

Эволюция схемы данных. Носим данные из реляционной СУБД в Hadoop

Чаптер инженеров данных и
разработчиков

Михаил Благов





Михаил Благов

- Руководитель чаптера дата-инженеров в билайне
- Старший преподаватель Кафедры Прикладной кибернетики СПбГУ

mikhail.blagov@gmail.com
@vogal

10 лет в IT

7,5 лет обрабатываю данные

4 года управляю командами и проектами
в дата-инжиниринге

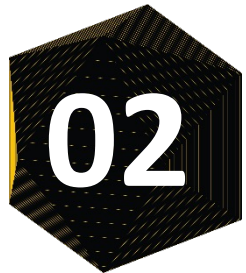
Программа доклада



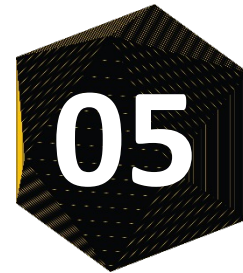
Проблема эволюции схемы данных



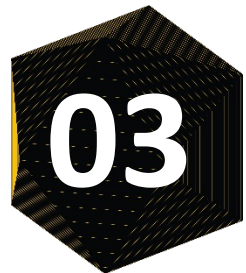
Мониторинг эволюции схемы



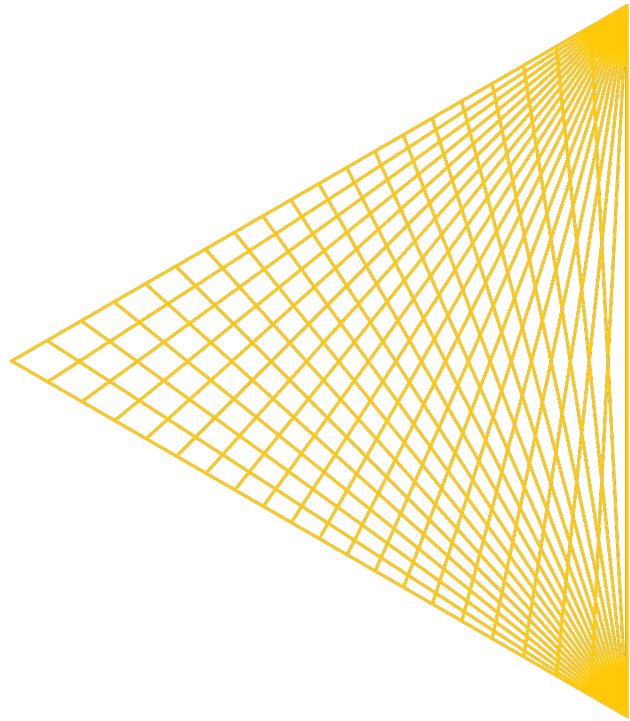
Управление эволюцией схемы для популярных форматов хранения в Hadoop



Интерфейсные соглашения, SLA и другие страшные слова

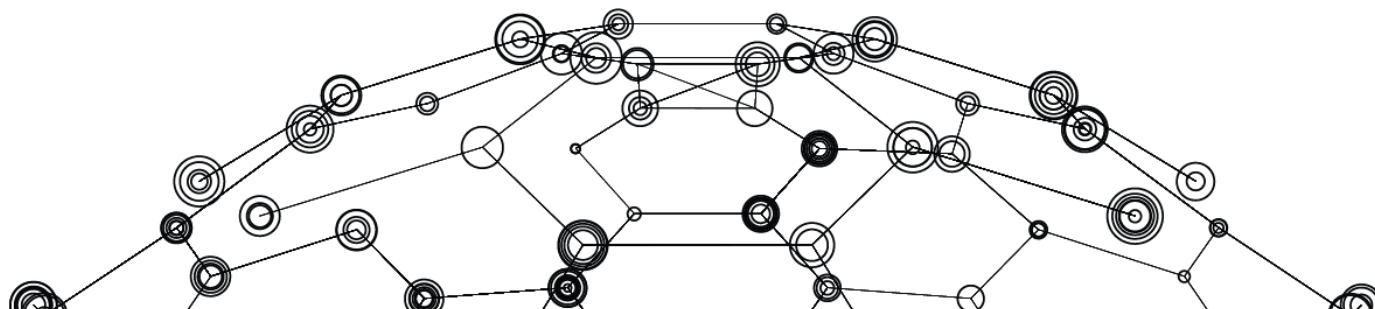
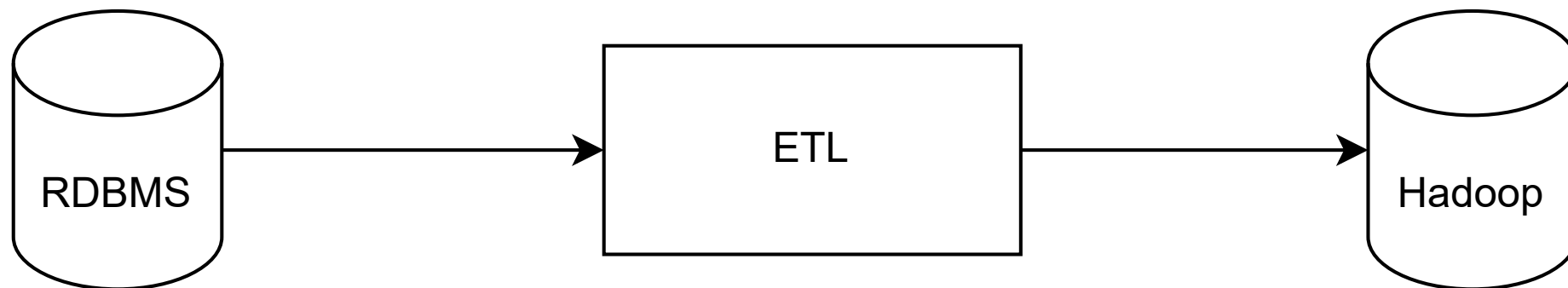


Инструменты для управления схемой данных

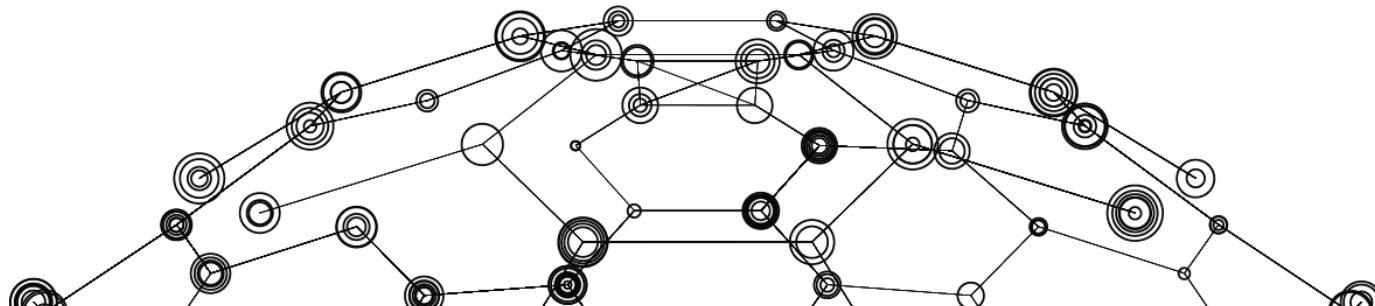
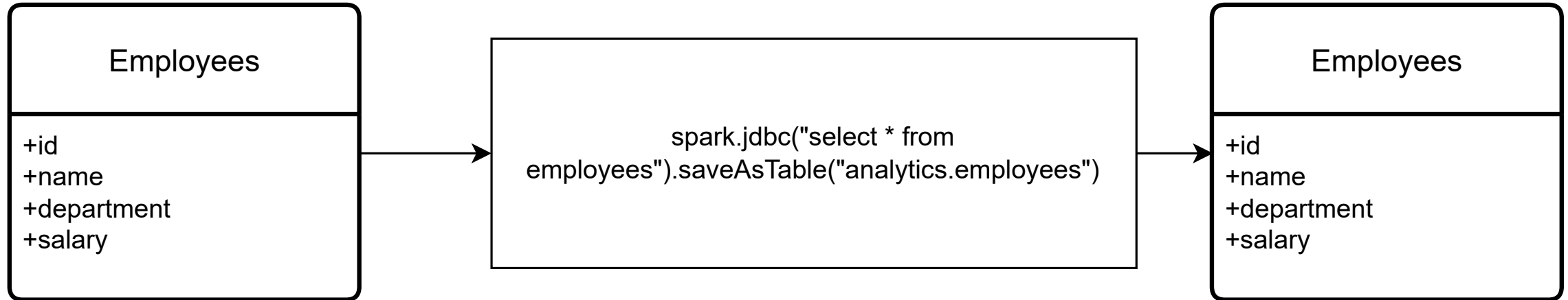


