|  |
| --- |
| **Триггеры. Обеспечение активной целостности данных базы данных** |

**Цель работы**

Реализовать для своей базы данных триггеры для всех событий (insert,delete, update) до и после.(6 триггеров) Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение из других предложенных , но не менее 2 различных

(- Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей); - логирование (запись) изменений; - обеспечения безопасности данных; - логическое (мягкое) удаление данных - проверка корректности проводимых действий.).

Вычислимые поля можно добавить при необходимости.

**Вариант 12**

вакансии: название вакансии, организация работодатель, адрес работодателя, диапазон

зарплаты, требования к образованию, Обязанности, график работы, требования обязательные,

желательные, дата выставления вакансии

а. вакансии, имеющие в названии «разработчик», но не начинающиеся на него

б. обязанность, не присутствующая ни в одной вакансии

в. работодатели в Москве, выставившие вакансии и программиста Java и системного

администратора

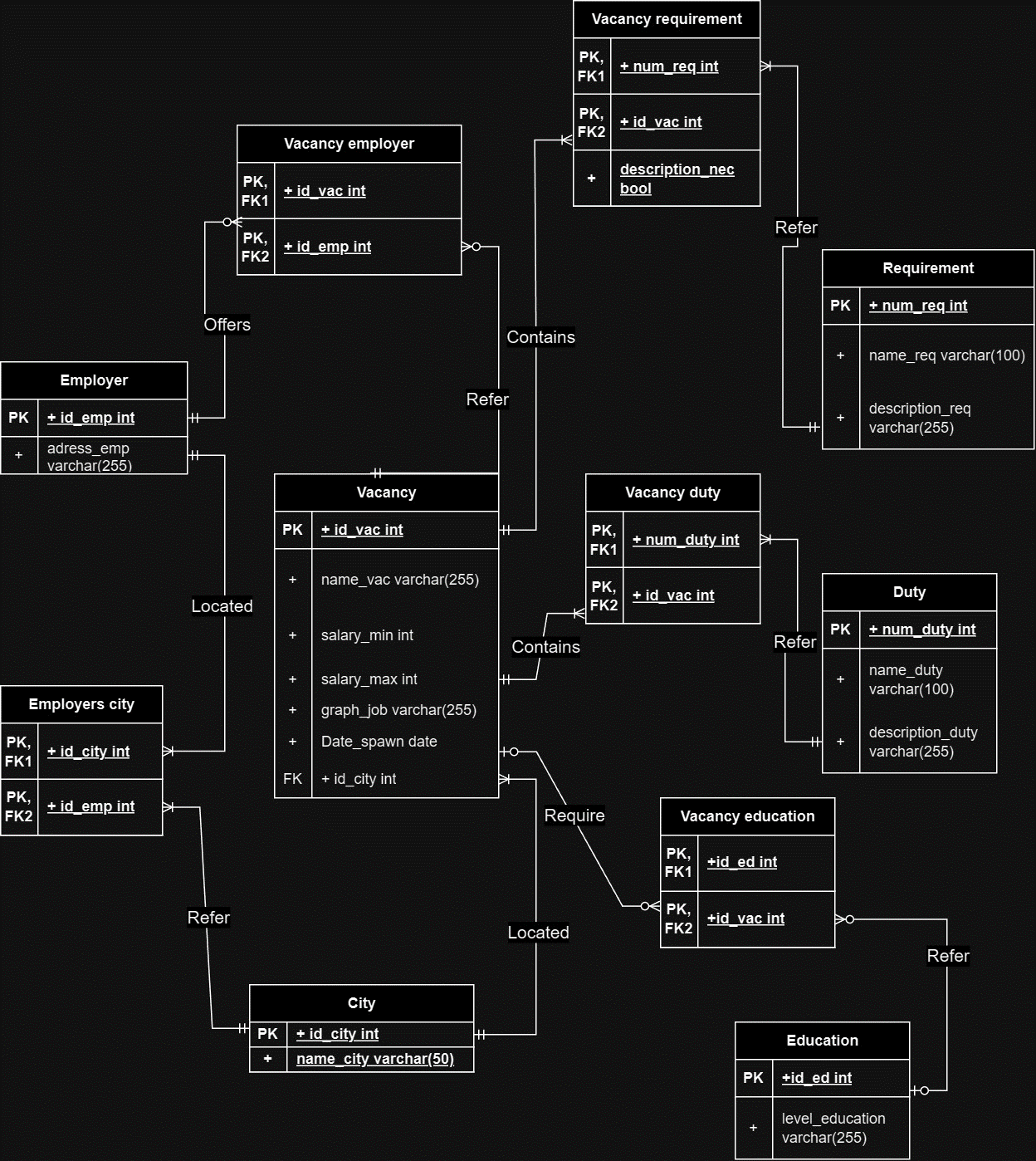
г. вакансия с зарплатой ниже среднего

д. вакансии с максимальным количеством обязательных требований

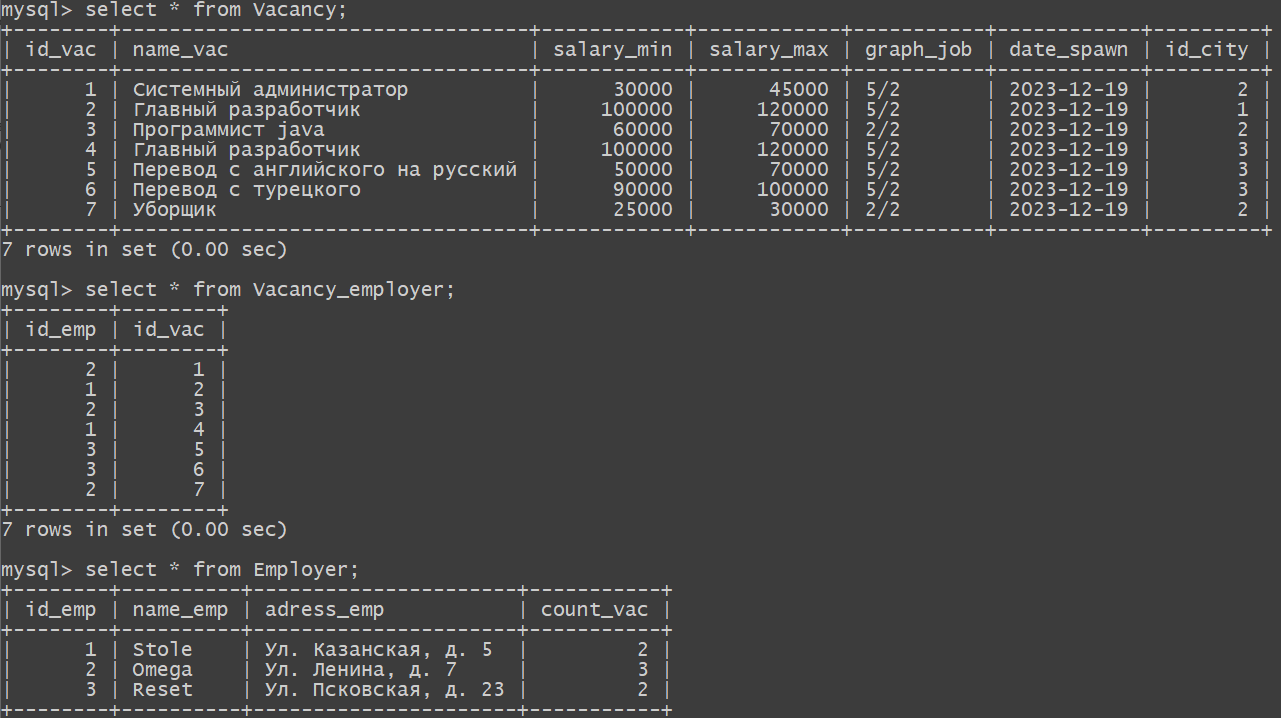
е. требование, присутствующее во всех вакансиях на переводчика (любые вакансии со словом перевод)

ж. вакансии, которые есть в Санкт-Петербурге или Пскове, но которых нет в Москве

**Физическая модель БД**



1. **Поддержание в актуальном состоянии количества вакансий у работодателя (after insert). При добавлении вакансии у работодателя увеличивается счетчик вакансий на единицу.**

