

# Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна  
Кафедра теоретичної та прикладної інформатики

Звіт по дисципліні  
Вступ до SQL баз даних

Додадкове завдання № 1

Студента: Чистякова Артема  
Групи: МФ-31

Необхідний термін здачі завдання: \_\_\_\_\_

Фактичний термін здачі завдання: \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Харків 2020

## Постановка задачи

Разработать базу данных для хранения и обработки информации о внутреннем устройстве некоторой небольшой аутсорс IT-компании.

Компания хоть и небольшая, но может располагать несколькими филиалами. В каждом из офисов работают: менеджеры, программисты и дизайнеры. У каждого менеджера в подчинении могут находиться другие менеджеры, а также программисты и дизайнеры. У каждого программиста может быть программист ментор высшей должности.

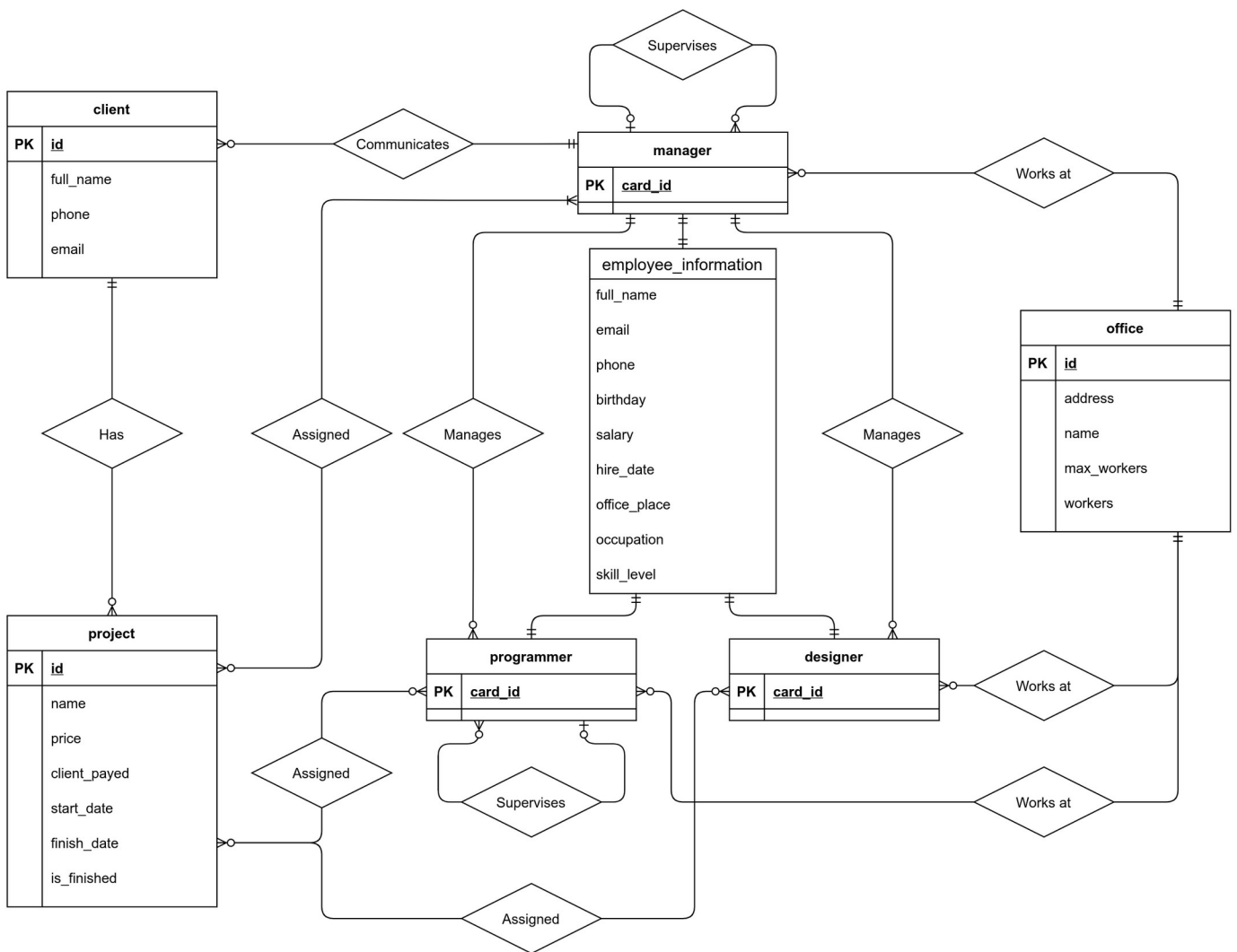
В компанию поступают заказы от клиентов в качестве проектов. Каждому клиенту предоставляется менеджер для общения, и над поступившими проектами начинается работа. В проекте могут быть задействованы менеджеры, программисты и дизайнеры.

В базе данных должна храниться информация о каждом сотруднике, это личные данные, зарплата, место в офисе и направление работы (для программиста, например, Java или C++).