Проблема эволюции схемы данных

Типичный ETL



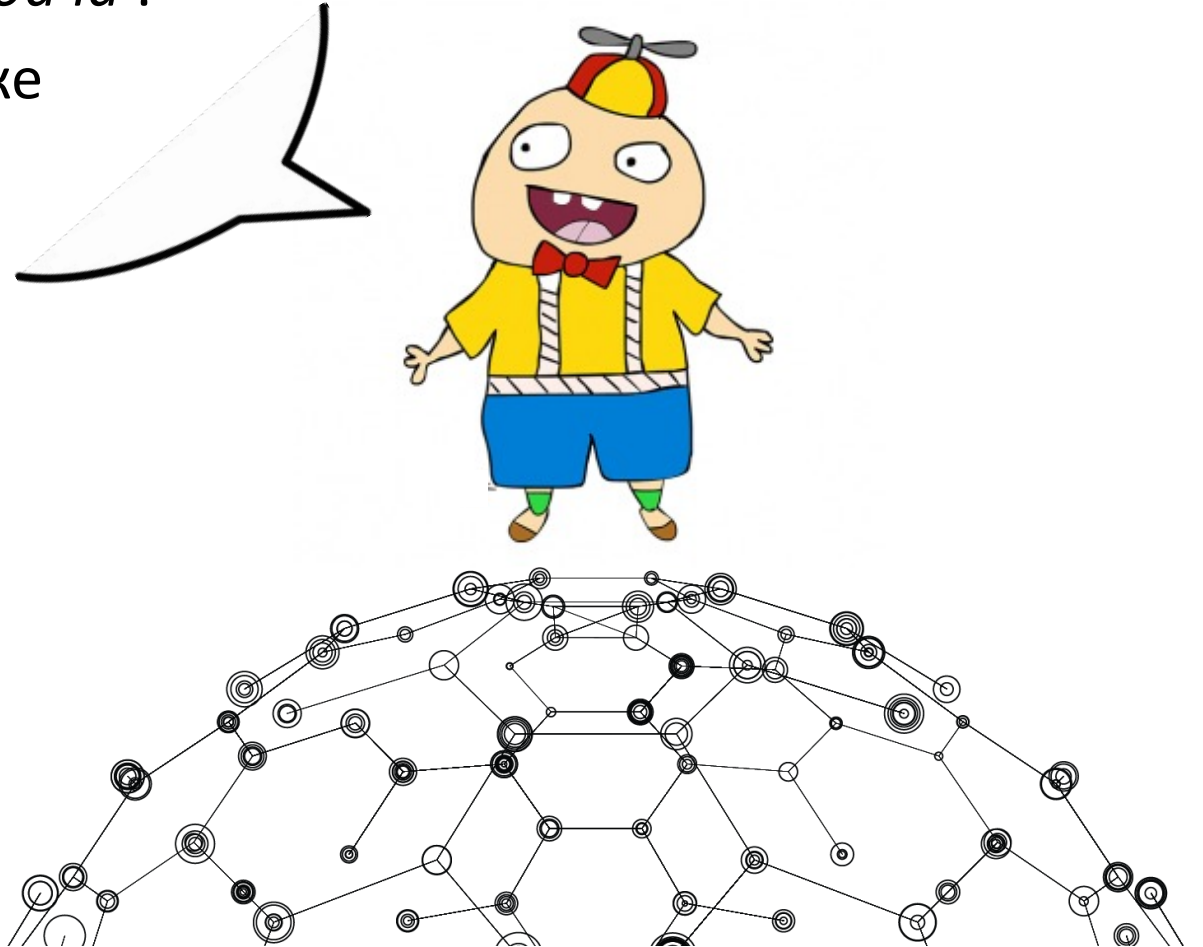
Типичный ETL



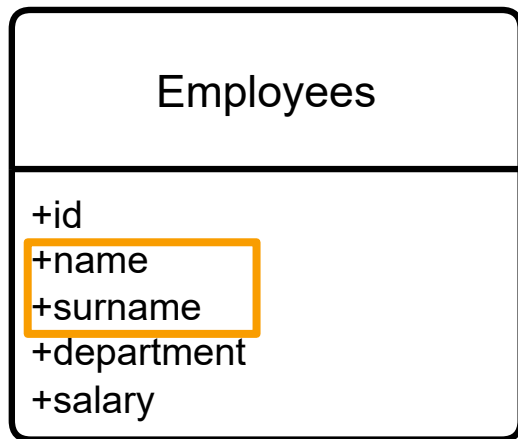
Джуниор задача



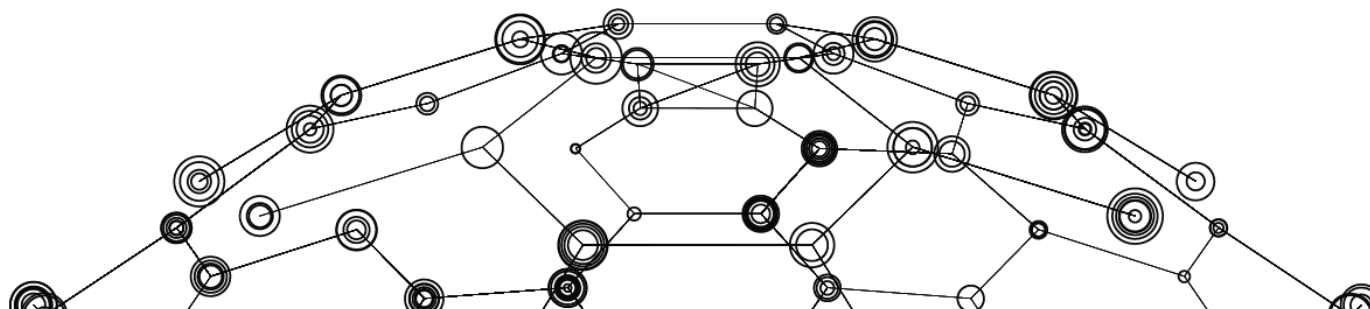
У меня есть для тебя *простая задача* :
раздели поле name в табличке
employees на 2...



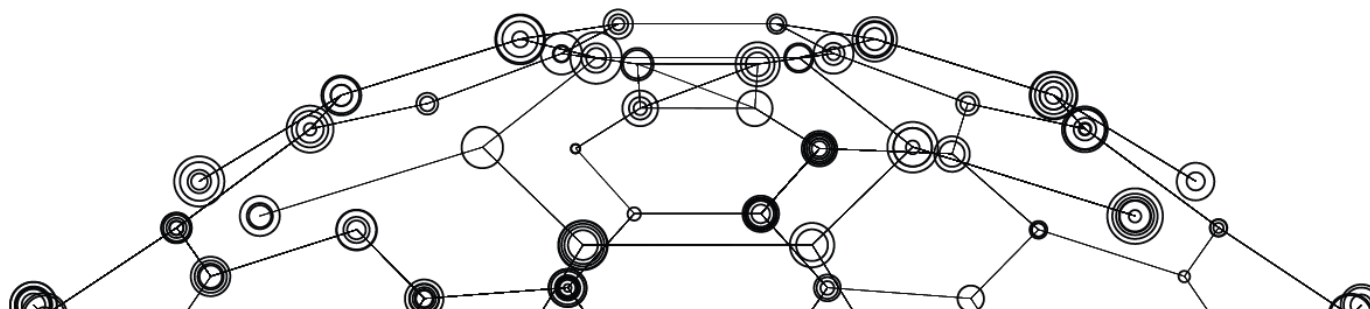
Типичное изменение в ETL



```
spark.jdbc("select * from employees").saveAsTable("analytics.employees")
```

