



ТЕХНОТРЕК

Занятие №5

Проектирование Баз Данных

Дина Сафина

Проектирование Баз Данных



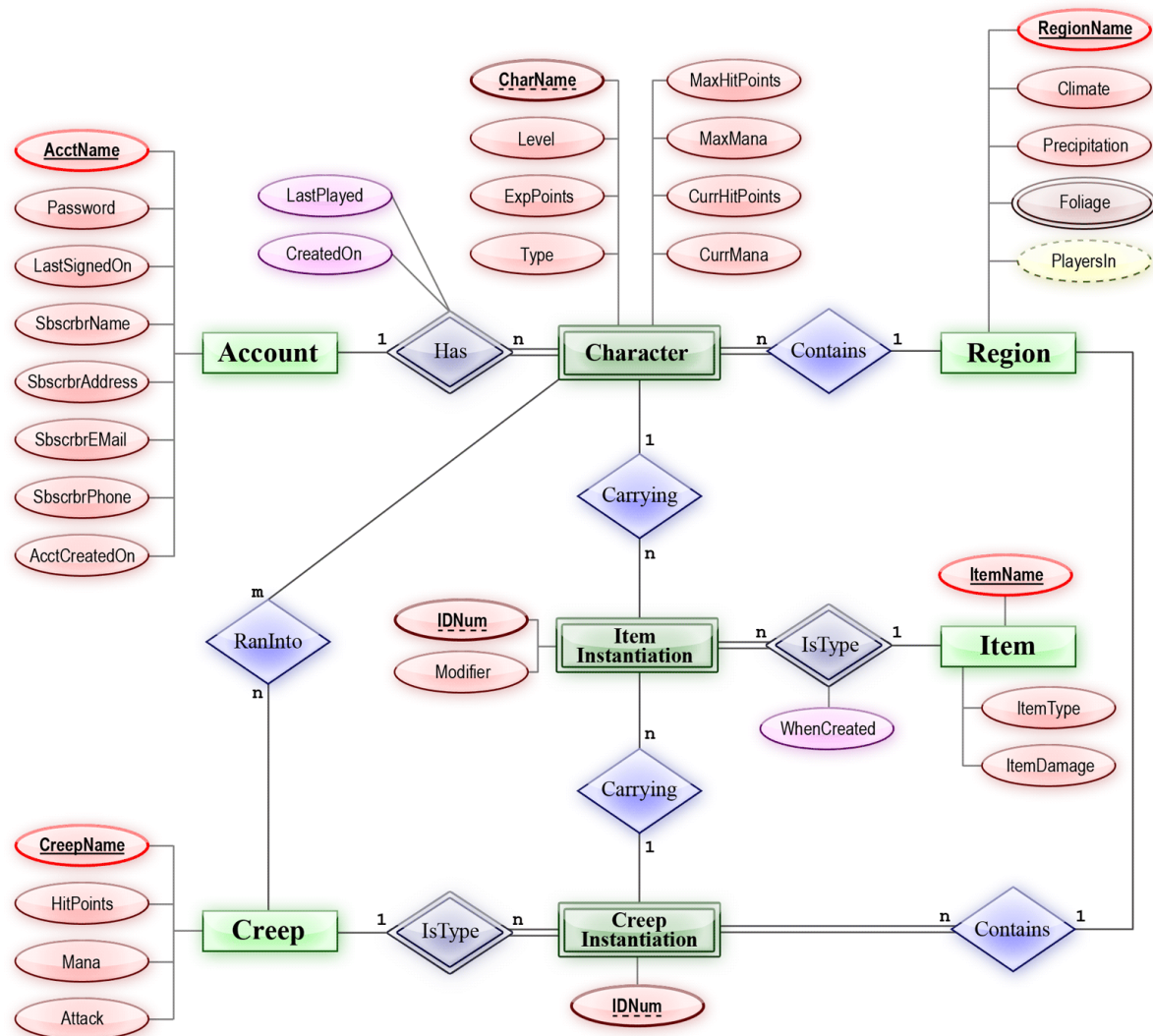
1. Этапы проектирования БД
2. Нормальные формы
3. Хранилища данных
4. Data Vault
5. Anchor modeling
6. Работа с базой данных

