Основные методики моделирования

Курс “Аналитический SQL”
2020H2 А. И. Майсурадзе

Реализация основных типов связей

От отношения к логической модели

Для логической модели необходимо описать способы реализации основных типов связей:

1. Один к одному
2. Один ко многим
3. Многие ко многим

Реализацию проводим с использованием первичных и внешних ключей.

Один ко многим: 1,1-0,M

Удобнее всего с отношением 1,1-0,M

- каждый счет связан с одним и только одним клиентом
- у каждого клиента может быть 0 и более счетов

Это внешний ключ из *дочерней* таблицы “счета” в *родительскую* таблицу “клиенты”

Метку NULL для такого внешнего ключа стандартно интерпретируем: “у счета клиент есть, но не помним”

- В работе банка это ошибка, соответственно, такой внешний ключ определяется как NOT NULL

Один ко многим: 0,1-0,M

Если в банке есть обезличенные счета, то можно говорить об отношении 0,1-0,M (хотя точнее это 2 подкласса счетов)

И вот тут среди людей есть **принципиально разные мнения**

- Кто-то говорит, что NULL для внешнего ключа - это и есть обезличенный счет, ведь вариант “не знаю” запрещён.

Если для 1,1-0,M nullable FK обычно неуместно, то для 0,1-0,M иметь nullable FK просто опасно, т. к. “ошибочные” NULL получают ложную интерпретацию.

Если проект совсем маленький и короткоживущий, то такое использование nullable FK может ускорить моделирование и упростить написание запросов.

Один ко многим: 0,1-0,M

- Кто-то говорит, что должен быть фиктивный клиент “unassigned”. При этом FK NOT NULL.

Для многих СУБД вариант “unassigned” быстрее всего работает, но при моделировании обычно еще рано оптимизацией заниматься.

- В теории рекомендуют создать третью развязочную таблицу с FK UNIQUE NOT NULL на счет, т. е. shared PK у развязочной таблицы и счетов. Оба FK NOT NULL, никаких фиктивных клиентов нет.

Проблема nullable FK бесконечно обсуждается в теории моделирования. Понятно, что без них можно обойтись. Вопрос в том, бывает ли от них прок. В любом случае надо проявлять с ними большую осторожность.

Один ко многим: 1,1-1,M

Для каждой заявки ровно один заявитель
Для каждого заявителя хотя бы одна заявка

“Лучший” ответ

Отношение вопрос-ответ:

- Каждый ответ отвечает на конкретный вопрос
- На каждый вопрос может быть (0,M) ответов

Вариант 1:

- На вопрос среди своих ответов **может быть** один “лучший ответ”

Вариант 2:

- Если на вопрос есть хоть один ответ, то среди своих ответов **должен быть** один “лучший ответ”

Один к одному

Распространены 3 способа моделировать отношение 1:1

- у сущностей один и тот же первичный ключ (shared PK)
  - типично при вертикальном секционировании

Только shared PK дают отношение 0,1-0,1

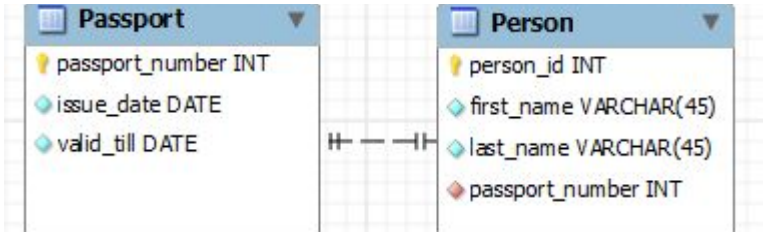
Если в одной из таблиц shared PK одновременно сделать FK (UNIQUE NOT NULL), то будет отношение 1,1-0,1. Как раз при последовательном возникновении частей сущности работает.

Mutual FK возможны, но на практике другие механизмы

Один к одному

- у одной из сущностей FK UNIQUE NOT NULL на другую
  - FK UNIQUE NOT NULL даёт 1,1-0,1
  - FK UNIQUE NULL не рекомендуется

У человека ссылка на уникальный паспорт
Бывают паспорта без людей
Стандартный NULL: паспорт есть, но склероз



- развязочная таблица, каждый из FK UNIQUE NOT NULL
  - каждый FK есть потенциальный ключ

Один к одному: 0,1-1,1 (Weak Entity)

У университета есть Learning Management Systems. Пользователь в ней может быть студентом. Студент - это пользователь.

