

# Хранимые процедуры

## Цель работы

По аналогии с примерами, приведенными в п. 1 , создать в БД ХП, реализующие:

- вставку с пополнением справочников (вставляется информация о студенте, если указанный номер группы отсутствует в БД, запись добавляется в таблицу с перечнем групп) (получаем ссылку на внешний ключ по значению данного из родительской таблицы);
- удаление с очисткой справочников – удаление всех зависимых данных(удаляется информация о студенте, если в его группе нет больше студентов, запись удаляется из таблицы с перечнем групп);
- каскадное удаление (перед удалением записи о группе удаляются записи обо всех студентах этой группы);
- вычисление и возврат значения агрегатной функции (т.к. агрегатная функция дает единственный результат);
- формирование статистики во временной таблице (например, для рассматриваемой БД — для каждого факультета: количество групп, количество обучающихся студентов, количество изучаемых дисциплин, количество сданных дисциплин, средний балл по факультету).

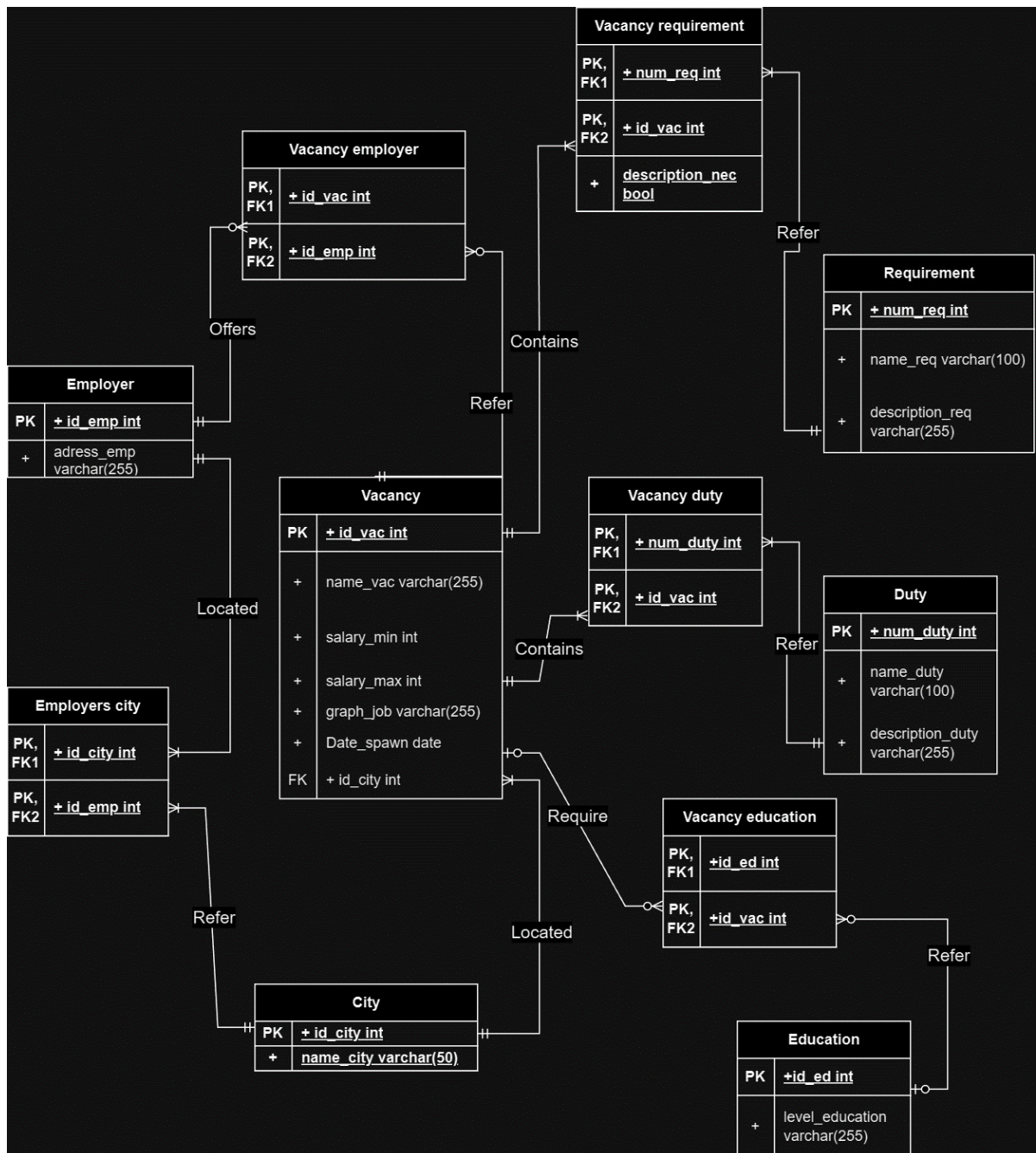
## Вариант 12

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. Процедура вставки с пополнением справочника.

```
DELIMITER //

CREATE PROCEDURE ins_vacancy (IN name_city_ VARCHAR(50),
IN name_vac_ VARCHAR(255),
IN salary_min_ INT,
IN salary_max_ INT,
IN graph_job_ VARCHAR(255),
IN date_spawn_ DATE)
BEGIN
DECLARE id_vac_new INT;
```

```

DECLARE id_city_new INT;
if exists(select * from City where name_city = name_city_)
    then select id_city into id_city_new from City where name_city = name_city_;
else begin
    INSERT INTO City(name_city) VALUES (name_city_);
    set id_city_new = (select LAST_INSERT_ID());
end;
end if;

insert into Vacancy (name_vac, salary_min, salary_max, graph_job, date_spawn, id_city)
VALUES (name_vac_, salary_min_, salary_max_, graph_job_, date_spawn_, id_city_new);

END;

DELIMITER ;

CALL ins_vacancy('Челябинск', 'Директор', 60500, 150000, '5/2', NOW());

CALL ins_vacancy('Воркута', 'Директор', 60500, 150000, '5/2', NOW());

```

Наборы данных до вызова:

```
mysql> select * from City;
```

id_city	name_city
1	Санкт-Петербург
2	Москва
3	Псков

```
mysql> select * from Vacancy;
```

id_vac	name_vac	salary_min	salary_max	graph_job	date_spawn	id_city
1	Системный администратор	30000	45000	5/2	2023-12-10	2
