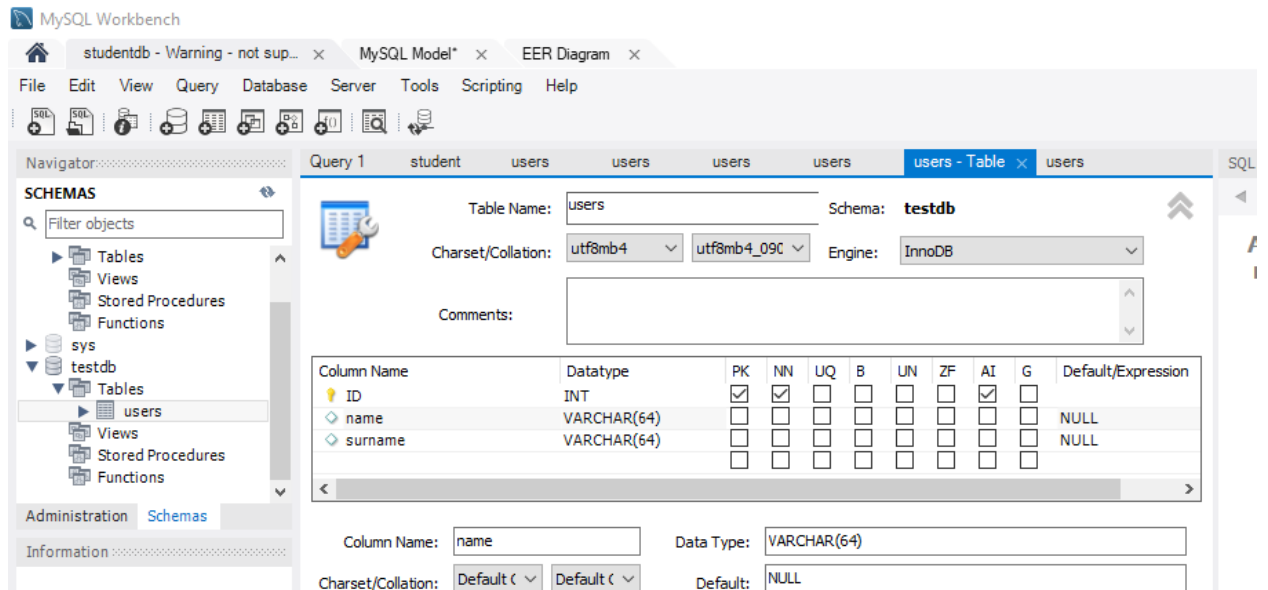
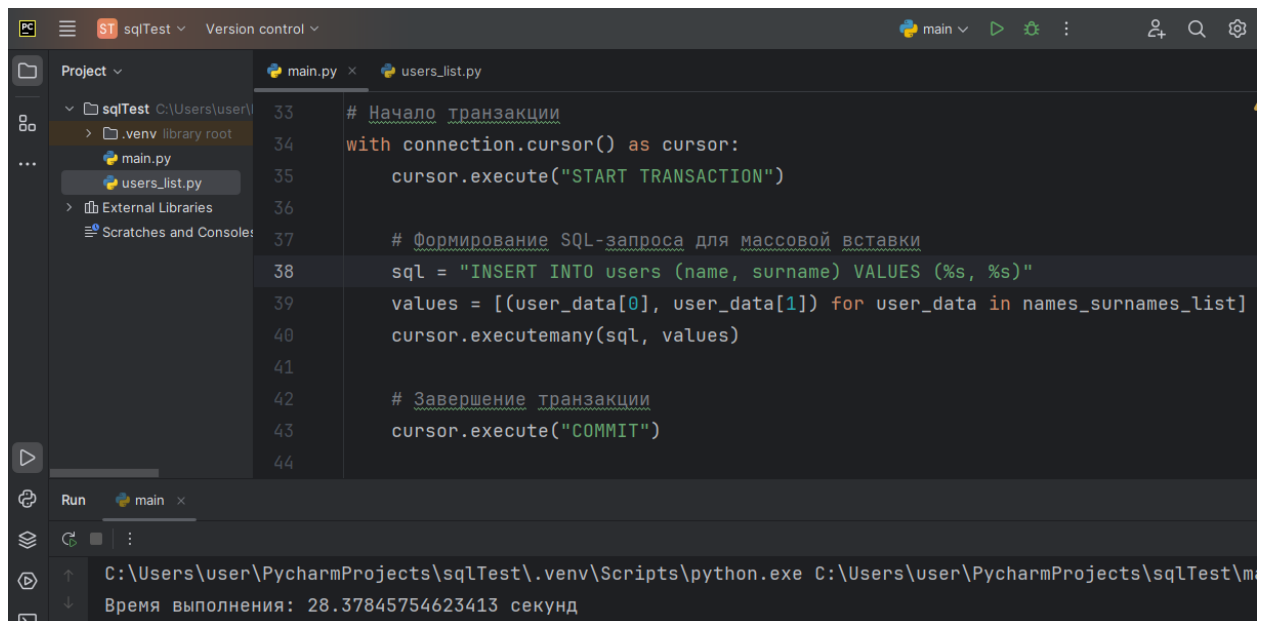