- Shared PK: PK в таблице студентов одновременно является FK UNIQUE NOT NULL на пользователей.
- Не все пользователи являются студентами

Каждый пользователь LMS также может быть преподавателем. Быть студентом и преподавателем одновременно можно.

- Нет проблем, то же самое делаем

Один к одному: подклассы

У всех пациентов есть ФИО, но есть сугубо мужские и сугубо женские атрибуты. Быть мужчиной и женщиной одновременно **нельзя**.

Один к одному: супруги

- Мужчина может быть женат только на одной женщине
- Женщина может быть замужем только за одним мужчиной
- И женщины, и мужчины - люди

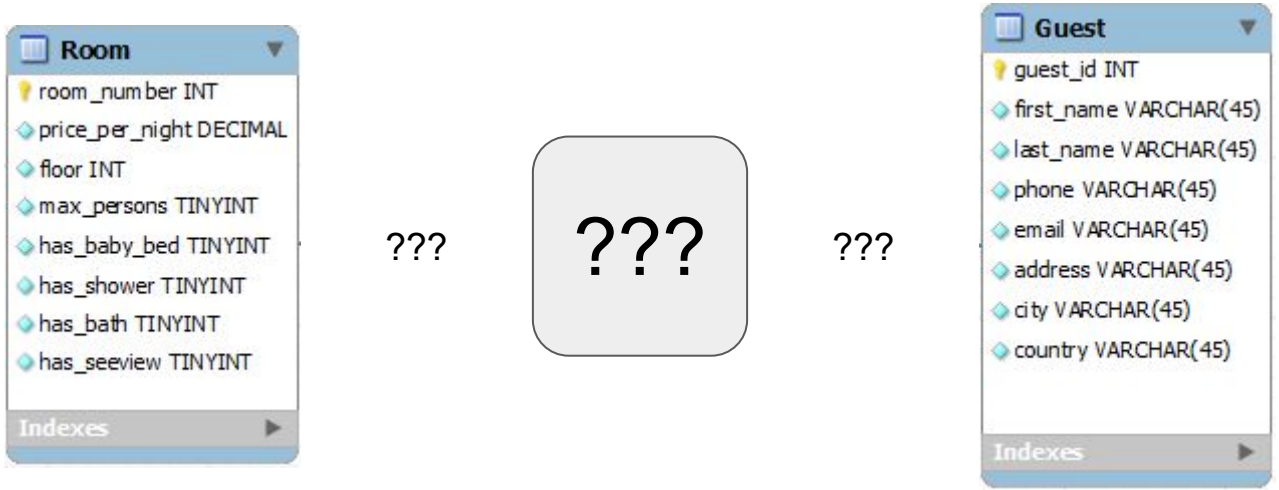
Многие ко многим

Для каждого клиента забронированы комнаты
Каждую комнату бронируют клиенты

Многие ко многим

Через развязочную таблицу (“таблицу-связь”) можно любое бинарное отношение представить

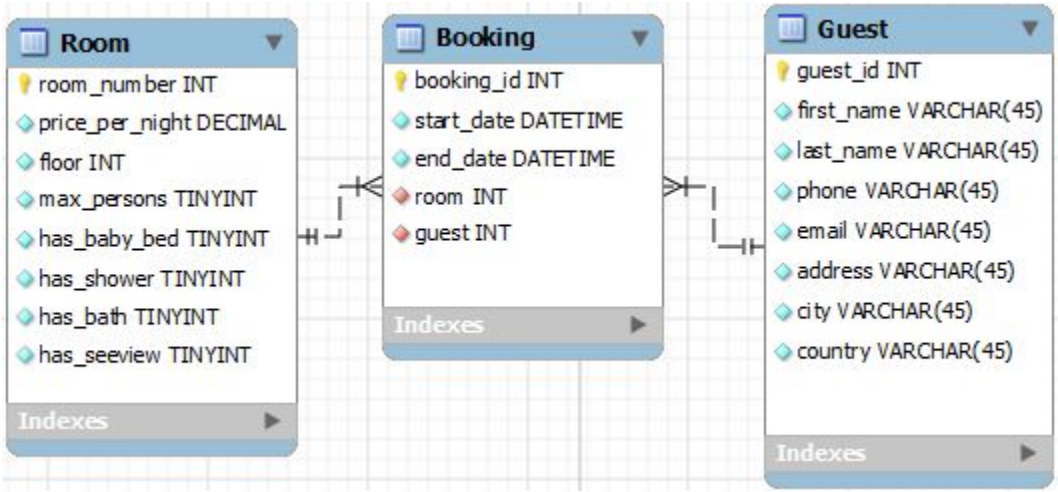
- явно храним подмножество декартова произведения
- очевидно обобщение на отношения больших арностей