****

DELIMITER //

CREATE TRIGGER ins\_vac\_cntEmp AFTER INSERT ON Vacancy\_employer

FOR EACH ROW BEGIN

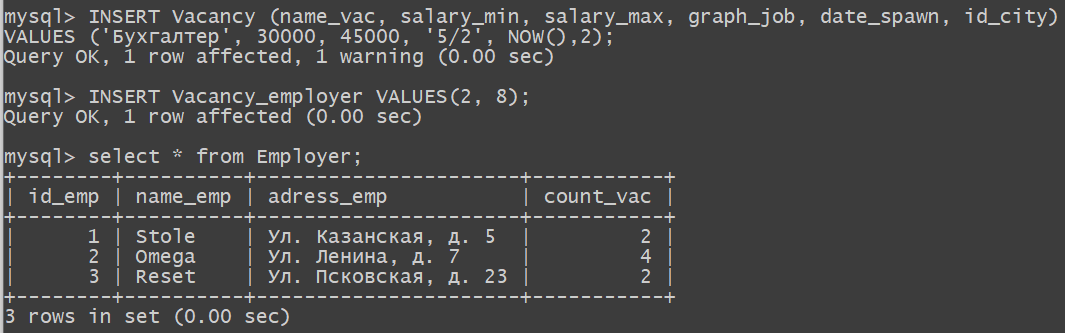
UPDATE Employer

Set count\_vac = (IFNULL(count\_vac, 0) + 1)

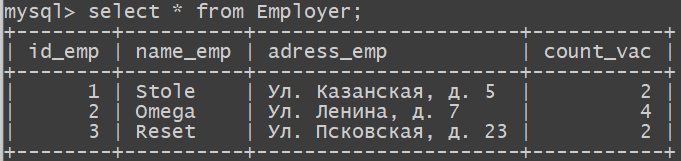
Where id\_emp = new.id\_emp;

END;//

DELIMITER ;



1. **Поддержание в актуальном состоянии количества вакансий у работодателя (after delete). При удалении вакансии у работодателя уменьшается счетчик вакансий на единицу.**

****

DELIMITER //

CREATE TRIGGER del\_Vac\_emp\_cntEmp AFTER DELETE ON Vacancy\_employer

FOR EACH ROW BEGIN

UPDATE Employer

Set count\_vac = count\_vac -1

Where id\_emp = OLD.id\_emp;

END;//

DELIMITER ;

DELIMITER //

CREATE TRIGGER del\_Vac\_cntEmp BEFORE DELETE ON Vacancy

FOR EACH ROW BEGIN

DELETE from Vacancy\_employer WHERE id\_vac = OLD.id\_vac;

DELETE from Vacancy\_requirement WHERE id\_vac = OLD.id\_vac;

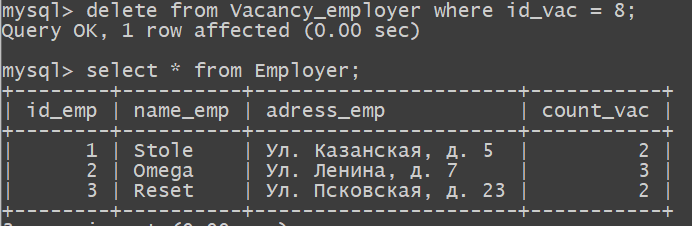
DELETE from Vacancy\_duty WHERE id\_vac = OLD.id\_vac;

DELETE from Vacancy\_education WHERE id\_vac = OLD.id\_vac;

SET @sum = @sum – OLD.salary\_min;

END;//

DELIMITER ;



1. **Вычисление суммы минимальной зарплаты всех вакансий (BEFORE INSERT)**

DELIMITER //

CREATE TRIGGER ins\_vac\_avgSalEmp

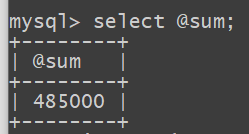
BEFORE INSERT ON Vacancy

FOR EACH ROW BEGIN

SET @sum = IFNULL(@sum, 0) + NEW.salary\_min;

END;//

DELIMITER ;



1. **Логирование событий (AFTER UPDATE). При обновлении информации о вакансии, создается резервная копия строки со старыми данными.**

CREATE TABLE Backup (

Id\_back INT NOT NULL AUTO\_INCREMENT,

id\_vac INT NOT NULL,

name\_vac VARCHAR(255) NOT NULL,

salary\_min INT NOT NULL,

salary\_max INT NOT NULL,

graph\_job VARCHAR(255) NOT NULL,

date\_spawn DATE NOT NULL,

id\_city INT NOT NULL,

PRIMARY KEY (id\_back));

