation.remaining

HAVING circulation.remaining > 50;

```
Query    Query History

1    SELECT book.id, book.name, circulation.remaining
2    FROM complit.book
3    JOIN complit.publication ON book.id = publication.book_id
4    JOIN complit.circulation ON publication.id = circulation.publication_id
5    WHERE book.name LIKE '%проектирование%'
6    GROUP BY book.id, book.name, circulation.remaining
7    HAVING circulation.remaining > 50;
8
9    |
```

Data Output   Messages   Notifications

id bigint	name character varying (70)	remaining integer
1	619 Механическое проектирование	86

- Покупателя, сделавшего заказ на максимальную сумму за последний месяц.

SELECT customer.id, customer.username, MAX(account.amount) AS max_amount

FROM complit.customer

JOIN complit."order" ON customer.id = "order".customer_id

JOIN complit.account ON "order".id = account.order_id

WHERE "order".date_of_execution > (NOW() - INTERVAL '1 month')

GROUP BY customer.id, customer.username

ORDER BY max_amount DESC

LIMIT 1;

```
Query   Query History
1  SELECT customer.id, customer.username, MAX(account.amount) AS max_amount
2  FROM complit.customer
3  JOIN complit."order" ON customer.id = "order".customer_id
4  JOIN complit.account ON "order".id = account.order_id
5  WHERE "order".date_of_execution > (NOW() - INTERVAL '1 month')
6  GROUP BY customer.id, customer.username
7  ORDER BY max_amount DESC
8  LIMIT 1;
9
```

Data Output   Messages   Notifications

	id [PK] bigint	username character varying (70)	max_amount numeric
1	204	Harold Adams	75921.00

- Список книг, не попавших ни в один из заказов в течение последнего года.

SELECT book.id, book.name

FROM complit.book

WHERE book.id NOT IN (

SELECT DISTINCT book.id

FROM complit.book b

JOIN complit.publication ON book.id = "publication".book_id

JOIN complit.circulation ON "publication".id = circulation.publication_id

JOIN complit.making_an_order ON circulation.id = making_an_order.circulation_id

JOIN complit."order" ON making_an_order.order_id = "order".id

WHERE "order".date_of_execution > (NOW() - INTERVAL '1 year')

);

```
Query   Query History
 1   SELECT book.id, book.name
 2   FROM complit.book
 3   WHERE book.id NOT IN (
 4   SELECT DISTINCT book.id
 5   FROM complit.book b
 6   JOIN complit.publication ON book.id = "publication".book_id
 7   JOIN complit.circulation ON "publication".id = circulation.publication_id
 8   JOIN complit.making_an_order ON circulation.id = making_an_order.circulation_id
 9   JOIN complit."order" ON making_an_order.order_id = "order".id
10   WHERE "order".date_of_execution > (NOW() - INTERVAL '1 year')
11   );
12
13
14
15
```

Data Output   Messages   Notifications

	id [PK] bigint	name character varying (70)
1	969	Virtual asymmetric toolset
2	35	Multi-channeled neutral neural-net
3	366	Optional maximized middleware
4	298	Balanced real-time software
5	407	Optional asynchronous collaboration
6	941	Universal upward-trending toolset
7	161	Polarized coherent time-frame
8	103	Adaptive system-worthy interface
9	815	De-engineered composite Graphic Interface

2. Представления

• содержащее сведения о количестве заказанных экземпляров каждой книги, изданной в текущем году;

CREATE VIEW order_current_year AS

SELECT book.id, book.name, COUNT("order".id) AS order_count

FROM complit.book

JOIN complit.publication ON book.id = "publication".book_id

JOIN complit.circulation ON "publication".id = circulation.publication_id

JOIN complit.making_an_order ON circulation.id = making_an_order.circulation_id

JOIN complit."order" ON making_an_order.order_id = "order".id

WHERE book.id IN (

SELECT book.id

FROM complit.book

WHERE book.year_of_writing = EXTRACT(YEAR FROM CURRENT_DATE))

GROUP BY book.id, book.name;