Сущности-транзакции

Через развязочную таблицу (“таблицу-связь”)

- и здесь довольно часто осознают ещё одну новую сущность-транзакцию
- **связь возникает как событие** - да ещё и **с атрибутами**



Популярные логические модели

Логическая модель

Существуют разные подходы к проектированию логической модели

- т.е. разные методики построения логической по концептуальной

Рассмотрим 3 наиболее популярные для проектирования хранилищ данных (база данных для аналитики):

1. Нормализованная (как правило, 3-я нормальная форма)
2. Звезда / снежинка
3. Data Vault

Нормализованная модель

Нормальная форма — свойство таблицы, определяющее избыточность и, как следствие, потенциальную противоречивость информации в БД:

- 1-я нормальная форма, 1NF
- 2-я нормальная форма, 2NF
- 3-я нормальная форма, 3NF
- Усиленная 3-я нормальная форма, BCNF
- 4-я нормальная форма, 4NF
- …

Процесс перехода от формы N к форме N+1 называется нормализацией.
Процесс перехода от формы N+1 к форме N называется денормализацией.

1-я нормальная форма

Все атрибуты атомарны

Это свойство таблицы, а не математического отношения.

Клиенты				
Ключ	Фамилия	Имя	Отчество	Детали
1	ИВАНОВ	ИВАН	ИВАНОВИЧ	Родился в 1985 году, получил высшее образование в МГУ

Клиенты						
Ключ	Фамилия	Имя	Отчество	Год рождения	Образование	ВУЗ
1	ИВАНОВ	ИВАН	ИВАНОВИЧ	1985	Высшее	МГУ

Просто по определению ключа каждый неключевой атрибут «должен предоставлять информацию о ключе».

2-я нормальная форма

Находится в 1NF

Каждый неключевой атрибут «должен предоставлять информацию о **полном** ключе»

Музыкальные диски				
Название группы	**Название СД-диска**	**Название песни**	**Автор слов**	**Композитор**
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

Музыкальные диски		
Название группы	**Название СД-диска**	**Название песни**
Scorpions	World Wide Live	Countdown
Scorpions	World Wide Live	Coming Home
Scorpions	World Wide Live	Blackout
Scorpions	Blackout	Blackout
The Big City	Blackout	Blackout

Песни			
Название группы	**Название песни**	**Автор слов**	**Композитор**
Scorpions	Countdown	Klaus Meine	Matthias Jabs
Scorpions	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Rudolf Schenker	Nige Roberts

Автор слов не зависит от названия диска

3-я нормальная форма

Находится в 2NF

Каждый неключевой атрибут «должен предоставлять информацию ... и **ни о чём, кроме** ключа».

Счета					
Ключ	Ключ клиента	Фамилия	Имя	Отчество	Баланс
1	1	ИВАНОВ	ИВАН	ИВАНОВИЧ	250
2	1	ИВАНОВ	ИВАН	ИВАНОВИЧ	1500

Счета		
Ключ	Ключ клиента	Баланс
1	1	250
2	1	1500

Клиент			
Ключ	Фамилия	Имя	Отчество
1	ИВАНОВ	ИВАН	ИВАНОВИЧ

Фамилия зависит от счёта, но **транзитивно** через клиента.

Нормальная форма Бойса—Кодда

Таблица находится в нормальной форме Бойса—Кодда тогда и только тогда, когда детерминанты всех её функциональных зависимостей являются потенциальными ключами.

BCNF во всех отношениях сильнее ранее определённой 3NF

Отличие от 3NF только для нетранзитивных функциональных зависимостей. На практике такие отношения встречаются достаточно редко, для всех прочих отношений 3NF и BCNF эквивалентны.

Типичные примеры 3NF→BCNF связаны с тем, что значение одного из полей является составным, а это заставляет задуматься об атомарности.

Нормализованная модель

Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.

