

Лекция 10

Методика проектирования базы данных

Проектирование БД – процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи:

- Хранение в БД всей необходимой информации.
- Получение данных по всем требующимся запросам.
- Сокращение избыточности и дублирования данных.
- Обеспечение целостности данных (исключение противоречий, предотвращение потерь и т.д.).

Проектирование реляционной БД

Основные задачи:

- Определить нужные таблицы с учетом нормализации БД
- Указать первичные и внешние ключи
- Связать нужные таблицы друг с другом
- Определить ограничения для обеспечения целостности данных

Ожидаемый результат:

- ER-диаграмма базы данных
- Схема базы данных в нотации IDEF1x
- SQL-скрипты для создания БД в конкретной СУБД

Нужна методика, алгоритм

Методика проектирования реляционной БД

1. Изучаем процессы предметной области
 2. Определяем сущности и их атрибуты
 3. Устанавливаем связи между сущностями
 4. Определяем типы связей
 5. Переходим от сущностей к таблицам
 6. Строим связи между таблицами
 7. Проверяем нормализацию
 8. Определяемся с СУБД
-
- ER-диаграмма
- Диаграмма IDEF1x
- SQL-скрипты

Терминология

- **Предметная область** – ограниченная область реального мира. База данных - это модель (абстракция, упрощение) предметной области.
- **Сущность предметной области** – абстрактное понятие, которое объединяет группу реальных объектов, имеющих одинаковые свойства.
- **Экземпляры сущности** - реальные объекты, объединенные в одну сущность.
- **Атрибуты сущности** – свойства, которыми описывается сущность.

Пример

Предметная область – факультет математики и ИТ Мордовского госуниверситета.

Нужна БД для учета студентов, преподавателей, администрации и успеваемости студентов.

Шаг 1. Изучаем процессы предметной области

Кто и какие действия выполняет, какие документы используются в процессе выполнения.

Студенты:

- учатся на определённой специальности;
- изучают дисциплины, входящие в эту специальность;
- в течение семестра набирают баллы;
- в сессию сдают экзамены.

Преподаватели:

- работают на определённой кафедре;
- читают лекции по дисциплинам;
- проводят л/р;
- принимают экзамены.

Шаг 1. Изучаем процессы предметной области

На какие вопросы нужно ответить с помощью базы данных

- Статистика по студентам и преподавателям
- Анализ успеваемости
- Расписание занятий и экзаменов
- ...

Шаг 2. Определяем сущности

Сущность предметной области – абстрактное понятие, которое объединяет группу реальных объектов, имеющих одинаковые свойства.

СТУДЕНТ

ПРЕПОДАВАТЕЛЬ

ДЕКАН

ДЕКАНАТ

СЕКРЕТАРЬ ДЕКАНАТА

КАФЕДРА

ЛАБОРАТОРИЯ

АУДИТОРИЯ

ЭКЗАМЕН

ОЦЕНКА

КОМПЬЮТЕР В ЛАБОРАТОРИИ

КОНДИЦИОНЕР

Шаг 2. Определяем сущности

Сущность предметной области – абстрактное понятие, которое объединяет группу реальных объектов, имеющих одинаковые свойства.

СТУДЕНТ

ПРЕПОДАВАТЕЛЬ

ДЕКАН

ДЕКАНАТ

СЕКРЕТАРЬ ДЕКАНАТА

КАФЕДРА

ЛАБОРАТОРИЯ

АУДИТОРИЯ

ЭКЗАМЕН

ОЦЕНКА

КОМПЬЮТЕР В ЛАБОРАТОРИИ

КОНДИЦИОНЕР

Шаг 2. Выделяем атрибуты сущностей

СТУДЕНТ

Номер зачетной книжки
ФИО
Дата рождения
Место проживания
Дата поступления
Форма обучения

ПРЕПОДАВАТЕЛЬ

Табельный номер
ФИО
Дата рождения
Место проживания
Кафедра
Должность

КАФЕДРА

Название
Заведующий

Дисциплина

Код дисциплины
Название

Шаг 3. Устанавливаем связи между сущностями

Структура описания связи

Подлежащее сказуемое дополнение

Студент изучает дисциплину.