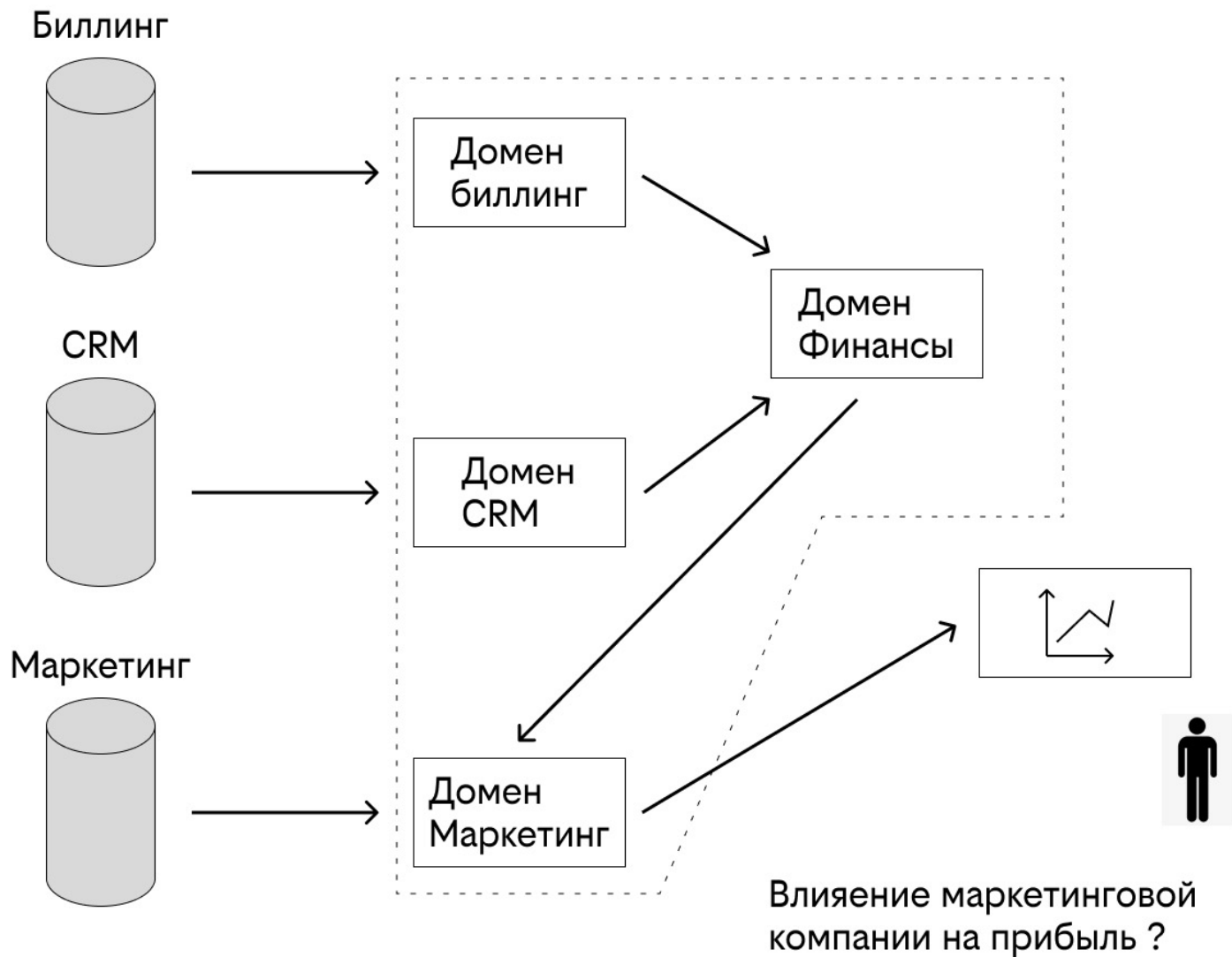


Типичное изменение в ETL



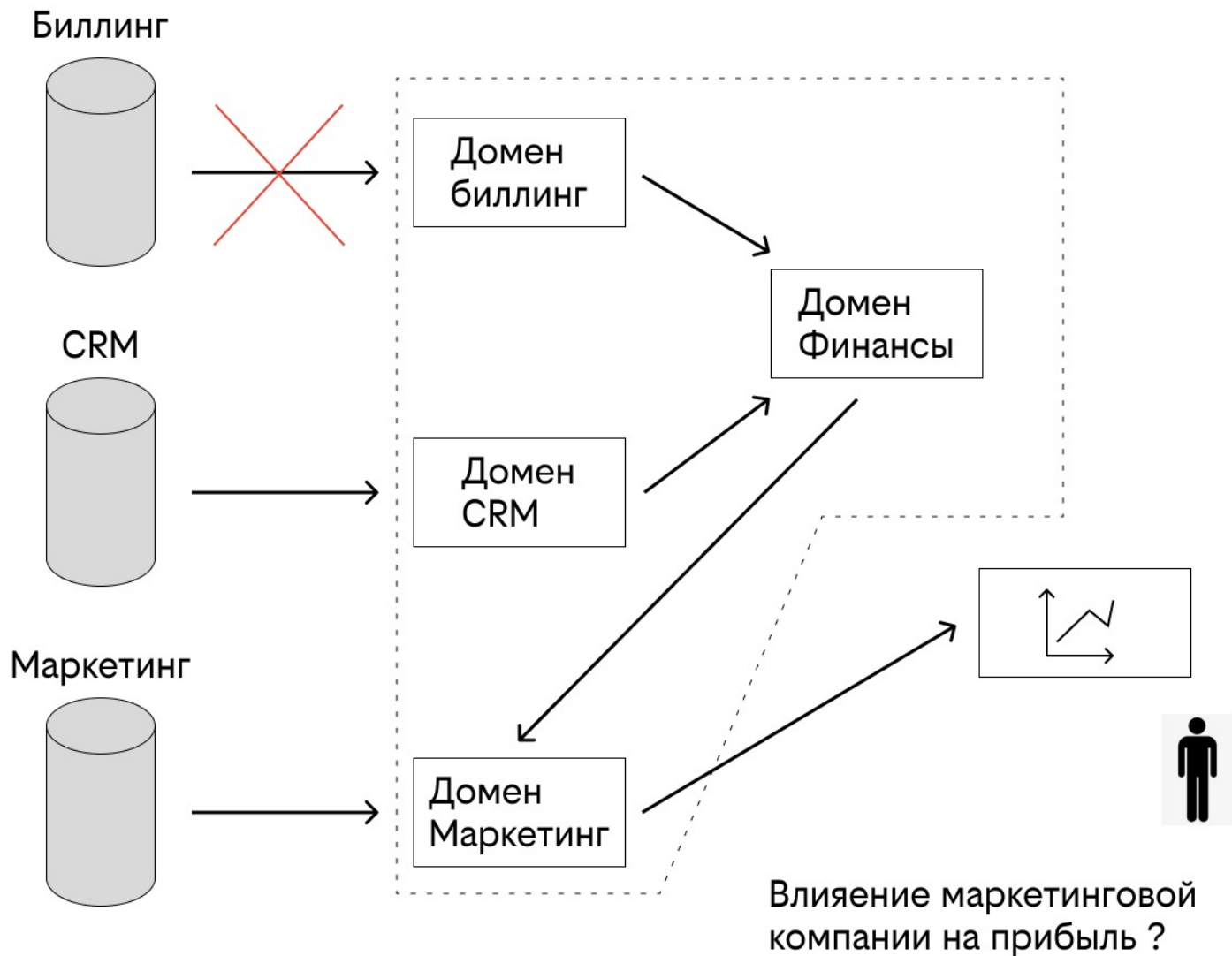
Пример посложнее

Data Mesh



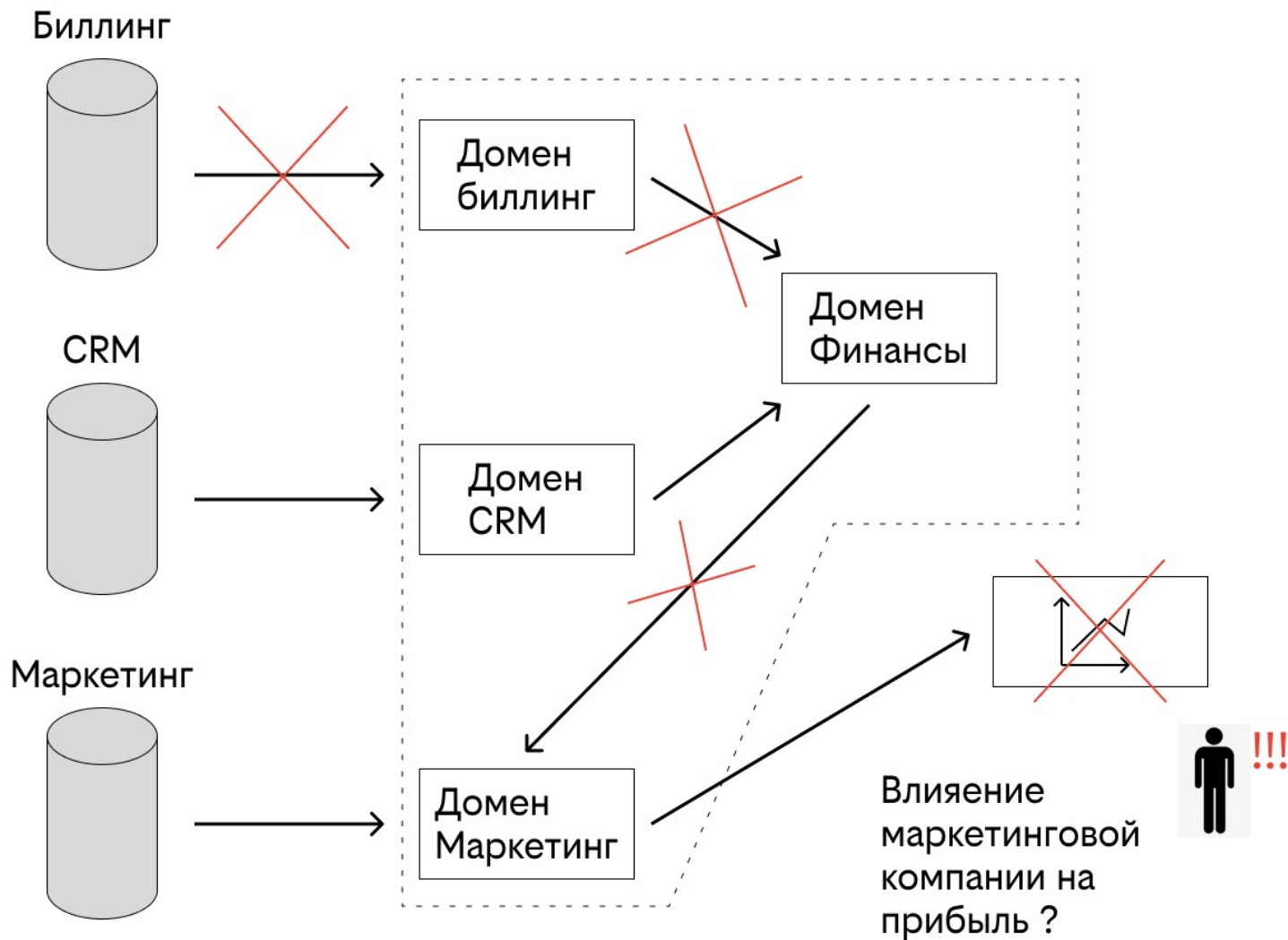
Пример посложнее

Data Mesh



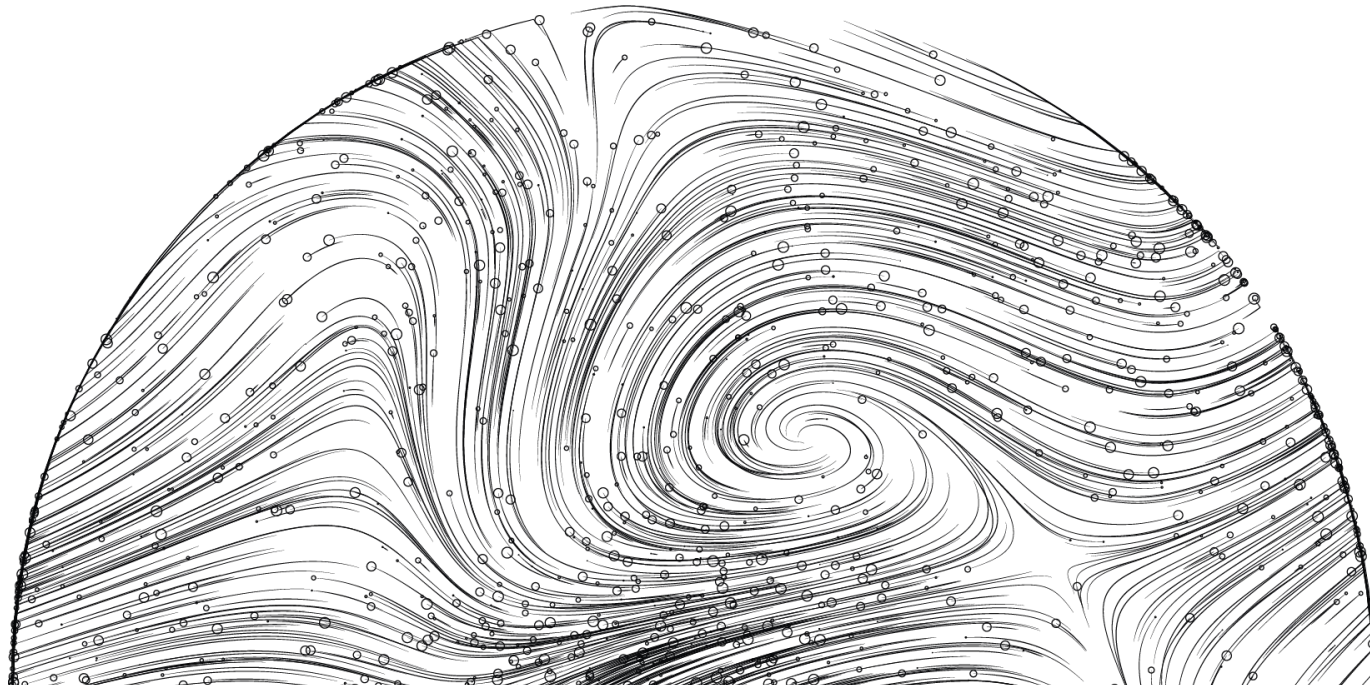
Пример посложнее

Data Mesh



К чему проблемы с данными могут привести?

- Потеря денег
- Упущенная прибыль
- Налоговые/законодательные/регуляторные риски
- Мошенничество
- Ухудшение имиджа компании



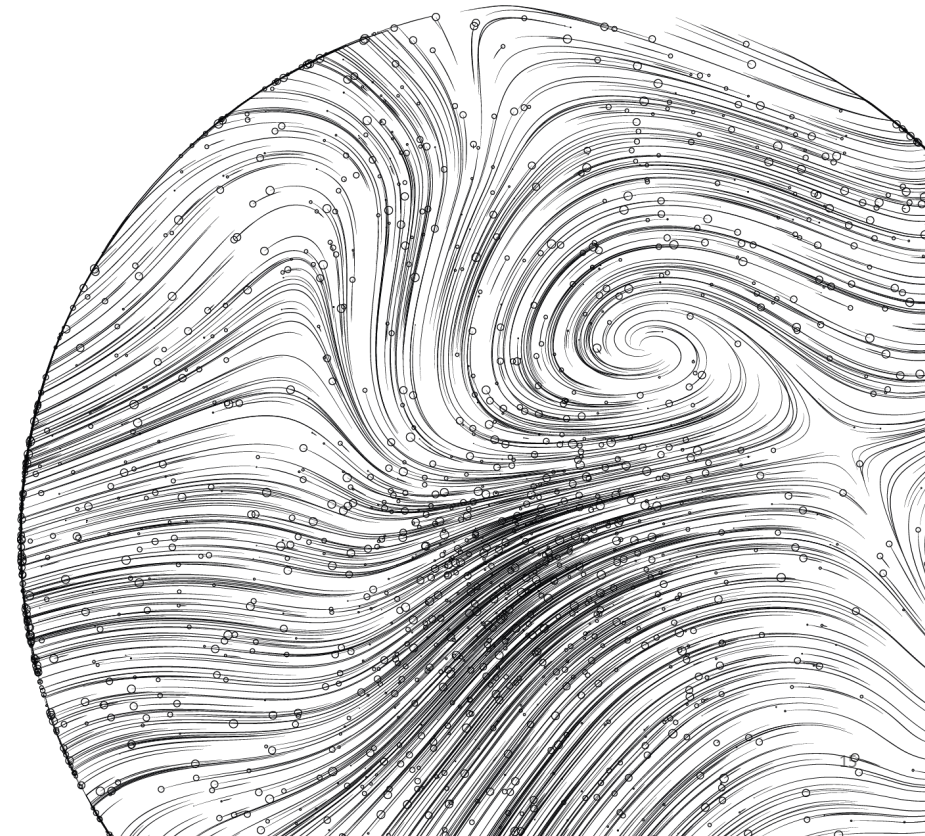
