

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

Індивідуальне завдання № 6

Студента: Чистякова Артема
Групи: МФ-31

Необхідний термін здачі завдання: 04.12.2020

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.

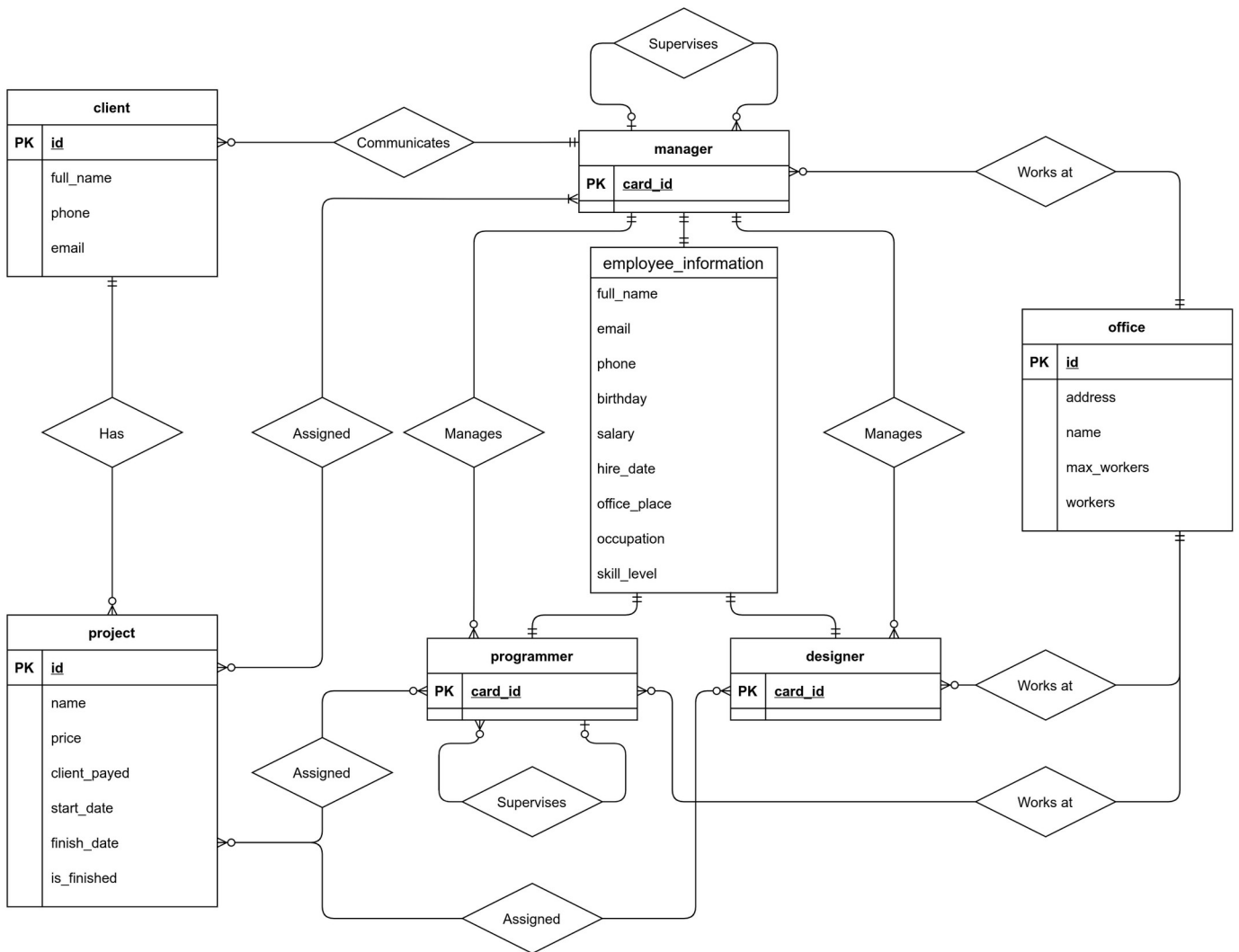
В компанию поступают заказы от клиентов в качестве проектов. Каждому клиенту предоставляется менеджер для общения, и над поступившими проектами начинается работа. В проекте могут быть задействованы менеджеры, программисты и дизайнеры.