2	Главный разработчик	100000	120000	5/2	2023-12-10	1
3	Программист java	60000	70000	2/2	2023-12-10	2
4	Главный разработчик	100000	120000	5/2	2023-12-10	3
5	Перевод с английского на русский	50000	70000	5/2	2023-12-10	3
6	Перевод с турецкого	90000	100000	5/2	2023-12-10	3
7	Уборщик	25000	30000	2/2	2023-12-10	2

Наборы данных после вызова:

```
mysql> select * from City;
```

id_city	name_city
1	Санкт-Петербург
2	Москва
3	Псков
4	Челябинск

```
mysql> select * from Vacancy;
```

id_vac	name_vac	salary_min	salary_max	graph_job	date_spawn	id_city
1	Системный администратор	30000	45000	5/2	2023-12-10	2
2	Главный разработчик	100000	120000	5/2	2023-12-10	1
3	Программист java	60000	70000	2/2	2023-12-10	2
4	Главный разработчик	100000	120000	5/2	2023-12-10	3
5	Перевод с английского на русский	50000	70000	5/2	2023-12-10	3
6	Перевод с турецкого	90000	100000	5/2	2023-12-10	3
7	Уборщик	25000	30000	2/2	2023-12-10	2
8	Директор	60500	150000	5/2	2023-12-10	4

## 2. Процедура удаления с очисткой справочника (Удаляется город, если в этом городе нет вакансий и работодателей)

```
DELIMITER //
```

```
CREATE PROCEDURE del_vac_clear_city (IN id_vac_del INT)
```

```
BEGIN
```

```
DECLARE id_city_del INT;
```

```
SELECT id_city INTO id_city_del FROM Vacancy where id_vac = id_vac_del;
```

```
Delete from Vacancy where id_vac = id_vac_del;
```

```
If not exists(select * from Vacancy WHERE id_city = id_city_del) AND
```

```
not exists(select * from Employer_city WHERE id_city = id_city_del)
```

```
then DELETE from City WHERE id_city = id_city_del;
```

```
end if;
```

```
END; //
```

```
DELIMITER ;
```

```
CALL del_vac_clear_city(8);
```

Наборы данных до вызова:

```
mysql> select * from City;
```

id_city	name_city
1	Санкт-Петербург
2	Москва
3	Псков
4	Челябинск

```
mysql> select * from Vacancy;
```

id_vac	name_vac	salary_min	salary_max	graph_job	date_spawn	id_city
1	Системный администратор	30000	45000	5/2	2023-12-10	2
2	Главный разработчик	100000	120000	5/2	2023-12-10	1
3	Программист java	60000	70000	2/2	2023-12-10	2
4	Главный разработчик	100000	120000	5/2	2023-12-10	3
5	Перевод с английского на русский	50000	70000	5/2	2023-12-10	3
6	Перевод с турецкого	90000	100000	5/2	2023-12-10	3
7	Уборщик	25000	30000	2/2	2023-12-10	2
8	Директор	60500	150000	5/2	2023-12-10	4