DELIMITER //

CREATE TRIGGER upd\_vac\_addBack AFTER UPDATE ON Vacancy

FOR EACH ROW BEGIN

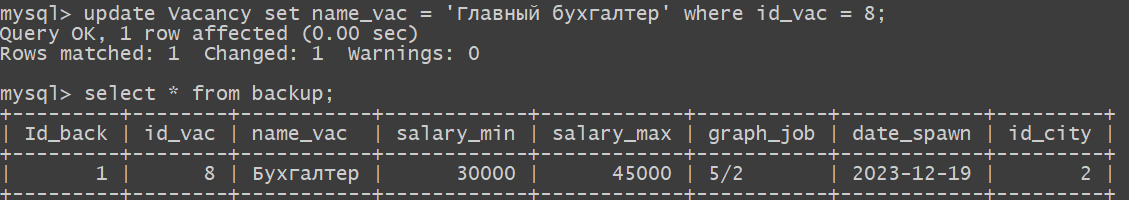
INSERT INTO Backup set id\_vac = OLD.id\_vac, name\_vac = OLD.name\_vac, salary\_min = OLD.salary\_min, salary\_max = OLD.salary\_max, graph\_job = OLD.graph\_job, date\_spawn = OLD.date\_spawn, id\_city = OLD.id\_city;

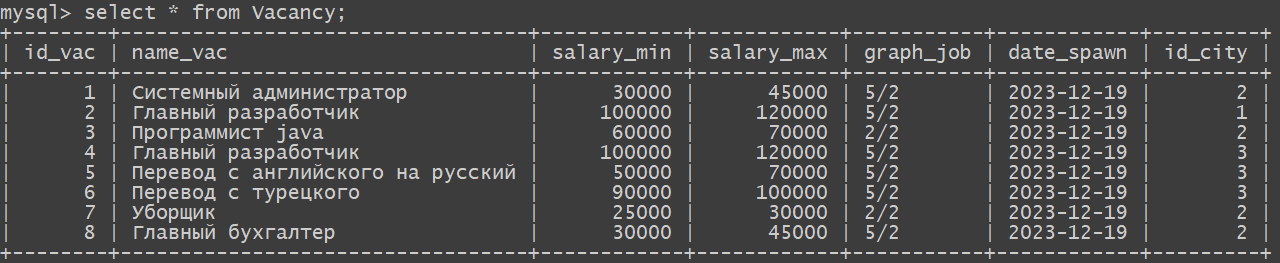
Set @raz = NEW.salary\_min – OLD.salary\_min;

SET @sum = @sum + @raz;

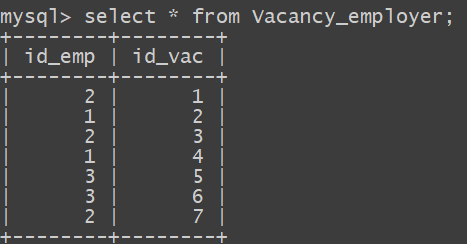
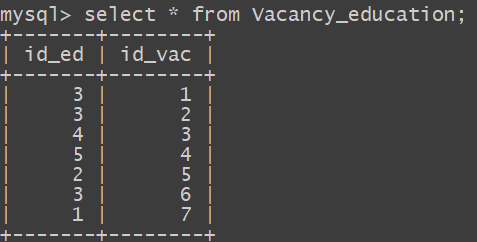
END;//

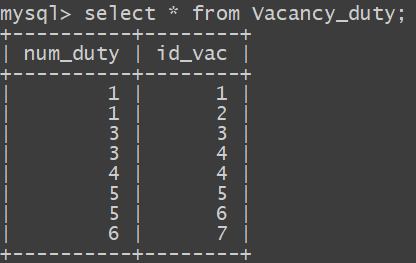
DELIMITER ;





1. **Обеспечение ссылочной целостности (BEFORE UPDATE). При обновлении id вакансии, id вакансии изменяется в связанных таблицах.**

**** ****

****

DELIMITER //

CREATE TRIGGER upd\_vac

BEFORE UPDATE ON Vacancy

FOR EACH ROW BEGIN

Update Vacancy\_employer

Set id\_vac = NEW.id\_vac

WHERE id\_vac = OLD.id\_vac;

Update Vacancy\_requirement

Set id\_vac = NEW.id\_vac

WHERE id\_vac = OLD.id\_vac;

Update Vacancy\_duty

Set id\_vac = NEW.id\_vac

WHERE id\_vac = OLD.id\_vac;

