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 № 4

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 находится другие менеджеры, а также программисты и дизайнеры. У каждого программиста может быть программист ментор высшей должности.

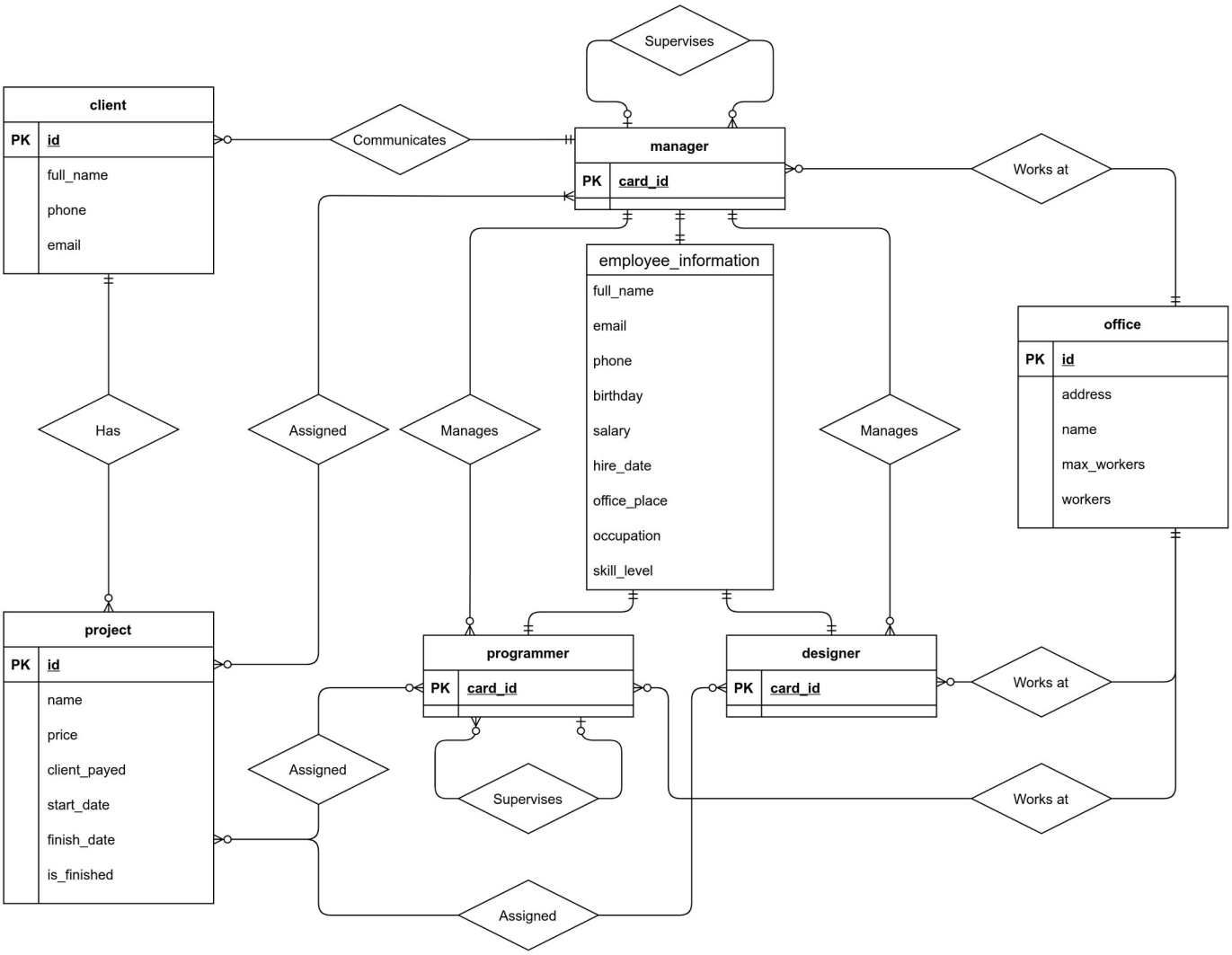
В компанию поступают заказы от клиентов в качестве проектов. Каждому клиенту предоставляется менеджер для общения, и над поступившими проектами начинается работа. В проекте могут быть задействованы менеджеры, программисты и дизайнеры.