SELECT * FROM order_current_year;

```
Query    Query History
1   CREATE VIEW order_current_year AS
2   SELECT book.id, book.name, COUNT("order".id) AS order_count
3   FROM complit.book
4   JOIN complit.publication ON book.id = "publication".book_id
5   JOIN complit.circulation ON "publication".id = circulation.publication_id
6   JOIN complit.making_an_order ON circulation.id = making_an_order.circulation_id
7   JOIN complit."order" ON making_an_order.order_id = "order".id
8   WHERE book.id IN (
9   SELECT book.id
10  FROM complit.book
11  WHERE book.year_of_writing = EXTRACT(YEAR FROM CURRENT_DATE)
12  )
13  GROUP BY book.id, book.name;
14
15  SELECT * FROM order_current_year;
```

Data Output   Messages   Notifications

	id bigint	name character varying (70)	order_count bigint
1	739	Programming in 5 steps	1
2	974	Databases in our life	1

- Количество заказов по покупателям за последний год.

CREATE VIEW customer_year AS

SELECT customer.id, customer.username, COUNT("order".id) AS order_count

FROM complit.customer

JOIN complit."order" ON customer.id = "order".customer_id

WHERE "order".date_of_execution > (NOW() - INTERVAL '1 year')

GROUP BY customer.id, customer.username

ORDER BY order_count DESC;

SELECT * FROM customer_year;


```
Query   Query History
 1  CREATE VIEW customer_year AS
 2  SELECT customer.id, customer.username, COUNT("order".id) AS order_count
 3  FROM complit.customer
 4  JOIN complit."order" ON customer.id = "order".customer_id
 5  WHERE "order".date_of_execution > (NOW() - INTERVAL '1 year')
 6  GROUP BY customer.id, customer.username
 7  ORDER BY order_count DESC;
 8
 9  SELECT * FROM customer_year;
10
```

Data Output   Messages   Notifications

	id bigint	username character varying (70)	order_count bigint
1	204	Harold Adams	1
2	962	Dylan Thompson	1
3	677	Kim Wilson	1
4	880	Shawn Whitaker	1
5	185	Lori Norton	1
6	868	Rhonda Hughes	1
7	26	Patrick Delgado	1
8	190	Mark Mcdonald	1
9	303	Alan Evans	1

Запросы на модификацию данных

- INSERT (добавление новых данных в таблицу complit.belongs)

INSERT INTO complit.belongs (book_id, book_category_id, status, id)

VALUES

((SELECT id FROM complit.book WHERE name = 'Polarized coherent time-frame'),

(SELECT id FROM complit.book_category WHERE name = 'Fiction'),

'true', 23);

Скриншот до выполнения запроса

	book_id bigint	book_category_id bigint	status character varying (10)	id [PK] bigint
1	366	13	true	1
2	741	2	true	2
3	826	6	false	3
4	941	20	true	4
5	826	17	false	5
6	103	5	false	6
7	257	9	true	7
8	969	12	false	8
9	35	21	true	9
10	748	16	false	10
11	35	6	false	11
12	458	11	true	12
13	204	3	true	13
14	407	2	false	14
15	941	4	true	15
16	668	4	false	16
17	739	16	false	17
18	741	18	true	18
19	376	14	true	19
20	458	19	true	20
21	974	21	true	21
22	257	22	true	22

13

Скриншот после выполнения запроса

	book_id bigint	book_category_id bigint	status character varying (10)	id [PK] bigint
2	741	2	true	2
3	826	6	false	3
4	941	20	true	4
5	826	17	false	5
6	103	5	false	6
7	257	9	true	7
8	969	12	false	8
9	35	21	true	9
10	748	16	false	10
11	35	6	false	11
12	458	11	true	12
13	204	3	true	13
14	407	2	false	14
15	941	4	true	15
16	668	4	false	16
17	739	16	false	17
18	741	18	true	18
19	376	14	true	19
20	458	19	true	20
21	974	21	true	21
22	257	22	true	22
23	161	4	true	23

- UPDATE (обновление email в таблице автора)

UPDATE complit.author

SET email = 'isaaccol@example.com'

WHERE id = (SELECT id FROM complit.author WHERE username = 'Isaac Collins');

