

Database Management Systems

01

LIBRARY

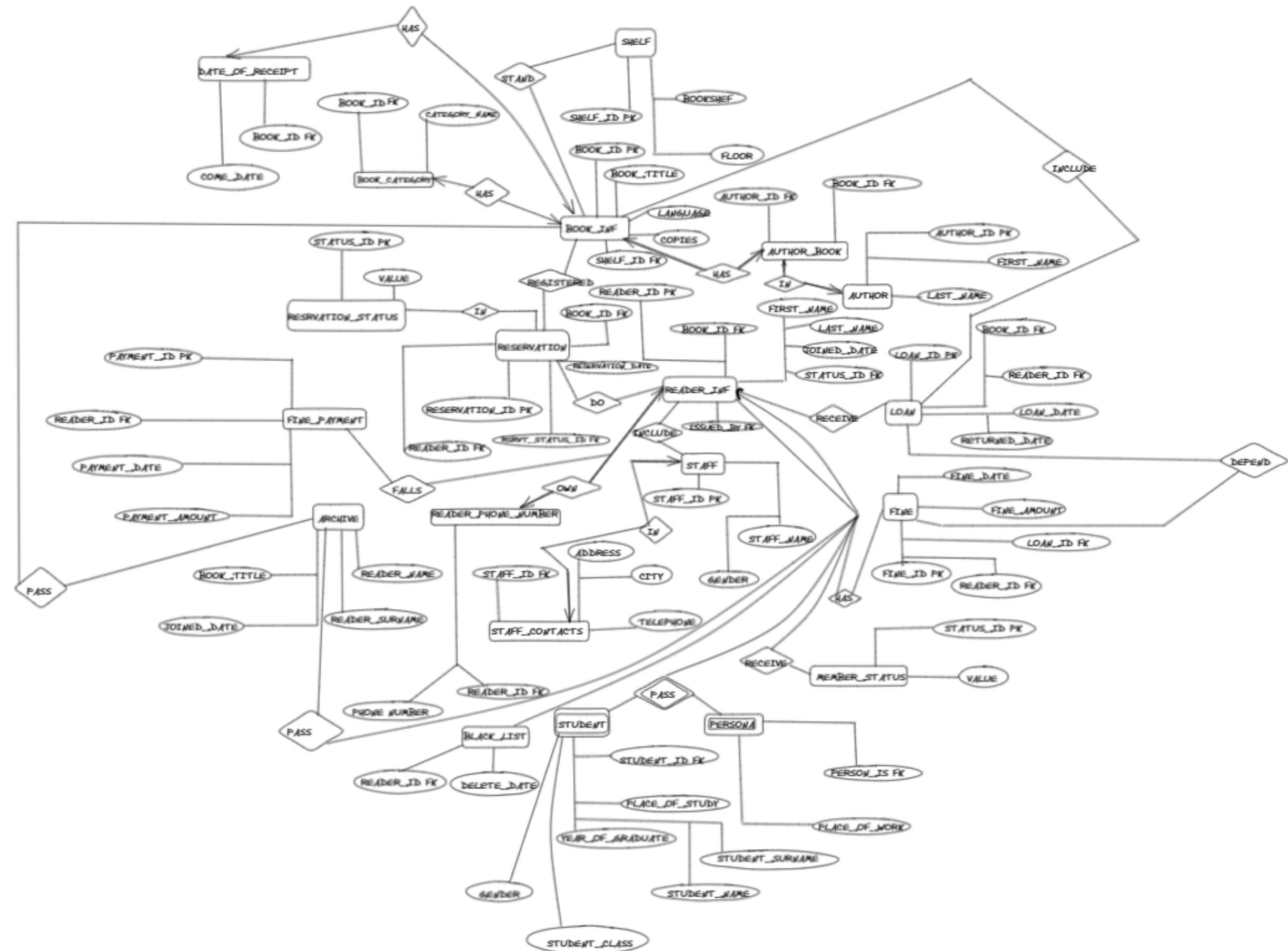
Arystan Temirlan
Abzhan Muhammedali
Makimova Dilnaz

1. OUR PROJECT IS A LIBRARY DATABASE. THAT IS, WE CAN SIMPLIFY WORK AND TIME THROUGH THIS DATABASE. THAT IS, FOR EXAMPLE, A LIBRARY EMPLOYEE CAN EASILY FIND A PERSON WHO TOOK A BOOK OR KNOW THE NUMBER OF BOOKS READ, AND SO ON
2. MY FUNCTION OUTPUTS INFORMATION ABOUT BOOKS THAT HAVE BEEN WRITTEN, THEIR CATEGORY AND LANGUAGE, OUTPUTS WHERE THE BOOK IS LOCATED, OUTPUTS THE NUMBER OF BOOKS WRITTEN BY THE AUTHOR, OUTPUTS PEOPLE WHO READ THIS BOOK, OUTPUTS A PERSON FROM THE READER, DISPLAYS BOOKS FROM THE ARCHIVE THAT WERE DELIVERED AFTER THIS DATE, DISPLAYS THE BOOK WITH THE LARGEST COPY AND SO ON
- 3) THE END USER IS A LIBRARY EMPLOYEE. IT WILL BE CONVENIENT FOR HIM TO WORK WITH DATABASES BECAUSE EVERYTHING IS MUCH EASIER TO FIND AND ADD SOMETHING.
4. I AM SENDING OLD DATA OR DATA THAT SHOULD BE DELETED TO THE ARCHIVE. THAT IS, I SAVE THIS DATA IN THE ARCHIVE
- 5) THERE WAS THIS ITEM IN THE DEPUTY DESCRIPTION OF THE PROJECT. OF ALL THE OPTIONS, THE TOPIC “LIBRARY” WAS CLOSE TO US. BECAUSE OUR UNIVERSITY DOES NOT YET HAVE A GOOD DATABASE. AND IN THE FRONT END LESSON, WE ALSO HAVE A PROJECT AND WE WANTED TO RECREATE THE WEBSITE OF THE SDU LIBRARY. ALL THIS WAS CONNECTED AND THEREFORE WE CHOSE THIS TOPIC IN THE FUTURE, IF THIS PROJECT IS IMPROVED, IT WILL BE ABLE TO HELP VARIOUS LARGE LIBRARIES, OR IF OUR UNIVERSITY EXPANDS THE TERRITORY OF THE LIBRARY, WE WILL NEED SUCH A DATABASE.
THANKS TO THE PROJECT, WE WERE ABLE TO REPEAT ALL THE TOPICS AND THINK BETTER IN THE FIELD OF THE DATABASE.

