

Sql 1 лабораторная работа

Задание 1. Назначение пунктов меню MySQL Workbench

Раздел "Instance" ("Экземпляр БД")

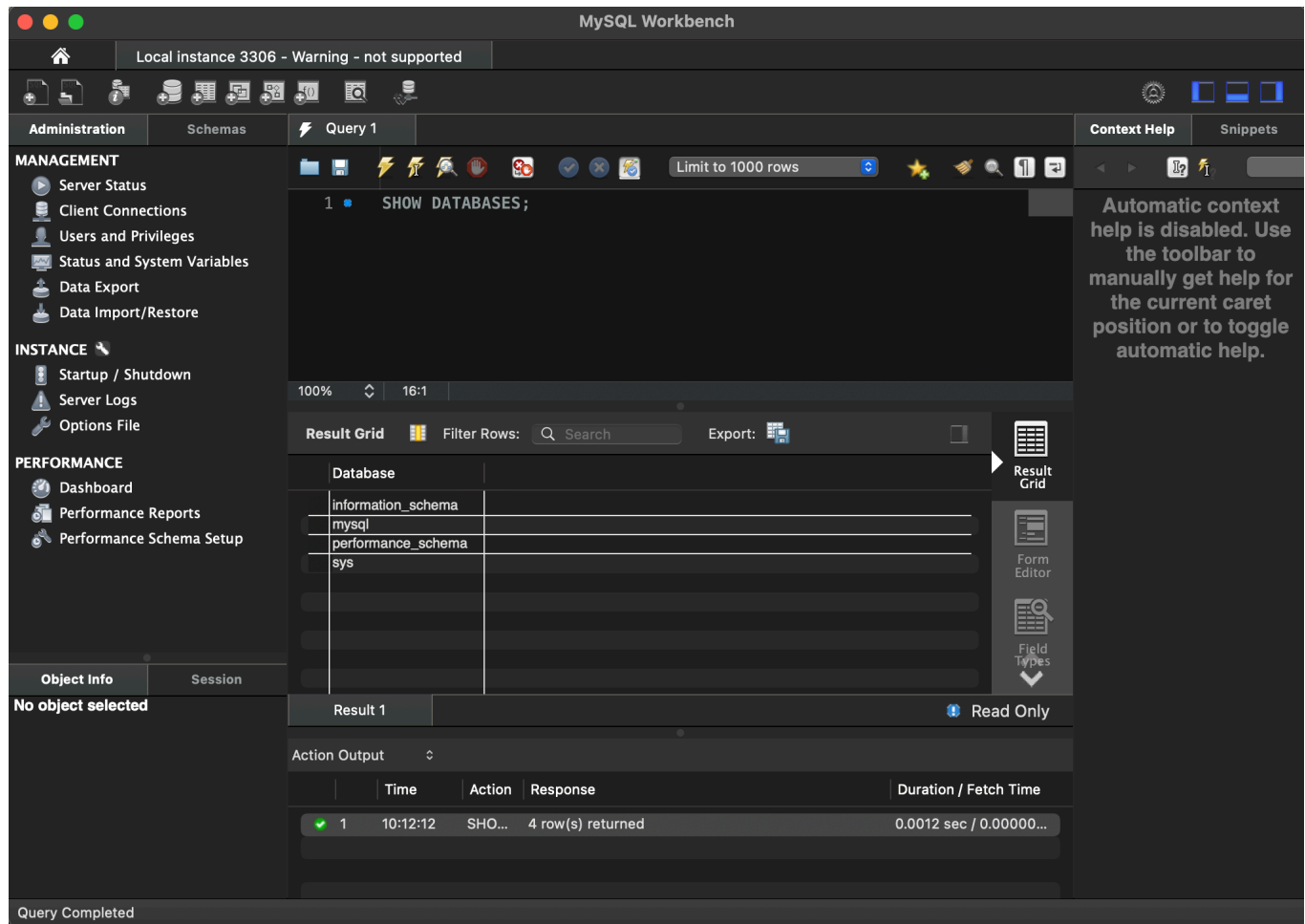
1. **Startup/Shutdown** – управление запуском и остановкой сервера MySQL.
2. **Server Logs** – доступ к журналам ошибок, общим журналам, журналам медленных запросов.
3. **Client Connections** – просмотр активных подключений к серверу.
4. **Status and System Variables** – отображение статистики и конфигурации сервера.
5. **Users and Privileges** – управление пользователями и их правами доступа.