14

Скриншот до выполнения запроса

	id [PK] bigint	username character varying (70)	email character varying (70)
1	776	Michael Hunt	spencerdebbie@example.net
2	508	Michael Taylor	loriowens@example.net
3	592	John Johnson	rhernandez@example.net
4	984	Michael Clark	clarence01@example.net
5	222	Patricia Simpson	lbrown@example.net
6	377	Isaac Collins	llin@example.com
7	89	Ricky Santos	nicoletorres@example.net
8	422	Michelle Richardson	jamesstewart@example.org
9	778	Seth Craig	sethclayton@example.net
10	163	Steven Tyler	rbrady@example.com
11	884	Jonathan Howard	shannongeorge@example.org
12	241	Danielle Mercado	michaelkeller@example.net
13	853	Jeffery Terrell	hgarcia@example.org
14	921	John Fisher	christophernguyen@example.c...
15	56	Shane James	fitzpatrickjennifer@example.org
16	593	Samuel Hughes	melendezsarah@example.net
17	29	Bridget Mitchell	jennifer22@example.org
18	242	Gabriela Campos	markherrera@example.net
19	591	Tina Taylor	mbrown@example.net
20	706	Erik Rangel	gbates@example.net
21	167	Shawn Mendess	mendess@gmail.com
22	596	Camila Mendess	cami084@gmail.com

Скриншот после выполнения запроса

	id [PK] bigint	username character varying (70)	email character varying (70)
1	776	Michael Hunt	spencerdebbie@example.net
2	508	Michael Taylor	loriowens@example.net
3	592	John Johnson	rhernandez@example.net
4	984	Michael Clark	clarence01@example.net
5	222	Patricia Simpson	lbrown@example.net
6	89	Ricky Santos	nicoletorres@example.net
7	422	Michelle Richardson	jamesstewart@example.org
8	778	Seth Craig	sethclayton@example.net
9	163	Steven Tyler	rbrady@example.com
10	884	Jonathan Howard	shannongeorge@example.org
11	241	Danielle Mercado	michaelkeller@example.net
12	853	Jeffery Terrell	hgarcia@example.org
13	921	John Fisher	christopherguyen@example.c...
14	56	Shane James	fitzpatrickjennifer@example.org
15	593	Samuel Hughes	melendezsarah@example.net
16	29	Bridget Mitchell	jennifer22@example.org
17	242	Gabriela Campos	markherrera@example.net
18	591	Tina Taylor	mbrown@example.net
19	706	Erik Rangel	gbates@example.net
20	167	Shawn Mendess	mendess@gmail.com
21	596	Camila Mendess	cami084@gmail.com
22	377	Isaac Collins	isaaccol@example.com

- **DELETE** (удаление всех заказчиков, не сделавших ни одного заказа)

DELETE FROM complit.customer

WHERE customer.id NOT IN (

SELECT DISTINCT customer.id

FROM complit.customer

JOIN complit."order" ON customer.id = "order".customer_id

);