01

ERD

02



Functional Dependency

1 TABLE: reservation

reservation id-> reader id, book id, reservation date, rsrv status id

2 TABLE: book category

book id->category name

3 TABLE: person

person id->place of work

4 TABLE: author

author id->first name, last name

5 TABLE: reader inf

reader id-> book id, last name, first name, status id, joined date, issued by

04

Functional Dependency

6 TABLE: reader phone number

reader id->phone number

7 TABLE: loan

loan id->book id, reader id, returned date, loan date

8 TABLE: fine

fine id-> fine date, reader id, fine amount, loan id

9 TABLE: fine payment

payment id ->, payment date, reader id, payment amount

10 TABLE: reservation status

status id->value

11 TABLE: staff contacts

staff id -> telephone, city, address

04

Functional Dependency

13 TABLE: student

student_id->place_of_study, year_of_graduate, gender, student_class,
student_name, student_surname

14 TABLE:member_status

m_status_id->value

15 TABLE:book_inf

book_id->shelf_id, book_title, language, copies

16 TABLE:staff

staff_id -> staff_name, gender

17 TABLE:shelf

shelf_id->floor, bookshelf

18 TABLE:data_of_receipt

Book_id->come_date

19 TABLE:blacklist

Reader_id->delete_date

20 TABLE: archive

Reader_name->reader_surname, book_title, joined_date

05

NORMALIZATION

1 TABLE: reservation

reservation_id->book_id, reader_id, reservation_date, rsrv_status_id

old Scheme: { reservation_id, book_id, reader_id, reservation_date, rsrv_status_id }

2 NF New Scheme: { reservation_id, reservation_date, rsrv_status_id }

New Scheme: { reservation_id, book_id, reader_id }

3 NF: reservation_id-> rsrv_status_id, rsrv_status_id-> reservation_date

reader_id-> book_id, book_id-> reservation_id

BCNF: reservation_id-> rsrv_status_id, rsrv_status_id-> reservation_date, DURATION

reader_id-> book_id, book_id-> reservation_id, FOR HOW MANY DAYS

4 TABLE: author

author_id->first_name, last_name

Old Scheme: { author_id, first_name, last_name }

2 NF New Scheme: { author_id, first_name }

New Scheme: { author_id, last_name }

06

NORMALIZATION

5 TABLE: reader_inf

reader_id-> book_id, last_name, first_name, status_id, joined_date, issued_by

Old Scheme: { reader_id, book_id, last_name, first_name, status_id, joined_date, issued_by }

2 NF New Scheme: { reader_id, book_id, last_name, first_name }

New Scheme: { reader_id, status_id, joined_date, issued_by }

3 NF: reader_id->book_id, book_id-> last_name, first_name

reader_id->status_id, status_id-> joined_date, issued_by

BCNF: reader_id->book_id, book_id-> last_name, first_name, UNIVERSITY
EMAIL

reader_id->status_id, status_id-> joined_date, issued_by, HAVE A FINE?

|

7 TABLE: loan

loan_id->book_id, reader_id, returned_date, loan_date

Old Scheme: { loan_id, book_id, reader_id, returned_date, loan_date }

2 NF New Scheme: { loan_id, reader_id }

New Scheme: { loan_id, book_id, returned_date, loan_date }

3 NF: loan_id-> book_id, returned_date, returned_date->loan_date

BCNF: loan_id-> book_id, returned_date, returned_date->loan_date, HOW
MANY DAY LOAN

07

NORMALIZATION

8 TABLE: fine

fine_id->fine_date, reader_id, fine_amount, loan_id

Old Scheme: {fine_id, fine_date, reader_id, fine_amount, loan_id}

2 NF New Scheme: {fine_id, fine_date, fine_amount }

New Scheme: {fine_id, reader_id, loan_id}

3 NF: fine_id->fine_date, fine_date ->fine_amount

fine_id->reader_id, fine_id->loan_id

BCNF: fine_id->fine_date, fine_date ->fine_amount, PHONE NUMBER

fine_id->reader_id, fine_id->loan_id, LOAN_DATE

|

9 TABLE: fine_payment

payment_id -> payment_date, reader_id, payment_amount

Old Scheme: {payment_id, payment_date, reader_id, payment_amount}

2 NF New Scheme: {payment_id, payment_date, payment_amount }

New Scheme: {payment_id, reader_id }

3 NF: payment_id->payment_date, payment_date->payment_amount

BCNF: payment_id->payment_date, payment_date->payment_amount, BONUS

08

NORMALIZATION

11 TABLE: staff_contacts

staff_id -> telephone, city, address

Old Scheme: { staff_id, telephone, city, address }

2 NF New Scheme: { staff_id, telephone }

New Scheme: { staff_id, city, address }

3 NF: staff_id->city, city->address

BCNF: staff_id->city, city->address, INDEX

13 TABLE: student

student_id->place_of_study, year_of_graduate, gender, student_class,
student_name, student_surname

Old Scheme: { student_id, place_of_study, year_of_graduate, gender,
student_class, student_name, student_surname }

2 NF New Scheme: { student_id, place_of_study, year_of_graduate, student_class }

New Scheme: { student_id, student_name, student_surname, gender }

3NF: student_id->place_of_study, year_of_graduate, year_of_graduate->student_class;

student_id->student_name, student_surname, student_id->gender

BCNF: student_id->place_of_study, year_of_graduate, year_of_graduate->student_class, DEGREE

student_id->student_name, student_surname, STUDENT_EMAIL, student_id->gender

NORMALIZATION

17 TABLE:shelf

shelf_id->floor, bookshelf

Old Scheme: { shelf_id, floor, bookshelf }

2 NF New Scheme: { shelf_id, bookshelf }

New Scheme: { floor, bookshelf }

20 TABLE: archive

Reader_name->reader_surname, book_title, joined_date

Old Scheme: { reader_name, reader_surname, book_title, joined_date }

2 NF New Scheme: { reader_name, reader_surname }

New Scheme: { reader_name, book_title, joined_date }

3 NF: reader_name-> book_title, book_title->joined_date

BCNF: reader_name-> book_title, book_title->joined_date, ISSUED DATE

10

NORMALIZATION

15 TABLE:book_inf

book_id-> shelf_id, book_title, language, copies

Old Scheme: { book_id, shelf_id, book_title, language, copies }

2 NF New Scheme: { book_id, shelf_id, copies }

New Scheme: { book_id, book_title, language }

3 NF: book_id->shelf_id, book_id-> copies

book_id->book_title, book_title->language

BCNF: book_id->shelf_id.NUMBER_OF_SHELF, book_id-> copies

book_id->book_title, book_title->language, NUMBER PAGE

16 TABLE:staff

staff_id -> staff_name, gender

Old Scheme: { staff_id, staff_name, gender }

2 NF New Scheme: { staff_id, staff_name }

New Scheme: { staff_name, gender }

13

Query

INFORMATION ABOUT BOOKS WHO IS WRITTEN, HIS CATEGORY AND LANGUAGE.

```

CREATE VIEW ALLBOOKS AS
SELECT bi.book_id, a.first_name, a.last_name, bi.book_title, bc.category_name, bi.language
FROM book_inf AS bi, author_book AS ab
INNER JOIN author AS a ON a.author_id=ab.author_id
JOIN book_category AS bc ON ab.book_id=bc.book_id
WHERE bi.book_id=ab.book_id;

```

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0033 сек.)

1. $\pi_{bi.book_id, a.first_name, a.last_name, bi.book_title, bc.category_name, bi.language}$

$$\sigma_{bi.book_id = ab.book_id}$$

$$(\rho_{bi.book_inf} \times$$

$$(\rho_{ab.author_book} \bowtie a.author_id = ab.author_id)$$

$$\rho_a.author \bowtie ab.book_id = bc.book_id)$$

$$\rho_{bc.book_category})$$

14

Query WHERE BOOKS

The screenshot shows the MySQL Workbench interface with two tabs. The top tab displays the SQL code for creating a view:

```
1 CREATE VIEW whereBook AS
2 SELECT bi.book_title, sh.bookshelf, sh.floor, bi.copies, bi.language
3 FROM book_inf AS bi, shelf AS sh
4 WHERE bi.shelf_id=sh.shelf_id;
```

The bottom tab shows the results of the query execution:

```
CREATE VIEW whereBook AS SELECT bi.book_title, sh.bookshelf, sh.floor, bi.copies, bi.language FROM book_inf AS bi, shelf AS sh WHERE bi.shelf_id=sh.shelf_id;
```

A message in the status bar indicates: "MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0033 сек.)".