Наборы данных после вызова:

```
mysql> select * from City;
```

id_city	name_city
1	Санкт-Петербург
2	Москва
3	Псков

```
mysql> select * from Vacancy;
```

id_vac	name_vac	salary_min	salary_max	graph_job	date_spawn	id_city
1	Системный администратор	30000	45000	5/2	2023-12-10	2
2	Главный разработчик	100000	120000	5/2	2023-12-10	1
3	Программист java	60000	70000	2/2	2023-12-10	2
4	Главный разработчик	100000	120000	5/2	2023-12-10	3
5	Перевод с английского на русский	50000	70000	5/2	2023-12-10	3
6	Перевод с турецкого	90000	100000	5/2	2023-12-10	3
7	Уборщик	25000	30000	2/2	2023-12-10	2

### 3. Процедура каскадного удаления (При удалении города, удаляются связанные вакансии)

```
DELIMITER //
```

```
CREATE PROCEDURE del_city_cascade (IN id_city_del INT)
```

```
BEGIN
```

```
Delete from Vacancy_duty WHERE id_vac in (SELECT id_vac from Vacancy WHERE id_city = id_city_del);
```

```
Delete from Vacancy_education WHERE id_vac in (SELECT id_vac from Vacancy WHERE id_city = id_city_del);
```

```
Delete from Vacancy_requirement WHERE id_vac in (SELECT id_vac from Vacancy WHERE id_city = id_city_del);
```

```
Delete from Vacancy_employer WHERE id_vac in (SELECT id_vac from Vacancy WHERE id_city = id_city_del);
```

```
Delete from Employer_city WHERE id_city = id_city_del;
```

```
Delete from Vacancy WHERE id_city = id_city_del;
```

```
Delete from City WHERE id_city = id_city_del;
```

```
END; //
```

```
DELIMITER ;
```

```
CALL del_city_cascade(3);
```

Наборы данных до вызова:

```
mysql> select * from City;
```

id_city	name_city
1	Санкт-Петербург
2	Москва
3	Псков

```
mysql> select * from Vacancy;
```

id_vac	name_vac	salary_min	salary_max	graph_job	date_spawn	id_city
1	Системный администратор	30000	45000	5/2	2023-12-10	2
2	Главный разработчик	100000	120000	5/2	2023-12-10	1
3	Программист java	60000	70000	2/2	2023-12-10	2
4	Главный разработчик	100000	120000	5/2	2023-12-10	3
5	Перевод с английского на русский	50000	70000	5/2	2023-12-10	3
6	Перевод с турецкого	90000	100000	5/2	2023-12-10	3
7	Уборщик	25000	30000	2/2	2023-12-10	2

```
mysql> select * from Vacancy_requirement order by id_vac;
```

num_req	id_vac	description_nec
1	1	0
2	1	1
3	1	0
5	1	0
1	2	0
2	2	1
3	2	0
5	2	0
2	3	1
5	3	0
1	4	0
2	4	1
3	4	1
5	4	1
7	4	1
1	5	0
2	5	1
4	5	1
5	5	1
7	5	0
4	6	1
6	6	1
2	7	1

```
mysql> select * from Vacancy_duty order by id_vac;
```

num_duty	id_vac
1	1
1	2
3	3
3	4
4	4
5	5
5	6
6	7

```
mysql> select * from Vacancy_education;
```

id_ed	id_vac
3	1
3	2
4	3
5	4
2	5
3	6
1	7

```
mysql> Select * from vacancy_employer order by id_vac;
```

id_emp	id_vac
2	1
1	2
2	3
1	4
3	5
3	6
2	7

Наборы данных после вызова:

```
mysql> select * from City;
```

id_city	name_city
1	Санкт-Петербург
2	Москва



```
mysql> select * from Vacancy;
```

id_vac	name_vac	salary_min	salary_max	graph_job	date_spawn	id_city
1	Системный администратор	30000	45000	5/2	2023-12-10	2
2	Главный разработчик	100000	120000	5/2	2023-12-10	1
3	Программист java	60000	70000	2/2	2023-12-10	2
7	Уборщик	25000	30000	2/2	2023-12-10	2

```
mysql> Select * from Vacancy_requirement order by id_vac;
```