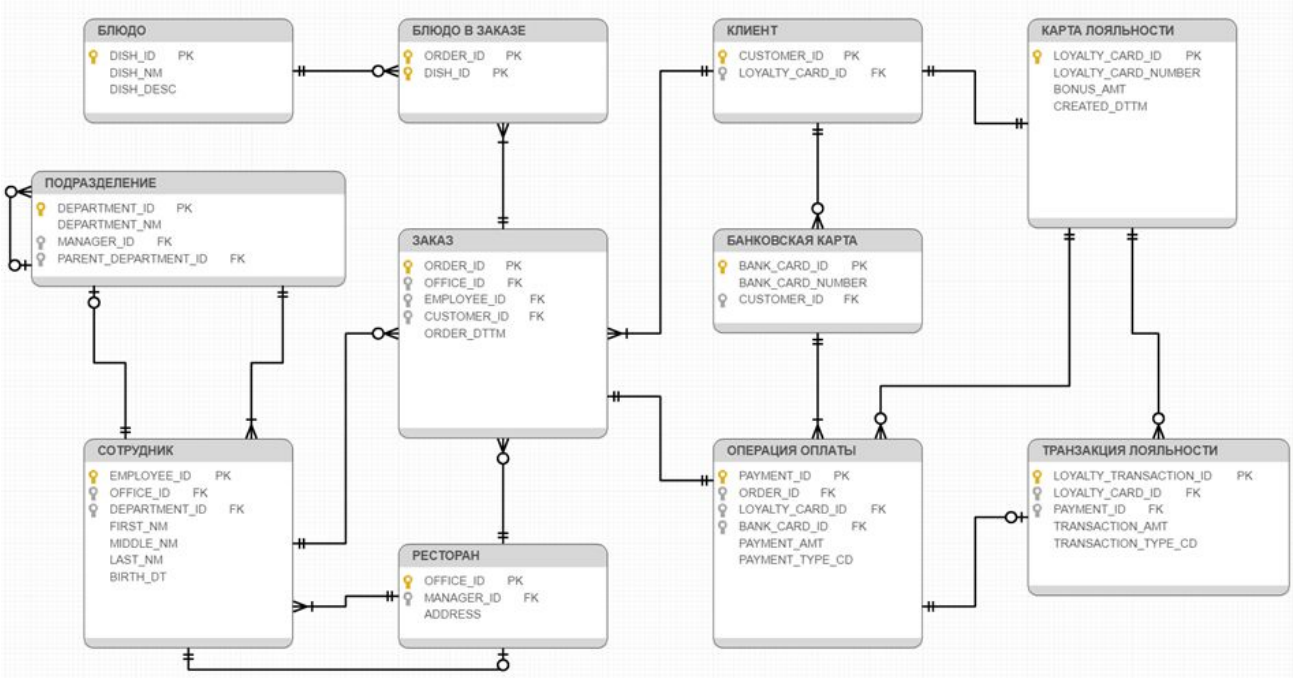
Все таблицы находятся, как правило, в 3NF или BCNF.
Сущности сильно связаны между собой.

Свойства:

- высокая устойчивость (особенно к ошибкам обновления)
- высокая масштабируемость
- необходимость длительного проектирования
- анализировать данные сложнее (нужно много соединений, прибегают к денормализации)

Нормализованная модель

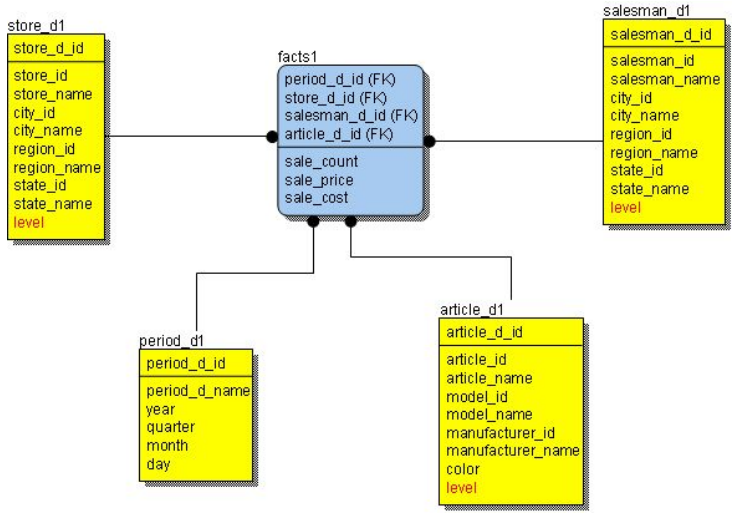
Пример “сеть ресторанов”



Модель звезда

Два типа таблиц:

- Таблица фактов (событий), FACTS
- Таблицы измерений, DIMENSIONS



Факты - объекты анализа.
Как правило, таблица фактов содержит числовые показатели и ссылки на измерения.