2. $\pi_{bi_book_title, sh_bookshelf, sh_floor, bi_copies, bi_language}$

$\sigma_{bi_shelf_id = sh_shelf_id}$

$(\rho_{bi_book_inf} \times$

$\rho_{sh_shelf})$

15

Query

The screenshot shows the MySQL Workbench interface. The top window is titled 'База данных: library' and contains a SQL editor with the following code:

```

1 CREATE VIEW reader_info AS SELECT r.FIRST_name, r.LAST_name, bi.book_title, l.loan_date
2 FROM reader_inf as r, book_inf bi, loan as l
3 JOIN student as s on s.student_id=l.reader_id
4 WHERE bi.book_id = l.book_id;

```

The bottom window is also titled 'База данных: library' and shows the results of the query execution:

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0023 сек.)

```

CREATE VIEW reader_info AS SELECT r.FIRST_name, r.LAST_name, bi.book_title, l.loan_date FROM reader_inf as r, book_inf bi, loan as l JOIN student as s on s.student_id=l.reader_id WHERE bi.book_id = l.book_id;

```

3. $\pi_{r_first_name, r_last_name, bi_book_title, l_loan_date}$
 $\sigma_{bi_book_id = l_book_id}$
 $(\rho_r reader_inf \times$
 $\rho_{bi} book_inf \times$
 $(\rho_l loan \bowtie s_student_id = l_reader_id \times$
 $\rho_s student))$

16

Query

Outputs the number of books that the author has written

The screenshot shows a database management system interface with two main windows.

Top Window (SQL Editor):

- Server: localhost:3307
- Database: library
- Tab: SQL
- Text area contains the following SQL query:

```
1 select row_number() over(), book_title
2 from book_inf
3 where book_id IN (SELECT ab.book_id from author_book ab, author a WHERE a.author_id = ab.author_id and a.first_name = 'Wilie' and last_name = 'Stack');
```

Bottom Window (Table Viewer):

- Server: localhost:3307
- Database: library
- Table: book_inf
- Tab: SQL
- Message: Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.
- Text area shows the same SQL query as the top window.
- Data grid shows the results:

row_number() over()	book_title
1	Intimate Enemies
2	Mars Attacks!

17

Query

WHO IS TAKE/READ THIS BOOK

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Оп...

Выполнить SQL-запрос(ы) к таблице library.book_inf: [?](#)

```

1 select FIRST_name, LAST_name, book_title, LANGUAGE
2 from book_inf b JOIN reader_inf r
3 on b.book_id = r.book_id;

```

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Оп... Триггеры

Отобразить поле запроса

⚠ Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 24 (120 всего. Запрос занял 0.0010 сек.)

```
select FIRST_name, LAST_name, book_title, LANGUAGE from book_inf b JOIN reader_inf r on b.book_id = r.book_id;
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> Показать все Количество строк: 25 Фильтровать строки: Поиск в таблице

Extra options

FIRST_name	LAST_name	book_title	LANGUAGE
Alexandrina	De Launde	Talaash	Kannada
Kinny	Smerdon	One Mile Away	Danish
Anica	Mellonby	Home Alone: The Holiday Heist	Northern Sotho
Maynard	Pleasants	Sunfish, The (Klumpfisken)	Lithuanian
Dido	Stodhart	The Butterfly Effect 2	New Zealand Sign Language
Jeniffer	Hebner	Three Way	Kyrgyz
Scarface	Ilannone	Charlie Countryman	Bulgarian
Clint	Finlaison	Nicholas on Holiday	Amharic
Mordy	Wartnaby	World According to Dick	Dzongkha
Aharon	Kerrane	Berlin Calling	Arabic
Addy	Braven	Dorian Gray	Northern Sotho
Vera	Coopland	Tom & Viv	Maltese
Michail	Jarrel	Dance of Reality, The	Sotho
Zeb	Apperley	Day the Fish Came Out, The	Maltese
Vergil	Wilberley	Line of Sight	English
Dagny	Cars	Yellow Sea, The	Afrikaans
Nimetta	Pochin	Jaws 3-D	Azeri
Hector	Pizsey	Alien Nation: Millennium	Greek
Gawith		Witches, The (Le streghe)	Hindi

Консоль

5. $\pi_{first_name, last_name, book_title, language}$
 $(\rho_b book_inf \bowtie \rho_r reader_inf)$

18

Query

WILL TAKE THE PERSON OUT OF READER

```
1 select r.FIRST_name, r.LAST_name, p.place_of_work
2 from reader_inf r, persona p
3 where r.reader_id = p.persona_id and r.reader_id in (select persona_id FROM persona);
```

SQL results:

```
select r.FIRST_name, r.LAST_name, p.place_of_work from reader_inf r, persona p where r.reader_id = p.persona_id and r.reader_id in (select persona_id FROM persona);
```

FIRST_name	LAST_name	place_of_work
Addy	Braven	Leexo
Vera	Cooplard	Skimia
Michail	Jarrel	Tazz
Zeb	Apperley	Realfire
Vergil	Wibberley	Yabox
Dagny	Cars	Twitterbeat
Ninetta	Pochin	Podcat
Hector	Pizzeyp	Bubbletube
Cesar	Gawith	Quinu
Millisent	Bolesma	Demizz

6. $\pi_{r_first_name, r_last_name, p_place_of_work}$

$\sigma_{r_reader_id = p_persona_id \text{ AND } r.reader_id \in (select persona_id FROM persona)}$

Query

Сервер: localhost:3307 » База данных: library » Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Выполнить SQL-запрос(ы) к таблице library.book_inf: ?

```

1 SELECT s.staff_name, s.gender, r.joined_date, r.FIRST_name, r.LAST_name
2 from staff s, reader_inf r
3 where s.staff_id = r.issued_by;

```

Сервер: localhost:3307 » База данных: library » Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отобразить поле запроса

⚠ Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 24 (120 всего. Запрос занял 0.0007 сек.)

```
SELECT s.staff_name, s.gender, r.joined_date, r.FIRST_name, r.LAST_name from staff s, reader_inf r where s.staff_id = r.issued_by;
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> Показать все Количество строк: 25 Фильтровать строки: Поиск в таблице

Extra options

staff_name	gender	joined_date	FIRST_name	LAST_name
Marco Channon	Male	2022-12-10	Alexandrina	De Launde
Aline Rumble	Female	2022-12-08	Kinny	Smerdon
Rahal Pringell	Female	2022-12-08	Anica	Mellonby
Gavin Grantham	Male	2022-12-02	Maynard	Pleasants
Linea Meake	Female	2022-12-06	Dido	Stodhart
Danielle Lamberts	Female	2022-12-01	Jeniffer	Hebner
Rahal Pringell	Female	2022-12-06	Scarface	Iiannone
Webster McMurrough	Male	2022-12-07	Clint	Finlaison
Linea Meake	Female	2022-12-08	Mordy	Wartnaby
Scarface Morrell	Male	2022-12-06	Aharon	Kerrane
Alison McMurphy	Female	2022-12-08	Addy	Braven
Lane Fynan	Male	2022-12-08	Vera	Coopland
Rahal Pringell	Female	2022-12-10	Michail	Jarrel
Rosene Clewlow	Female	2022-12-10	Zeb	Apperley
Anne-corinne Gedge	Female	2022-12-03	Vergil	Wibberley
Rahal Pringell	Female	2022-12-07	Dagny	Cars
Vannie Kiera	Agender	2022-12-06	Ninetta	Pochin
Anne-corinne Gedge	Female	2022-12-08	Hector	Pizsey
Anne-corinne Gedge	Male	2022-12-09	Cesar	Gavith