Причины проблем

- «Внезапные» изменения
- Неожиданные изменения
- Изменения, которые никто не согласовал
- Изменения, которые сделали не вовремя

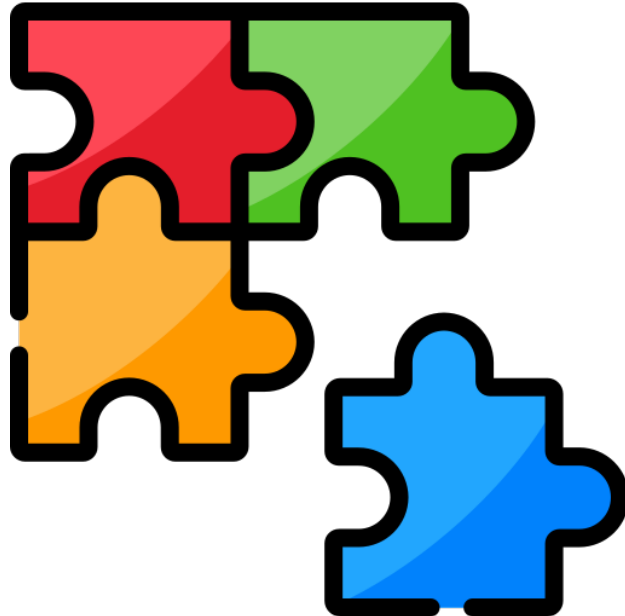




**Данные меняются!
Что делать?**



Прямая и обратная совместимость



Backward compatibility = Обратная совместимость

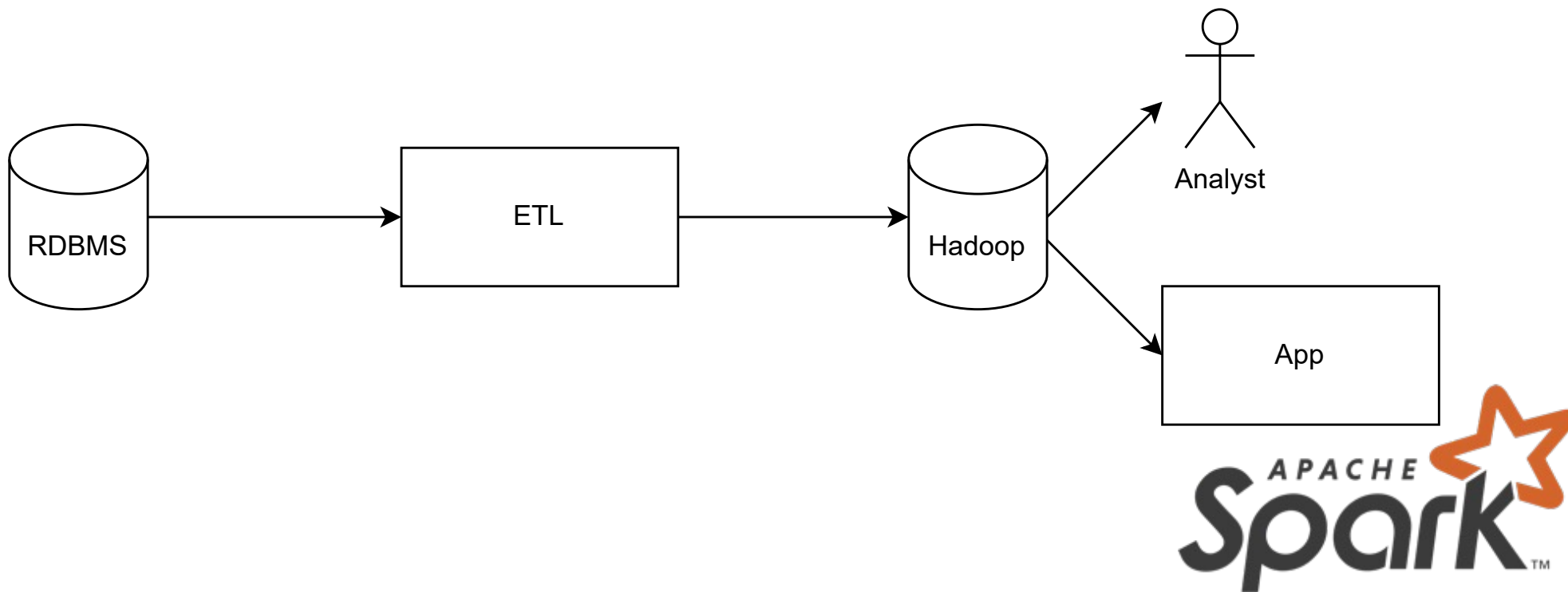
Новая версия кода **потребителя** может прочитать данные, записанные старой версией кода **производителя**