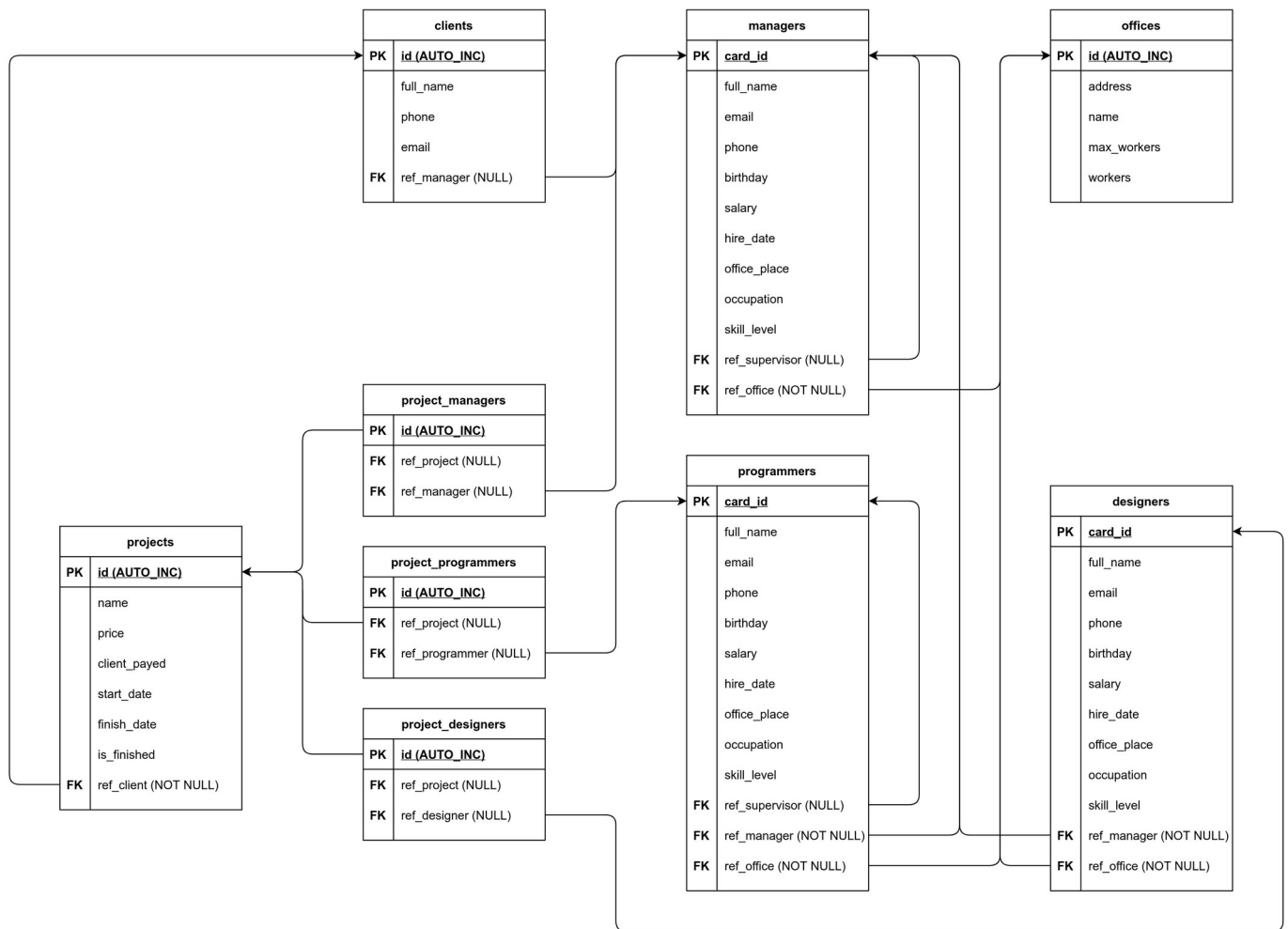
О проектах в базе данных тоже должна храниться информация, это стоимость проекта, текущая оплата клиента, время начала работы и дедлайн.

В базе данных также должна храниться информация о офисе.

# Концептуальная модель БД



# Представление БД в качестве таблиц и связей



## Перечень запросов

1. Создать ограничения на зарплаты и цену проекта (чтобы стоимость была не отрицательной).

```
ALTER TABLE managers ADD CONSTRAINT managers_salary_con CHECK
(salary >= 0);
ALTER TABLE programmers ADD CONSTRAINT programmers_salary_con
CHECK (salary >= 0);
ALTER TABLE designers ADD CONSTRAINT designers_salary_con
CHECK (salary >= 0);
```

```
ALTER TABLE projects ADD CONSTRAINT projects_price_con CHECK
(price >= 0);
ALTER TABLE projects ADD CONSTRAINT projects_payed_con CHECK
(client_payed >= 0);
```

### Проверка

```
INSERT INTO managers VALUES (6, "Random Name", "email",
"number", "1970.01.01", -123, "1970.01.01", 1, "", "", NULL,
2); // should fail
```

2. Создать базовую проверку на валидность email.

```
ALTER TABLE managers ADD CONSTRAINT managers_email_con CHECK
(email LIKE '%@%.%');
ALTER TABLE programmers ADD CONSTRAINT programmers_email_con
CHECK (email LIKE '%@%.%');
ALTER TABLE designers ADD CONSTRAINT designers_email_con CHECK
(email LIKE '%@%.%');
ALTER TABLE clients ADD CONSTRAINT clients_email_con CHECK
(email LIKE '%@%.%');
```

### Проверка

```
INSERT INTO managers VALUES (6, "Random Name", "bad email",
"number", "1970.01.01", 2000, "1970.01.01", 1, "", "", NULL,
2); // should fail
```

3. Создать проверку, что количество работников в офисе не больше их максимального количества.

```
ALTER TABLE offices ADD CONSTRAINT offices_workers_con CHECK
(workers <= max_workers);
```

## Проверка

```
UPDATE offices SET workers = 1000000 WHERE id = 1; // should fail
```

4. Установить default параметр для даты найма сотрудника (сегодняшнее число).

```
ALTER TABLE managers ALTER hire_date SET DEFAULT  
(CURRENT_DATE);  
ALTER TABLE programmers ALTER hire_date SET DEFAULT  
(CURRENT_DATE);  
ALTER TABLE designers ALTER hire_date SET DEFAULT  
(CURRENT_DATE);
```