Консоль: nam

7. $\pi_{\underline{s \cdot staff_name}, s \cdot gender, r \cdot joined_date, r \cdot first_name, r \cdot last_name}$
 $\sigma_{s \cdot staff_id = r \cdot issued_by}$
 $(\rho_s staff \times$
 $\rho_r reader_inf)$

20

Query

Выполнить SQL-запрос(ы) к таблице library.book_inf:

```

1 SELECT reader_id, FIRST_name, LAST_name
2 from reader_inf
3 WHERE EXISTS (SELECT * from loan WHERE returned_date is NULL);SELECT * FROM `book_inf` WHERE 1

```

Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

Отображение строк 0 - 24 (120 всего, Запрос занял 0.0004 сек.)

```
SELECT reader_id, FIRST_name, LAST_name from reader_inf WHERE EXISTS (SELECT * from loan WHERE returned_date is NULL);
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

reader_id	FIRST_name	LAST_name
1	Alexandrina	De Launde
2	Kinny	Smerdon
3	Anica	Mellonby
4	Maynard	Pleasant
5	Dido	Stodhart
6	Jeniffer	Hebner
7	Scarface	Iannone
8	Clint	Finlaison
9	Mordy	Wartnaby
10	Aharon	Kerrane
11	Addy	Braven
12	Vera	Coopland
13	Michail	Jarrel
14	Zeb	Apperley
15	Vergil	Wilberley
16	Dagny	Cars
17	Ninetta	Pochin
18	Hector	Pizzey
9	Cesar	Gawith

8. $\pi_{\text{reader_id}, \text{first_name}, \text{last_name}} \text{reader_inf}(\sigma_{\text{returned_date} = \text{NULL}} \text{loan})$

21

Query

Сервер: localhost:3307 » База данных: library » Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт

Выполнить SQL-запрос(ы) к таблице library.book_inf: ⚡

```

1 SELECT re.FIRST_name, re.LAST_name
2 from reservation r, reader_inf re
3 UNION
4 SELECT re.FIRST_name, re.LAST_name
5 from loan l, reader_inf re;

```

Сервер localhost:3307 > База данных library > Таблица: reader_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отобразить поле запроса

⚠ Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 24 (120 всего, Запрос занял 0.0052 сек.)

```
SELECT re.FIRST_name, re.LAST_name from reservation r, reader_inf re UNION SELECT re.FIRST_name, re.LAST_name from loan l, reader_inf re;
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 >> Показать все Количество строк: 25 Фильтровать строки: Поиск в таблице

Extra options

FIRST_name	LAST_name
Alexandrina	De Launde
Kinny	Smerdon
Anica	Mellony
Maynard	Pleasant
Dido	Stodhart
Jeniffer	Hebner
Scarface	Iannone
Clin	Finkinson
Mordy	Wartnaby
Aharon	Kerrane
Addy	Braven
Vera	Coopland
Michail	Jarrel
Zeb	Apperley
Vergil	Wiberley
Dagny	Cars
Ninetta	Pochin
Hector	Pizzey
Gawith	

Консоль

Query

```

1. SELECT b.book_id, book_title, bookshelf, floor
2. from book_inf b join shelf s
3. on b.book_id = s.shelf_id;

```

book_id	book_title	bookshelf	floor
1	Outer Space	19	7
10	Great Gatsby, The	10	5
100	Galileo	9	3
11	Salomé	19	4
12	Alexander and the Terrible	1	8
13	Tom & Viv	4	1
14	Talaash	9	1
15	Loverboy	2	8
16	Yellow Sea, The	20	7
17	Beat Street	6	8
18	Act in Question, The (Acto en cuestión, El)	8	8
19	Jimmy Show, The	13	5
2	Tristram Shandy: A Cock and Bull Story	16	5
20	Kickboxer	16	5
21	Red Heat	12	7
22	We Need a Vacation	6	4
23	Generation, A (Pokolenie)	10	6
24	World According to Dick	5	2
	Kouznetsov, N-Laws, The	8	9

10. $\pi_{\underline{b_book_id}, \underline{book_title}, \underline{bookshelf}, \underline{floor}}$
 $(\rho_{\underline{b_book_inf}} \bowtie \rho_{\underline{b_book_id} = \underline{s_shelf_id}} \rho_{\underline{s_shelf}})$

23

Query

```

1 select a.first_name, a.last_name, COUNT(ab.book_id) as quantity
2 FROM author a, author_book ab
3 WHERE a.author_id = ab.author_id
4 GROUP BY ab.author_id
5 ORDER by quantity;

```

first_name	last_name	quantity
Millie	Creser	1
Berny	Woolard	1
Beannard	Ticcombe	1
Charlotte	Pughe	1
Findlay	Arpin	1
Alphonso	Batterham	1
Giacinta	Vassilachev	1
Lothaire	Tissell	1
Estevan	Rabl	1
Nelle	Accombe	1
Lawton	Rigmond	1
Dore	Isabell	1
Roberto	Scawton	1
Tim	Buffin	1
Had	Dulinity	1
Olwen	Minty	1
Emlynne	Leinlecker	1
Danette	Perry	1
Lynskey		1

11. $\tau_{quantity}$ $\pi_{\underline{a.first_name}, \underline{a.last_name}, COUNT(book_id)} \rightarrow quantity$ $\gamma_{\underline{author_id}, COUNT(book_id)}$ $\sigma_{\underline{a.author_id} = \underline{ab.author_id}}$ $(\rho_{\underline{a}author} \times$ $\rho_{\underline{ab}author_book})$

24

Query

Сервер: localhost3307 » База данных: library » Таблица: student

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггер

Выполнить SQL-запрос(ы) к таблице library.student:

```
1 select row_number() over(),r.first_name, r.last_name, f.reader_id, sum(fine_amount)
2 from reader_inf r, fine f
3 where f.reader_id = r.reader_id
4 GROUP by f.reader_id;
```

Сервер: localhost 3307 » База данных: library » Таблица: student

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отобразить поле запроса

⚠ Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 6 (7 всего. Запрос занял 0.0008 сек.)

```
select row_number() over(),r.first_name, r.last_name, f.reader_id, sum(fine_amount) from reader_inf r, fine f where f.reader_id = r.reader_id GROUP by f.reader_id;
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

Extra options

row_number() over()	first_name	last_name	reader_id	sum(fine_amount)
6	Maynard	Pleasants	4	10000
7	Jeniffer	Hebner	6	15000
1	Clint	Finlaison	8	5000
2	Addy	Braven	11	5000
3	Michail	Jarrel	13	5000
4	Zeb	Apperley	14	5000
5	Vergil	Wibberley	15	15000

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

Использование результатов запроса

Печать В буфер обмена Экспорт Отобразить график Создать представление

14. $\pi_{\text{row_number}, \text{over}, \underline{r}.\text{first_name}, \underline{r}.\text{last_name}, \underline{f}.\text{reader_id}, \text{SUM}(\text{fine_amount})}$

$\gamma_{\underline{f}.\text{reader_id}, \text{SUM}(\text{fine_amount})}$

$\sigma_{\underline{f}.\text{reader_id} = \underline{r}.\text{reader_id}}$

$(\rho_{\underline{r}.\text{reader_inf}} \times$

$\rho_{\underline{f}.\text{fine}})$

25

Query

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with tabs: Структура (Structure), SQL, Поиск (Search), Запрос по шаблону (Template Query), Экспорт (Export), Импорт (Import), and Операции (Operations). The SQL tab is active.

In the main area, there's a text input field labeled "Выполнить SQL-запрос(ы) к базе данных library: ?". Below it is the SQL query:

```
1 Select r.first_name, r.last_name, b.book_title
2 from reader_inf r, book_inf b
3 where exists
4 (select * from loan where reader_id = r.reader_id and
5 book_id=b.book_id and book_id =
6 any(select book_id from author_book where author_id=95))
```

Below the query, the results pane shows the output of the query:

first_name	last_name	book_title
Maynard	Pleasants	Sunfish, The (Klumpfisken)

At the bottom of the results pane, there are buttons for Показать все (Show All), Количество строк (Number of rows), and Фильтровать строки (Filter rows). There's also a search bar for Поиск в таблице (Search in table).

26

Query

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the server is set to 'localhost:3307' and the database is 'library'. The main window has tabs for 'Структура' (Structure), 'SQL', 'Поиск' (Search), 'Запрос по шаблону' (Template Query), 'Экспорт' (Export), 'Импорт' (Import), and 'Операции' (Operations). The 'SQL' tab is selected, displaying the following SQL code:

```

1 DELIMITER $$;
2 CREATE TRIGGER push_fine AFTER UPDATE ON loan
3 FOR EACH ROW BEGIN
4     if datediff(now(),new.returned_date) > 7 then
5         INSERT into fine (reader_id, loan_id, fine_date, fine_amount)
6             VALUES (new.reader_id, new.loan_id, new.returned_date, 5000);
7     END if;
8 END $$;
9 Delimiter ;;

```

Below the code, a message from MySQL states: "MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.0051 сек.)". At the bottom, there are buttons for 'Построчное редактирование' (Row-based editing), 'Изменить' (Edit), and 'Создать PHP-код' (Create PHP code).

The screenshot shows the MySQL Workbench interface with the 'fine' table selected. The top navigation bar includes 'Обзор' (Overview), 'Структура' (Structure), 'SQL', 'Поиск' (Search), 'Вставить' (Insert), 'Экспорт' (Export), 'Импорт' (Import), 'Привилегии' (Privileges), 'Операции' (Operations), and 'Триггеры' (Triggers). The 'SQL' tab is selected, showing the query: 'SELECT * FROM `fine`'. The results pane displays the following data:

fine_id	reader_id	loan_id	fine_date	fine_amount
2	15	2	2022-12-11	5000
3	6	5	2022-12-11	5000
4	4	1	2022-12-12	5000
5	15	2	2022-12-12	5000
6	6	5	2022-12-12	5000
7	4	1	2022-12-12	5000
8	15	2	2022-12-12	5000
9	6	5	2022-12-12	5000
10	8	9	2022-12-12	5000
11	11	14	2022-12-12	5000
12	13	18	2022-12-12	5000
13	14	20	2022-12-12	5000

Query

The screenshot shows two instances of the MySQL Workbench interface. The top window is titled 'Сервер: localhost:3307 » База данных: library' and has a tab bar with 'Структура', 'SQL', 'Поиск', 'Запрос по шаблону', 'Экспорт', 'Импорт', 'Операции', and 'Пр'. It contains a text area with the following SQL code:

```
1 DELIMITER $$  
2 CREATE TRIGGER copies_minus AFTER INSERT ON reader_inf  
3 FOR EACH ROW BEGIN  
4 UPDATE book_inf set copies = copies - 1  
5 where new.book_id = book_inf.book_id;  
6 END $$  
7 DELIMITER ;;
```

The bottom window is also titled 'Сервер: localhost:3307 » База данных: library' and has a more extensive tab bar including 'Привилегии', 'Процедуры', 'События', 'Тrigгеры', and 'Дизайнер'. It displays a message: 'MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0054 сек.)'. Below this, the same SQL code is shown again, followed by '[Построчное редактирование] [Изменить] [Создать PHP-код]'.

Query

The screenshot shows two instances of the MySQL Workbench interface. Both windows have the title 'Сервер: localhost:3307 » База данных: library'. The top window is in 'SQL' mode, displaying the following SQL code:

```
1 DELIMITER $$  
2 CREATE TRIGGER copies_plus AFTER DELETE ON reader_inf  
3 FOR EACH ROW BEGIN  
4 UPDATE book_inf set copies = copies + 1  
5 where old.book_id = book_inf.book_id;  
6 INSERT into archive  
7 select old.first_name,old.last_name,b.book_title,now()  
8 from book_inf b  
9 where old.book_id = b.book_id;  
10 END $$  
11 Delimiter ;;  
12
```

Below the code are buttons for 'Очистить' (Clear), 'Формат' (Format), and 'Получить автосохранённый запрос' (Get Autosaved Query). The bottom window is also in 'SQL' mode, showing the same trigger creation code. A green message bar at the top indicates: 'MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0035 сек.)'. Below the message, the trigger definition is repeated, followed by a row of buttons: '[Построчное редактирование]' (Line-by-line editing), '[Изменить]' (Edit), and '[Создать PHP-код]' (Create PHP code).

29

Query

The screenshot shows two windows of the MySQL Workbench application. Both windows have the title bar "Сервер: localhost:3307 » База данных: library". The top window displays the following SQL code:

```
1 DELIMITER $$  
2 CREATE TRIGGER pay_fine AFTER INSERT ON fine_payment  
3 FOR EACH ROW BEGIN  
4 DELETE FROM fine  
5 where new.reader_id = reader_id;  
6 END $$  
7 Delimiter ;;
```

The bottom window shows the result of the execution:

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0057 сек.)

```
CREATE TRIGGER pay_fine AFTER INSERT ON fine_payment FOR EACH ROW BEGIN DELETE FROM fine where new.reader_id = reader_id; END;
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

30

Query

DISPLAYS BOOKS FROM THE ARCHIVE THAT WERE DELIVERED AFTER THIS DATE

The screenshot shows a database management interface with two main panes. The top pane is a SQL editor window titled 'Выполнить SQL-запрос(ы) к таблице library.archive:' containing the following SQL code:

```
1 SELECT first_name, last_name from archive where joined_date > 2022-08-02;
```

The bottom pane displays the results of the query in a table format. The table has two columns: 'first_name' and 'last_name'. The data is as follows:

first_name	last_name
Candida	Alleburton
Carlyle	MacConnell
Joella	Baglow
Conway	Epple
Jedidiah	Flaubert
Cassie	Tummasutti
Korie	Golland
Kali	Bastian
Kai	Cordle
Bruno	Thatcham
Birgit	Burns
Kimmy	Lindholm
Emmet	Grisley
Laney	Fulmen
Cosmo	MacKenzie
Roseline	Rillstone
Zsazsa	Neave
Barbette	Picheford
	Kneafsey

16. $\pi_{\text{first_name}, \text{last_name}}$
 $\sigma_{\text{joined_date} > 2022 - 8 - 2} \text{archive}$

31

Query
Displays the book with the largest copy

```

1 SELECT * FROM book_inf where copies =
2 (Select max(copies) from book_inf)

SELECT * FROM book_inf where copies = (Select max(copies) from book_inf);

book_id  book_title  LANGUAGE  shelf_id  copies
      21  Red Heat    Albanian     27        299
      86  Pumzi       Hindi       55        299
  
```

12. $\sigma_{copies = \underline{(\pi_{MAX(copies)} book_inf)}}$

book inf

32

Query

List the number of boys and girls among the staff.

