



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

Базы данных

Лекция 3 Проектирование БД

Агафонов Антон Александрович
к.т.н., доцент кафедры ГИИБ

Самара



- Этапы разработки БД
- ER-диаграммы
- Нормализация отношений





Информационная система (Information System) – это система, предназначенная для хранения, поиска и обработки информации.

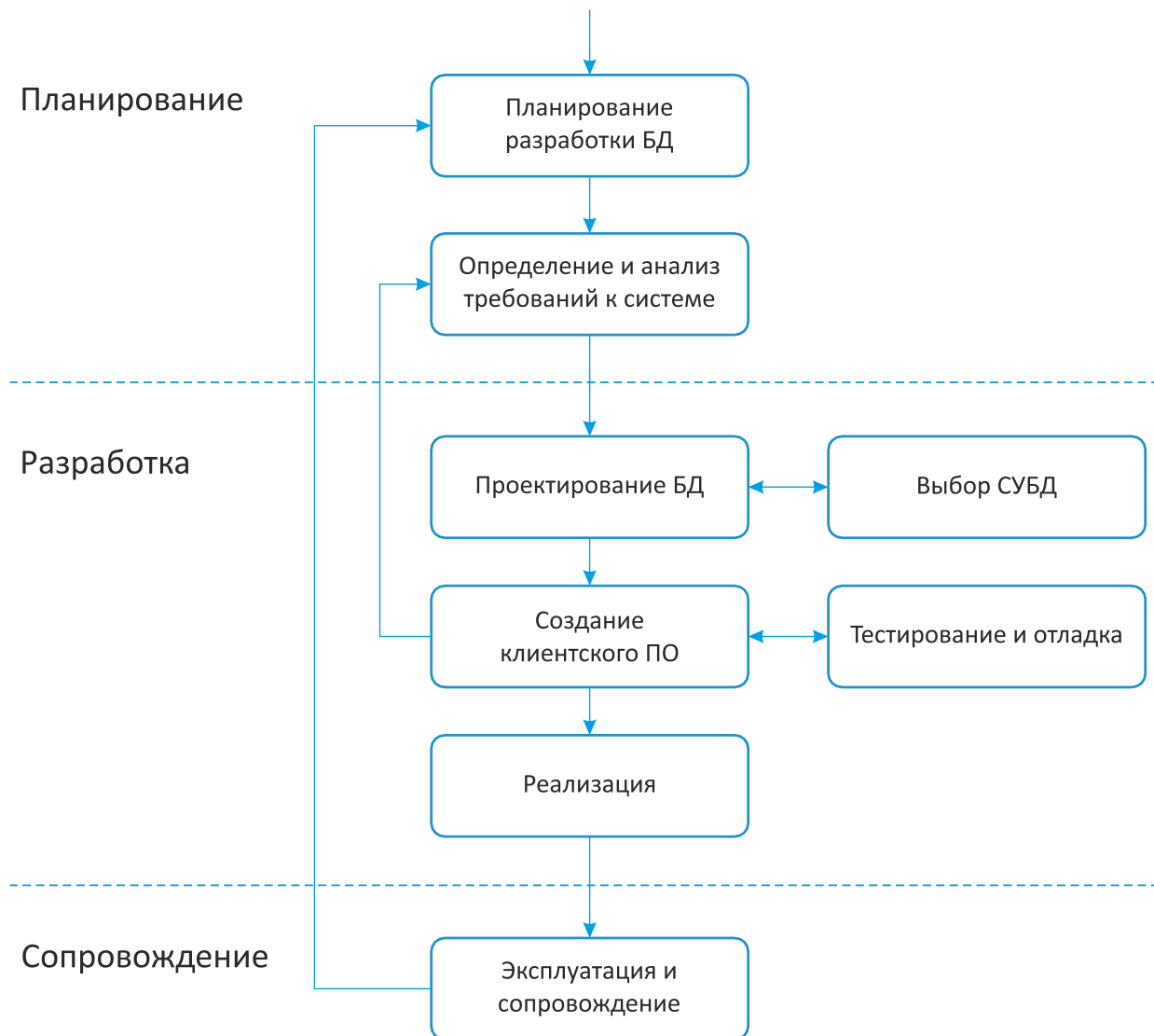
Процесс создания информационной системы в общем случае включает этапы:

- 1) планирование системы;
- 2) анализ задач системы и требований к системе;
- 3) проектирование системы;
- 4) реализация и ввод системы в эксплуатацию;
- 5) сопровождение.