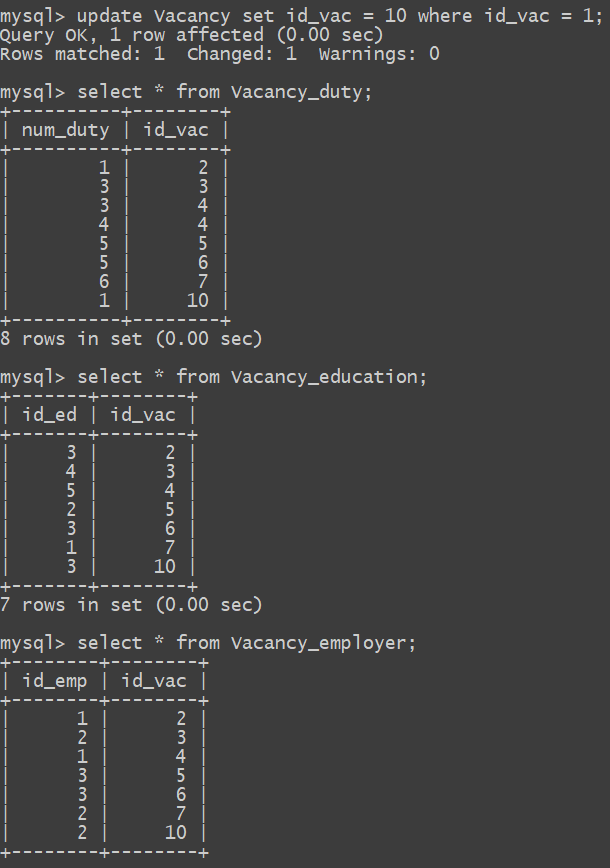
Update Vacancy\_education

Set id\_vac = NEW.id\_vac

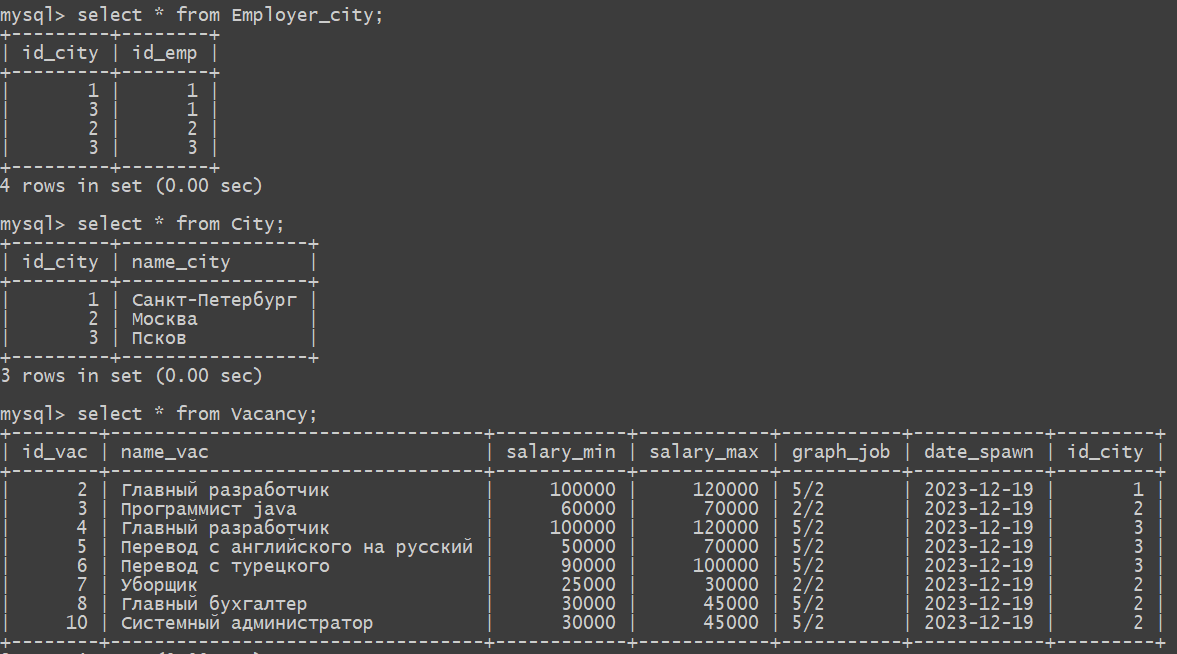
WHERE id\_vac = OLD.id\_vac;

END;//

DELIMITER ;



1. **Ссылочная целостность (before delete). При удалении города удаляются вакансии в этом городе и строки в таблице Employer\_city**



DELIMITER //

CREATE TRIGGER del\_Cit\_Vac BEFORE DELETE ON City

FOR EACH ROW BEGIN

DELETE FROM Vacancy WHERE id\_city = OLD.id\_city;

DELETE FROM Employer\_city WHERE id\_city = OLD.id\_city;

END;//

DELIMITER ;

