

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 №2

Создание и модификация базы данных и таблиц базы данных

По дисциплине: СОЗДАНИЕ БАЗ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4936

подпись, дата

Назаров М. Р.

фамилия, инициалы

Санкт-Петербург 2021

Лабораторная работа 2. Создание и модификация базы данных и таблиц базы данных

Цель работы:

В соответствии с моделью, разработанной в предыдущей работе, создать базу данных. Продемонстрировать умение добавить и удалить столбец командой alter table.

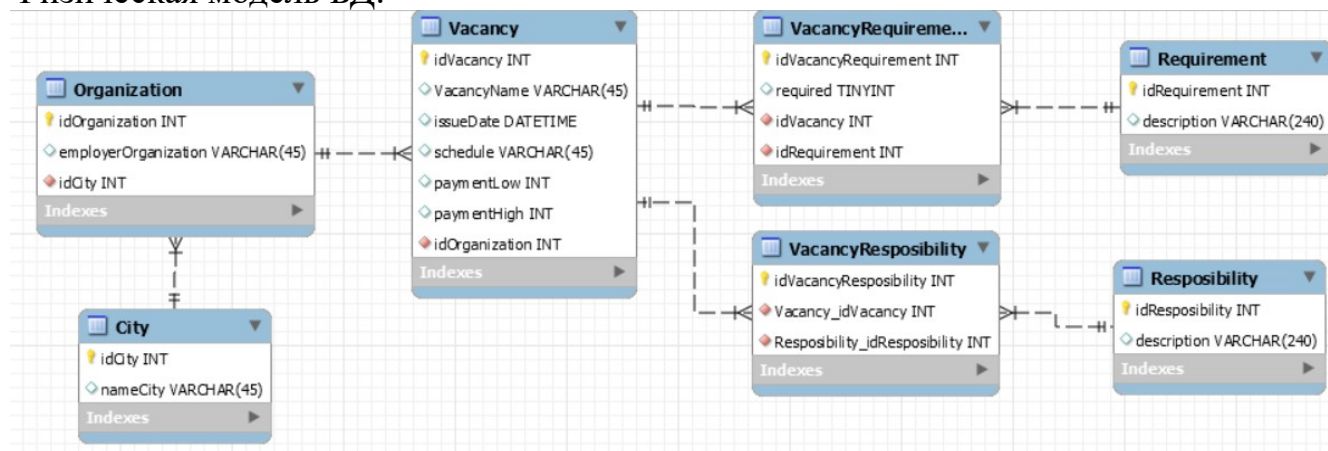
Вариант лабораторной работы

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:



Опишем ссылочную целостность:

Дочерняя таблица (с внешним ключом)	Внешний ключ	Родительская таблица	Как поддерживается ссылочная целостность при удалении	Описание ссылочной целостности при удалении	Как поддерживается ссылочная целостность при обновлении	Описание ссылочной целостности при обновлении	Обоснование
Organization	id_city_int	City	Устанавливается	При удалении	Каскадируется	При обновлении	При удалении

				данных из таблицы «City», значения ссылающихся на них данных из таблицы «Organization» получают «null»		и данных из таблицы «City», обновятся данные в таблице «Organization»	города компании должны будут перемещены в другой город, организация продолжит существовать. Если id города будет изменен, то он будет изменен и в организации.
Vacancy	id_organization_int	Organization	Каскадируется	При удалении данных из таблицы «Organization» удаляются данные в таблице «Vacancy»	Каскадируется	При обновлении и данных из таблицы «Organization», обновятся данные в таблице «Vacancy»	При удалении организации удалятся и вакансии в организации. При обновлении id организации должен обновиться и id в вакансии
Vacancy requirement	id_vacancy_int	Vacancy	Устанавливается	При удалении данных из таблицы «Vacancy», значения ссылающихся	Каскадируется	При обновлении и данных из таблицы «Vacancy», обновятся	При удалении вакансии, требования остаются, но связь с

				хся на них данных из таблицы «Vacancy requirement» получают «null»		данные в таблице «Vacancy requirement»	удаленной вакансией обнуляется. При обновлении и вакансии обновляется и связь с требованием
Vacancy requirement	id_requirement_int	Requirement	Устанавливается	При удалении данных из таблицы «Requirement», значения ссылающихся на них данных из таблицы «Vacancy requirement» получают «null»	Каскадируется	При обновлении и данных из таблицы «Requirement», обновятся данные в таблице «Vacancy requirement»	При удалении требования, вакансии остаются, но связь с удаленным требованием обнуляется. При обновлении и требования обновляется и связь с вакансией
Vacancy responsibility	id_vacancy_int	Vacancy	Устанавливается	При удалении данных из таблицы «Vacancy», значения ссылающихся на них данных из таблицы «Vacancy responsibility»	Каскадируется	При обновлении и данных из таблицы «Vacancy», обновятся данные в таблице «Vacancy responsibility»	При удалении вакансии, обязанности остаются, но связь с удаленной вакансией обнуляется. При

				ity» получат «null»			обновлени и вакансии обновляет ся и связь с требовани ем
Vacancy responsibil ity	id_respons ibility_int	Responsibi lity	Устанавли вается	При удалении данных из таблицы « Responsibi lity», значения ссылающи хся на них данных из таблицы «Vacancy responsibil ity» получат «null»	Каскадиру ется	При обновлени и данных из таблицы « Responsibi lity», обновятся данные в таблице «Vacancy responsibil ity»	При удалении обязаннос ти, вакансии остаются, но связь с удаленной обязаннос тью обнуляетс я. При обновлени и обязаннос ти обновляет ся и связь с вакансией

Скрипт для создания таблиц в соответствии с физической моделью: (1)

```

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
);

```

```
SHOW TABLES;
```

Добавляем колонку с именем обязанности (responsibilityName) в таблицу responsibility: (2)

```
ALTER TABLE responsibility ADD responsibilityName
varchar(45) NOT NULL;
```

Удаляем колонку responsibilityName из таблицы responsibility: (3)

```
ALTER TABLE responsibility DROP COLUMN responsibilityNAME;
```

Результат работы скрипта:

Выполняем скрипт 1.

Список таблиц:

Table: **city**

Columns:

idCity	int AI PK
nameCity	varchar(45)

Table: **organization**

Columns:

idOrganization	int AI PK
employerOrganization	varchar(45)
idCity	int

Table: **requirement**

Columns:

idRequirement	int AI PK
descriptionRequirement	varchar(240)

Table: **responsibility**

Columns:

idResponsibility	int AI PK
descriptionResponsibility	varchar(240)

Table: **vacancy**

Columns:

idVacancy	int AI PK
vacancyName	varchar(45)
idOrganization	int
issueDate	datetime
schedule	varchar(45)
paymentLow	int
paymentHigh	int

Table: **vacancyrequirement**

Columns:

idVacancyRequirement	int AI PK
idRequirement	int
idVacancy	int
required	tinyint

Table: **vacancyresponsibility**

Columns:

idVacancyResponsibility	int AI PK
idResponsibility	int
idVacancy	int

Выполняем скрипт 2.

Table: **responsibility**

Columns:

idResponsibility	int AI PK
descriptionResponsibility	varchar(240)
responsibilityName	varchar(45)

Выполняем скрипт 3.

Table: responsibility

Columns:

idResponsibility	int AI PK
descriptionResponsibility	varchar(240)

Вывод:

Мы научились создавать таблицы с различными параметрами с помощью функции CREATE TABLE, связывать из друг с другом по внешнему ключу функцией FOREIGN KEY и вносить изменения в их структуру функцией ALTER TABLE.