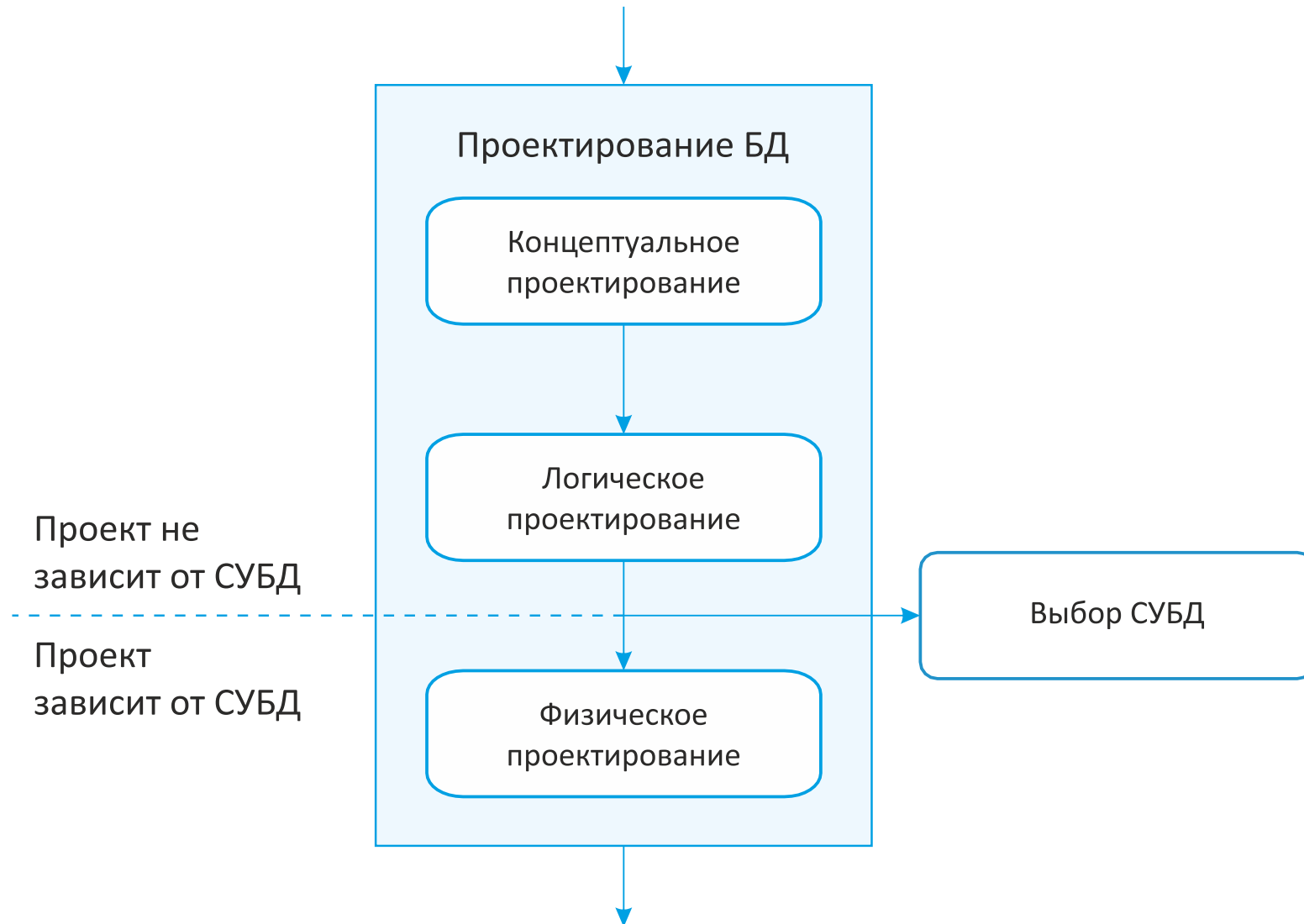


Жизненный цикл БД





Этап проектирования БД





Основным средством построения концептуальной модели БД выступает **модель «сущность-связь» (ER – Entity-Relation)** или родственные модели.

Графические нотации создания ER-диаграмм:

- нотация Чена;
- Crow's Foot;
- IDEF1X;
- и т.д.

Результатом фазы концептуального проектирования является ER-модель БД, включающая в себя описание:

- типов сущностей;
- связей между типами сущностей;
- атрибутов (желательно с предварительным описанием доменов и ограничений);
- первичных ключей.





Логическое проектирование предназначено для преобразования обобщенной концептуальной модели в завершенную логическую.

На этом этапе уточняются требования, выявленные на концептуальной стадии проектирования, и выполняется упрощение модели (без снижения функциональных возможностей).

На логическом этапе осуществляются следующие действия:

- выполняется нормализация таблиц для получения не избыточных реляционные таблицы, свободные от присущих ненормализованным данным аномалий вставки, редактирования и удаления;
- уточняются ограничения на данные;
- определяются домены данных;
- вводятся бизнес-правила и корпоративные ограничения целостности;
- определяется местоположение будущих таблиц (в случае если речь идет о распределенной БД).





На этапах концептуального и логического проектирования обычно пользуются одной из двух стратегий проектирования БД: стратегией **восходящего проектирования** (bottom-up design) или стратегией **нисходящего проектирования** (top-down design).

Суть восходящего метода заключается в том, что сначала формируется полный список атрибутов (столбцов таблиц), подлежащих хранению, в одном отношении. Позднее атрибуты группируются в типы сущностей, которые попадают в модель, т.е. выполняется процесс нормализации.

В нисходящей стратегии проектирование начинается с выявления основных типов сущностей, и затем для сущностей определяются атрибуты. Классический пример нисходящего метода – модель сущность-связь (ER-модель).





Физическое проектирование – это уточнение решения с учетом имеющихся в наличии технологий, возможности реализации и требуемой производительности.