Этапы проектирования БД

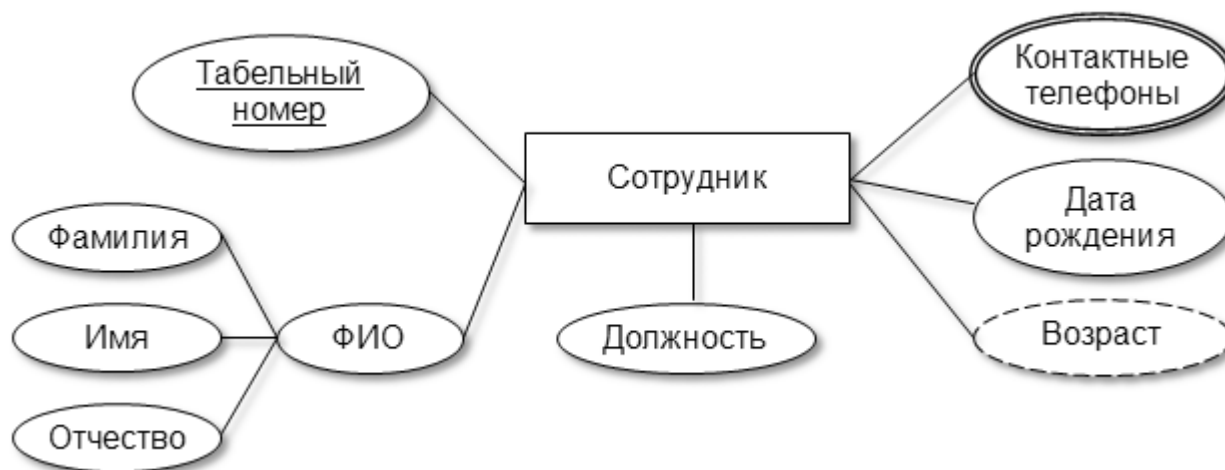


1. Сбор требований
2. Концептуальное проектирование (нотация П.Чена):
 - объекты предметной области и связи между ними
 - определение атрибутов и их допустимых значений
 - ограничения целостности
3. Логическое проектирование
 - набор отношений
 - первичные и внешние ключи
4. Физическое проектирование:
 - схема базы данных для конкретной СУБД
 - DDL-скрипты