Преподаватель принимает экзамен.

Студент сдает экзамен.

Шаг 4. Определяем типы связей между сущностями

- **Связь 1:1.** Одиночный экземпляр сущности одного класса связан с одиночным экземпляром сущности другого класса.
- **Связь 1:M.** Единый экземпляр сущности одного класса связан со многими экземплярами сущности другого класса.
- **Связь M:N.** Несколько экземпляров сущности одного класса связаны с несколькими экземплярами сущности другого класса.

Структура описания связи

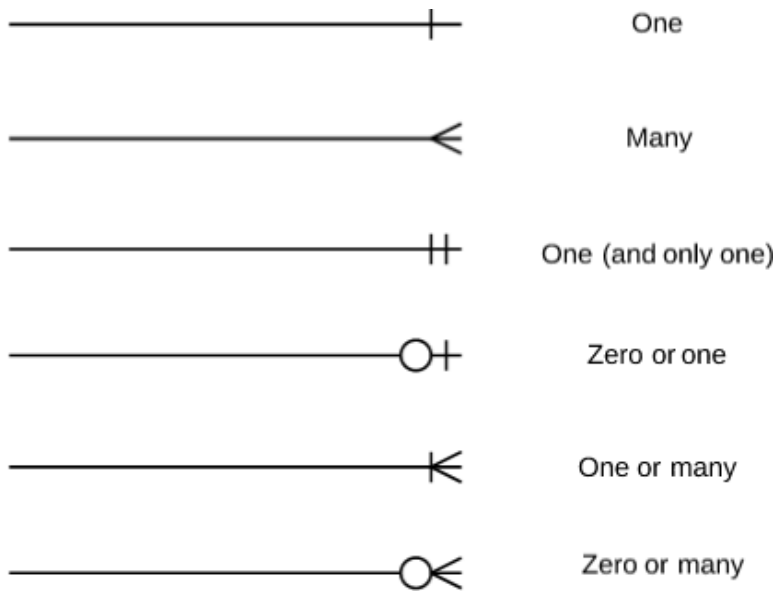
Один студент изучает много дисциплин.

Одну дисциплину изучает много студентов.

Один преподаватель работает на одной кафедре.

Одна кафедра включает много преподавателей.