Пример создания индексов в MySQL

Создадим БД с одной таблицей:



Используем скрипт на Питоне, чтобы вставить несколько миллионов строк:



Проверяем таблицу, 3 млн. строк:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'testdb' database with a 'users' table. The central SQL editor contains the following queries:

```
1 • USE testdb;
2 • SELECT COUNT(*) FROM users;
3 • SELECT * FROM users LIMIT 5;
```

The 'Result Grid' at the bottom shows the result of the second query:

COUNT(*)
3000000

The 'Information' pane on the left provides details about the 'users' table:

Table: users

Columns:

- ID int AI PK
- name varchar(64)
- surname varchar(64)

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'testdb' database with a 'users' table. The central SQL editor contains the following queries:

```
1 • USE testdb;
2 • SELECT COUNT(*) FROM users;
3 • SELECT * FROM users LIMIT 10;
```

The 'Result Grid' at the bottom shows the result of the third query:

ID	name	surname
1	Устин	Юльевич
2	Маргарита	Семеновна
3	Валерия	Афанасьевна
4	Аникей	Егорович
5	г-н	Рожков
6	Софон	Борисович
7	Потапова	Зоя
8	Степанова	Мария
9	Алла	Захаровна
10	Белоусов	Валерий
NULL	NULL	NULL

The 'Information' pane on the left provides details about the 'users' table:

Table: users

Columns:

- ID int AI PK
- name varchar(64)
- surname varchar(64)

Теперь выполним запросы на поиск случайного юзера по ID и по имени:

```
поиск по ключу 251361
user_name - Игоревна
Время выполнения: 0.0019989013671875 секунд
поиск по имени
Время выполнения: 2.626962661743164 секунд
```

Индексы.

Посмотрим какие индексы существуют в таблице:

The screenshot shows a database client interface with a SQL editor and a results pane. The SQL editor contains the following queries:

- 1 • `USE testdb;`
- 2 • `SELECT COUNT(*) FROM users;`
- 3 • `SELECT * FROM users LIMIT 10;`
- 4 • `SHOW INDEX FROM users;`

The results pane displays the output of the fourth query, showing the index information for the 'users' table:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Ir
users	0	PRIMARY	1	ID	A	299369				BTREE		

Видим, что проиндексирован первичный ключ, это объясняет небольшое время поиска.

Создадим индекс на поле "name":

The screenshot shows a database client interface with a SQL editor. The SQL editor contains the following queries:

- 1 • `USE testdb;`
- 2 • `SELECT COUNT(*) FROM users;`
- 3 • `SELECT * FROM users LIMIT 10;`
- 4 • `SHOW INDEX FROM users;`
- 5 • `CREATE INDEX index_name ON users (name);`

Процесс занимает некоторое время,

The screenshot shows the 'Output' pane of the database client, displaying the execution of the 'CREATE INDEX' query. The output is as follows:

#	Time	Action	Message
1	16:15:20	SHOW INDEX FROM users	1 row(s) returned
4	16:15:25	CREATE INDEX index_name ON users (name)	Running...

затем в строке состояния должно появиться:

Action Output				
#	Time	Action	Message	
✓ 27	15:56:33	SELECT * FROM users LIMIT 10	10 row(s) returned	
✓ 28	16:05:31	SHOW INDEX FROM users	1 row(s) returned	
✓ 29	16:11:00	CREATE INDEX index_name ON users (name)	0 row(s) affected Records: 0 Duplicates: 0 Warning	

Проверим, как происходит поиск случайного пользователя после индексации:

```
C:\Users\user\PycharmProjects\sqlTest\.venv\Scripts\python.exe C:\
поиск по ключу 446621
user_name - Яковлевич
Время выполнения: 0.0009999275207519531 секунд
поиск по имени
Время выполнения: 0.000997781753540039 секунд
```

Разница очевидна.

Настройка логирования.

Чтобы отслеживать «длинные» или медленные запросы, необходимо настроить логирование.