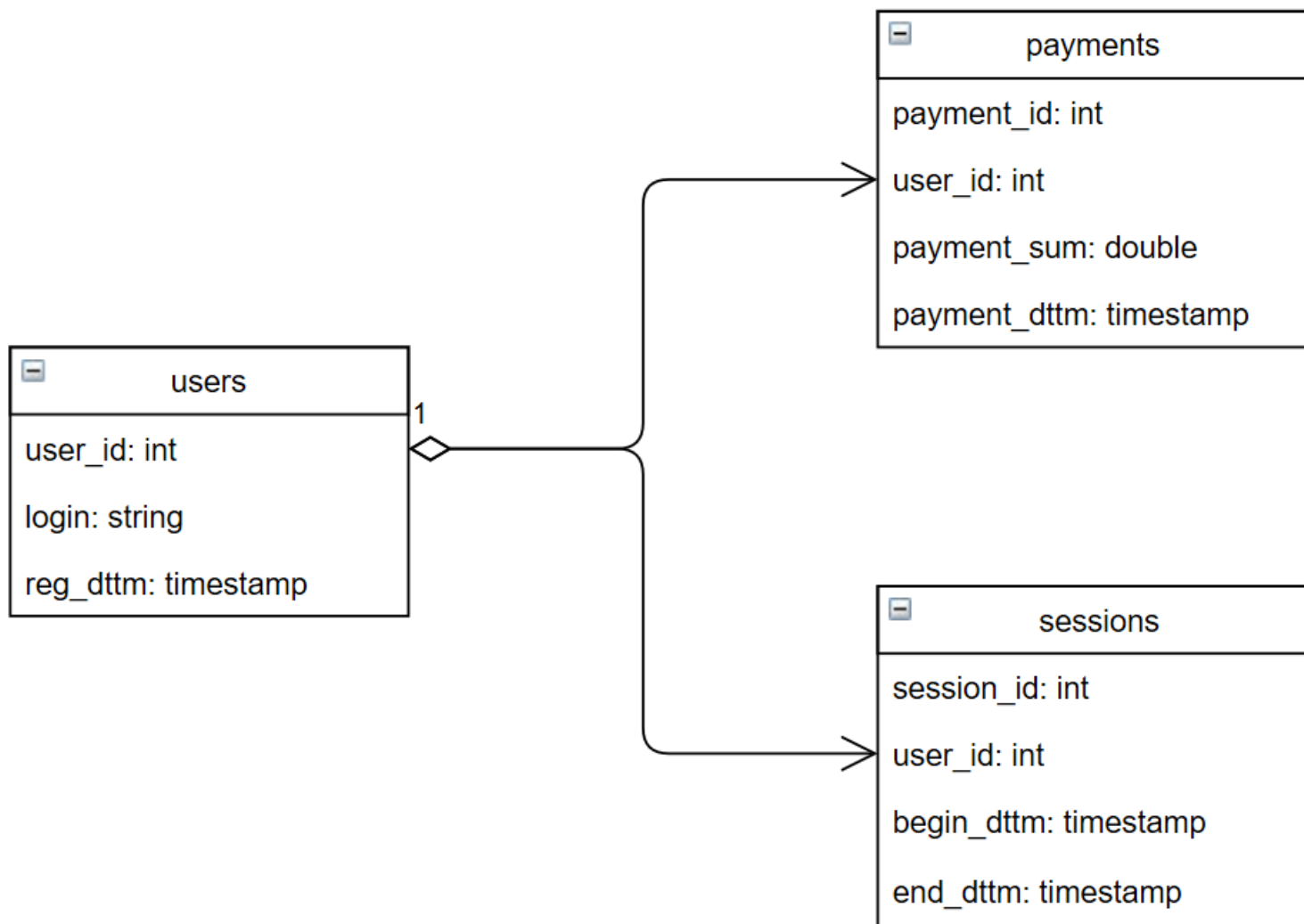
Нотация Питера Чена



Нотация Питера Чена



Логическая модель базы данных



Почему проект БД может быть плохим?



- Избыточность
- Потенциальная противоречивость (аномалии обновления)
- Аномалии включения
- Аномалии удаления



ТЕХНОТРЕК

Нормальные формы

Нормализация Базы Данных



Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами.

Нормализация – процесс преобразования отношений базы данных к виду, отвечающему нормальным формам.

Первая нормальная форма (1NF)



Переменная отношения находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

Первая нормальная форма (1NF)



До нормализации:

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

После нормализации:

Фирма	Модель
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

Вторая нормальная форма (2NF)



Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа.

Вторая нормальная форма (2NF)



До нормализации:

Фирма	Модель	Цена	Скидка
BMW	M5	5500000	5%
BMW	X5M	6000000	5%
BMW	M1	2500000	5%
Nissan	GT-R	5000000	10%

После нормализации:

Фирма	Модель	Цена
BMW	M5	5500000
BMW	X5M	6000000
BMW	M1	2500000
Nissan	GT-R	5000000

Фирма	Скидка
BMW	5%
Nissan	10%

Третья нормальная форма (3NF)



Переменная отношения находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

Третья нормальная форма (3NF)



До нормализации:

Модель	Магазин	Телефон
BMW	Риал-Авто	87-33-98
Audi	Риал-Авто	87-33-98
Nissan	Некст-Авто	94-54-12

После нормализации:

Модель	Магазин
BMW	Риал-Авто
Audi	Риал-Авто
Nissan	Некст-Авто
Модель	Магазин

Фирма	Скидка
Риал-Авто	87-33-98
Некст-Авто	94-54-12

Нормальная форма Бойса – Кодда (BCNF)



Переменная отношения находится в нормальной форме Бойса — Кодда (иначе — в усиленной третьей нормальной форме) тогда и только тогда, когда каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

BCNF, до нормализации



Номер стоянки	Время начала	Время окончания	Тариф
1	09:30	10:30	Бережливый
1	11:00	12:00	Бережливый
1	14:00	15:30	Стандарт
2	10:00	12:00	Премиум-В
2	12:00	14:00	Премиум-В
2	15:00	18:00	Премиум-А

- находится в третьей нормальной форме
- тариф зависит от номера стоянки

BCNF, после нормализации



Тариф	Номер стоянки	Имеет льготы
Бережливый	1	Да
Стандарт	1	Нет
Премиум-А	2	Да
Премиум-В	2	Нет

Тариф	Время начала	Время окончания
Бережливый	09:30	10:30
Бережливый	11:00	12:00
Стандарт	14:00	15:30
Премиум-В	10:00	12:00
Премиум-В	12:00	14:00
Премиум-А	15:00	18:00

Четвёртая нормальная форма (4NF)



Переменная отношения находится в четвёртой нормальной форме, если она находится в нормальной форме Бойса — Кодда и не содержит нетривиальных многозначных зависимостей.

Четвёртая нормальная форма (4NF)



До нормализации:

{Ресторан, Вид пиццы, Район доставки}

После нормализации:

{Ресторан} → {Вид пиццы}

{Ресторан} → {Район доставки}

Не нуждается в нормализации:

{Ресторан, Вид пиццы, Район доставки} → Цена

Пятая нормальная форма (5NF)



Переменная отношения находится в пятой нормальной форме (иначе — в проекционно-соединительной нормальной форме) тогда и только тогда, когда каждая нетривиальная *зависимость соединения* в ней определяется потенциальным ключом (ключами) этого отношения.