Результат шагов 1-4: ER-диаграмма



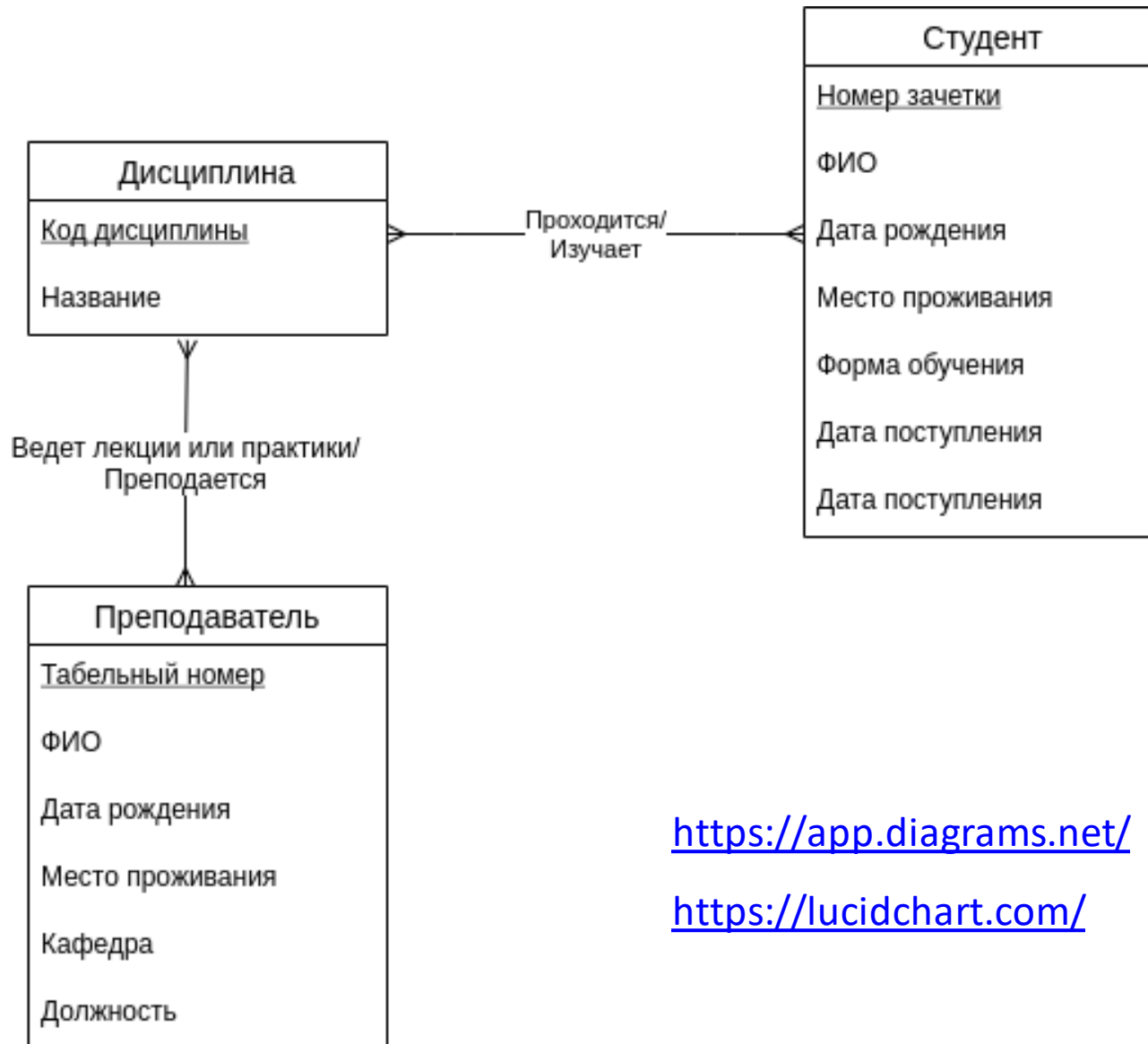
Сущности

- атрибуты
- первичные ключи

Связи между сущностями

- типы

Результат шагов 1-4: ER-диаграмма



<https://app.diagrams.net/>

<https://lucidchart.com/>

Шаг 5. Переходим от сущностей к таблицам

Одна сущность - одна таблица

Для каждой сущности выделяем **первичный ключ** (Primary Key, PK).

Естественный одиночный PK - это атрибут, значение которого:

- не может быть пустым
- уникально для каждого экземпляра сущности

Составной PK - состоит из комбинации атрибутов

Синтетический PK - искусственный атрибут с порядковой нумерацией

Шаг 6. Строим связи между таблицами

Правило 1. Для построения связи «**один ко многим**» нужно скопировать первичный ключ из одной главной таблицы в подчиненную.

Новое неуникальное поле в подчиненной таблице называется **внешним ключом** (Foreign Key, FK)

Правило 2. Для построения связи «**многие ко многим**» нужно создать дополнительную ассоциирующую таблицу и скопировать туда первичные ключи двух исходных таблиц в качестве внешних ключей.

Правило 3. Для построения связи «**один к одному**» нужно скопировать первичный ключ из одной таблицы в другую, причем во второй таблице это поле должно стать первичным ключом.