num_req	id_vac	description_nec
1	1	0
2	1	1
3	1	0
5	1	0
1	2	0
2	2	1
3	2	0
5	2	0
2	3	1
5	3	0
2	7	1

```
mysql> select * from Vacancy_duty;
```

num_duty	id_vac
1	1
1	2
3	3
6	7

```
mysql> Select * from Vacancy_education;
```

id_ed	id_vac
3	1
3	2
4	3
1	7



```
mysql> select * from Vacancy_employer;
```

id_emp	id_vac
2	1
1	2
2	3
2	7

### Возврат количества вакансий:

```
DELIMITER //
CREATE PROCEDURE count_vac (out cnt_vac INT)
BEGIN
SELECT ifnull(count(id_vac),0) into cnt_vac from Vacancy;
END;//
DELIMITER ;
```

Call count\_vac( @cnt);

```
SELECT @cnt;
```

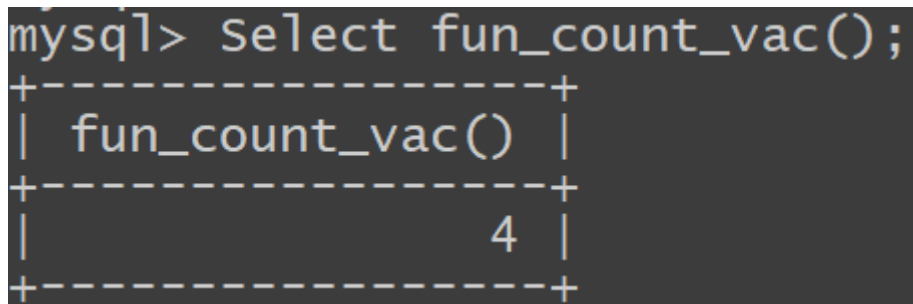
```
mysql> SELECT @cnt;
```

@cnt
4

### Возврат количества вакансий - функция:

```
DELIMITER //
CREATE FUNCTION fun_count_vac() returns int DETERMINISTIC
BEGIN
DECLARE cnt_vac int DEFAULT 0;
Set cnt_vac=(SELECT ifnull(count(id_vac),0) from Vacancy);
Return cnt_vac;
END;//
DELIMITER ;
```

Select fun\_count\_vac();



```
mysql> select fun_count_vac();
+-----+
| fun_count_vac() |
+-----+
| 4               |
+-----+
```

### Формирование статистики во временной таблице

DELIMITER //

CREATE PROCEDURE vac\_statistics ()

BEGIN

Create temporary table if not exists vac\_stat

(

id\_stat int auto\_increment primary key,

id\_vac int,

min\_salary int,

avg\_min\_salary int,

diff\_min\_avg\_salary int,

count\_req int,

count\_req\_avg double default 0,

diff\_cnt\_avg double default 0

);

insert into vac\_stat (id\_vac, count\_req, min\_salary)

select Vacancy.id\_vac, count(num\_req) as count\_req, salary\_min from Vacancy

left join Vacancy\_requirement on Vacancy.id\_vac=Vacancy\_requirement.id\_vac group by

Vacancy.id\_vac;

update vac\_stat set avg\_min\_salary=(select avg(salary\_min) from Vacancy);

update vac\_stat set diff\_min\_avg\_salary=min\_salary-avg\_min\_salary;

update vac\_stat set count\_req\_avg=(select avg(count\_req) from

(select Vacancy.id\_vac, count(num\_req) as count\_req from Vacancy

left join Vacancy\_requirement on Vacancy.id\_vac=Vacancy\_requirement.id\_vac group by

Vacancy.id\_vac)q);

```
update vac_stat set diff_cnt_avg=count_req-count_req_avg;
```

```
select * from vac_stat;
```

```
select avg(diff_min_avg_salary*diff_min_avg_salary) from vac_stat;
```

```
drop table vac_stat;
```

```
END; //
```

```
DELIMITER ;
```

```
Call vac_statistics;
```

```
mysql> call vac_statistics;
```

id_stat	id_vac	min_salary	avg_min_salary	diff_min_avg_salary	count_req	count_req_avg	diff_cnt_avg
1	1	30000	53750	-23750	4	2.75	1.25
2	2	100000	53750	46250	4	2.75	1.25
3	3	60000	53750	6250	2	2.75	-0.75
4	7	25000	53750	-28750	1	2.75	-1.75

```
4 rows in set (0.01 sec)
```

avg(diff_min_avg_salary*diff_min_avg_salary)
892187500.0000