Целью физического проектирования становится перенос логической модели на платформу выбранной СУБД. Для этого необходимо:

- создать таблицы и связи между ними;
- создать вторичные индексы таблиц;
- разработать представления;
- реализовать бизнес-логику БД (в первую очередь, с помощью триггеров и хранимых процедур);
- определить функциональные характеристики транзакций;
- внедрить механизмы защиты (как минимум, предусмотреть авторизацию пользователей и назначить права доступа к данным).





Сущность – это класс однотипных объектов, информация о которых должна быть учтена в модели.

Экземпляр сущности – это конкретный представитель данной сущности. Экземпляры сущностей должны быть *различимы*, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

Атрибут сущности – это именованная характеристика, являющаяся некоторым свойством сущности.

Покупатель
Номер покупателя
Имя покупателя
Адрес
Банковские реквизиты



Ключ сущности – это *неизбыточный* набор атрибутов, значения которых в совокупности являются *уникальными* для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность.

Связь – это некоторая ассоциация между *двумя* сущностями. Одна сущность может быть связана с другой сущностью или сама с собою. Связи позволяют по одной сущности находить другие сущности, связанные с нею.

1:1 —————

Может- - - - -

1:M —————<

Должен —————

N:M >—————<



При разработке ER-моделей необходимо получить следующую информацию о предметной области:

1. Список сущностей предметной области.
2. Список атрибутов сущностей.
3. Описание взаимосвязей между сущностями.

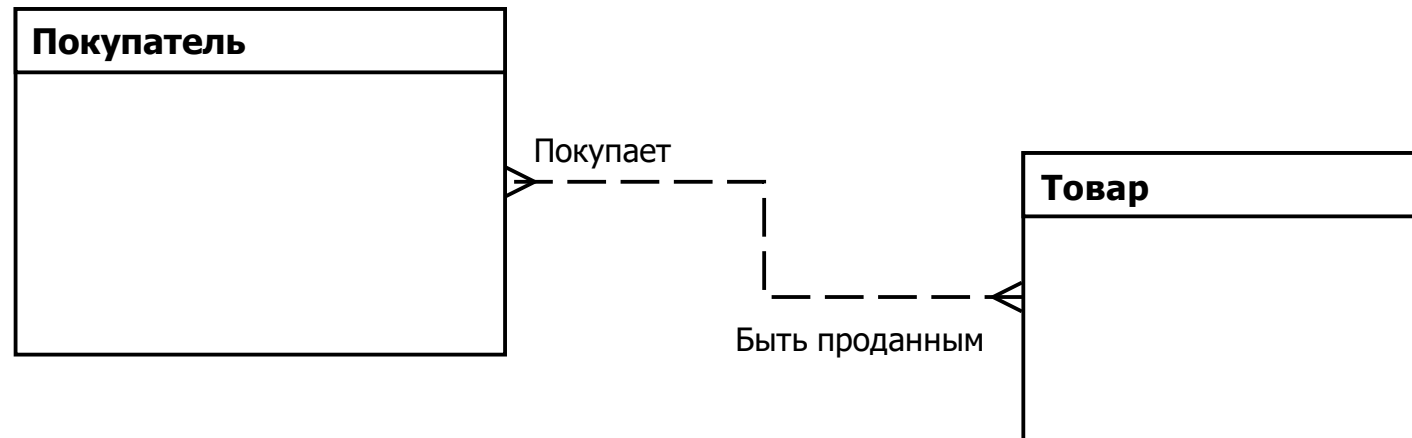
Неформальная модель предметной области «Оптовая торговая фирма»:

- Хранить информацию о покупателях.
- Печатать накладные на отпущенные товары.
- Следить за наличием товаров на складе.