Шаг 6. Строим связи между таблицами

Связь «Один ко многим»

Таблица «Кафедры»

Код кафедры (PK)	Название
100	Кафедра фундаментальной информатики
101	Кафедра систем автоматизированного проектирования
102	Кафедра математического анализа, алгебры и геометрии



Табельный номер (PK)	Код кафедры (FK)	Фамилия
1234	100	Иванов
1237	102	Петров
3456	100	Сидоров

Таблица «Сотрудники»

Шаг 6. Строим связи между таблицами

Таблица «Дисциплины»

Код дисциплины (PK)	Название
100	Базы данных
101	ООП
102	Языки программирования

Таблица «Студенты»

Код студента (PK)	Фамилия
200	Антонов
202	Белов
210	Волков

Связь «Многие ко многим»

Код дисциплины (FK)	Код студента (FK)
100	200
101	200
100	202

Шаг 6. Строим связи между таблицами

Таблица «Дисциплины»

Код дисциплины (PK)	Название
100	Базы данных
101	ООП
102	Языки программирования

Таблица «Студенты»

Код студента (PK)	Фамилия
200	Антонов
202	Белов
210	Волков

Связь «Многие ко многим»

Таблица «Экзамены»

Код экзамена (PK)	Код дисциплины (FK)	Код студента (FK)	Дата	Баллы
100	100	200	25.05.2020	86
101	101	200	28.05.2020	58
210	100	202	30.05.2020	76

**Дополнительная
сущность**

Шаг 6. Строим связи между таблицами

Связь «один к одному» используется редко, т.к. две таблицы с одинаковым первичным ключом можно объединить.

Нужна для построения **отношения категоризации**.

Таблица «Оборудование»

Код единицы (PK)	Категория	Процессор	ОЗУ	ПЗУ	Формат	Тип
1	Компьютер	Intel Core i5	DDR3 8	SSD 500		
2	Компьютер	Intel Core i7	DDR4 8	HDD 1000		
3	Компьютер	AMD A10	DDR4 16	SSD 250		
4	Принтер				A4	Лазерный
5	Принтер				A4	Струйный
6	Принтер				A3	Лазерный

Шаг 6. Строим связи между таблицами

Таблица «Оборудование»

Код единицы (ПК)	Категория
1	Компьютер
2	Компьютер
3	Компьютер
4	Принтер
5	Принтер
6	Принтер

**Связь «Один к одному»
для категоризации**

Таблица «Компьютеры»

Код компьютера (ПК)	Процессор	ОЗУ	ПЗУ
1	Intel Core i5	DDR3 8	SSD 500
2	Intel Core i7	DDR4 8	HDD 1000
3	AMD A10	DDR4 16	SSD 250

Таблица «Принтеры»