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Обзор (Preview), Структура (Structure), SQL, Поиск (Search), Вставить (Insert), Экспорт (Export).
- Query Editor:** Title: Выполнить SQL-запрос(ы) к таблице library.staff: ⓘ
Content:

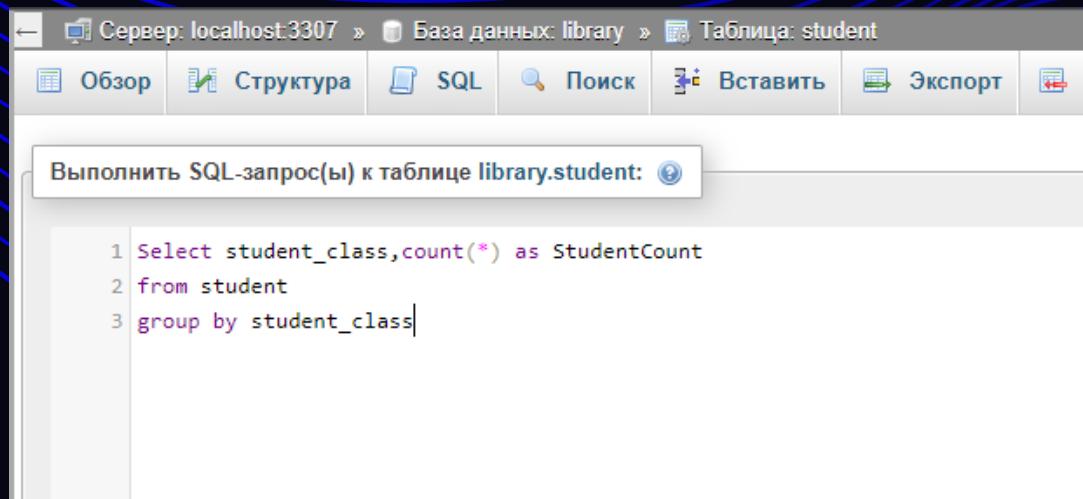
```
1 Select gender,count(*) as StaffCount
2 from staff
3 group by gender|
```
- Message Bar:** Отобразить поле запроса (Show Query Field). Warning: Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и у
- Result Area:** Confirmation: Отображение строк 0 - 4 (5 всего, Запрос занял 0,0004 сек.)
SQL query: `Select gender,count(*) as StaffCount from staff group by gender;`
Buttons: Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]
Options: Показать все (Show All), Количество строк: 25, Фильтровать строки: Поиск в таблице (Search in Table).
- Table Results:** A table titled "Extra options" showing the count of staff by gender.

gender	StaffCount
Agender	1
Bigender	1
Female	9
Genderfluid	2
Male	7

33

Query

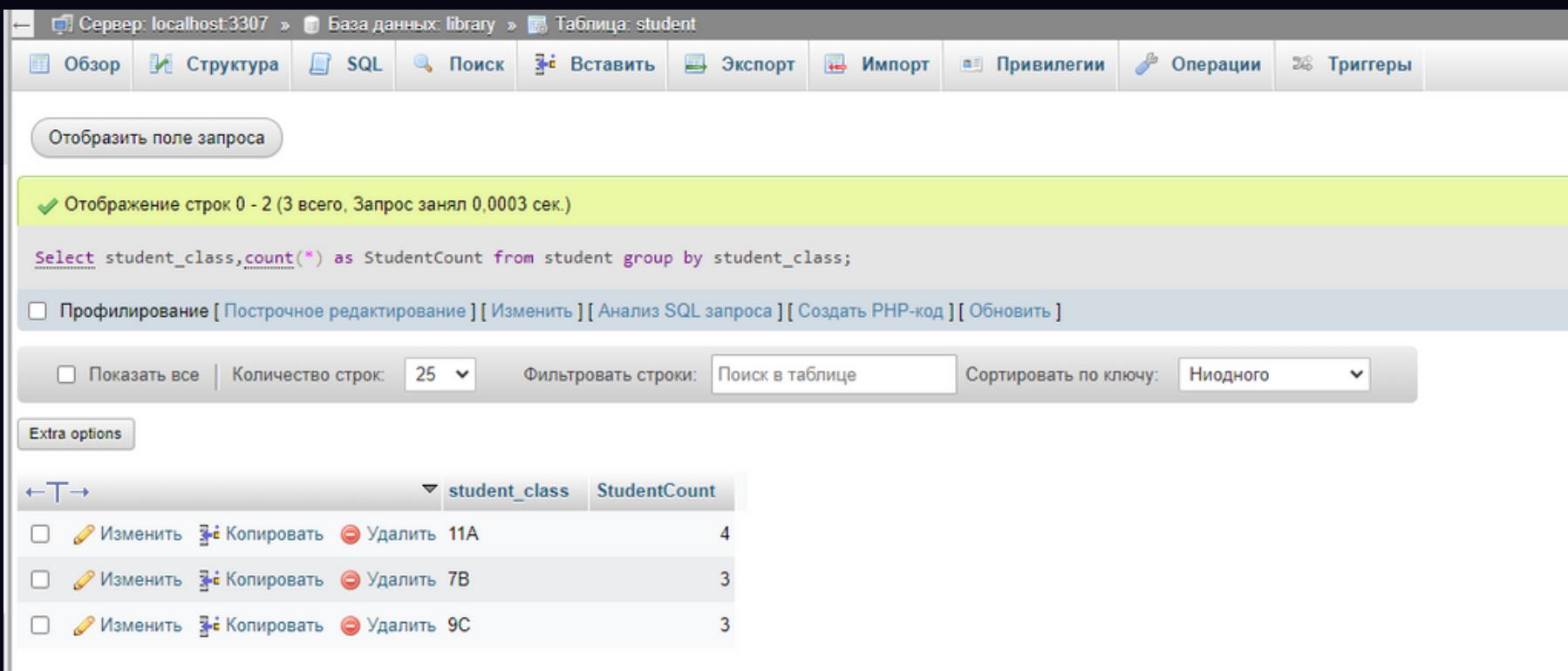
Displays the names of classes and the number of students in each class



The screenshot shows a database interface with a navigation bar at the top. The SQL tab is selected. Below it, a text input field contains the following SQL code:

```
1 Select student_class, count(*) as StudentCount
2 from student
3 group by student_class
```

21. $\pi_{student_class, COUNT(*)} \rightarrow_{studentcount}$
 $\gamma_{student_class, COUNT(*)} student$