Forward compatibility = Прямая совместимость

Старая версия кода **потребителя** может прочитать данные, записанные новой версией кода **производителя**

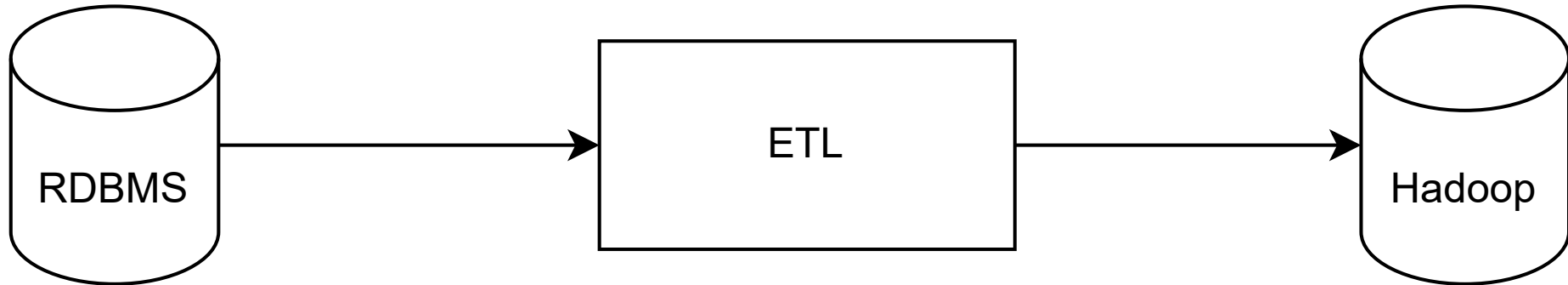
Designing Data-Intensive Applications by **Martin Kleppmann** Released
March 2017 Publisher(s): O'Reilly Media, Inc. ISBN: 9781491903100

Допущения о потребителях

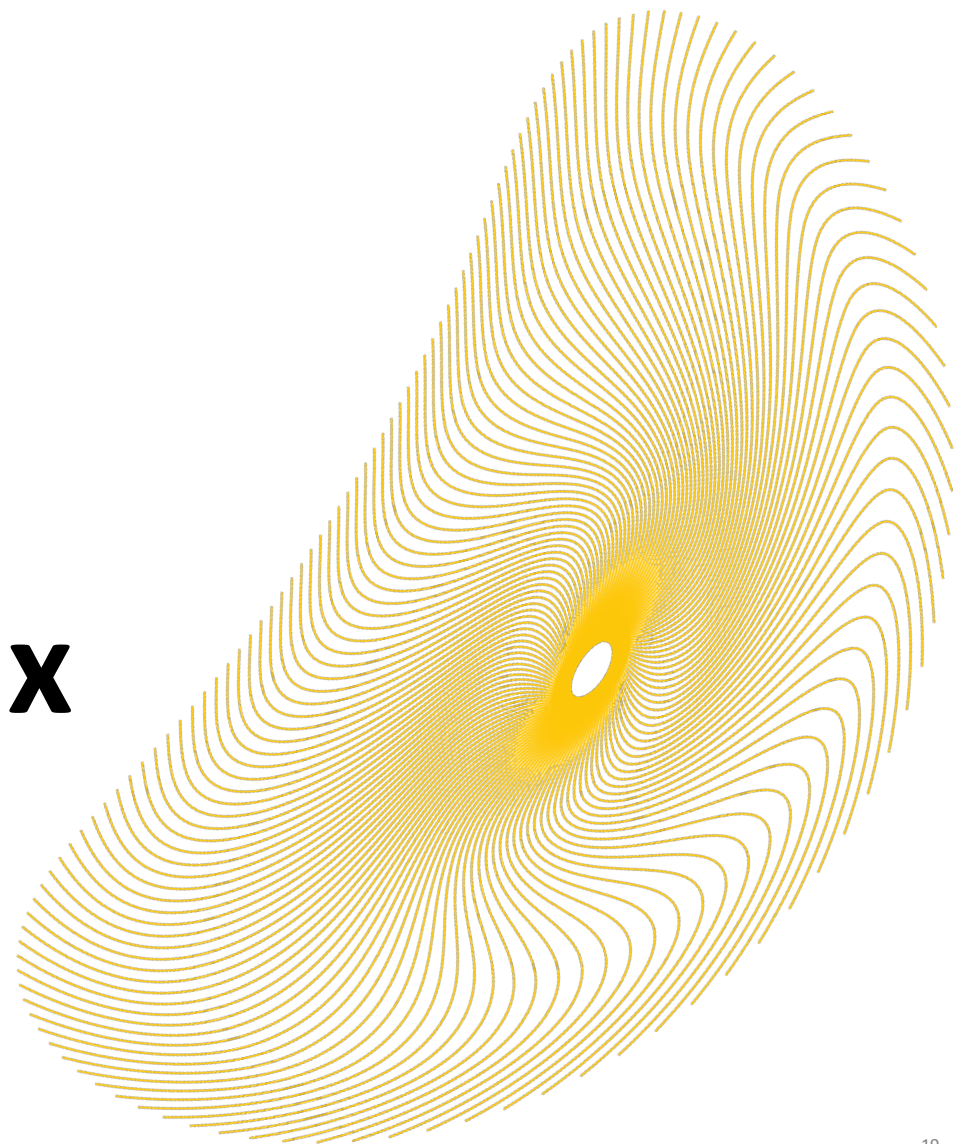


Возможные изменения

- Добавление колонки в конец таблицы = справа
- Добавление колонки в середину таблицы
- Изменение названия колонки
- Изменение порядка колонок
- Изменение типа данных колонки
- Удаление колонки



Форматы файлов для хранения реляционных данных

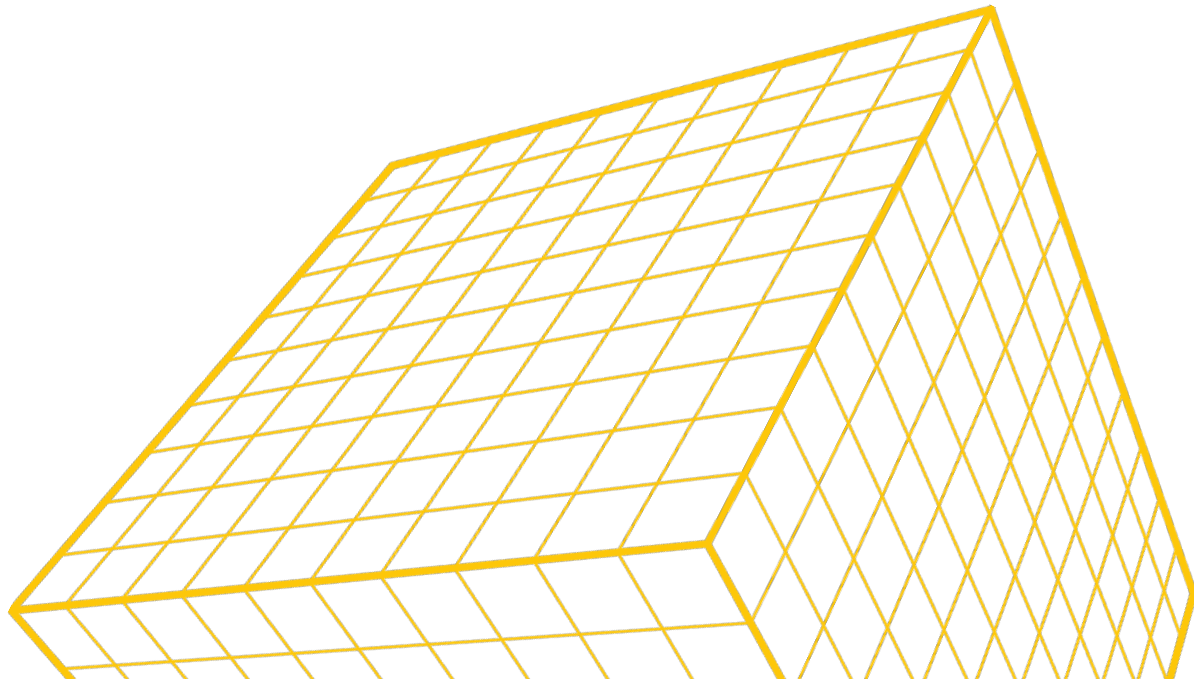


CSV



- Человеко-читаемый
- Строковый
- Схема в первой строке
- Отсутствует строгая типизация

```
id,name,department,salary
1,Ivan Ivanov,Sales,100
2,Petr Petrov,Marketing,90
3,Sidor Sidorov,R&D,110
```