Код принтера (ПК)	Формат	Тип
4	A4	Лазерный
5	A4	Струйный
6	A3	Лазерный

Шаг 7. Проверяем нормализацию

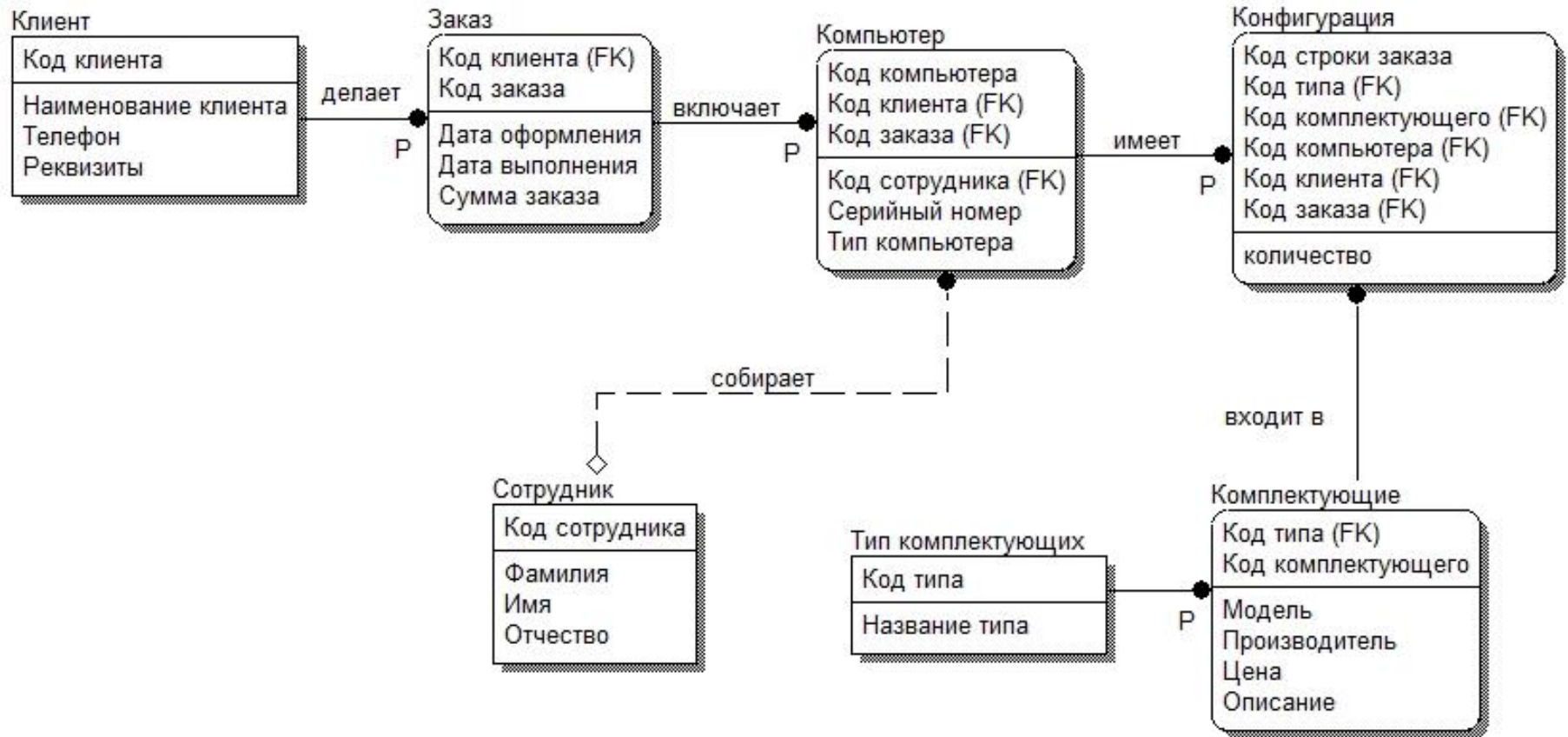
Первая НФ требует, чтобы каждое поле таблицы было неделимым (не должно делиться на более мелкие значения) и не содержало повторяющихся групп.

Таблица находится во **второй НФ**, если она находится в первой НФ и не содержит полей, которые зависят от части первичного ключа. Те поля, которые зависят только от части первичного ключа, должны быть выделены в отдельные таблицы.

Третья НФ требует, чтобы таблица была во второй НФ и в ней не было транзитивных зависимостей между неключевыми полями.