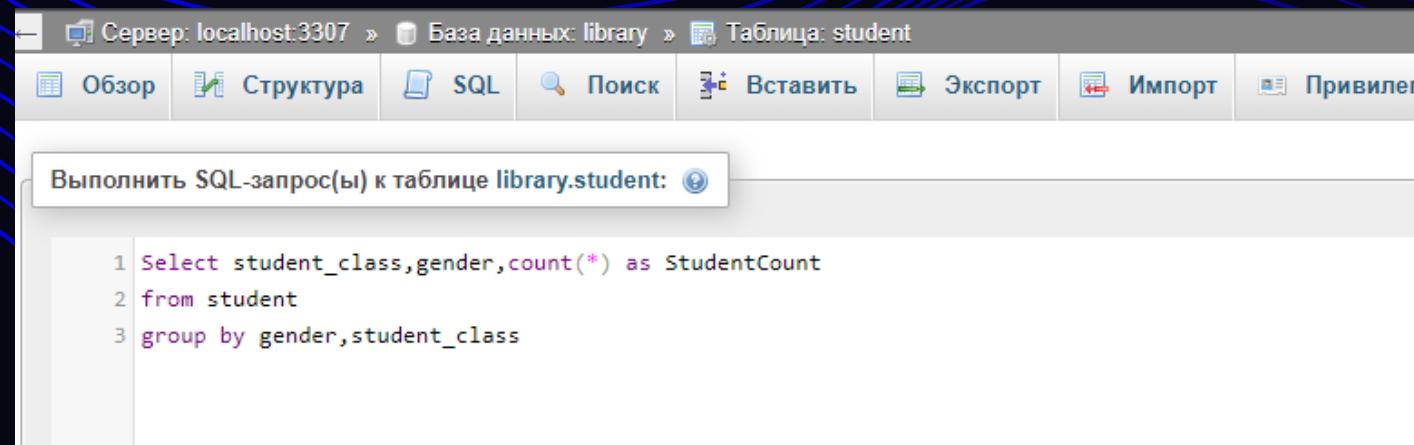
The screenshot shows the results of the executed SQL query. The results are displayed in a table with two columns: `student_class` and `StudentCount`. The data is as follows:

student_class	StudentCount
11A	4
7B	3
9C	3

34

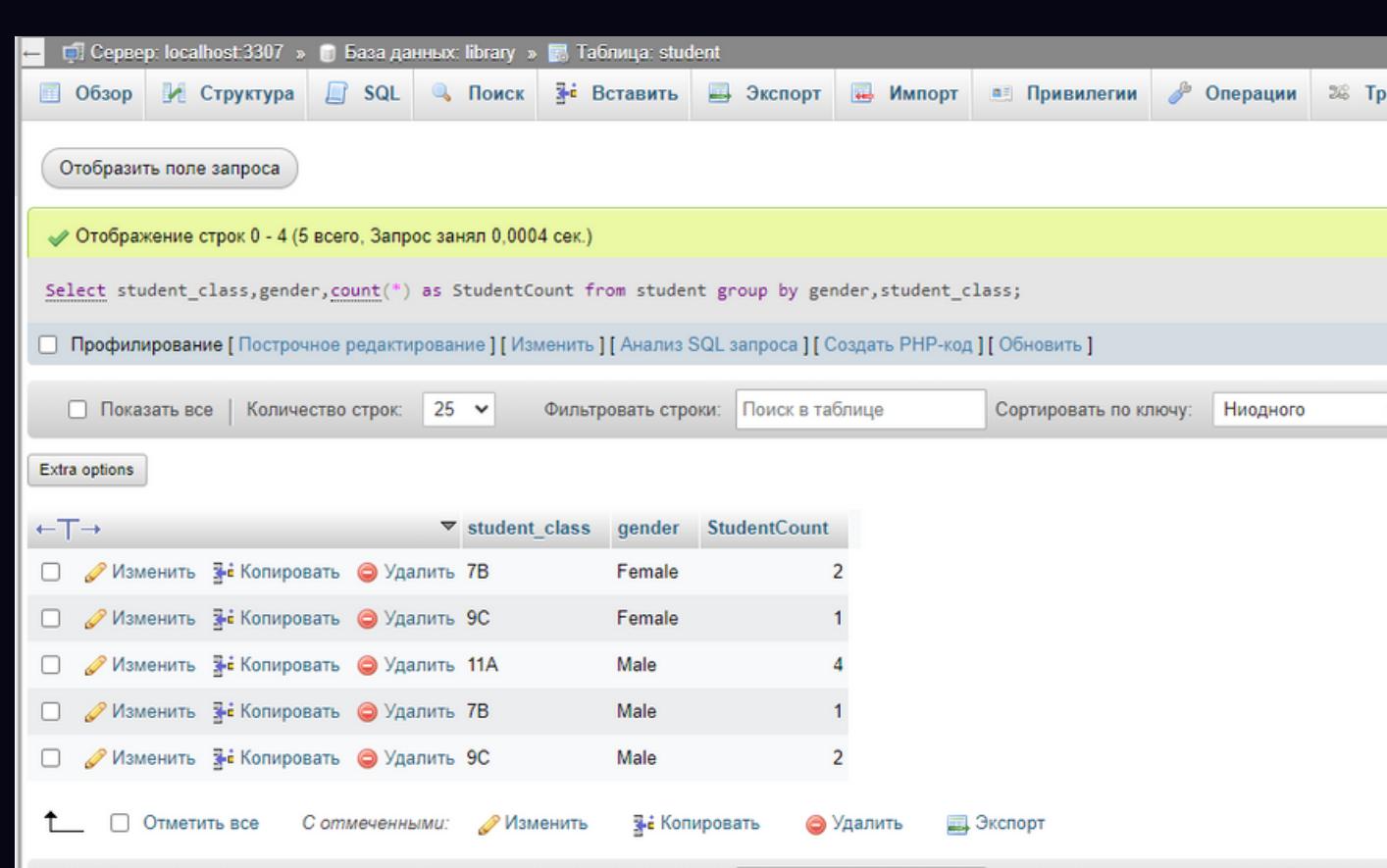
Query

Displays the number of boys and girls in each class.



Выполнить SQL-запрос(ы) к таблице library.student:

```
1 Select student_class,gender,count(*) as StudentCount
2 from student
3 group by gender,student_class
```



Отображение строк 0 - 4 (5 всего, Запрос занял 0,0004 сек.)

```
Select student_class,gender,count(*) as StudentCount from student group by gender,student_class;
```

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Н�одного

student_class	gender	StudentCount
7B	Female	2
9C	Female	1
11A	Male	4
7B	Male	1
9C	Male	2

Изменить | Копировать | Удалить | Изменить | Копировать | Удалить | Экспорт

22. $\pi_{\text{student_class}, \text{gender}, \text{COUNT}(*)} \rightarrow \text{studentcount}$
 $\gamma_{\text{gender}, \text{student_class}, \text{COUNT}(*)} \text{student}$

35

Transaction

The screenshot shows two instances of MySQL Workbench. The left instance displays a SQL editor with the following code:

```
1 START TRANSACTION;
2 SELECT * FROM author;
3 insert into fine select row_number() over(), reader_id, loan_id, now(), 5000 from loan where datediff(now(), loan_date) > 7;
4 COMMIT;
```

The right instance shows the results of the query in the SQL tab, which returns an empty result set as indicated by the status message:

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0001 сек.)