Измерения - атрибуты событий. Любые значения, характеризующие не один, а группу объектов.

Это результат нормализации одной таблицы данных.

Модель звезда

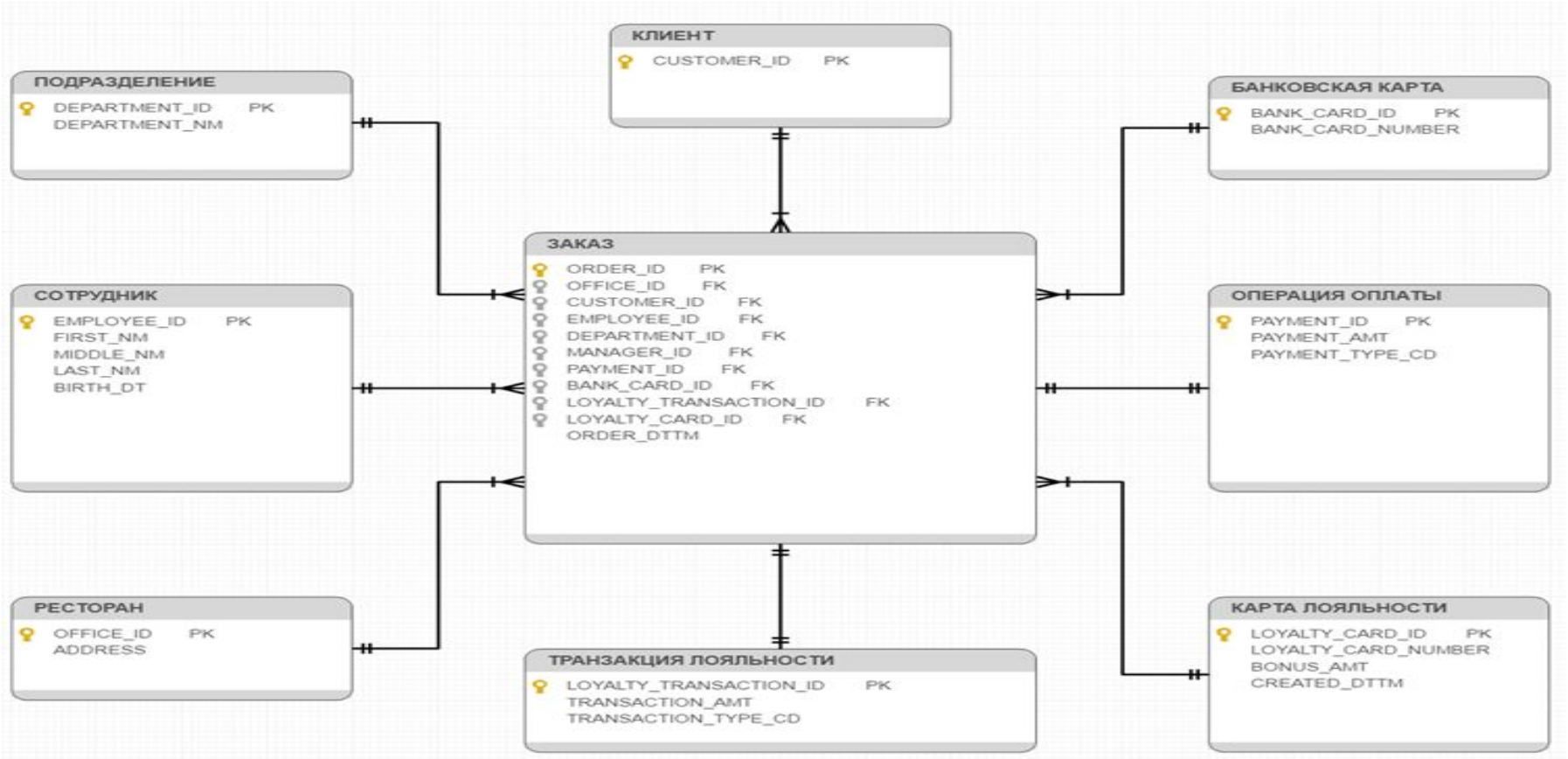
Не должно быть циклов

В частности, в таблице фактов не должно быть двух внешних ключей на одно измерение. Например, если для человека указывается страна рождения и страна смерти, то надо обеспечить “разнотипность” этих стран.