Скриншот до выполнения запроса

	id [PK] bigint	username character varying (70)	address character varying (70)	email character varying (70)	phone_number character varying (50)
1	851	Shawna Stout	4750 Jaclyn Plains Apt. 490	stephaniemccoy@example.net	+9(815)-718-19-44
2	868	Rhonda Hughes	2122 Brown Mill Apt. 006	steven29@example.net	+4(172)-431-19-11
3	303	Alan Evans	248 Miller Point Apt. 911	zmartinez@example.net	+4(183)-423-37-82
4	592	Travis Mack	249 Moore Port Apt. 844	kellybrian@example.net	+5(600)-104-38-22
5	559	Jonathan Smith	Unit 0207 Box 4847	lnichols@example.com	+9(231)-162-95-35
6	677	Kim Wilson	461 Flores Spurs Suite 838	wnewman@example.org	+1(653)-171-17-96
7	784	Jessica Beck	162 Medina Park	marshdanielle@example.net	+4(499)-306-31-28
8	26	Patrick Delgado	69614 Smith Street	cherylhernandez@example.c…	+1(783)-348-47-57
9	190	Mark Mcdonald	PSC 5387, Box 5797	stephenlynn@example.net	+5(582)-376-37-75
10	290	Jason Montgomery	3095 Stephanie Centers	jasonweaver@example.com	+5(116)-294-63-52
11	880	Shawn Whitaker	7590 Bishop Radial Apt. 942	pamelamitchell@example.co…	+6(556)-774-58-87
12	326	Joshua Johnson	312 Diana Springs	qchavez@example.com	+6(323)-358-30-19
13	185	Lori Norton	01234 Teresa Lake Suite 239	lisabeck@example.net	+6(134)-660-49-48
14	822	Nathan Gilmore	32827 Wells Points Apt. 892	ivaughn@example.net	+4(475)-912-38-17
15	997	Marie Daugherty	218 Valencia View	amandahunt@example.com	+3(607)-440-53-25
16	459	Sydney Hardy	07374 Lee Views	westmichael@example.org	+1(181)-669-23-93
17	431	Christina Allen	08489 Vazquez Land	mgonzalez@example.org	+8(796)-601-43-27
18	810	Kelli Key	809 Parsons Points Suite 370	lisa57@example.net	+2(344)-880-13-71
19	962	Dylan Thompson	6919 Larry Pines	garciadiane@example.org	+6(720)-834-36-70
20	204	Harold Adams	9998 Melissa Turnpike	carolyncallahan@example.org	+5(116)-760-36-36

Скриншот после выполнения запроса

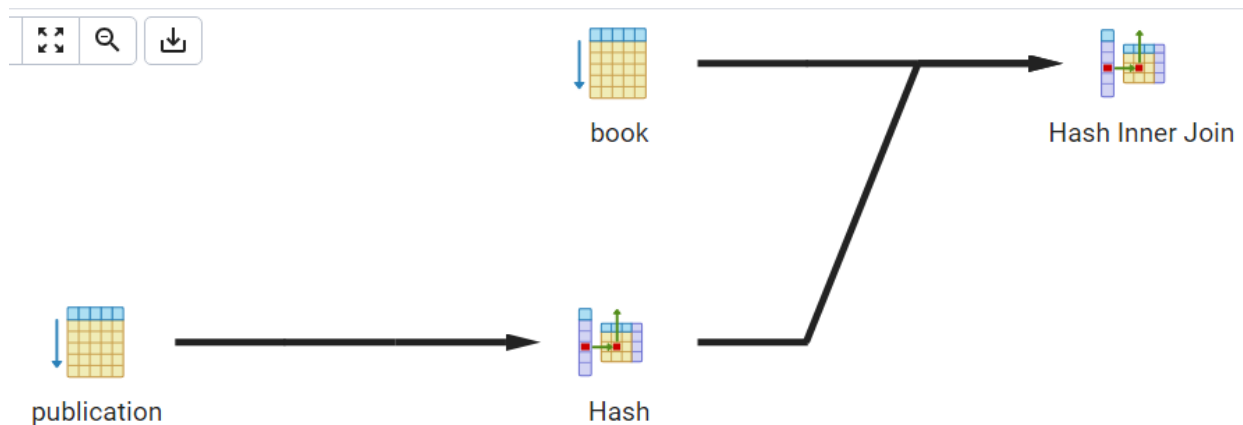|	id [PK] bigint	username character varying (70)	address character varying (70)	email character varying (70)	phone_number character varying (50)
1	851	Shawna Stout	4750 Jaclyn Plains Apt. 490	stephaniemccoy@example.net	+9(815)-718-19-44
2	868	Rhonda Hughes	2122 Brown Mill Apt. 006	steven29@example.net	+4(172)-431-19-11
3	303	Alan Evans	248 Miller Point Apt. 911	zmartinez@example.net	+4(183)-423-37-82
4	592	Travis Mack	249 Moore Port Apt. 844	kellybrian@example.net	+5(600)-104-38-22
5	559	Jonathan Smith	Unit 0207 Box 4847	lnichols@example.com	+9(231)-162-95-35
6	677	Kim Wilson	461 Flores Spurs Suite 838	wnewman@example.org	+1(653)-171-17-96
7	784	Jessica Beck	162 Medina Park	marshdanielle@example.net	+4(499)-306-31-28
8	26	Patrick Delgado	69614 Smith Street	cherylhernandez@example.c…	+1(783)-348-47-57
9	190	Mark Mcdonald	PSC 5387, Box 5797	stephenlynn@example.net	+5(582)-376-37-75
10	290	Jason Montgomery	3095 Stephanie Centers	jasonweaver@example.com	+5(116)-294-63-52
11	880	Shawn Whitaker	7590 Bishop Radial Apt. 942	pamelamitchell@example.co…	+6(556)-774-58-87
12	185	Lori Norton	01234 Teresa Lake Suite 239	lisabeck@example.net	+6(134)-660-49-48
13	822	Nathan Gilmore	32827 Wells Points Apt. 892	ivaughn@example.net	+4(475)-912-38-17
14	459	Sydney Hardy	07374 Lee Views	westmichael@example.org	+1(181)-669-23-93
15	810	Kelli Key	809 Parsons Points Suite 370	lisa57@example.net	+2(344)-880-13-71
16	962	Dylan Thompson	6919 Larry Pines	garciadiane@example.org	+6(720)-834-36-70
17	204	Harold Adams	9998 Melissa Turnpike	carolyncallahan@example.org	+5(116)-760-36-36