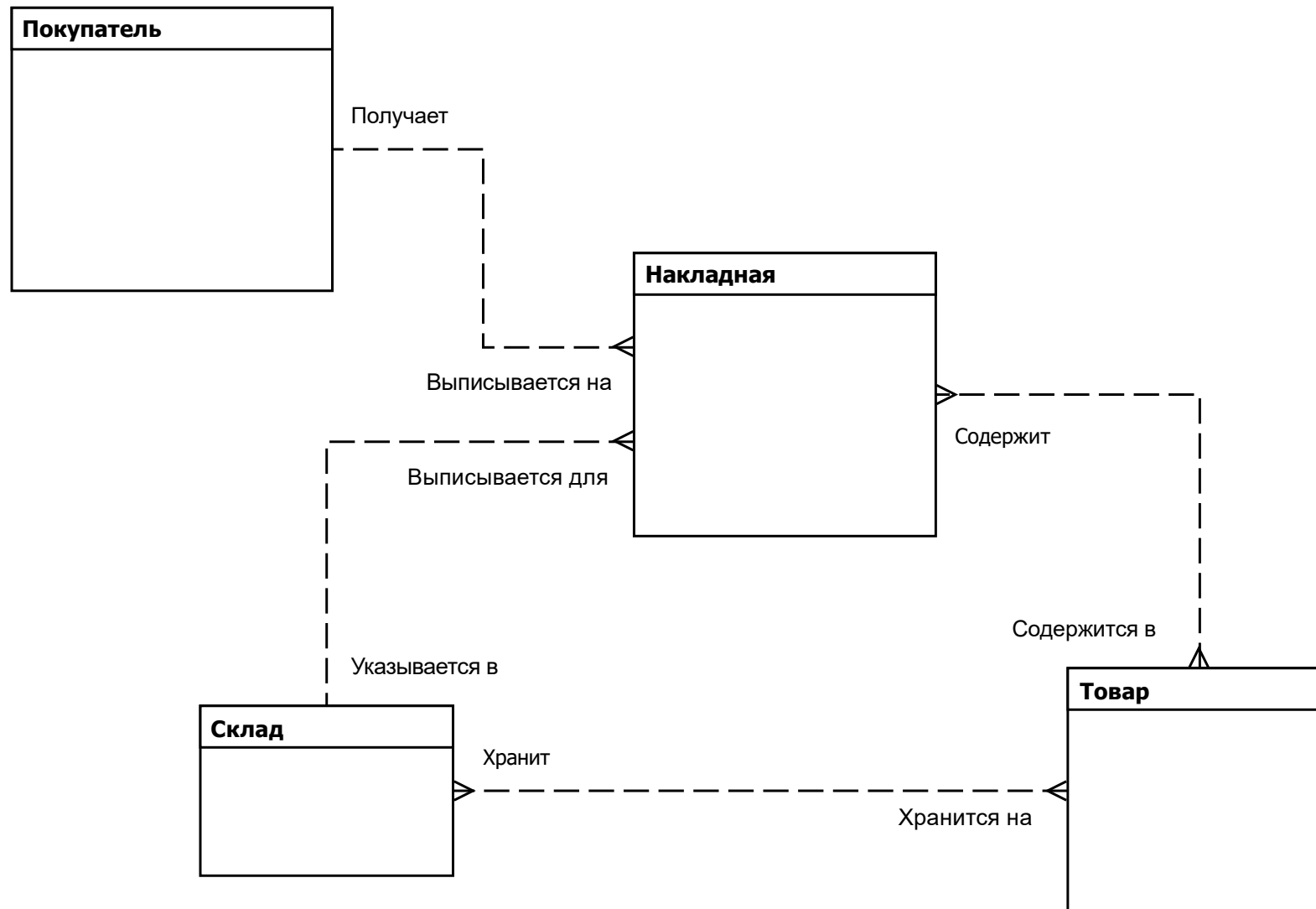
Потенциальные сущности:

- *Покупатель* – явный кандидат на сущность.
- *Накладная* – явный кандидат на сущность.
- *Товар* – явный кандидат на сущность
- (?) *Склад* – сколько складов имеет фирма? Если несколько, то это будет кандидатом на новую сущность.
- (?) *Наличие товара* – это, скорее всего, атрибут, но атрибут какой сущности?