В Linux обычно конфигурационный файл MySQL находится в каталоге /etc/mysql/my.cnf или /etc/mysql/mysql.conf.

Найдите или добавьте следующие строки:

slow_query_log = 1

long_query_time = 2

slow_query_log_file = /var/log/mysql/slow.log

В Windows нужно отредактировать конф. файл C:\ProgramData\MySQL\MySQL Server 8.0\my.ini (нужны права администратора)

```
119
120 slow-query-log=1
121
122 slow_query_log_file="WIN-1UR16D02E19-slow.log"
123
124 long_query_time=2
125
126 # Error Logging.
127 log_error="WIN-1UR16D02E19_err"
```

long_query_time определяет порог времени выполнения запроса в секундах.

После этого нужно перезапустить MySQL-server:

- Открыть окно "Выполнить" (Win+R).
- Ввести команду: services.msc и нажать Enter.
- Найти службу MySQL в списке служб.
- Щелкнуть правой кнопкой мыши по службе MySQL и выбрать "Перезапустить".

Теперь все «длинные» запросы будут отображаться в лог-файле, например:

```
Time          Id Command      Argument
# Time: 2024-09-14T13:17:56.562972Z
# User@Host: root[root] @ localhost [127.0.0.1] Id:      9
# Query_time: 2.558427 Lock_time: 0.000002 Rows_sent: 0  Rows_examined: 3000000
use testdb;
SET timestamp=1726319874;
SELECT * FROM `users` WHERE name = 'Харитоновна';
```

Команда EXPLAIN в MySQL

Скопируем запрос из лога и вставим в EXPLAIN

7 • EXPLAIN SELECT * FROM `users` WHERE name = 'Харитоновна';

8

9

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	users	NULL	ALL	NULL	NULL	NULL	NULL	2988448	10.00	Using where

Пояснение к столбцам таблицы:

id: Идентификатор сегмента запроса.

select_type: Тип сегмента (SIMPLE, UNION, SUBQUERY и т.д.).

table: Имя таблицы, к которой применяется операция.

type: Тип соединения (system, const, eq_ref, ref, range, index_merge, unique_subquery, index_subquery, fulltext, ref_or_null, index_merge, random).

possible_keys: Список индексов, которые могут быть использованы для выполнения запроса.

key: Индекс, который фактически используется.

key_len: Число байтов, используемых из ключа.

ref: Значение, используемое для поиска строки в индексе.

rows: Приблизительное количество строк, которые должны быть прочитаны из таблицы.

Extra: Дополнительная информация, такая как использование временных таблиц, файлов сортировки и т.

В примере выше видно, что индекс отсутствует, приблизительно потребуется перебрать 3 млн строк.

Тот же запрос, но после добавления индекса (внимание на поля rows и possible_keys):

```
7 • EXPLAIN SELECT * FROM `users` WHERE name = 'Харитоновна';
8
9
```


<



Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	users	NULL	ref	index_name	index_name	259	const	1	100.00	NULL


Индексы для нескольких таблиц

Добавим еще одну таблицу, с названиями городов пользователей:

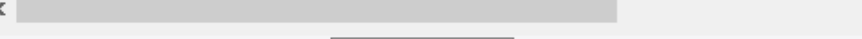
	Table Name: <input type="text" value="cities"/>	Schema: testdb
	Charset/Collation: <input type="text" value="Default Charset"/> <input type="text" value="Default Collation"/>	Engine: <input type="text" value="InnoDB"/>
	Comments: <input type="text"/>	




Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expressi
 idcities	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 city	VARCHAR(128)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

С помощью скрипта на Python добавим несколько городов:




```
1 • use testdb;
2 • SELECT * FROM cities;
3 • SELECT COUNT(*) FROM cities;
```







Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	COUNT(*)
▶	35746

Теперь добавим поле city_id (внешний ключ) в таблицу users (перед этим очистив ее):

	Table Name: <input type="text" value="users"/>	Schema: testdb
	Charset/Collation: <input type="text" value="utf8mb4"/> <input type="text" value="utf8mb4_0900_ai_ci"/>	Engine: <input type="text" value="InnoDB"/>
	Comments: <input type="text"/>	

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expressi
 ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 name	VARCHAR(64)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 surname	VARCHAR(64)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 city_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

И вставим несколько строк, используем Питон:

testdb* x Administration - Data Export users - Table

Limit to 1000 rows

```

1 use testdb;
2 • SELECT COUNT(*) FROM users;
3 • SELECT * FROM users LIMIT 10;

```

Result Grid

	COUNT(*)
▶	3000000

Теперь обе таблицы заполнены, посчитаем, сколько пользователей живет в каждом городе:

Limit to 1000 rows

```

2 • SELECT COUNT(*) FROM users;
3 • SELECT * FROM users LIMIT 10;
4 • SELECT c.city, COUNT(u.id) as user_count
5 FROM cities c
6 LEFT JOIN users u ON c.idcity = u.city_id
7 GROUP BY c.city;

```

Result Grid

	city	user_count
▶	Toddborough	163
	Lake Denis	165
	New Christopher	151
	Port Janice	188
	South Rachaelmouth	169
	South Ronald	165
	South Frances	163
	Daviesmouth	176

Запрос выполняется довольно долго:

```

Time                Id Command      Argument
# Time: 2024-09-14T16:59:33.097282Z
# User@Host: root[root] @ localhost [127.0.0.1] Id:      8
# Query_time: 5.805051  Lock_time: 0.000033 Rows_sent: 1000  Rows_examined: 3035746
use testdb;
SET timestamp=1726333167;
SELECT c.city, COUNT(u.id) as user_count
FROM cities c
LEFT JOIN users u ON c.idcity = u.city_id
GROUP BY c.city
LIMIT 0, 1000;

```

Проверим запрос в EXPLAIN:

The screenshot shows a database IDE with a toolbar at the top. Below the toolbar, there are four SQL queries listed:

```
2 • SELECT COUNT(*) FROM users;  
3 • SELECT * FROM users LIMIT 10;  
4 • EXPLAIN SELECT c.city, COUNT(u.id) as user_count  
5 FROM cities c  
6 LEFT JOIN users u ON c.idcity = u.city_id  
7 GROUP BY c.city;
```

Below the queries is a "Result Grid" section. It has a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The grid contains the following data:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	c	<small>NULL</small>	ALL	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	<small>NULL</small>	35652	100.00	Using temporary
	1	SIMPLE	u	<small>NULL</small>	ref	<small>idx</small>	<small>idx</small>	4	testdb.c.idcity	83	100.00	Using index

Чтобы ускорить выполнение запроса, нужно выяснить какие столбцы участвуют в запросе:

- cities.idcity: Это первичный ключ таблицы cities и уже проиндексирован, т.к. это первичный ключ.
- users.city_id: Этот столбец используется в условии соединения JOIN и должен быть проиндексирован для ускорения поиска соответствующих записей в таблице users.
- cities.city: Поскольку мы группируем результаты по столбцу city, индекс на этом столбце поможет ускорить группировку данных.

Итого, нужно создать два индекса:

The screenshot shows a database IDE with a toolbar at the top. Below the toolbar, there are four SQL queries listed:

```
4 • EXPLAIN SELECT c.city, COUNT(u.id) as user_count  
5 FROM cities c  
6 LEFT JOIN users u ON c.idcity = u.city_id  
7 GROUP BY c.city;  
8 • CREATE INDEX idx_users_city_id ON users(city_id);  
9 • CREATE INDEX idx_cities_city ON cities(city);
```

Повторим запрос:

The screenshot shows a database IDE with a toolbar at the top. Below the toolbar, there are four SQL queries listed:

```
2 • SELECT COUNT(*) FROM users;  
3 • SELECT * FROM users LIMIT 10;  
4 • SELECT c.city, COUNT(u.id) as user_count  
5 FROM cities c  
6 LEFT JOIN users u ON c.idcity = u.city_id  
7 GROUP BY c.city;
```

Below the queries is a "Result Grid" section. It has a "Filter Rows:" input field, an "Export:" button, a "Wrap Cell Content:" checkbox, and a "Fetch rows:" button. The grid contains the following data:

	city	user_count
▶	Aaronberg	109
	Aaronborough	80
	Aaronburgh	75
	Aaronbury	93
	Aaronchester	87

И посмотрим на время:

Output			
Time	Action	Message	Duration / Fetch
22:17:28	drop index idx_users_city_id ON users	Error Code: 1553. Cannot drop index 'idx_uesy' constraint	0.000 sec
22:18:12	CREATE INDEX idx_cities_city ON cities(city)	0 row(s) affected Records: 0 Duplicates: 0	0.578 sec
22:18:16	SHOW INDEX FROM cities	2 row(s) returned	0.016 sec / 0.000 sec
22:18:23	SHOW INDEX FROM users	2 row(s) returned	0.000 sec / 0.000 sec
22:18:38	SELECT c.city, COUNT(u.id) as user_count FROM cities c LEFT JOIN users u ON c.idcity = ...	1000 row(s) returned	0.141 sec / 0.015 sec