Раздел "Performance" ("Производительность")




1. **Performance Dashboard** – визуализация метрик производительности в реальном времени.
 2. **Performance Reports** – детальные отчёты о производительности.
 3. **Query Statistics** – анализ выполнения SQL-запросов.
 4. **Host Metrics** – мониторинг системных ресурсов сервера.
 5. **InnoDB Status** – информация о работе движка InnoDB.
-

Задание 2. Создание базы данных simpledb

Создана база данных **simpledb** с кодировкой **utf8** и правилом сортировки **utf8_general_ci**.



Задание 3. Создание таблицы new_table

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 name	VARCHAR(45)	<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"/>	
 email	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
CREATE TABLE new_table (  
  id int NOT NULL,  
  name varchar(45) NOT NULL,  
  email varchar(45) DEFAULT NULL,
```

```
PRIMARY KEY ( id ),  
UNIQUE KEY email_UNIQUE ( email )  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Задание 4. Добавление и обновление данных в new_table

```
1 SELECT * FROM simplifiedb.new_table;  
2 INSERT INTO `simplifiedb`.`new_table` (`id`, `name`, `email`)  
3 VALUES  
4   (1, 'Иван', 'ivan@example.com'),  
5   (2, 'Мария', 'maria@test.com'),  
6   (3, 'Алексей', 'alex@company.org');
```

100% 45:6

Result Grid Filter Rows: Search Edit: Export/Import:

	id	name	email
	1	Иван	ivan...
	2	Мария	mari...
	3	Але...	alex...
	NULL	NULL	NULL

Добавлены три записи:

```
INSERT INTO new_table ( id, name, email )  
VALUES  
(1, 'Иван', 'ivan@example.com'),  
(2, 'Мария', 'maria@test.com'),  
(3, 'Алексей', 'alex@company.org');
```

Обновление записи:

```
UPDATE new_table  
SET email = 'новый_email@example.com'  
WHERE id = 1;
```

Задание 5. Создание таблицы users

sql

```
CREATE TABLE simpledb.users (  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(50) NULL,  
  email VARCHAR(45) NULL,  
  gender ENUM('M', 'F') NULL,  
  bday DATE NULL,  
  postal_code VARCHAR(10) NULL,  
  rating FLOAT NULL DEFAULT 0.0,  
  created TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Пояснения:

- **ENUM('M', 'F')** – хранит только 'M' или 'F'.
 - **TIMESTAMP DEFAULT CURRENT_TIMESTAMP** – автоматически устанавливает дату и время создания записи.
 - **NULL поля** (name, email, gender, bday, postal_code) – для приватности пользователей.
 - **rating DEFAULT 0.0** – рейтинг по умолчанию 0.0.
-

Задание 6. Добавление данных в таблицу users

Данные добавлены двумя способами:

1. Вручную через интерфейс MySQL Workbench.
2. SQL-запросами:

id	name	email	gender	bday	postal_code	rating	created
1	Ekaterina	ekaterina.petrova@...	F	2000-02-11	145789	1.123	2026-02-0...
2	Paul	paul@superpochta.ru	M	1998-08-12	123789	1	2026-02-0...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

users 1

Apply

Revert

Action Output

	Time	...	Response	Duration / Fetch Time
✖ 10	13:26:23	I...	Error Code: 1062. Duplicate entry '1' for key 'new_tabl...	0.0014 sec
✔ 11	14:39:39	S...	3 row(s) returned	0.00049 sec / 0.000...
✖ 12	14:39:39	I...	Error Code: 1062. Duplicate entry '1' for key 'new_tabl...	0.00038 sec
✔ 13	14:39:42	S...	3 row(s) returned	0.00041 sec / 0.0000...
✖ 14	14:39:42	I...	Error Code: 1062. Duplicate entry '1' for key 'new_tabl...	0.00044 sec
✔ 15	14:47:47	C...	0 row(s) affected	0.0070 sec
✖ 16	14:48:14	C...	Error Code: 1050. Table 'users' already exists	0.0016 sec
! 17	14:48:57	A...	No changes detected	
✔ 18	14:50:21	S...	3 row(s) returned	0.00047 sec / 0.0000...
✔ 19	14:51:09	S...	0 row(s) returned	0.00046 sec / 0.000...
✔ 20	17:58:02	S...	0 row(s) returned	0.00057 sec / 0.0000...