Доменно-ключевая нормальная форма (DKNF)



Переменная отношения находится в ДКНФ тогда и только тогда, когда каждое наложенное на неё ограничение является логическим следствием ограничений доменов и ограничений ключей, наложенных на данную переменную отношения.

Шестая нормальная форма (6NF)



Переменная отношения находится в шестой нормальной форме тогда и только тогда, когда она удовлетворяет всем нетривиальным зависимостям соединения. Из определения следует, что переменная находится в 6НФ тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6НФ, также находится и в 5НФ.

Нормализованная схема



- + Быстрее обновляются
- + Меньше данных приходится изменять
- + Таблицы меньше по размеру
- + Реже приходится агрегировать и использовать DISTINCT

- Увеличивается количество соединений
- Не позволяет использовать некоторые стратегии индексирования
- Проблемы с расширением

Примеры нормальной денормализации



- Кэширование столбца из одной таблицы в другую
- Кэширование производных значений
- Кэшированная таблица
- Сводная таблица
- Таблицы счётчиков
- Материализованные представления (не поддерживаются в MySQL)



ТЕХНОТРЕК

Хранилище Данных

OLTP – OnLine Transaction Processing



Ввод, хранение и обработка информации в реальном времени. Операционное хранилище.

Требования:

- Сильно нормализованные модели данных
- Возможность откатить транзакцию
- Обработка данных в реальном времени

+ Высокая надёжность и достоверность данных

– Оптимизированы для небольших транзакций.

Витрина



- Используется аналитиками
- Решает одну конкретную задачу
- Коллекция денормализованных таблиц или Звезда
- Можно делать виртуальными

OLAP – OnLine Analytical Processing



Подготовка агрегированной информации на основе многомерных данных. Аналитическое хранилище.

Типы OLAP:

- Multidimensional OLAP
- Relational OLAP
- Hybrid OLAP
- Real-time ROLAP

+ Скорость

– Сложность

Типы таблиц в OLAP



Таблица фактов:

- информация о событиях
- составной ключ из внешних ключей
- несколько полей с числовыми значениями

Типы:

- Transaction facts
- Snapshot facts
- Line-item facts
- Event or state facts

Таблица измерений – содержат атрибуты событий

Схема звезды



- Таблица фактов
- Несколько таблиц измерений
- Таблицы измерений денормализованы

Схема работы:

- Соединений таблицы фактов с таблицами измерений
- Фильтрация
- Группировка и агрегация