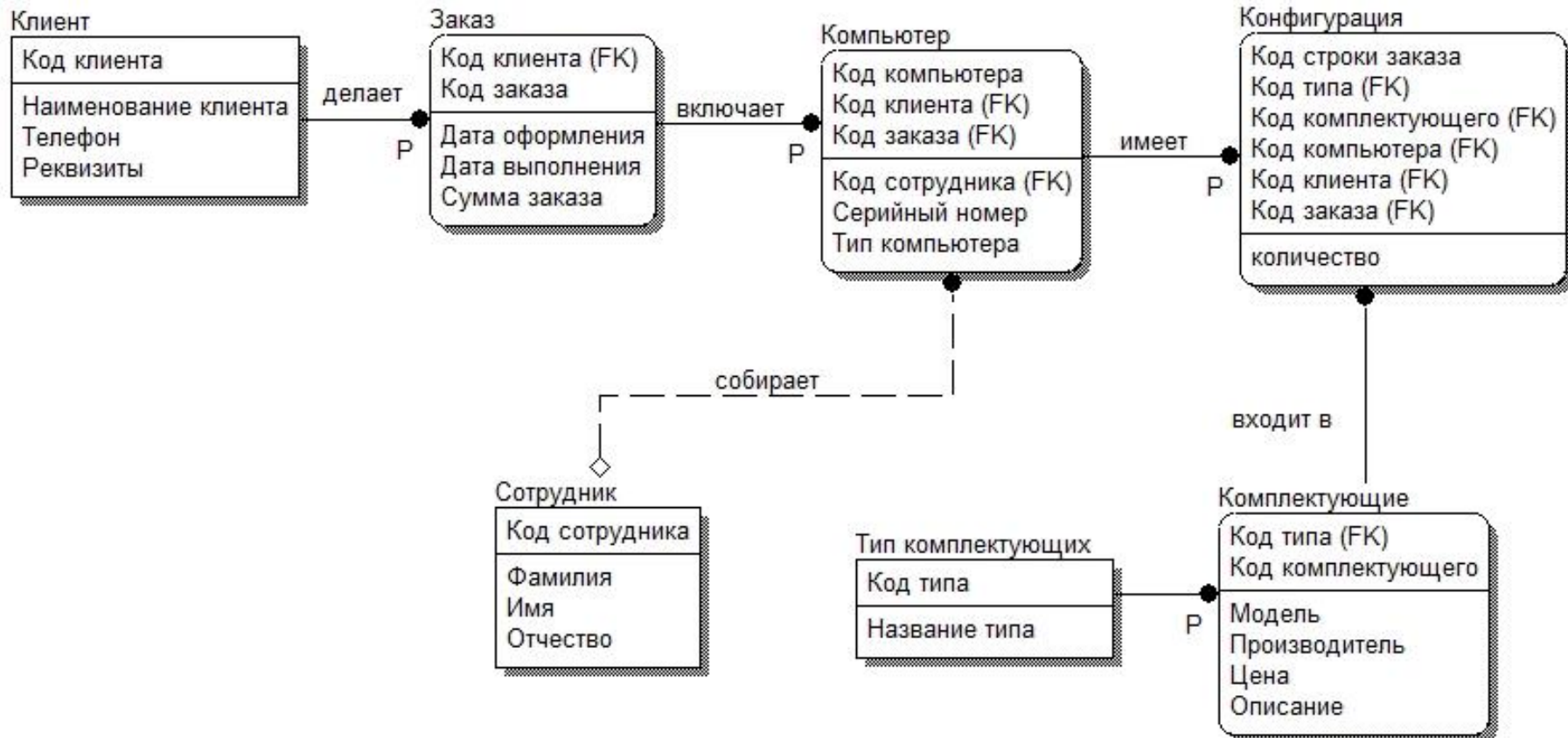
Нарушение второй или третьей НФ означает, что при проектировании была пропущена сущность или атрибуты были распределены по сущностям неправильно.

Результат шагов 5-7: диаграмма IDEF1X



Нотация IDEF1X

Зависимая (дочерняя) таблица изображается с закругленными углами.



