

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО  
ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель

Путилова Н. В.

подпись, дата

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Заполнение таблиц и модификация данных

По дисциплине: СОЗДАНИЕ БАЗ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4936

подпись, дата

Назаров М. Р.

фамилия, инициалы

Санкт-Петербург 2021

## Лабораторная работа 2. Заполнение таблиц и модификация данных

### Цель работы:

Изучить основные команды по созданию и модификации баз данных

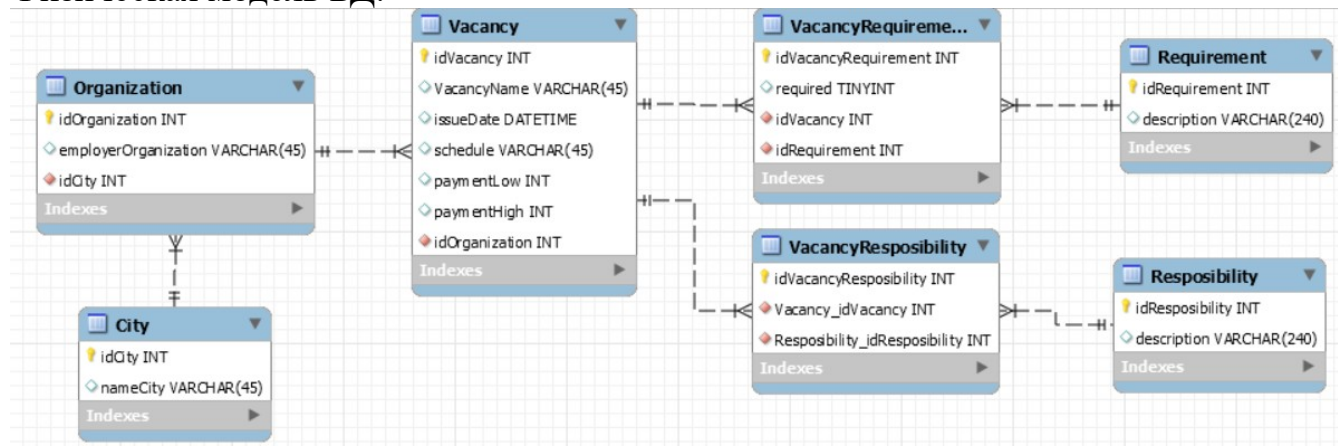
### Вариант лабораторной работы

**Вариант 13.** Вакансии: волонтерские позиции, название вакансии, организация работодатель, адрес работодателя, диапазон зарплаты, требования к образованию, Обязанности, график работы, требования обязательные, желательные, дата выставления вакансии.

- вакансии, имеющие в названии SQL;
- работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора;
- вакансия с наименьшей зарплатой;
- вакансии с минимальным количеством обязательных требований;
- вакансии, в которых нет требования к опыту работы.

### Ход работы:

Физическая модель БД:



Заполненные таблицы:

	idCity	nameCity
▶	1	Санкт-Петербург
	2	Москва
•	NULL	NULL

	idOrganization	employerOrganization	idCity
▶	1	Эпам Системс	1
	2	Flexbby	2
	3	Terrasoft	2
•	NULL	NULL	NULL

	idVacancy	vacancyName	idOrganization	issueDate	schedule	paymentLow	paymentHigh
▶	1	Senior программист	1	2021-09-11 00:00:00	Пн-Пт 10:00-20:00	30000	35000
	2	Аналитик	2	2021-08-21 00:00:00	Пн-Сб 10:00-20:00	35000	50000
	3	Тестировщик	2	2021-10-07 00:00:00	Пн-Вт 10:00-20:00 Пт-Вс 10:00-20:00	50000	60000
	4	Разработчик БД на SQL	3	2021-08-30 00:00:00	Пн-Пт 10:00-20:00	70000	80000
	5	Backend разработчик	3	2021-09-25 00:00:00	Пн-Вт 9:00-18:00 Пт-Вс 9:00-18:00	40000	50000
	6	Системный администратор	1	2021-09-25 00:00:00	Пн-Вт 9:00-18:00 Пт-Вс 9:00-18:00	40000	50000
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	idRequirement	descriptionRequirement
▶	1	Опыт работы больше 2-х лет
	2	Знание Java
	3	Знание Java Script
	4	Знание SQL
	5	Опыт работы в MySQL
	6	Знание библиотек Spring, Maven, Guava, JUnit
	7	Знание английского языка
	8	Знание Python
•	NULL	NULL