Создание индексов

- Простой индекс

Запрос без индекса

```
Query    Query History
1  SELECT book.id, book.name, publication.start_year
2  FROM complit.book
3  JOIN complit.publication ON book.id = publication.book_id
4  WHERE start_year > 1970;
```

Data Output   Messages   Explain ×   Notifications

	id bigint	name character varying (70)	start_year integer
1	35	Multi-channeled neutral neural-net	2022
2	366	Optional maximized middleware	2007
3	407	Optional asynchronous collaboration	1975
4	407	Optional asynchronous collaboration	1995
5	881	Integrated uniform benchmark	1995
6	748	Visionary asynchronous leverage	2021
7	668	Phased systemic orchestration	1978
8	668	Phased systemic orchestration	2000
9	668	Phased systemic orchestration	2001
10	974	Databases in our life	2023
11	739	Programming in 5 steps	2023
12	893	Проектирование дизайна проекта	2002

Data Output | Messages | Notifications

```
Successfully run. Total query runtime: 68 msec.
12 rows affected.
```

Создание индекса

```
Query    Query History
1    CREATE INDEX publication_start_idx ON complit.publication (start_year);
2
```

Время выполнения запроса после создания индекса

```
Data Output    Messages    Explain  ×    Notifications

Successfully run. Total query runtime: 39 msec.
12 rows affected.
```

- Составной индекс

Запрос без индекса

```
Query    Query History
1    SELECT book.id, book.name, publication.start_year, publication.number_of_pages
2    FROM complit.book
3    JOIN complit.publication ON book.id = publication.book_id
4    WHERE start_year > 1970 AND number_of_pages < 200;
5
```

	id bigint	name character varying (70)	start_year integer	number_of_pages integer
1	366	Optional maximized middleware	2007	99
2	407	Optional asynchronous collaboration	1975	30
3	407	Optional asynchronous collaboration	1995	49
4	748	Visionary asynchronous leverage	2021	25
5	668	Phased systemic orchestration	1978	197
6	668	Phased systemic orchestration	2000	181

Successfully run. Total query runtime: 42 msec.
6 rows affected.

Создание индекса

```
Query   Query History
1   CREATE INDEX publication_start_number ON complit.publication (start_year, number_of_pages);
```

Время выполнения запроса после создания индекса

Data Output   Messages   Explain ×   Notifications

Successfully run. Total query runtime: 35 msec.
6 rows affected.

Вывод

В ходе данной лабораторной работы были созданы различные запросы на выборку данных к базе данных PostgreSQL, так же с использованием подзапросов. В работе с подзапросами удалось осуществить их применение в запросах на запись, удаление и обновление данных. Так же удалось узнать, что при использовании индексов операция чтения запроса быстрее, чем без них.