The results grid shows the following data:

author_id	first_name	last_name
34	Adrien	Ogden
24	Alphonso	Batterham
26	Andromache	Tue
98	Annadiane	Cranston
97	Any	Georgius
50	Arleta	Beavington
37	Bea	Harget
6	Beamard	Titcombe
86	Berny	Woolard
66	Blaire	Fibbings
81	Bradan	Kasper
82	Brynn	Lovell
80	Brynna	Justice
96	Calida	Yurin

36

Transaction

The screenshot shows two windows of MySQL Workbench. The top window displays a SQL editor with the following transaction script:

```
1 START TRANSACTION;
2   DELETE FROM reader_inf WHERE reader_id = 100;
3   DELETE FROM persona WHERE persona_id = 100;
4 COMMIT;
```

The bottom window shows the execution results of the transaction:

- MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.0003 сек.)
- START TRANSACTION;
- [Построчное редактирование] [Изменить] [Создать PHP-код]
- Затронута 1 строка. (Запрос занял 0,0020 сек.)
- DELETE FROM reader_inf WHERE reader_id = 100;
- [Построчное редактирование] [Изменить] [Создать PHP-код]
- Затронуто 0 строк. (Запрос занял 0,0002 сек.)
- DELETE FROM persona WHERE persona_id = 100;
- [Построчное редактирование] [Изменить] [Создать PHP-код]
- MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.0012 сек.)
- COMMIT;
- [Построчное редактирование] [Изменить] [Создать PHP-код]

37

Transaction

The screenshot shows the MySQL Workbench interface with the title bar "Сервер: localhost:3307 > База данных: library > Таблица: persona". The SQL tab is selected. A modal window titled "Выполнить SQL-запрос(ы) к таблице library.persona:" contains the following SQL code:

```
1 START TRANSACTION;
2     INSERT INTO fine_payment VALUES(18, 150, 2022-10-10, 5000);
3 COMMIT;
```

The screenshot shows the MySQL Workbench interface with the same title bar. The SQL tab has a message "Отобразить поле запроса". Below it, the results of the transaction are displayed in a scrollable list:

- MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.0002 сек.)
- START TRANSACTION;
- [Построчное редактирование] [Изменить] [Создать PHP-код]
- Добавлена 1 строка. (Запрос занял 0.0004 сек.)
- INSERT INTO fine_payment VALUES(18, 150, 2022-10-10, 5000);
- [Построчное редактирование] [Изменить] [Создать PHP-код]
- Warning: #1265 Data truncated for column 'payment_date' at row 1
- MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.0024 сек.)
- COMMIT;
- [Построчное редактирование] [Изменить] [Создать PHP-код]

38

INDEX

Сервер: localhost:3307 > База данных: library > Таблица: author

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Тrigгеры

Отобразить поле запроса
Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 24 (100 всего. Запрос занял 0.0003 сек.)

```
SELECT first_name, last_name FROM author;
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> Показать все Количество строк: 25 Фильтровать строки: Поиск в таблице Сортировать по ключу: Нисходящего

Extra options

first_name	last_name
Adrien	Ogden
Alphonso	Batterham
Andromache	Tue
Annadiane	Cranston
Any	Georgius
Arleta	Beavington
Bea	Harget
Bearnard	Titcombe
Berny	Woolard
Blaire	Fibbins
Bradan	Kasper
Brynn	Lovell
Brywna	Justice
Calida	Yurin
Camey	McNaughton
Catie	Casari
Charlotte	Pughe
Chrissie	Eccleston
Concordia	Archanbault

Консоль

Сервер: localhost:3307 > База данных: library > Таблица: fine_payment

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Тrigгеры

Отобразить поле запроса
Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

SQL-запрос успешно выполнен.

```
SHOW INDEX FROM author;
```

Профилирование [Построчное редактирование] [Изменить] [Создать PHP-код] [Обновить]

Extra options

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
author	1	AUTHORS	1	first_name	A	100	NULL NULL	YES	BTREE			
author	1	AUTHORS	2	last_name	A	100	NULL NULL	YES	BTREE			

Использование результатов запроса

Печать В буфер обмена Создать представление

Сервер: localhost:3307 > База данных: library > Таблица: author

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии

Выполнить SQL-запрос(ы) к таблице library.author:

```
1 CREATE INDEX AUTHORS ON AUTHOR(FIRST_NAME, LAST_NAME);
```

Сервер: localhost:3307 > База данных: library > Таблица: author

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Тrigгеры

Отобразить поле запроса

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0102 сек.)

```
CREATE INDEX AUTHORS ON AUTHOR(FIRST_NAME, LAST_NAME);
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

39

INDEX

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Тrigгеры

Отобразить поле запроса

Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 24 (100 всего, Запрос занял 0,0003 сек.)

SELECT book_title FROM book_inf;

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> Показать все Количество строк: 25 Фильтровать строки: Поиск в таблице Сортировать по ключу: Ниодного

Extra options

book_title
3 dev adam (Three Giant Men)
47 Ronin
6954 Kilometria Kotlin
A Bright Shining Lie
Absence of Malice
Acid House, The
Act in Question, The (Acto en cuestión, El)
Alexander and the Terrible
Alien Nation: Millennium
Anatomy of a Murder
Apartment, The
Beat Street
Beowulf
Berlin Calling
Blank Check
BorderTown
Breach, The (Rupture, La)
Charlie Countryman
Watching Us

Консоль Watching Us

Сервер: localhost:3307 > База данных: library > Таблица: fine_payment

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Тrigгеры

Отобразить поле запроса

Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

SQL-запрос успешно выполнен.

SHOW INDEX FROM book_inf;

Профилирование [Построчное редактирование] [Изменить] [Создать PHP-код] [Обновить]

Extra options

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
book_inf	1	TITLES	1	book_title	A	100	NULL NULL	BTREE				
book_inf	1	all_copies	1	copies	A	100	NULL NULL	YES	BTREE			

Использование результатов запроса

Печать В буфер обмена Создать представление

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии

Выполнить SQL-запрос(ы) к таблице library.book_inf:

```
1 CREATE INDEX TITLES ON BOOK_INF(BOOK_TITLE);
```

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Тrigгеры

Отобразить поле запроса

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0079 сек.)

```
CREATE INDEX TITLES ON BOOK_INF(BOOK_TITLE);
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

40

INDEX

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отобразить поле запроса

Данное выделение не содержит уникального столбца. Изменение сетки, выставление галочки, редактирование, копирование и удаление невозможно.

✓ Отображение строк 0 - 24 (100 всего. Запрос занял 0.00003 сек.)

`SELECT copies from book_inf;`

Просмотреть [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> Показать все Количество строк: 25 Фильтровать строки: Поиск в таблице Сортировать по ключу: Ниодного

copies

254
170
284
266
175
92
219
199
144
158
155
56
149
104
38
193
106
104

Консоль

Добавить 1 поле(я) после copies Вперед

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
Изменить Rename Удалить	TITLES	BTREE	Нет	Нет	book_title	100	А	Нет	
Изменить Rename Удалить	all_copies	BTREE	Нет	Нет	copies	100	А	Да	

Создать индекс для 1 столбцов Вперед

Сервер: localhost:3307 > База данных: library > Таблица: book_inf

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отобразить поле запроса

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0065 сек.)

`CREATE INDEX all_copies ON book_inf(copies);`

[Построчное редактирование] [Изменить] [Создать PHP-код]

Merry
Christmas

ERD

<https://excalidraw.com/#json=2mPYEcztFFIHBUVz86ca,WhApuetf1KHb-TH5uGvQMA>

Merry
Christmas!

GITHUB

<https://github.com/aarystanovv/DBMS-PROJECT.git>



Merry Christmas
& A Happy New Year!