```
INSERT INTO users ( name , email , postal_code , gender , bday , rating )
VALUES ('Ekaterina', 'ekaterina.petrova@outlook.com', '145789', 'f', '2000-02-11', '1.123');
```

```
INSERT INTO users ( name , email , postal_code , gender , bday , rating )
VALUES ('Paul', 'paul@superpochta.ru', '123789', 'm', '1998-08-12', '1');
```

Задание 7. Экспорт данных из таблицы users

Экспортированные INSERT-запросы:

sql

```
INSERT INTO users ( id , name , email , gender , bday , postal_code , rating ,
created ) VALUES
(1, 'Ekaterina', 'ekaterina.petrova@outlook.com', 'f', '2000-02-11', '145789', 1.123, '2024-01-15
10:30:00'),
(2, 'Paul', 'paul@superpochta.ru', 'm', '1998-08-12', '123789', 1.0, '2024-01-15 10:30:00');
```

Анализ синтаксиса:

Запросы содержат явное указание столбцов, значения в кавычках, даты в формате 'ГГГГ-

MM-ДД', время в формате 'ГГГГ-ММ-ДД ЧЧ:ММ:СС'.

Задание 8. Создание таблицы resume с внешним ключом

sql

```
CREATE TABLE simpledb.resume (  
  resumeid INT NOT NULL AUTO_INCREMENT,  
  userid INT NOT NULL,  
  title VARCHAR(100) NOT NULL,  
  skills TEXT NULL,  
  created TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (resumeid),  
  INDEX fk_resume_user_idx (userid),  
  CONSTRAINT fk_resume_user  
  FOREIGN KEY (userid)  
  REFERENCES simpledb.users (id)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE  
);
```

Пояснение:

Внешний ключ `fk_resume_user` связывает таблицы `resume` и `users`.

- **ON DELETE CASCADE:** при удалении пользователя удаляются все его резюме.
- **ON UPDATE CASCADE:** при изменении `id` пользователя автоматически обновляется `userid` в резюме.

Задание 9. Наполнение таблицы resume данными

Добавление данных:

sql

```
INSERT INTO resume (userid, title, skills) VALUES  
(1, 'Frontend Developer', 'HTML, CSS, JavaScript, React'),  
(1, 'UI/UX Designer', 'Figma, Adobe XD, User Research'),  
(2, 'Backend Developer', 'Python, Django, PostgreSQL, Docker'),
```

```
(2, 'DevOps Engineer', 'AWS, Kubernetes, CI/CD, Terraform'),  
(1, 'Full Stack Developer', 'Node.js, Express, MongoDB, React');
```

Экспорт данных:

sql

```
INSERT INTO resume VALUES  
(1,1,'Frontend Developer','HTML,CSS,JavaScript,React','2026-02-05 19:41:57'),  
(2,1,'UI/UX Designer','Figma,Adobe XD,User Research','2026-02-05 19:41:57'),  
(3,2,'Backend Developer','Python,Django,PostgreSQL,Docker','2026-02-05 19:41:57'),  
(4,2,'DevOps Engineer','AWS,Kubernetes,CI/CD,Terraform','2026-02-05 19:41:57'),  
(5,1,'Full Stack Developer','Node.js,Express,MongoDB,React','2026-02-05 19:41:57');
```

Ответы на вопросы:

- **Сколько резюме у пользователя?**

Минимум: 0, максимум: неограниченно.

- **Попытка добавить резюме с несуществующим userid=999:** sql INSERT INTO resume (userid, title, skills) VALUES (999, 'Test', 'Skills'); **Результат:** Ошибка Error 1452: Foreign key constraint fails.

Вывод: Внешний ключ защищает целостность данных, не позволяя добавить резюме для несуществующего пользователя.

Задание 10. Удаление пользователей и изменение id

1 `SELECT * FROM resume;`

100% 22:1

Result Grid Filter Rows: Search Edit: Export/Import:

resumeid	userid	title	skills	created
3	2	Backend Developer	Python, Django, PostgreSQL, Docker	2026-02-05 19:41:57
4	2	DevOps Engineer	AWS, Kubernetes, CI/CD, Terraform	2026-02-05 19:41:57
NULL	NULL	NULL	NULL	NULL

1. Удаление пользователя с id=1:

```
DELETE FROM users WHERE id = 1;
```

Результат: Автоматически удалились 3 резюме этого пользователя (ON DELETE CASCADE).

2. Проверка резюме:

```
sql
```

```
SELECT * FROM resume;
```

Результат: Осталось 2 резюме (пользователя id=2).

3. Изменение id пользователя:

```
sql
```

```
UPDATE users SET id = 100 WHERE id = 2;
```

Результат: Все `userid=2` в таблице `resume` стали `userid=100` (ON UPDATE CASCADE).

Вывод:

Внешний ключ с опциями CASCADE автоматически поддерживает целостность данных при удалении или обновлении связанных записей.