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++).

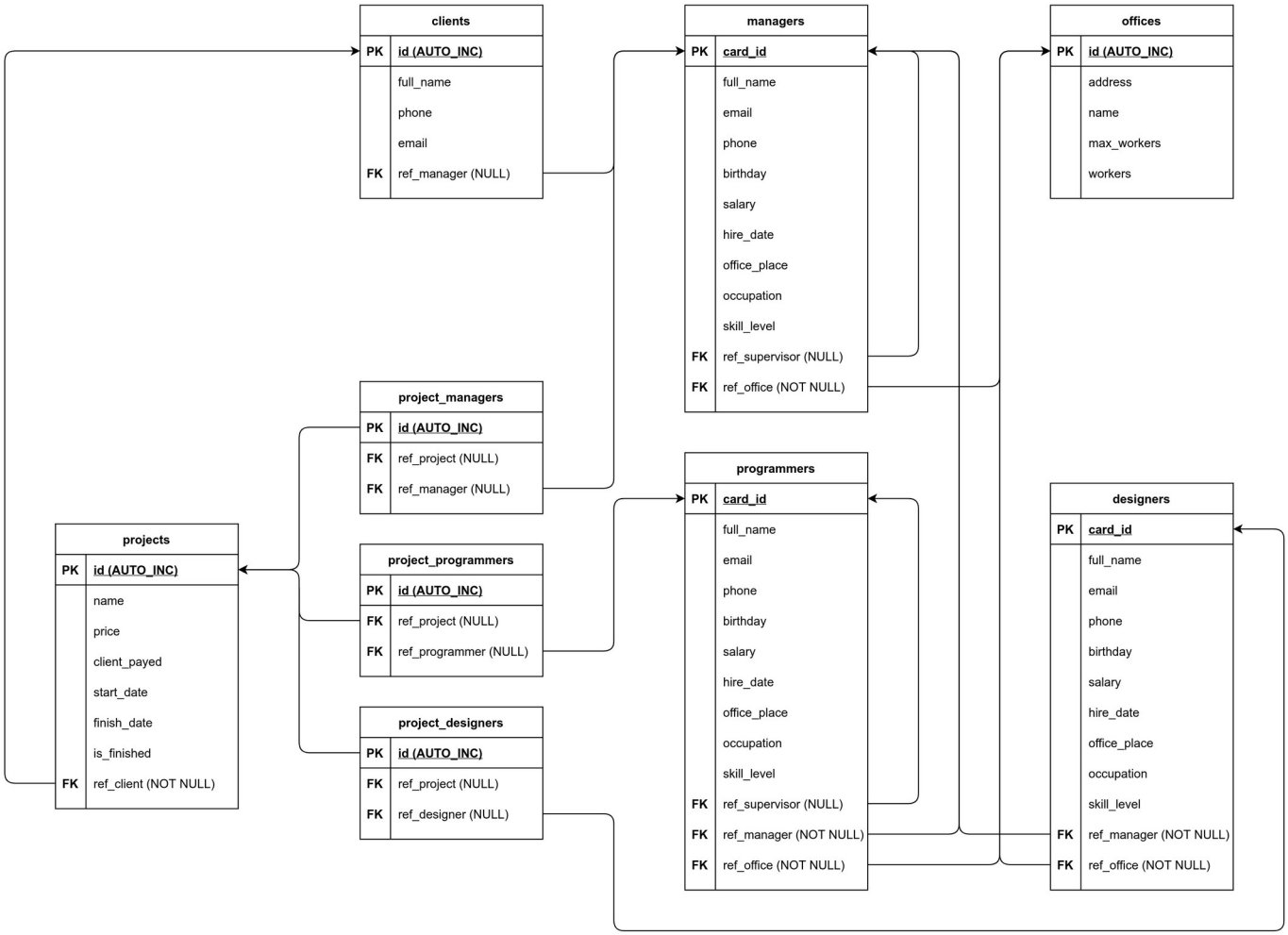
О проектах в базе данных тоже должна храниться информация, это стоимость проекта, текущая оплата клинта, время начала работы и дедлайн.