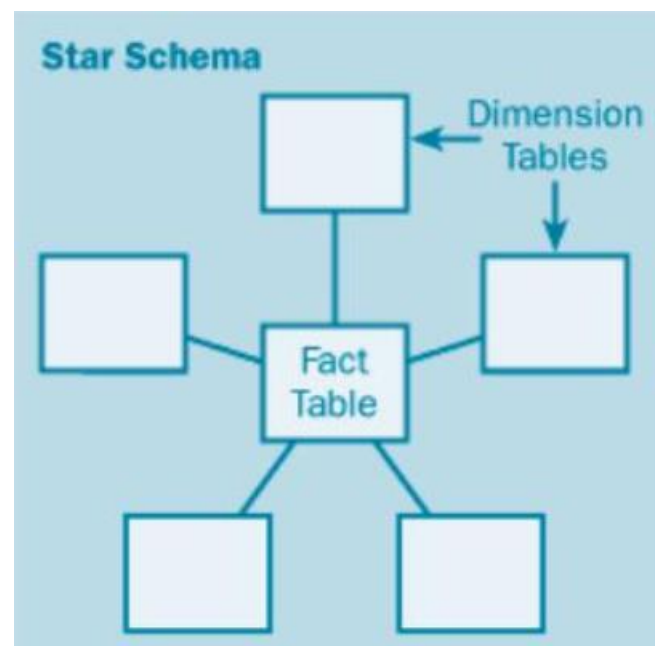
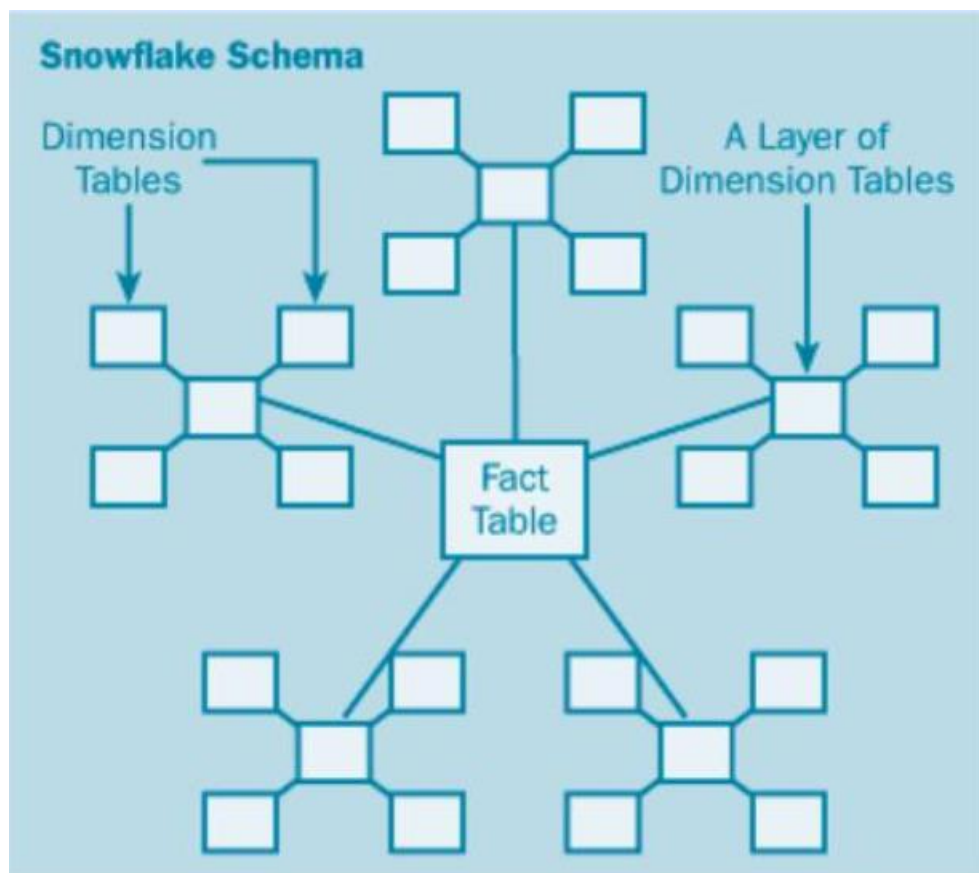


Схема снежинки















TEXHOTPEK

Data Vault



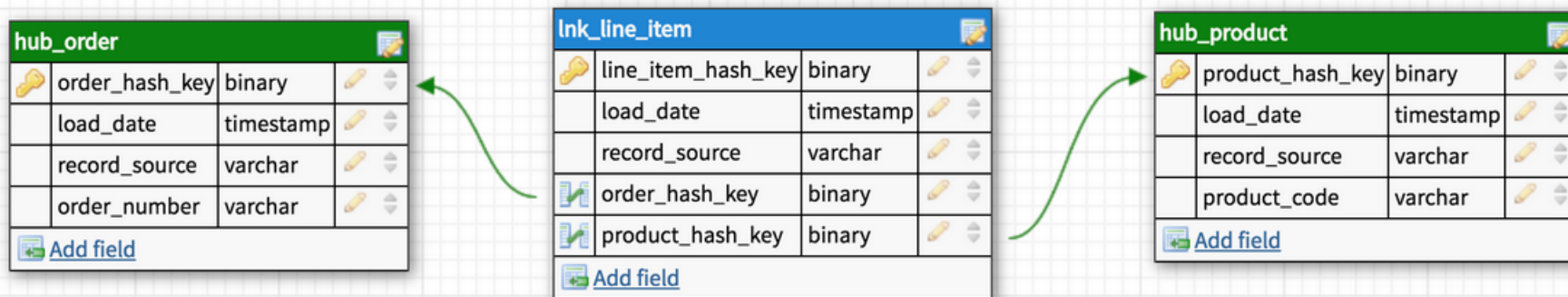
- Основная сущность
- Бизнес-ключ – одно или несколько основных полей
- Ключ уникальный и неизменяемый
- Первичный ключ – MD5 или SHA-1 от бизнес-ключа
- Мета-поля *load timestamp* и *record source*

hub_order			
	order_hash_key	binary	
	load_date	timestamp	
	record_source	varchar	
	order_number	varchar	

hub_product			
	product_hash_key	binary	
	load_date	timestamp	
	record_source	varchar	
	product_code	varchar	