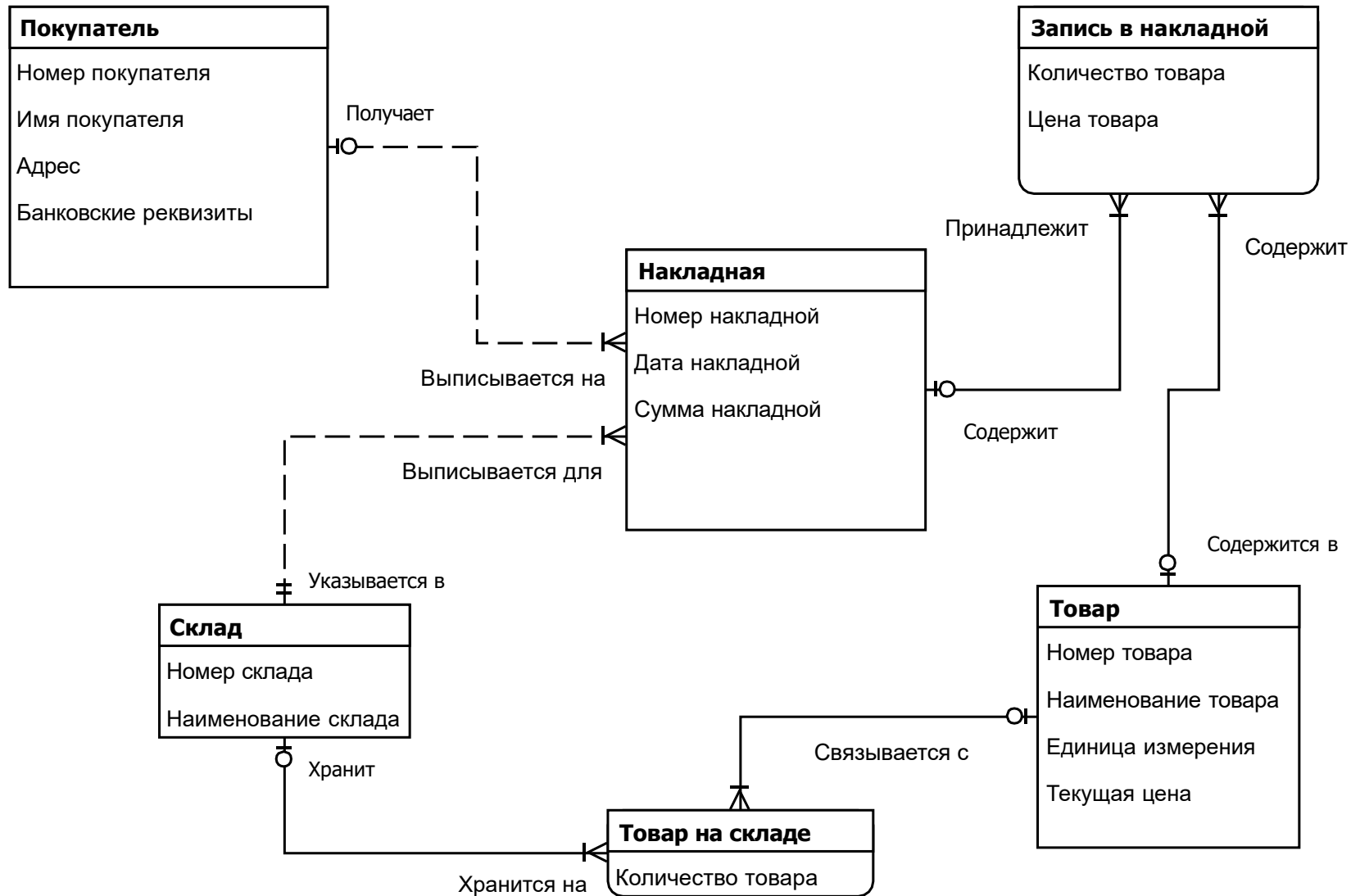


Пример разработки ER-модели



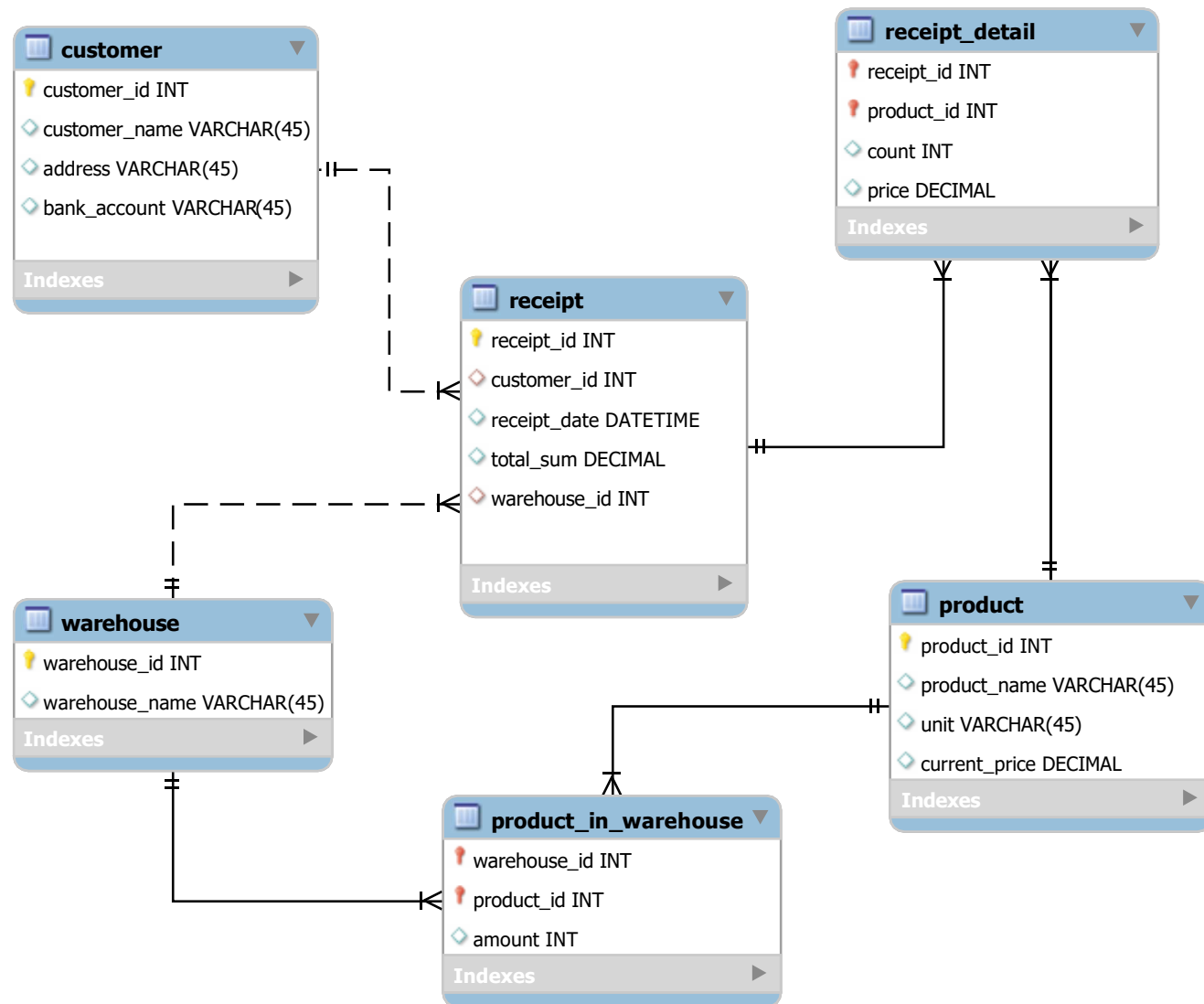


Пример разработки ER-модели





Пример разработки ER-модели





Процесс нормализации и ER-моделирование выступают разными сторонами одной медали. Оба подхода приводят к созданию модели БД:

- ER-моделирование осуществляется на концептуальном уровне и основано на принципе от общего к частному (так называемый нисходящий подход);
- нормализация производится на этапе логического проектирования и, в отличие от ER-моделирования, использует восходящий подход (от частного к общему).



Неформальная модель предметной области «Организация, выполняющая проекты»:

1. Сотрудники организации выполняют проекты.
2. Проекты состоят из нескольких задач.
3. Каждый сотрудник может участвовать в одном или нескольких проектах, или временно не участвовать ни в каких проектах.
4. Над каждым проектом может работать несколько сотрудников, или временно проект может быть приостановлен, тогда над ним не работает ни один сотрудник.
5. Над каждой задачей в проекте работает ровно один сотрудник.
6. Каждый сотрудник числится в одном отделе.
7. Каждый сотрудник имеет телефон, находящийся в отделе сотрудника.

Дополнительные сведения:

1. О каждом сотруднике необходимо хранить табельный номер (уникальный) и фамилию.
2. Каждый отдел имеет уникальный номер.
3. Каждый проект имеет номер и наименование. Номер проекта является уникальным.
4. Каждая задача из проекта имеет номер, уникальный в пределах проекта. Задачи в разных проектах могут иметь одинаковые номера.





Нормализация отношений. Предметная область

Номер проекта	Название проекта	Номер сотрудника	Фамилия сотрудника	Номер отдела	Телефон	Номер задания
1	Космос	1	Иванов	1	11-22-33	1
		2	Петров	1	11-22-33	2
		3	Сидоров	2	33-22-11	3
2	Климат	3	Сидоров	2	33-22-11	2
		1	Иванов	1	11-22-33	1





Первая нормальная форма (1НФ)

Отношение находится в **первой нормальной форме** тогда и только тогда, когда каждый кортеж отношения содержит только одно значение для каждого из атрибутов.

Согласно определению отношений, любое отношение автоматически уже находится в 1НФ согласно свойствам отношений:

- в отношении нет одинаковых кортежей;
- кортежи не упорядочены;
- атрибуты не упорядочены и различаются по наименованию;
- все значения атрибутов атомарны.

Таблица приведена к первой нормальной форме, если в ней отсутствуют повторяющиеся группы и все значения, хранимые в ней, неделимы (атомарны).



Первая нормальная форма

В ходе логического моделирования на первом шаге предложено хранить данные в одном отношении, имеющем следующие атрибуты:

СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ (
 НОМЕР_СОТРУДНИКА, ФАМИЛИЯ, НОМЕР_ОТДЕЛА, ТЕЛЕФОН,
 НОМЕР_ПРОЕКТА, ПРОЕКТ, НОМЕР_ЗАДАЧИ
)

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН	НОМЕР_ПРОЕКТА	ПРОЕКТ	НОМЕР_ЗАДАЧИ
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2





НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН	НОМЕР_ПРОЕКТА	ПРОЕКТ	НОМЕР_ЗАДАЧИ
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2

Аномалия обновления – неадекватность модели данных предметной области или необходимость дополнительных усилий для реализации всех ограничений, определенных в предметной области.

Различают следующие виды аномалий:

- Аномалии вставки (INSERT)
- Аномалии обновления (UPDATE)
- Аномалии удаления (DELETE)



Аномалии вставки (INSERT)

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН	НОМЕР_ПРОЕКТА	ПРОЕКТ	НОМЕР_ЗАДАЧИ
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2
4	Семенов	2	33-22-11	NULL	NULL	NULL

В данное отношение нельзя вставить запись, которая не содержит значения в столбце НОМЕР_ПРОЕКТА, т.к. этот атрибут входит в состав потенциального ключа, и, следовательно, не может содержать NULL-значений.

Причина аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и о проектах, и о работах по проекту).

Вывод: логическая модель данных неадекватна модели предметной области. База данных, основанная на такой модели, будет работать неправильно.



Аномалии обновления (UPDATE)

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН	НОМЕР_ПРОЕКТА	ПРОЕКТ	НОМЕР_ЗАДАЧИ
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2

Обновление избыточных данных невозможно реализовать одним действием. Для поддержания отношения в целостном состоянии необходимо написать триггер, который при обновлении одной записи корректно исправлял бы данные и в других местах.

Причина аномалии - избыточность данных, также порожденная тем, что в одном отношении хранится разнородная информация.

Вывод: увеличивается сложность разработки базы данных. База данных, основанная на такой модели, будет работать правильно только при наличии дополнительного программного кода в виде триггеров.