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++).

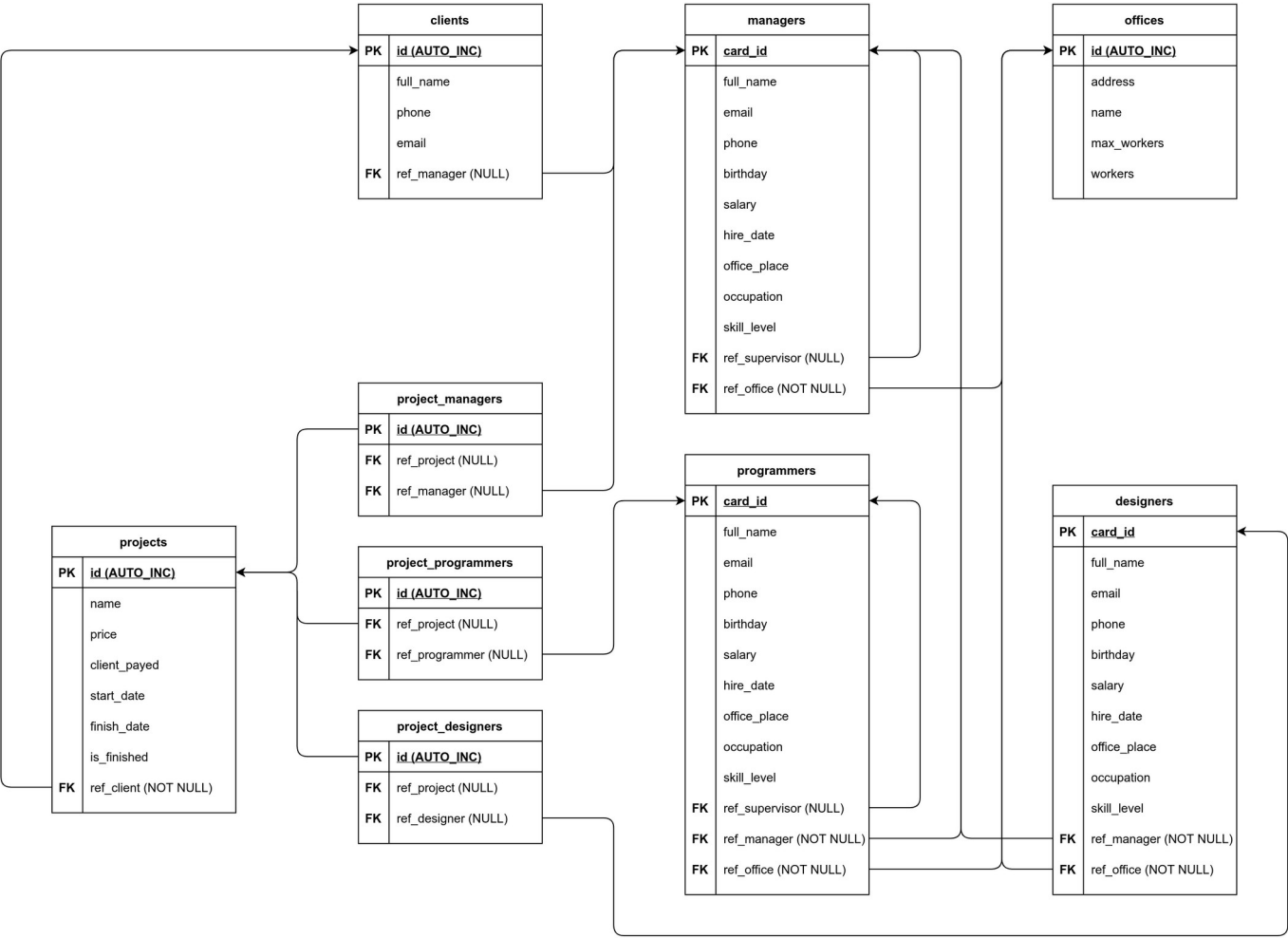
О проектах в базе данных тоже должна храниться информация, это стоимость проекта, текущая оплата клиента, время начала работы и дедлайн.