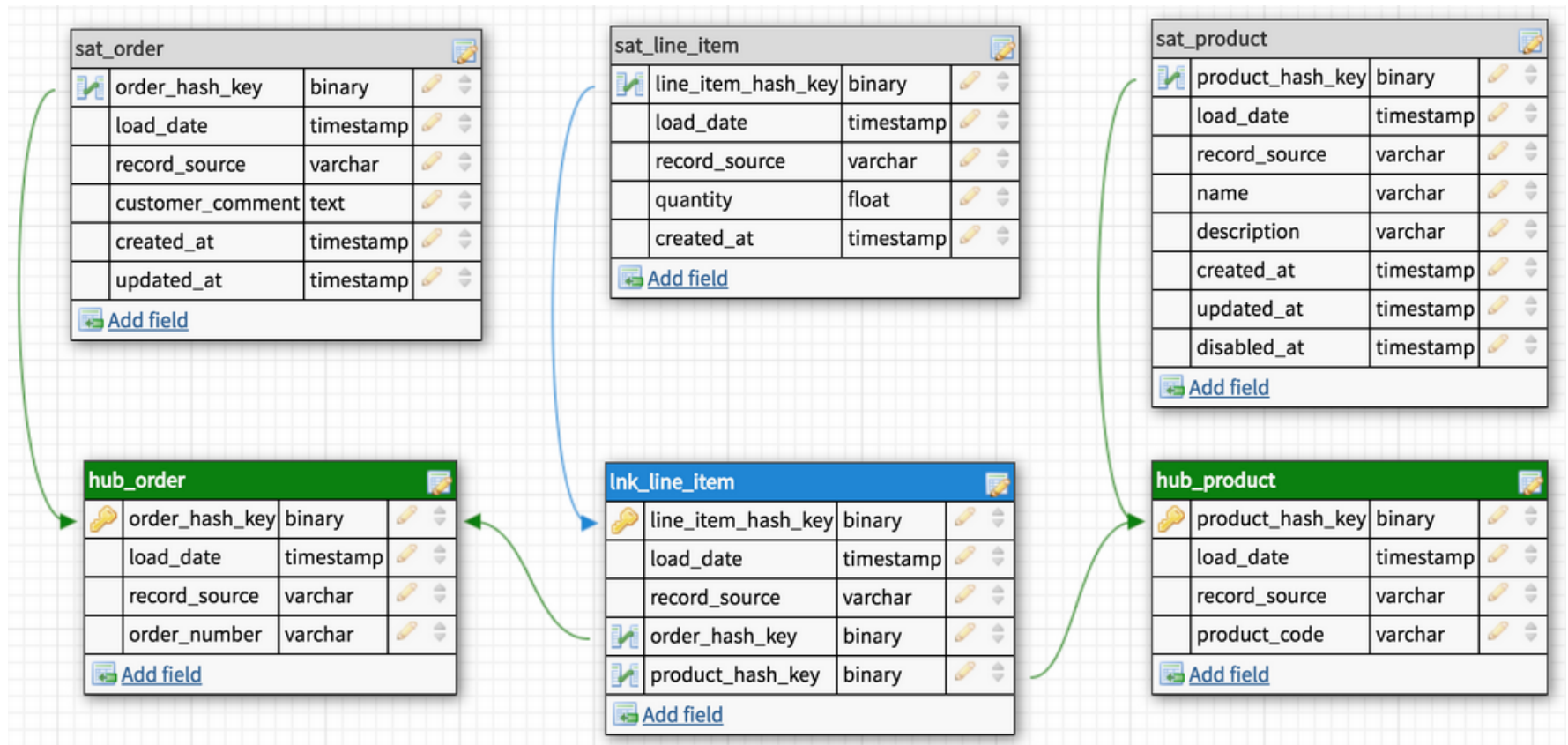
- Связывает Хабы связью многие-ко-многим
- Мета-поля *load timestamp* и *record source*