	idVacancyRequirement	idRequirement	idVacancy	required
▶	1	2	1	0
	2	6	1	1
	3	7	1	1
	4	1	2	1
	5	4	2	0
	6	1	3	1
	7	1	4	1
	8	4	4	1
	9	5	4	1
	10	7	4	1
	11	1	5	0
	12	7	5	1
	13	8	5	1
	14	7	6	1
•	NULL	NULL	NULL	NULL

	idResponsibility	descriptionResponsibility
▶	1	Писать код
	2	Тестировать введённые изменения
	3	Работать с большими объёмами данных
	4	Работать с базами данных и коллекциями
	5	Писать запросы
	6	Разрабатывать сервер сайта
•	NULL	NULL

	idVacancyResponsibility	idResponsibility	idVacancy
▶	1	1	1
	2	3	2
	3	2	3
	4	4	4
	5	5	4
	6	1	5
	7	6	5
★	NULL	NULL	NULL

Изменим таблицу vacancyResponsibility.

Исходная таблица:

	idVacancyResponsibility	idResponsibility	idVacancy
▶	1	1	1
	2	3	2
	3	2	3
	4	4	4
	5	5	4
	6	1	5
	7	6	5
★	NULL	NULL	NULL

Изменим поле с обязанностями, где id Вакансии = 5

UPDATE vacancyResponsibility

SET idResponsibility = 5

WHERE idVacancy = 5;

Таблица после изменений:

	idVacancyResponsibility	idResponsibility	idVacancy
▶	1	1	1
	2	3	2
	3	2	3
	4	4	4
	5	5	4
	6	5	5
	7	5	5
★	NULL	NULL	NULL

Теперь удалим эти поля по тому же условию:

DELETE FROM vacancyResponsibility

WHERE idVacancy = 5;

Таблица после изменений:

	idVacancyResponsibility	idResponsibility	idVacancy
▶	1	1	1
	2	3	2
	3	2	3
	4	4	4
	5	5	4
★	HULL	HULL	HULL

Для использования merge создадим копию таблицы обязанностей и частично заполним тестовыми данными:

```
CREATE TABLE responsibility_copy (
    idResponsibility INT AUTO_INCREMENT NOT NULL,
    descriptionResponsibility VARCHAR(240),
    PRIMARY KEY (idResponsibility)
);
```

```
INSERT responsibility_copy(descriptionResponsibility)
VALUES ("Писать код");
INSERT responsibility_copy(descriptionResponsibility)
VALUES ("Тестировать введённые изменения");
INSERT responsibility_copy(descriptionResponsibility)
VALUES ("Работать с большими объёмами данных");
```

Код использования merge:

```
MERGE db1.responsibility_copy AS T_Base
USING db1.responsibility AS T_Source
ON(T_Base.idResponsibility = T_Source.idResponsibility)
WHEN MATCHED THEN
    UPDATE SET descriptionResponsibility = T_Source.descriptionResponsibility
WHEN NOT MATCHED THEN
    INSERT (idResponsibility, descriptionResponsibility) VALUES(T_Source.idResponsibility,
T_Source.descriptionResponsibility)
OUTPUT $action AS [Операция], Inserted.idResponsibility,
        Inserted.descriptionResponsibility AS descriptionResponsibilityNEW,
        Deleted.descriptionResponsibility AS descriptionResponsibilityOLD;
```

Опишем данные, удовлетворяющие запросам по варианту, в таблице:

Текст запроса	данные удовлетворяющие условиям запросов	данные не удовлетворяющие условиям запросов
---------------	---	---

вакансии, имеющие в названии SQL;	Разработчик БД на SQL	Senior программист Аналитик Тестировщик Backend разработчик Системный администратор
работодатели в Санкт-Петербурге, выставившие вакансии программиста и системного администратора;	Эпам Системс	Flexbby Terrasoft
вакансия с наименьшей зарплатой;	Senior программист	Разработчик БД на SQL Аналитик Тестировщик Backend разработчик Системный администратор
вакансии с минимальным количеством обязательных требований;	Тестировщик	Senior программист Разработчик БД на SQL Аналитик Backend разработчик Системный администратор
вакансии, в которых нет требования к опыту работы.	Системный администратор	Senior программист Разработчик БД на SQL Аналитик Backend разработчик Тестировщик

Код скрипта:

```
CREATE DATABASE db1;
use db1;
CREATE TABLE city (
    idCity INT AUTO_INCREMENT NOT NULL,
    nameCity VARCHAR(45),
    PRIMARY KEY (idCity)
);
```

```

CREATE TABLE organization (
    idOrganization INT AUTO_INCREMENT NOT NULL,
    employerOrganization VARCHAR(45),
    idCity INT,
    PRIMARY KEY (idOrganization),
    FOREIGN KEY (idCity)
        REFERENCES city (idCity)
        ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE vacancy (
    idVacancy INT AUTO_INCREMENT NOT NULL,
    vacancyName VARCHAR(45),
    idOrganization INT NOT NULL,
    issueDate DATETIME,
    schedule VARCHAR(45),
    paymentLow INT,
    paymentHigh INT,
    PRIMARY KEY (idVacancy),
    FOREIGN KEY (idOrganization)
        REFERENCES organization (idOrganization)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE requirement (
    idRequirement INT AUTO_INCREMENT NOT NULL,
    descriptionRequirement VARCHAR(240),
    PRIMARY KEY (idRequirement)
);

CREATE TABLE vacancyRequirement (
    idVacancyRequirement INT AUTO_INCREMENT,
    idRequirement INT,
    idVacancy INT,
    required TINYINT,
    PRIMARY KEY (idVacancyRequirement),
    FOREIGN KEY (idRequirement)
        REFERENCES requirement (idRequirement)
        ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (idVacancy)
        REFERENCES vacancy (idVacancy)
        ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE responsibility (
    idResponsibility INT AUTO_INCREMENT NOT NULL,

```

```

        descriptionResponsibility VARCHAR(240),
        PRIMARY KEY (idResponsibility)
    );
CREATE TABLE vacancyResponsibility (
    idVacancyResponsibility INT AUTO_INCREMENT,
    idResponsibility INT,
    idVacancy INT,
    PRIMARY KEY (idVacancyResponsibility),
    FOREIGN KEY (idResponsibility)
        REFERENCES responsibility (idResponsibility)
        ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (idVacancy)
        REFERENCES vacancy (idVacancy)
        ON DELETE SET NULL ON UPDATE CASCADE
    );

INSERT city(nameCity) VALUES('Санкт-Петербург');
INSERT city(nameCity) VALUES('Москва');

INSERT organization(employerOrganization,idCity) VALUES('Эпам
Системс',1);
INSERT organization(employerOrganization,idCity) VALUES('Flexbby',2);
INSERT organization(employerOrganization,idCity)
VALUES('Terrasoft',2);

INSERT
vacancy(vacancyName,idOrganization,issueDate,schedule,paymentLow,paym
entHigh) VALUES('Senior программист',1,'20210911','Пн-Пт 10:00-
20:00',30000,35000);
INSERT
vacancy(vacancyName,idOrganization,issueDate,schedule,paymentLow,paym
entHigh) VALUES('Аналитик',2,'20210821','Пн-Сб 10:00-
20:00',35000,50000);
INSERT
vacancy(vacancyName,idOrganization,issueDate,schedule,paymentLow,paym
entHigh) VALUES('Тестировщик',2,'20211007','Пн-Вт 10:00-20:00 Пт-Вс
10:00-20:00',50000,60000);
INSERT
vacancy(vacancyName,idOrganization,issueDate,schedule,paymentLow,paym
entHigh) VALUES('Разработчик БД',3,'20210830','Пн-Пт 10:00-
20:00',70000,80000);
INSERT
vacancy(vacancyName,idOrganization,issueDate,schedule,paymentLow,paym
entHigh) VALUES('Backend разработчик',3,'20210925','Пн-Вт 9:00-18:00
Пт-Вс 9:00-18:00',40000,50000);

INSERT requirement(descriptionRequirement) VALUES('Опыт работы больше