Аномалии удаление (DELETE)

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН	НОМЕР_ПРОЕКТА	ПРОЕКТ	НОМЕР_ЗАДАЧИ
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2

При удалении некоторых данных может произойти потеря другой информации. Если по проекту временно прекращены работы, то при удалении данных о работах по этому проекту будут удалены и данные о самом проекте (наименование проекта). При этом если был сотрудник, который работал только над этим проектом, то будут потеряны и данные об этом сотруднике.

Причина аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и о проектах, и о работах по проекту).

Вывод: логическая модель данных неадекватна модели предметной области. База данных, основанная на такой модели, будет работать неправильно.





Вторая нормальная форма (2НФ)

Отношение находится во **второй нормальной форме** тогда и только тогда, когда оно находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от (каждого) её потенциального ключа. **Неключевой атрибут** - это атрибут, не входящий в состав никакого потенциального ключа.

Таблица приведена ко второй нормальной форме, если она соответствует 1НФ и в ней нет частичных зависимостей, т. е. нет неключевых столбцов, зависящих от части первичного ключа.

Если потенциальный ключ является простым, то есть состоит из единственного атрибута, то любая функциональная зависимость от него является неприводимой (полной). Если потенциальный ключ является составным, то, согласно определению второй нормальной формы, в отношении не должно быть неключевых атрибутов, зависящих от части составного потенциального ключа.

Другими словами, у таблицы в 2НФ не должно быть атрибутов, зависящих только от части первичного ключа.



Вторая нормальная форма (2НФ)

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН	НОМЕР_ПРОЕКТА	ПРОЕКТ	НОМЕР_ЗАДАЧИ
1	Иванов	1	11-22-33	1	Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	1	Космос	2
3	Сидоров	2	33-22-11	1	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2

Зависимость атрибутов от **ключа отношения**:

{НОМЕР_СОТРУДНИКА, НОМЕР_ПРОЕКТ} → ФАМИЛИЯ
{НОМЕР_СОТРУДНИКА, НОМЕР_ПРОЕКТ} → НОМЕР_ОТДЕЛА
{НОМЕР_СОТРУДНИКА, НОМЕР_ПРОЕКТ} → ТЕЛЕФОН
{НОМЕР_СОТРУДНИКА, НОМЕР_ПРОЕКТ} → ПРОЕКТ
{НОМЕР_СОТРУДНИКА, НОМЕР_ПРОЕКТ} → НОМЕР_ЗАДАЧИ

Зависимость атрибутов от **номера проекта**:

НОМЕР_ПРОЕКТА → ПРОЕКТ

Зависимость атрибутов от **номера сотрудника**:

НОМЕР_СОТРУДНИКА → ФАМИЛИЯ
НОМЕР_СОТРУДНИКА → НОМЕР_ОТДЕЛА
НОМЕР_СОТРУДНИКА → ТЕЛЕФОН

Зависимость атрибутов от **номера отдела**:

НОМЕР_ОТДЕЛА → ТЕЛЕФОН





Вторая нормальная форма (2НФ)

Для того, чтобы устранить зависимость атрибутов от части сложного ключа, нужно произвести **декомпозицию** отношения на несколько отношений. При этом *те атрибуты, которые зависят от части сложного ключа*, выносятся в отдельное отношение.

Отношение

СОТРУДНИКИ_ОТДЕЛЫ (НОМЕР_СОТРУДНИКА, ФАМИЛИЯ, НОМЕР_ОТДЕЛА, ТЕЛЕФОН):

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

Функциональные зависимости:

Зависимость атрибутов от **номера сотрудника**:

НОМЕР_СОТРУДНИКА → ФАМИЛИЯ

НОМЕР_СОТРУДНИКА → НОМЕР_ОТДЕЛА

НОМЕР_СОТРУДНИКА → ТЕЛЕФОН

Зависимость атрибутов от **номера отдела**:

НОМЕР_ОТДЕЛА → ТЕЛЕФОН





Вторая нормальная форма (2НФ)

Отношение
ПРОЕКТЫ (НОМЕР_ПРОЕКТА, ПРОЕКТ):

НОМЕР_ПРОЕКТА	ПРОЕКТ
1	Космос
2	Климат

Функциональные зависимости:

НОМЕР_ПРОЕКТА → ПРОЕКТ

Отношение
ЗАДАЧИ (НОМЕР_СОТРУДНИКА,
НОМЕР_ПРОЕКТА, НОМЕР_ЗАДАЧИ):

НОМЕР_СОТРУДНИКА	НОМЕР_ПРОЕКТА	НОМЕР_ЗАДАЧИ
1	1	1
1	2	1
2	1	2
3	1	3
3	2	2

Функциональные зависимости:

{НОМЕР_СОТРУДНИКА, НОМЕР_ПРОЕКТ} →
НОМЕР_ЗАДАЧИ



СОТРУДНИКИ_ОТДЕЛЫ

НОМЕР_ СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ ОТДЕЛА	ТЕЛЕФОН
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

ПРОЕКТЫ

НОМЕР_ ПРОЕКТА	ПРОЕКТ
1	Космос
2	Климат

ЗАДАЧИ

НОМЕР_ СОТРУДНИКА	НОМЕР_ ПРОЕКТА	НОМЕР_ ЗАДАЧИ
1	1	1
1	2	1
2	1	2
3	1	3
3	2	2



СОТРУДНИКИ_ОТДЕЛЫ

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11
4	Семенов	1	33-22-11

В данное отношение нельзя вставить запись, которая содержит противоречие в данных. Например, при добавлении указанной записи два сотрудника одного отдела будут иметь разные номера телефонов.