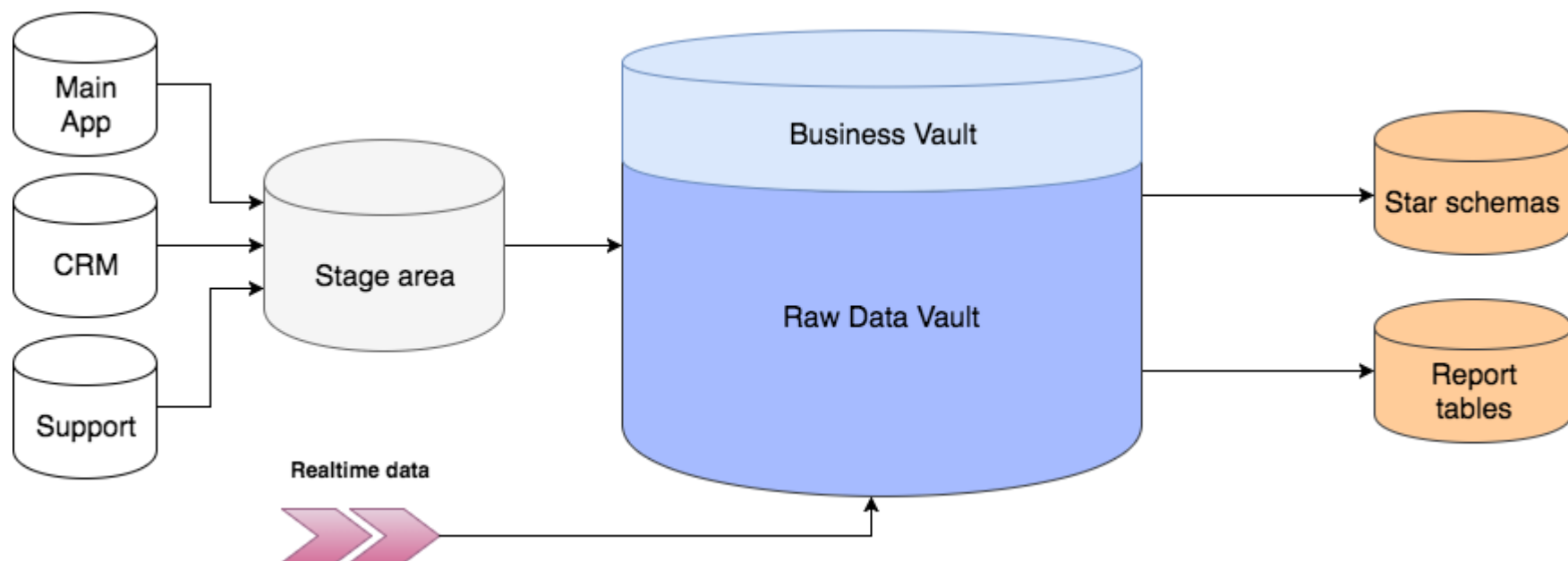


- Хранит контекст
- Все описательные атрибуты Хаба или Ссылки
- Единственный ключ родителя
- Мета-поля *load timestamp* и *record source*
- У Хаба и Ссылки может быть несколько Сателлитов
- Пример – хранение истории изменений

Satellite



Как с этим работать?



Data Vault?



- + Неразрушающее расширение модели
- + Agile-friendly
- + Лёгкий ETL
- + Хранение истории
- + Нет избыточности информации
- + Однозначное соответствие реляционным сущностям