```



```

2-х лет');
INSERT requirement(descriptionRequirement) VALUES('Знание Java');
INSERT requirement(descriptionRequirement) VALUES('Знание Java
Script');
INSERT requirement(descriptionRequirement) VALUES('Знание SQL');
INSERT requirement(descriptionRequirement) VALUES('Опыт работы в
MySQL');
INSERT requirement(descriptionRequirement) VALUES('Знание библиотек
Spring, Maven, Guava, JUnit');
INSERT requirement(descriptionRequirement) VALUES('Знание английского
языка');
INSERT requirement(descriptionRequirement) VALUES('Знание Python');

INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(2,1,0);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(6,1,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(7,1,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(1,2,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(4,2,0);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(1,3,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(1,4,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(4,4,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(5,4,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(7,4,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(1,5,0);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(7,5,1);
INSERT vacancyRequirement(idRequirement,idVacancy,required)
VALUES(8,5,1);

INSERT responsibility(descriptionResponsibility) VALUES("Писать
код");
INSERT responsibility(descriptionResponsibility) VALUES("Тестировать
введённые изменения");
INSERT responsibility(descriptionResponsibility) VALUES("Работать с
большими объёмами данных");
INSERT responsibility(descriptionResponsibility) VALUES("Работать с
базами данных и коллекциями");
INSERT responsibility(descriptionResponsibility) VALUES("Писать

```

```

запросы");
INSERT responsibility(descriptionResponsibility)
VALUES("Разрабатывать сервер сайта");

INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(1,1);
INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(3,2);
INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(2,3);
INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(4,4);
INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(5,4);
INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(1,5);
INSERT vacancyResponsibility(idResponsibility,idVacancy) VALUES(6,5);

UPDATE vacancyResponsibility
SET idResponsibility = 5
WHERE idVacancy = 5;

DELETE FROM vacancyResponsibility
WHERE idVacancy = 5;

CREATE TABLE responsibility_copy (
    idResponsibility INT AUTO_INCREMENT NOT NULL,
    descriptionResponsibility VARCHAR(240),
    PRIMARY KEY (idResponsibility)
);

INSERT responsibility_copy(descriptionResponsibility)
VALUES("Писать код");
INSERT responsibility_copy(descriptionResponsibility)
VALUES("Тестировать введённые изменения");
INSERT responsibility_copy(descriptionResponsibility)
VALUES("Работать с большими объёмами данных");

MERGE db1.responsibility_copy AS T_Base
USING db1.responsibility AS T_Source
ON(T_Base.idResponsibility = T_Source.idResponsibility)
WHEN MATCHEN THEN
    UPDATE SET descriptionResponsibility =
T_Source.descriptionResponsibility
WHEN NOT MATCHED THEN
    INSERT (idResponsibility,
descriptionResponsibility)
VALUES(T_Source.idResponsibility,
T_Source.descriptionResponsibility)
OUTPUT $action AS [Операция], Inserted.idResponsibility,
        Inserted.descriptionResponsibility AS

```

```
descriptionResponsibilityNEW,  
    Deleted.descriptionResponsibility AS  
descriptionResponsibilityOLD;
```