Совместимость изменений схемы для CSV



Изменение	Прямая совместимость	Обратная совместимость
Добавление колонки в конец	Потребитель отбрасывает лишние колонки	-
Добавление колонки в середину	Если читать header	Если читать header
Изменение названия колонки	Если не читать header	Если не читать header
Изменение порядка колонок	Если читать header	Если читать header
Изменение типа данных колонки	+	+
Удаление колонки	-	Потребитель отбрасывает лишние колонки

JSON

- Человеко-читаемый
- Сложной структуры
- Поддерживает внешнюю схему
- Отсутствует строгая типизация

```
{  
  "employees": [  
    {  
      "id": 1,  
      "name": "Ivan Ivanov",  
      "department": "Sales",  
      "salary": 100  
    },  
    {  
      "id": 2,  
      "name": "Petr Petrov",  
      "department": "Marketing",  
      "salary": 90  
    },  
    {  
      "id": 3,  
      "name": "Sidor Sidorov",  
      "department": "R&D",  
      "salary": 100  
    }  
  ]  
}
```





Пример схемы

```
{  
  "$schema": "http://json-schema.org/draft-04/schema#",  
  "type": "object",  
  "properties": {  
    "employees": {  
      "type": "array",  
      "items": [  
        {  
          "type": "object",  
          "properties": {  
            "id": {  
              "type": "integer"  
            },  
            "name": {  
              "type": "string"  
            },  
            "department": {  
              "type": "string"  
            },  
            "salary": {  
              "type": "integer"  
            }  
          }  
        }  
      ]  
    }  
  }  
}
```

```
    "department": {  
      "type": "string"  
    },  
    "salary": {  
      "type": "integer"  
    }  
  },  
  "required": [  
    "id",  
    "name",  
    "department",  
    "salary"  
  ]  
},
```

Эволюция схемы Json



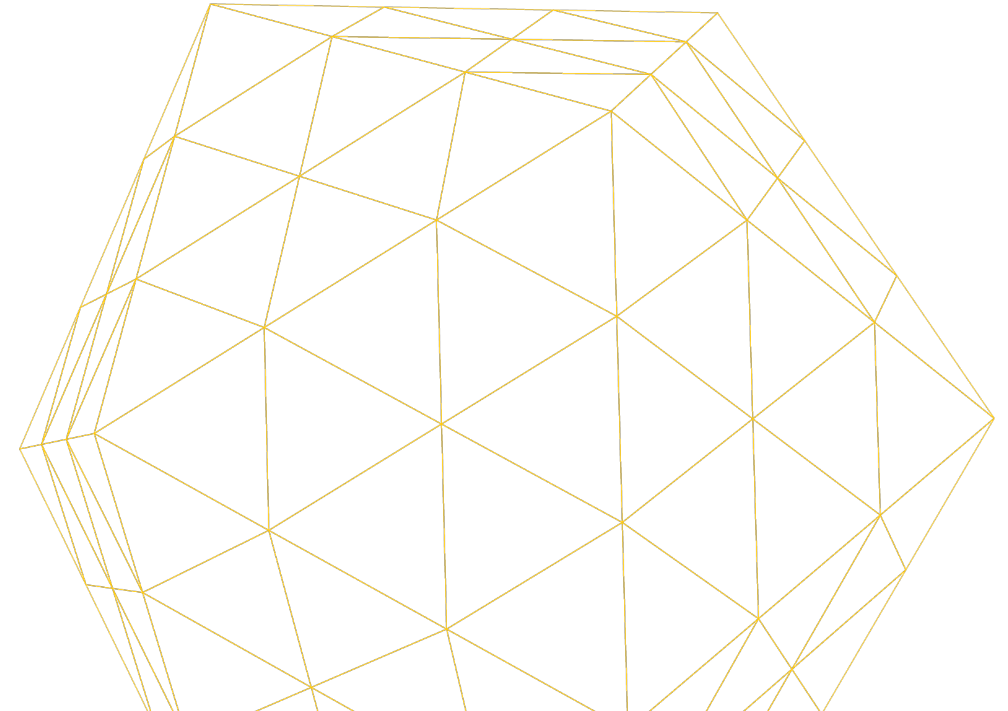
Изменение	Прямая совместимость	Обратная совместимость
Добавление колонки в конец	+, если потребитель отбрасывает лишние колонки	+
Добавление колонки в середину	+, если потребитель отбрасывает лишние колонки	+
Изменение названия колонки	-	-
Изменение порядка колонок	+	+
Изменение типа данных колонки	Для типов, которые могут быть приведены	Для типов, которые могут быть приведены
Удаление колонки	-	-

AVRO



- Бинарный
- Строковый
- С внешней схемой

```
?a Ivan $ov    snappy??v%1g?&f*  
Sales    (Y@Petr 0ovMarketing    "0?V@Sidor <ovR
```

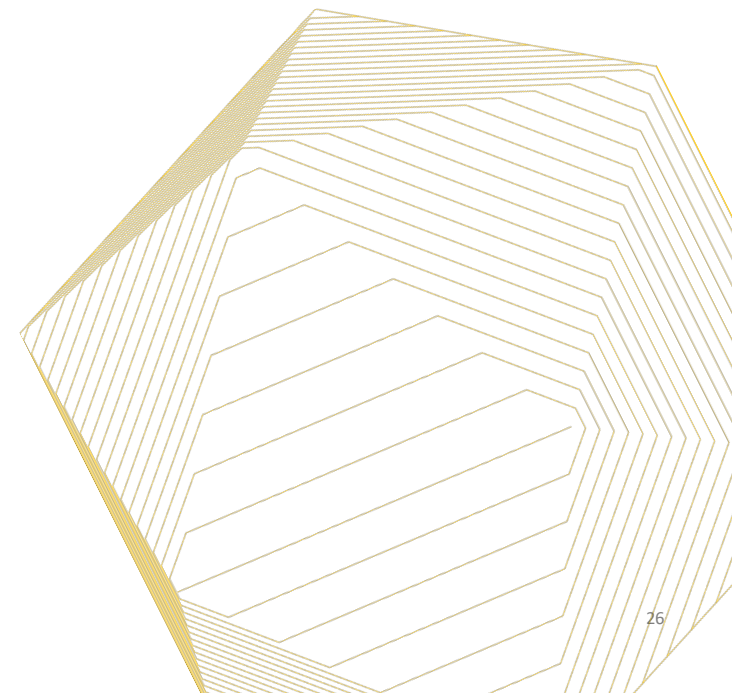


<https://avro.apache.org/docs/1.11.1/specification/>

Схема в AVRO



```
{  
  "type": "record",  
  "name": "topLevelRecord",  
  "fields": [  
    {"name": "id", "type": ["int", "null"]},  
    {"name": "name", "type": ["string", "null"]},  
    {"name": "department", "type": ["string", "null"]},  
    {"name": "salary", "type": ["double", "null"]} ]  
}
```



Совместимость изменений схемы для AVRO



Изменение	Прямая совместимость	Обратная совместимость
Добавление колонки в конец	+	Для колонок со значением по умолчанию
Добавление колонки в середину	+	Для колонок со значением по умолчанию
Изменение названия колонки	-	При помощи псевдонимов
Изменение порядка колонок	+	+
Изменение типа данных колонки	Для типов, которые могут быть приведены	Для типов, которые могут быть приведены
Удаление колонки	Для колонок со значением по умолчанию	+

Что за приведение типов такое?



Schema Resolution

A reader of Avro data, whether from an RPC or a file, can always parse that data because the original schema must be provided along with the data. However, the reader may be programmed to read data into a different schema. For example, if the data was written with a different version of the software than it is read, then fields may have been added or removed from records. This section specifies how such schema differences should be resolved.