Свойства:

- Факты, как правило, слабо связаны между собой
- Высокая скорость получения «первых результатов»
- Удобно анализировать
- Много ПО, ориентированного на эту модель

Модель “звезда”: пример “сеть ресторанов”

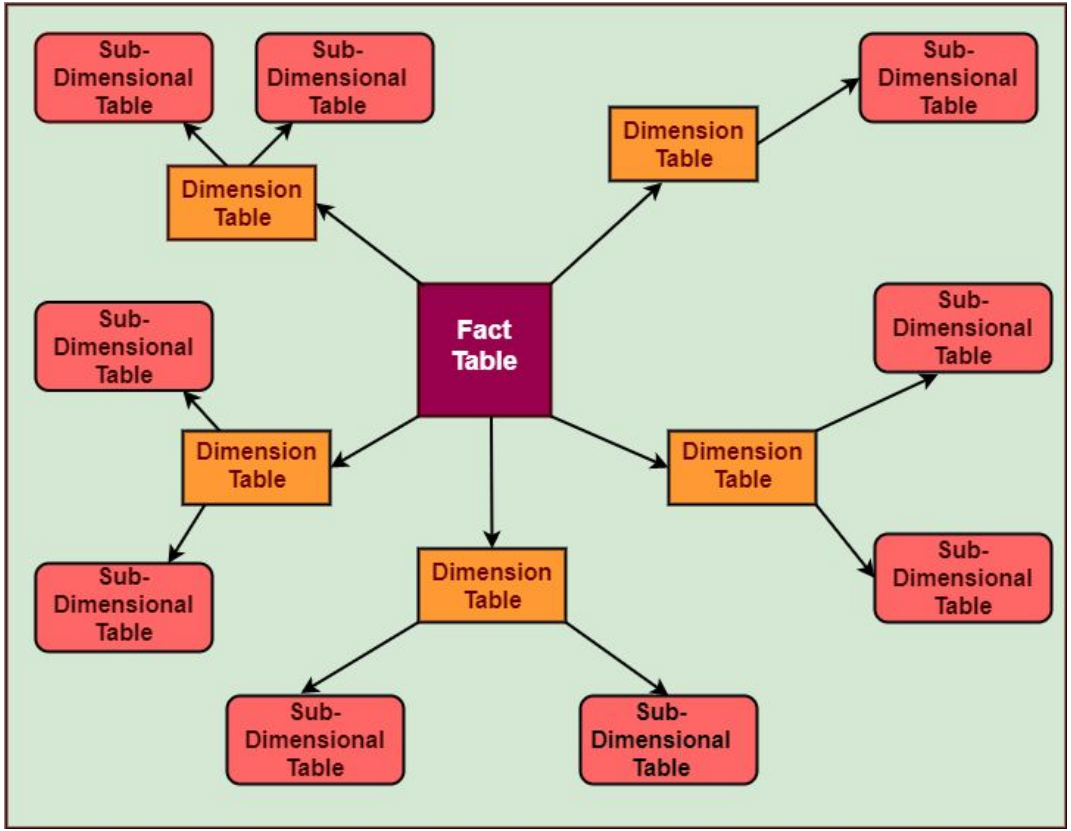


Модель “снежинка”

В модели “звезда” все измерения в одном шаге от фактов.

Иногда можно дополнительно нормализовать словарь, получатся новые словари - уже на расстоянии 2 от фактов.

И т. д.



С точки зрения измерений, факты - это развязочная таблица для отношения большой арности

Это отношение можно спроецировать на любое подмножество измерений, т. е. ситуация гораздо богаче, чем просто бинарные отношения между измерениями

Slowly Changing Dimension

Измерения можно хранить с историей или без истории. Историю можно хранить по-разному.

SCD type 0: Ничего не изменяется. Immutable entities. Истории нет. Например, “Календарь”

SCD type 1: Историю не храним. Помним только последние данные.