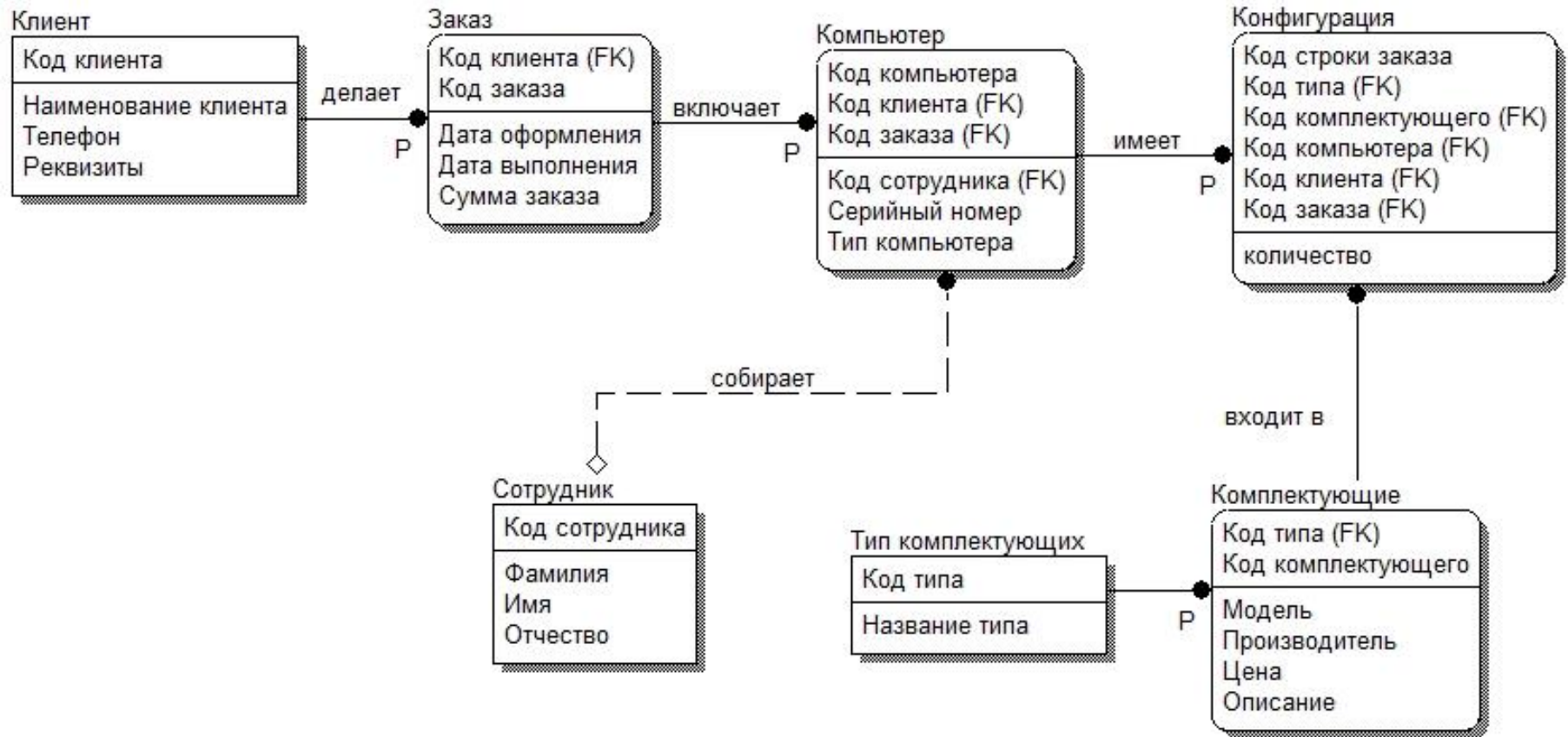
Нотация IDEF1X

Виды связей:

- **Идентифицирующая.** Внешний ключ в дочерней таблице входит в состав первичного ключа.
- **Обязательная.** При вводе в новой строки в дочернюю таблицу заполнение полей внешнего ключа обязательно, причем введённые значения должны совпадать с полями первичного ключа какой-либо записи в родительской таблице.
- **Необязательная.** Заполнение полей внешнего ключа в дочерней таблице необязательно или введённые значения могут не совпадать со значениями полей первичного ключа ни в одной записи родительской таблицы.

Внешний вид	Тип и обязательность связи	Мощность связи справа
—————	Обязательная, идентифицирующая	1
—————●	Обязательная, идентифицирующая	0 .. ∞
—————● Z	Обязательная, идентифицирующая	0 или 1
—————● P	Обязательная, идентифицирующая	1 .. ∞
—————● <число>	Обязательная, идентифицирующая	<число>
-----●	Обязательная, неидентифицирующая	0 .. ∞
◇-----●	Необязательная, неидентифицирующая	0 .. ∞

Результат шагов 5-7: диаграмма IDEF1X



Шаг 8. Определяемся с СУБД

- Выбираем СУБД, где будет создана база данных.
- Уточняем типы и размерности полей таблиц.
- Пишем SQL-скрипты создания таблиц для выбранной СУБД.

Методика проектирования реляционной БД

1. Изучаем процессы предметной области
 2. Определяем сущности и их атрибуты
 3. Устанавливаем связи между сущностями
 4. Определяем типы связей
 5. Переходим от сущностей к таблицам
 6. Строим связи между таблицами
 7. Проверяем нормализацию
 8. Определяемся с СУБД
-
- ER-диаграмма
- Диаграмма IDEF1x
- SQL-скрипты