We refer to the schema used to write the data as the writer's schema, and the schema that the application expects the reader's schema. Differences between these should be resolved as follows:

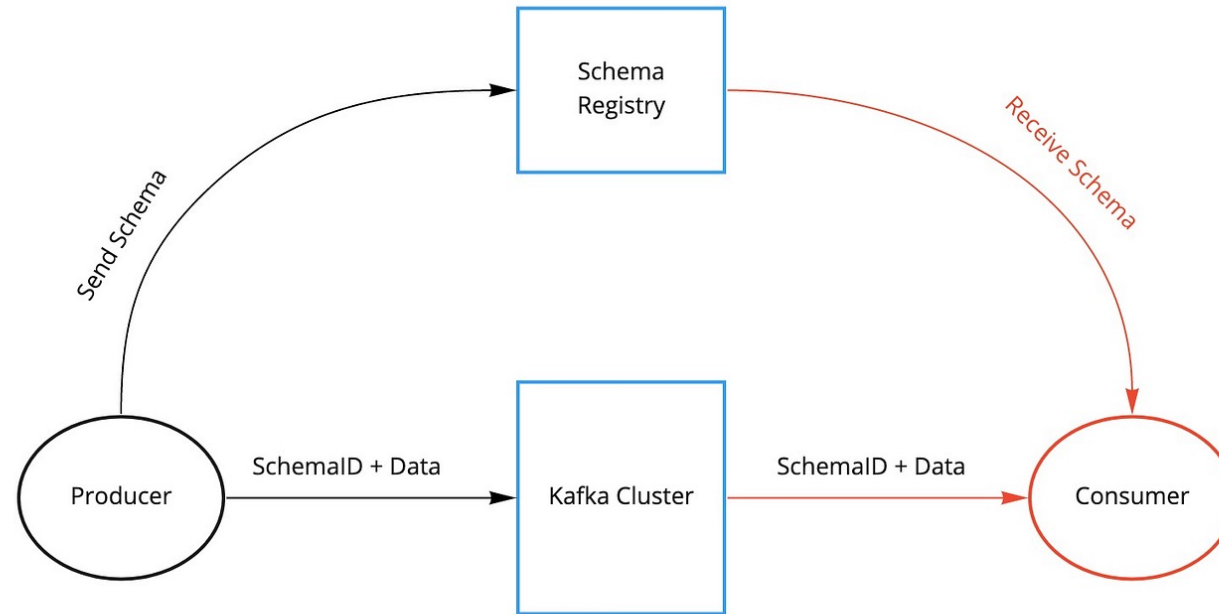
- It is an error if the two schemas do not *match*. To match, one of the following must hold:
 - both schemas are arrays whose item types match
 - both schemas are maps whose value types match
 - both schemas are enums whose (unqualified) names match
 - both schemas are fixed whose sizes and (unqualified) names match
 - both schemas are records with the same (unqualified) name
 - either schema is a union
 - both schemas have same primitive type
 - the writer's schema may be promoted to the reader's as follows:
 - int is promotable to long, float, or double
 - long is promotable to float or double
 - float is promotable to double
 - string is promotable to bytes
 - bytes is promotable to string
- **if both are records:**
 - the ordering of fields may be different: fields are matched by name.
 - schemas for fields with the same name in both records are resolved recursively.
 - if the writer's record contains a field with a name not present in the reader's record, the writer's value for that field is ignored.
 - if the reader's record schema has a field that contains a default value, and writer's schema does not have a field with the same name, then the reader should use the default value from its field.
 - if the reader's record schema has a field with no default value, and writer's schema does not have a field with the same name, an error is signalled.
- **if both are enums:** if the writer's symbol is not present in the reader's enum and the reader has a default value, then that value is used, otherwise an error is signalled.
- **if both are arrays:** This resolution algorithm is applied recursively to the reader's and writer's array item schemas.
- **if both are maps:** This resolution algorithm is applied recursively to the reader's and writer's value schemas.
- **if both are unions:** The first schema in the reader's union that matches the selected writer's union schema is recursively resolved against it. If none match, an error is signalled.
- **if reader's is a union, but writer's is not** The first schema in the reader's union that matches the writer's schema is recursively resolved against it. If none match, an error is signalled.
- **if writer's is a union, but reader's is not** If the reader's schema matches the selected writer's schema, it is recursively resolved against it. If they do not match, an error is signalled.

<https://avro.apache.org/docs/1.11.1/specification/#schema-resolution>



A schema's *doc* fields are ignored for the purposes of schema resolution. Hence, the *doc* portion of a schema may be dropped at serialization.

Confluent Schema Registry и автоматическая валидация



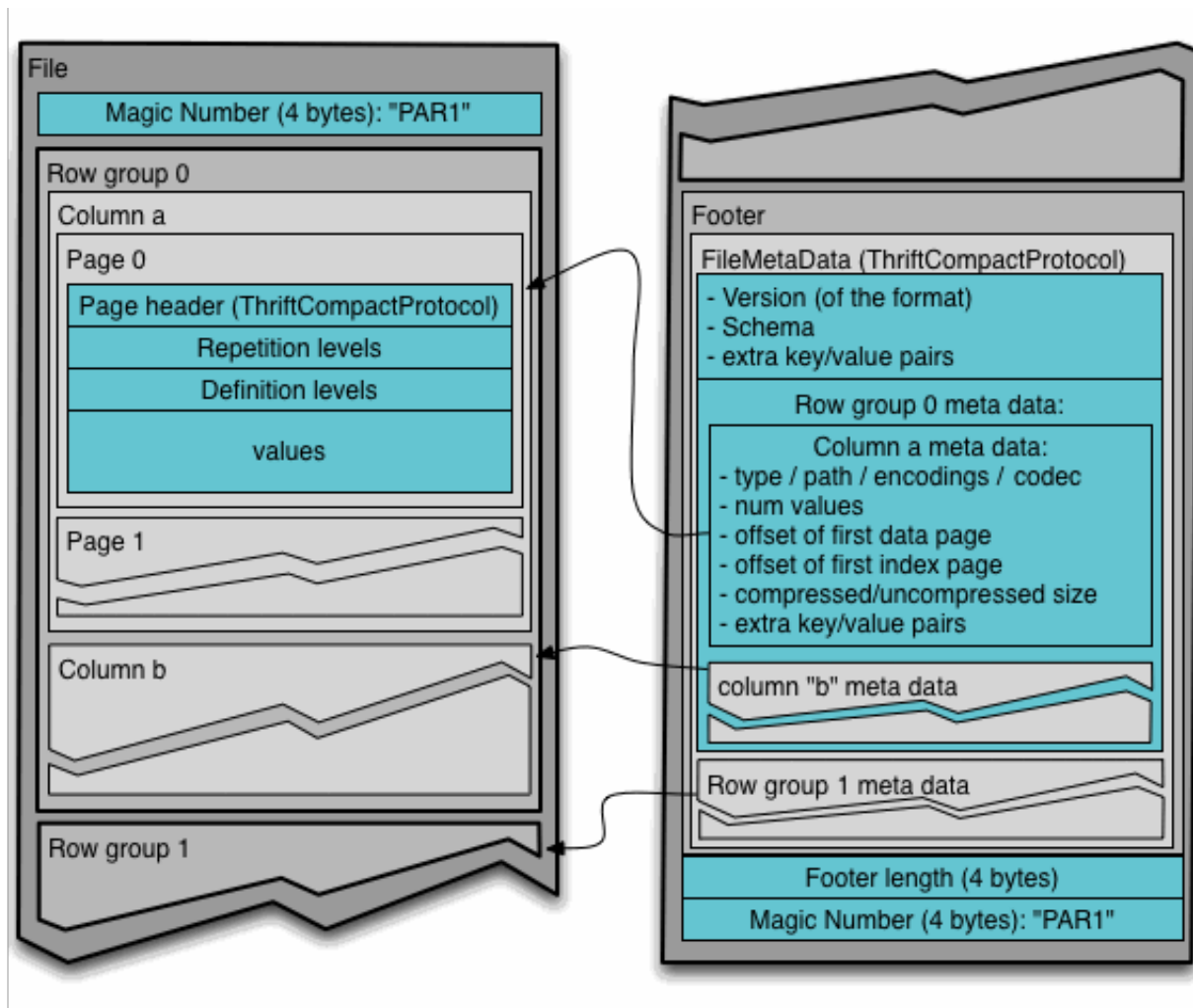
<https://docs.confluent.io/platform/current/schema-registry/fundamentals/schema-evolution.html>

<https://docs.confluent.io/platform/current/schema-registry/index.html>

Parquet



- Бинарный
- Колоночный
- Схема внутри файла
- Блочный





Пример файла и схемы Parquet

```

PAR1$(????      Djh?58Ѓ
Sidor SidorovFJ???#?Salesn ovPetMarketingR&D<8???Y@(?V@?[@
Sidor Sidorov      MarketingSale?V?[V^??x?d\H              Ivan Ivanov
                                spark_schem%id
                                %name%L
                                %

```

```

department%L
%salaryLidRV<(??.&^
Sidor Sidorov      name??&^<6(
      Ivan Ivanov??N&?
                                5

```

```

departmenttx&?<6(Sales Marketing??:&?
salaryhd&??[?V@?[?V@??>??,org.apache.spark.version3.4.1)org.apache.spark.sql.parquet.ro
w.metadata>{"type":"struct","fields":[{"name":"id","type":"integer","nullable":true,"me
tadata":{"scale":0}},{"name":"name","type":"string","nullable":true,"metadata":{"scale"
:0}},{"name":"department","type":"string","nullable":true,"metadata":{"scale":0}},{"nam
e":"salary","type":"double","nullable":true,"metadata":{"scale":17}}]}Jparquet-mr versi
on 1.12.3 (build f8dced182c4c1fbdec6ccb3185537b5a01e6ed6b)L?PAR1%