SCD type 2: Храним историю версий данных. **Суррогатный ключ меняется**

SCD type 1

- Для одной категории храним ровно одну “последнюю” версию
- При изменении категории обновляем существующую строчку
- Можно ограничиться бизнес-ключами сущностей
  - Суррогатные ключи лишь ускоряют соединения
- При связи невозможно выбрать правильную версию: она забыта
- Легко для сущности добавлять атрибуты

SCD type 2

- Для одной категории храним несколько версий
  - запись в такой таблице - это одна из версий категории
- Суррогатный ключ при каждом обновлении категории меняется
  - Получается суррогатный ключ версии
  - При связи не надо выбирать правильную версию, она точно известна
  - Проблема, если захотим добавить для сущности атрибут: ключи не разделить
- Каждая версия знает свой порядковый номер
  - Пара (ключ сущности, номер версии) - потенциальный ключ версии
- Или есть период актуальности этой версии данных о категории effective date (valid_from, valid_to)
  - standardized **surrogate high date** (e.g. 9999-12-31) may be used as an end date, so that the field can be included in an index, and so that NULL-value substitution is not required when querying
  - при добавлении версии надо менять период актуальности предыдущей версии

SCD Pure Type 6

- Для одной категории храним несколько версий
- Используем ключи для сущностей, они не меняются при обновлении
  - Можно ограничиться бизнес-ключами, но суррогатные эффективнее
  - При связи надо выбирать правильную версию, усложняются запросы
  - Таблица фактов не может явный внешний ключ определить
  - Легко добавлять атрибуты, разделять версии на несколько
- Строка имеет период актуальности этой версии данных о категории effective date (valid_from, valid_to)
  - Обеспечивает предикат текущей актуальности версии
  - Обеспечивает выбор правильной версии

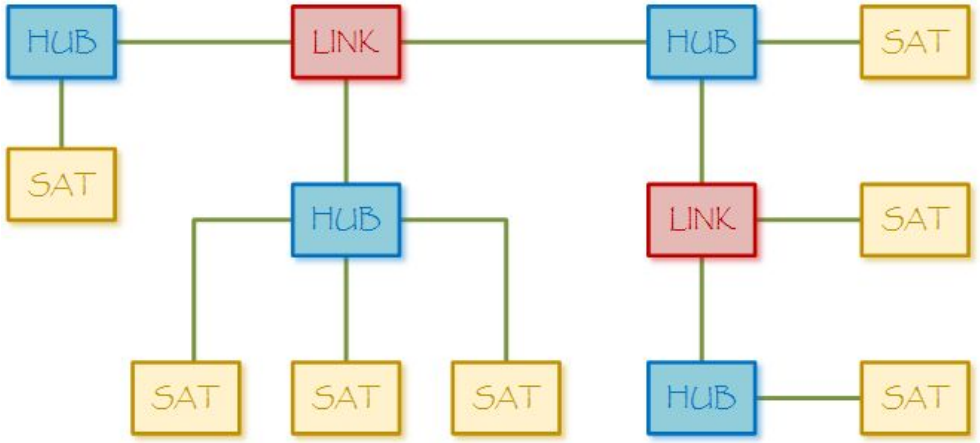
Например, таблица “Клиенты”

Модель Data Vault

Гибрид хороших свойств 3NF и Star Schema (полезная ссылка)

Три основных составляющих:

1. HUB (концентратор)
2. LINK (связь)
3. SATELLITE (спутник)



Свойства:

- гибкость - легко реагировать на изменения (эволюцию) самого бизнеса и информационной инфраструктуры
- простота построения отдельных таблиц
- анализировать данные сложнее

HUB

Определяет бизнес-сущность. Точка соединения данных об одной бизнес-сущности из разных источников. Содержит только:

- суррогатный ключ
- бизнес-ключ
- метка времени загрузки
- метка источника загрузки

обеспечивают возможность сверки с источником

для hub это информация о первом упоминании данного бизнес-ключа

H_CUSTOMER			
customer_rk	customer_code	processed_dttm	src
12314	1-FXFSFAD	2015-04-10 00:00:00	Siebel
43123	1-XFSSSDF	2015-04-10 00:00:00	Siebel

Бизнес-ключ позволяет узнавать один и тот же объект в разных источниках

- VIN для авто

LINK

Связи между сущностями. Любой арности. Сущности-транзакции (‘booking’).
Содержит:

- суррогатные ключи, определяющие связь
- метку времени/источника загрузки

L_CUSTOMER_ACCOUNT				
L_customer_account_rk	customer_rk	account_rk	processed_dttm	src
1	12314	2911	2015-04-10 00:00:00	Siebel
2	43123	7591	2015-04-10 00:00:00	Siebel
3	43123	1356	2015-04-10 00:00:01	Siebel

Все связи реализованы только как развязочные таблицы. Никаких внешних ключей между HUB’ами

- легко модифицировать и добавлять связи
- произвольная арность отношения

SATELLITE

Дополнительная описательная информация для сущности.