В базе данных также должна храниться информация о офисе.

Концептуальная модель БД



Представление БД в качестве таблиц и связей



Перечень запросов

1. Создать процедуру, с помощью которой можно было бы добавить любого сотрудника к любому проекту.

```
DELIMITER $$
```

```
CREATE PROCEDURE addToProject (
    IN position VARCHAR(30),
    IN fullName VARCHAR(50),
    IN projectName VARCHAR(50)
)
BEGIN
    DECLARE personID INT;
    DECLARE projectID INT;

    IF EXISTS (SELECT * FROM projects WHERE name = projectName) THEN
        SET projectID = (SELECT id FROM projects WHERE name = projectName);

        IF position = 'manager' THEN
            SET personID = (SELECT card_id FROM managers WHERE full_name =
                fullName);

            IF EXISTS (SELECT * FROM managers WHERE full_name = fullName) THEN
                IF NOT EXISTS (SELECT * FROM project_managers WHERE ref_project
                    = projectID AND ref_manager = personID) THEN
                    INSERT INTO project_managers VALUES (NULL, projectID,
                        personID);
                ELSE
                    SELECT 'Duplicate query' AS '';
                END IF;
            ELSE
                SELECT 'No such manager exists' AS '';
            END IF;
        ELSEIF position = 'programmer' THEN
            SET personID = (SELECT card_id FROM programmers WHERE full_name =
                fullName);

            IF EXISTS (SELECT * FROM programmers WHERE full_name = fullName) THEN
                IF NOT EXISTS (SELECT * FROM project_programmers WHERE
                    ref_project = projectID AND ref_programmer = personID) THEN
                    INSERT INTO project_programmers VALUES
                        (NULL, projectID, personID);
                ELSE
                    SELECT 'Duplicate query' AS '';
                END IF;
            ELSE
                SELECT 'No such programmer exists' AS '';
            END IF;
        ELSEIF position = 'designer' THEN
            SET personID = (SELECT card_id FROM designers WHERE full_name =
                fullName);

            IF EXISTS (SELECT * FROM designers WHERE full_name = fullName) THEN
                IF NOT EXISTS (SELECT * FROM project_designers WHERE
                    ref_project = projectID AND ref_designer = personID) THEN
                    INSERT INTO project_designers VALUES (NULL,
                        projectID, personID);
                ELSE
                    SELECT 'Duplicate query' AS '';
                END IF;
            ELSE
                SELECT 'No such designer exists' AS '';
            END IF;
        ELSE
            SELECT 'No such position exists' AS '';
        END IF;
    ELSE
        SELECT 'No such project exists' AS '';
    END IF;
END
```

```

                SELECT 'No such position exists' AS '';
            END IF;
        ELSE
            SELECT 'No such project exists' AS '';
        END IF;
    END$$

DELIMITER ;

```

2. Создать процедуру, которая возвращала бы суммарную зарплату всех сотрудников и их количество.

```

DELIMITER $$

CREATE PROCEDURE totalSalary (
    OUT sal FLOAT,
    OUT num INT
)
BEGIN
    SET sal = (SELECT (SELECT sum(salary) FROM managers) +
                (SELECT sum(salary) FROM programmers) +
                (SELECT sum(salary) FROM designers));

    SET num = (SELECT (SELECT count(*) FROM managers) +
                (SELECT count(*) FROM programmers) +
                (SELECT count(*) FROM designers));
END$$

DELIMITER ;

```

3. Создать процедуру, которая возвращала бы среднюю зарплату всех сотрудников (используя процедуру 2.).

```

DELIMITER $$

CREATE PROCEDURE averageSalary (
    OUT avSal FLOAT
)
BEGIN
    DECLARE sal FLOAT;
    DECLARE num INT;

    CALL totalSalary(sal, num);

    SET avSal = sal / num;
END$$

DELIMITER ;

```