Причина аномалии - избыточность данных, порожденная тем, что в одном отношении хранится разнородная информация (о сотрудниках и об отделах).

Вывод: увеличивается сложность разработки базы данных. База данных, основанная на такой модели, будет работать правильно только при наличии дополнительного программного кода в виде триггеров.



СОТРУДНИКИ_ОТДЕЛЫ

НОМЕР_СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ОТДЕЛА	ТЕЛЕФОН
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

Дублирование данных в кортежах отношения. Если в отделе меняется номер телефона, то такие изменения необходимо одновременно выполнить во всех местах, где этот номер телефона встречается, иначе отношение станет некорректным.

Причина аномалии - избыточность данных, также порожденная тем, что в одном отношении хранится разнородная информация.

Вывод: увеличивается сложность разработки базы данных. База данных, основанная на такой модели, будет работать правильно только при наличии дополнительного программного кода в виде триггеров.



СОТРУДНИКИ_ОТДЕЛЫ

НОМЕР_ СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ ОТДЕЛА	ТЕЛЕФОН
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

При удалении некоторых данных по-прежнему может произойти потеря другой информации.

Причина аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и об отделах).

Вывод: логическая модель данных неадекватна модели предметной области. База данных, основанная на такой модели, будет работать неправильно.



Третья нормальная форма (3НФ)

Отношение находится в **третьей нормальной форме** тогда и только тогда, когда отношение находится в 2НФ и *все неключевые атрибуты взаимно независимы*.

Атрибуты называются **взаимно независимыми**, если ни один из них не является функционально зависимым от другого.

Таблица приведена к третьей нормальной форме, если она соответствует 2НФ и в ней отсутствуют транзитивные зависимости.

Транзитивная зависимость $A \rightarrow B \rightarrow C$ устраняется единственным способом – за счет выделения атрибутов $B \rightarrow C$ (или $A \rightarrow B$) в другую таблицу.





Третья нормальная форма (3НФ)

Отношение
СОТРУДНИКИ (НОМЕР_СОТРУДНИКА,
ФАМИЛИЯ, НОМЕР_ОТДЕЛА):

НОМЕР_ СОТРУДНИКА	ФАМИЛИЯ	НОМЕР_ ОТДЕЛА
1	Иванов	1
2	Петров	1
3	Сидоров	2

Функциональные зависимости:

НОМЕР_СОТРУДНИКА → ФАМИЛИЯ
НОМЕР_СОТРУДНИКА → НОМЕР_ОТДЕЛА

Отношение
ОТДЕЛЫ (НОМЕР_ОТДЕЛА, ТЕЛЕФОН):

НОМЕР_ ОТДЕЛА	ТЕЛЕФОН
1	11-22-33
2	33-22-11

Функциональные зависимости:

НОМЕР_ОТДЕЛА → ТЕЛЕФОН





Шаг 1 (Приведение к 1НФ). На первом шаге задается одно или несколько отношений, отображающих понятия предметной области. По модели предметной области выписываются обнаруженные функциональные зависимости. Все отношения автоматически находятся в 1НФ.

Шаг 2 (Приведение к 2НФ). Если в некоторых отношениях обнаружена зависимость атрибутов от части сложного ключа, то проводим декомпозицию этих отношений на несколько отношений следующим образом: те атрибуты, которые зависят от части сложного ключа, выносятся в отдельное отношение вместе с этой частью ключа. В исходном отношении остаются все ключевые атрибуты.

Шаг 3 (Приведение к 3НФ). Если в некоторых отношениях обнаружена зависимость некоторых неключевых атрибутов от других неключевых атрибутов, то проводим декомпозицию этих отношений следующим образом: те неключевые атрибуты, которые зависят от других неключевых атрибутов, выносятся в отдельное отношение.



- Основным средством проектирования является использование **моделей «сущность-связь»** (ER-моделей). ER-диаграммы позволяют использовать наглядные графические обозначения для моделирования сущностей и их взаимосвязей.
- Различают **концептуальные** и **физические** ER-диаграммы. Концептуальные диаграммы не учитывают особенностей конкретных СУБД. Физические диаграммы строятся по концептуальным и представляют собой прообраз конкретной базы данных. Сущности, определенные в концептуальной диаграмме, становятся таблицами, атрибуты становятся колонками таблиц (при этом учитываются допустимые для данной СУБД типы данных и наименования столбцов), связи реализуются путем **миграции** ключевых атрибутов родительских сущностей и создания внешних ключей.
- При правильном определении сущностей, полученные таблицы будут сразу находиться в 3НФ. Основное достоинство метода состоит в том, что модель строится методом последовательных уточнений первоначальных диаграмм.



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

**БЛАГОДАРЮ
ЗА ВНИМАНИЕ**

Агафонов А.А.
к.т.н., доцент кафедры ГИИБ