Содержит:

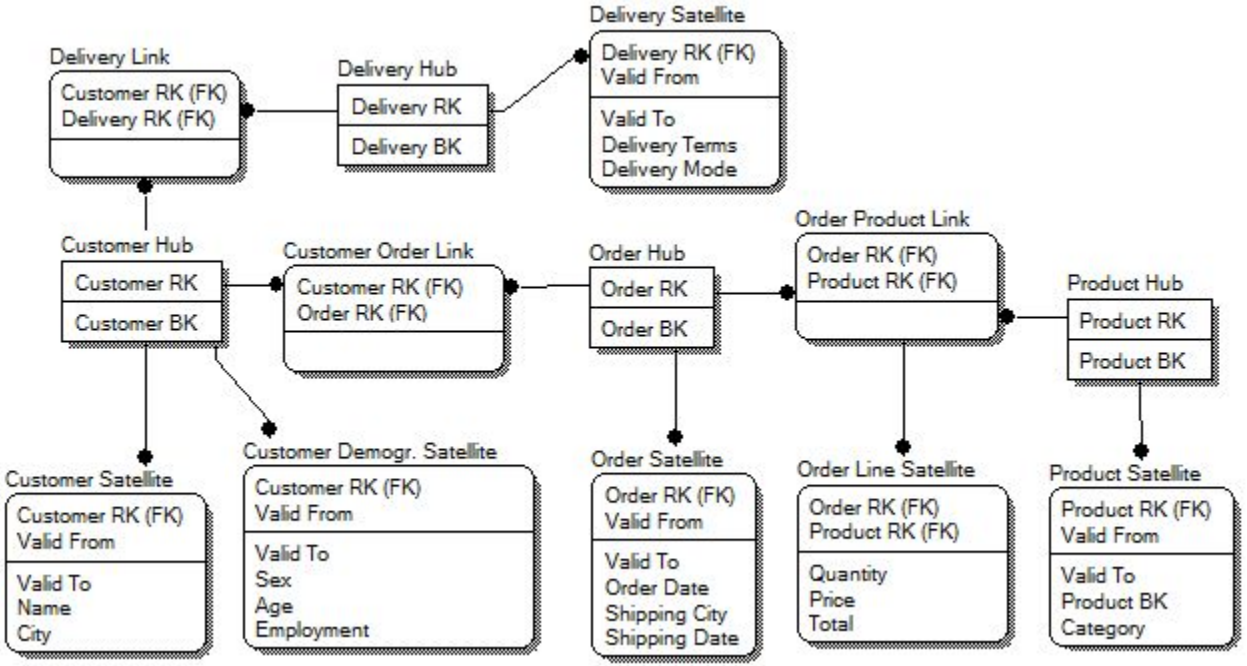
- суррогатный ключ сущности
- поля, описывающие сущность

S_CUSTOMER_PERSONAL				
customer_rk	first_nm	last_nm	middle_nm	birth_dt
12314	АЛЕКСАНДР	РАИСОВИЧ	ГРАЧЕВ	19.09.1929
43123	НИНА	ЕФИМОВНА	КИСЕЛЕВА	03.05.1963
23121	АРТУР	МАРАТОВИЧ	ЛОБАНОВ	23.02.1954

S_CUSTOMER_ADDRESS		
customer_rk	city	street
12314	Москва	Ленинградское шоссе
43123	Владивосток	Семеновская

Модель Data Vault

Пример - служба доставки



Развитие Data Vault

Базовый DATA VAULT (Raw DATA VAULT) имеет дальнейшее развитие, обусловленное высокой сложностью аналитических запросов к нему.

Следующим этапом эволюции является Business DATA VAULT. Здесь уже имеют место дополнительные сущности, например PIT и BRIDGE таблицы

Поскольку факты-события-транзакции ссылаются одновременно на много измерений, то возникает специфика работы с ними в рамках Data Vault.