В базе данных также должна храниться информация о офисе.

Концептуальная модель БД



Представление БД в качестве таблиц и связей



Перечень запросов

1. Создать проверочный триггер на добавление сотрудника в таблицу. Триггер должен проверять уникальность номера карточки и места в офисе. Также триггер должен обновить количество сотрудников в указанном офисе. (Здесь создан триггер только для менеджеров).

```
CREATE TRIGGER updateOfficeInsert BEFORE INSERT ON managers
FOR EACH ROW
BEGIN
      IF EXISTS (SELECT * FROM managers WHERE office_place =
      NEW.office_place AND ref_office = NEW.ref_office) THEN
            SIGNAL sqlstate '45001' SET message_text = "Duplicate office
            place";
      END IF;

      IF EXISTS (SELECT * FROM programmers WHERE office_place =
            NEW.office_place AND ref_office = NEW.ref_office) THEN
            SIGNAL sqlstate '45001' SET message_text = "Duplicate office
            place";
      END IF;

      IF EXISTS (SELECT * FROM designers WHERE office_place =
      NEW.office_place AND ref_office = NEW.ref_office) THEN
            SIGNAL sqlstate '45001' SET message_text = "Duplicate office
            place";
      END IF;

      IF EXISTS (SELECT * FROM managers WHERE card_id = NEW.card_id) THEN
            SIGNAL sqlstate '45001' SET message_text = "Duplicate card
            id";
      END IF;

      IF EXISTS (SELECT * FROM programmers WHERE card_id = NEW.card_id)
      THEN
            SIGNAL sqlstate '45001' SET message_text = "Duplicate card
            id";
      END IF;

      IF EXISTS (SELECT * FROM designers WHERE card_id = NEW.card_id) THEN
            SIGNAL sqlstate '45001' SET message_text = "Duplicate card
            id";
      END IF;

      UPDATE offices SET workers = workers + 1 WHERE id = NEW.ref_office;
END;
```

2. Создать триггер на увольнение (удаление) сотрудника. Триггер должен обновить количество сотрудников в указанном офисе. (Здесь создан триггер только для менеджеров).

```
CREATE TRIGGER updateOfficeDelete AFTER DELETE ON managers
FOR EACH ROW
BEGIN
```

```
     UPDATE offices SET workers = workers - 1 WHERE id =
     OLD.ref_office;
END;
```

3. Создать триггер на обновление места сотрудника. Триггер должен совершать проверку на коллизию места. (Здесь создан триггер только для менеджеров).

```
CREATE TRIGGER updateOfficeUpdate BEFORE UPDATE ON managers
FOR EACH ROW
BEGIN
     IF EXISTS (SELECT * FROM managers WHERE office_place =
     NEW.office_place AND ref_office = NEW.ref_office) THEN
          SIGNAL sqlstate '45001' SET message_text =
          "Duplicate office place";
     END IF;

     IF EXISTS (SELECT * FROM programmers WHERE office_place =
     NEW.office_place AND ref_office = NEW.ref_office) THEN
          SIGNAL sqlstate '45001' SET message_text =
          "Duplicate office place";
     END IF;

     IF EXISTS (SELECT * FROM designers WHERE office_place =
     NEW.office_place AND ref_office = NEW.ref_office)THEN
          SIGNAL sqlstate '45001' SET message_text =
          "Duplicate office place";
     END IF;
END;
```