- Слишком много JOIN'ов
- Маленькое комьюнити
- Много таблиц
- Ненаглядно



TEXHOTPEK

Anchor modeling



ANCHOR

H_Customer:
- customer_id
- load_date
- source_sys

ATTRIBUTE

-S_Customer_INN
-customer_id
-inn
-actual_date

H_Customer
-customer_id
-load_date
-source_sys

S_Customer_Name:
-customer_id
-name
-actual_date

S_Customer_Gender:
-customer_id
-gender
-actual_date

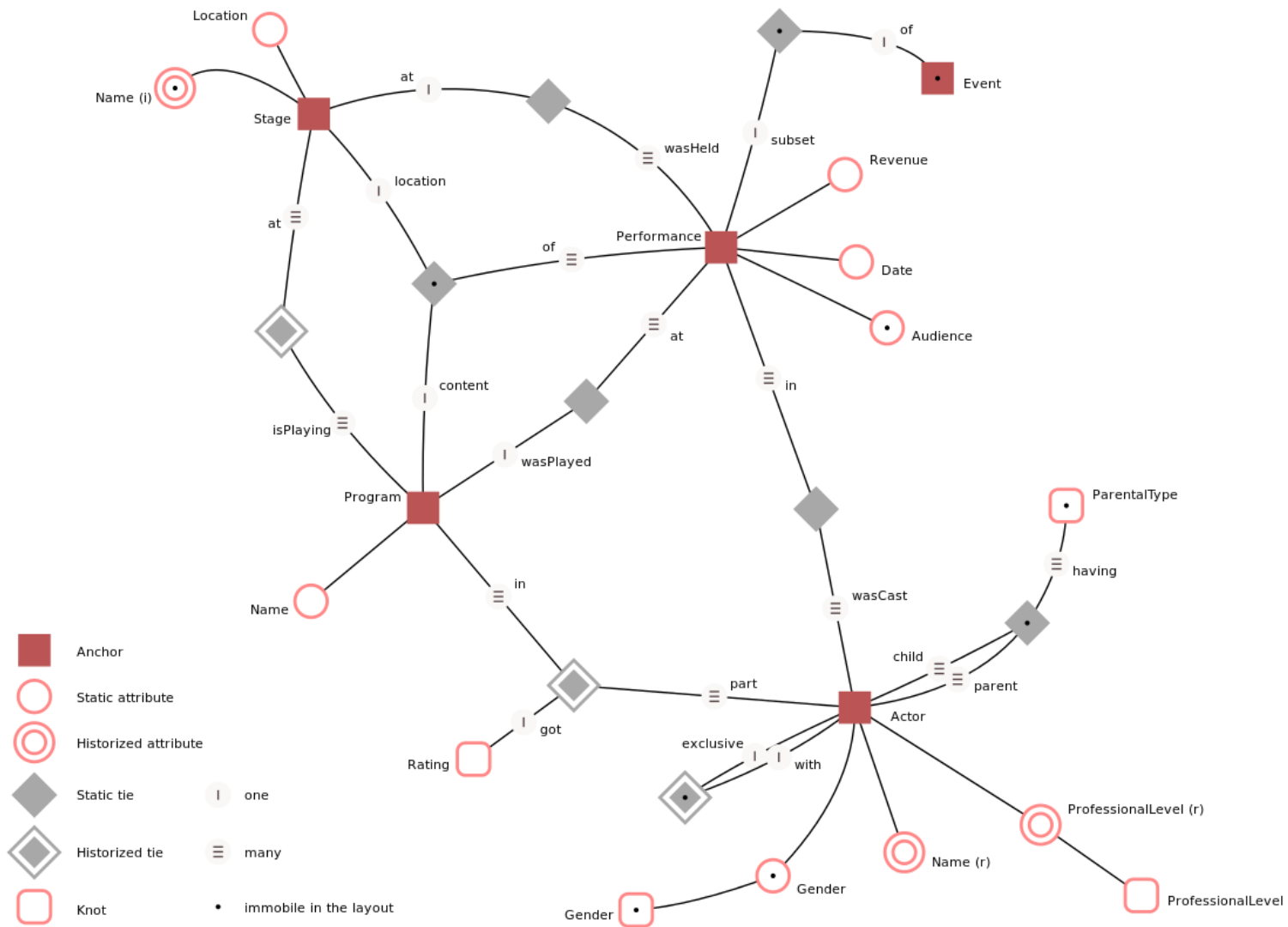


TIE

H_Customer:
-customer_id
-load_date
-source_sys

L_Locates:
-(fk) coustomer_id
-(fk) country_id
-load_date
-source_sys

H_Country:
-country_id
-load_date
-source_sys



Anchor Modeling



- + Преимущества 6NF без её недостатков
 - + Agile-friendly
 - + Неразрушающее расширение модели
 - + Хранение истории
 - + Избегание NULL-ов
 - + Нет избыточности информации
 - + Однозначное соответствие реляционным сущностям
 - + Можно пересечь на другие атрибуты
- Скорость соединений
 - Сложность восприятия
 - Очень, очень много таблиц

Data Vault VS Anchor



Data Vault

- 3NF
- Историчность: дата начала – дата конца
- Обновление + вставка
- Все атрибуты в одной таблице
- Более наглядное хранение атрибутов
- Связь как между Хабами, так и между Сателлитами

Anchor

- 6NF
- Историчность: только дата начала
- Только вставка
- Каждый атрибут в своей таблице
- Сложный сбор атрибутов
- Связь ТОЛЬКО между Якорями

Способы подключения к БД



1. UA (MySQL Workbench, Data Grip, DBeaver)
2. Cli
3. Скрипты (ODBC, JDBC)
4. ORM (Hibernate, Django ORM, Core.Data)

Литература для чтения



- Крис Дж. Дейт, «Введение в системы баз данных»
- William Inmon, «DW2.0. The Architecture for the Next Generation of Data Warehousing»
- Ralph Kimball, «*The Data Warehouse Toolkit: ...*»
- Dan Linstedt, Michael Olschimke «Building a Scalable Warehouse with Data Vault 2.0»

Домашнее задание №5



Разработать проект БД выбранной предметной области:

- Собрать список требований к проекту
- Привести сущности в третью нормальную форму
- Нарисовать диаграмму классов
- Создать скрипты создания таблиц в MySQL/PostgreSQL

Выложить на GitHub

Срок сдачи

24 октября 2018