```

Совместимость изменений схемы для Parquet

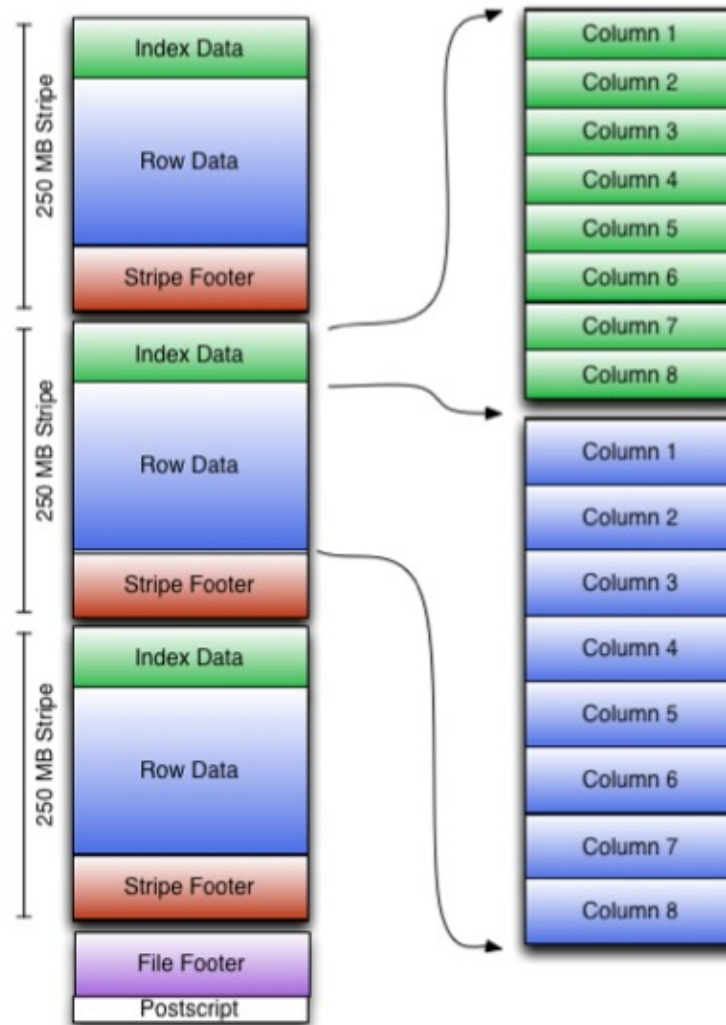


Изменение	Прямая совместимость	Обратная совместимость
Добавление колонки в конец	+	- *
Добавление колонки в середину	+, если маппинг осуществляется по имени колонки	- *
Изменение названия колонки	-	-
Изменение порядка колонок	+, если маппинг осуществляется по имени колонки	+, если маппинг осуществляется по имени колонки
Изменение типа данных колонки	Для типов, которые могут быть приведены	Для типов, которые могут быть приведены
Удаление колонки	Для nullable колонок	+

* +, если в Spark включена опция `mergeSchema=true`

ORC

- Бинарный
- Колоночный
- Схема внутри файла
- Блочный





```

ORC
P+

P_
-
"

Sidor SidorovFPK
#
"
    MarketingSales"PS
,
!    ?V@?[@?r@P    F$`GIvan IvanovPetr PetrovSidor Sidorov F??#SalesMarketingR&D    FY0,Y@(?V@?[@?1

(,
&
X
?
P

PX
"

Sidor SidorovFPX-
"
    MarketingSales"PX
    ?V@?[@?r@PX????
        ?h ]("&
            idname
departmentssalary"    ?V*!
org.apache.spark.version3.4.10P:
    PX:"

Sidor ?ovFPX-"
    MarketingSales"PX#    ?V@

?[T?r@PX@?NHXb1.8.???"
    (?0    ??ORC%

```

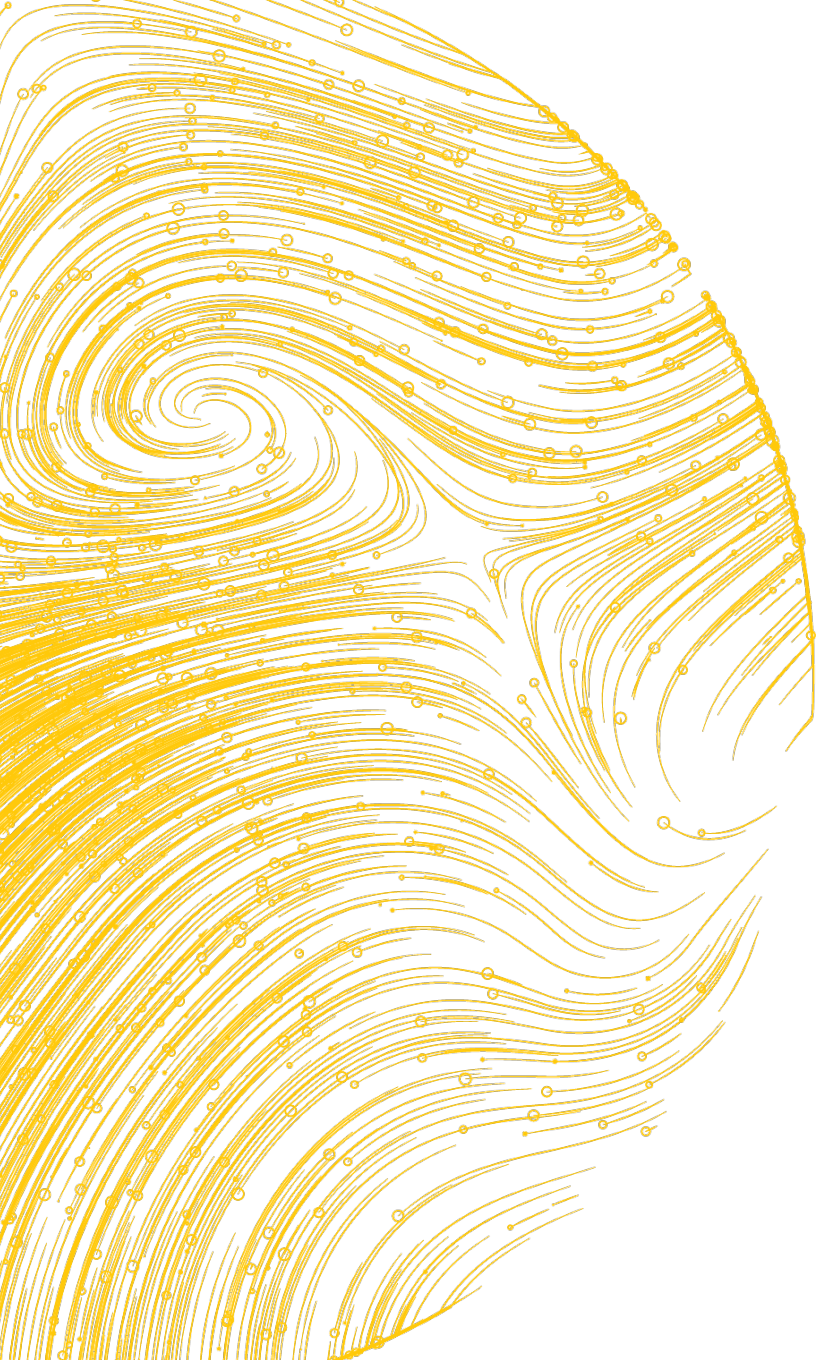


Совместимость изменений схемы для ORC



Изменение	Прямая совместимость	Обратная совместимость
Добавление колонки в конец	+	- *
Добавление колонки в середину	+, если маппинг осуществляется по имени колонки	- *
Изменение названия колонки	-	-
Изменение порядка колонок	+, если маппинг осуществляется по имени колонки	+, если маппинг осуществляется по имени колонки
Изменение типа данных колонки	Для типов, которые могут быть приведены	Для типов, которые могут быть приведены
Удаление колонки	Для nullable колонок	+

* +, если в Spark включена опция `mergeSchema=true`

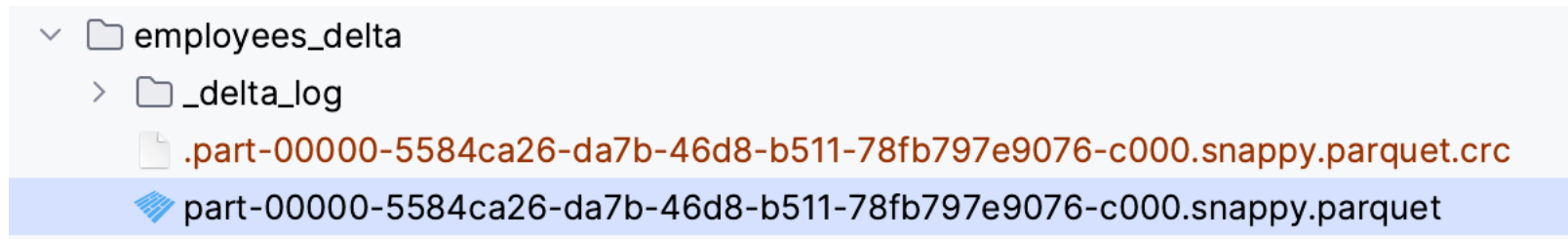


Инструменты для управления схемой данных

Delta lake



- ACID транзакции на Spark
- Расширенные возможности управления метаданными
- Один интерфейс для потоковой и пакетной обработки
- Принудительное соблюдение совместимости схемы данных
- Работа со срезами данных = «машина времени»
- Поддержка обновления и удаления данных



<https://delta.io/blog/2023-02-08-delta-lake-schema-evolution/>

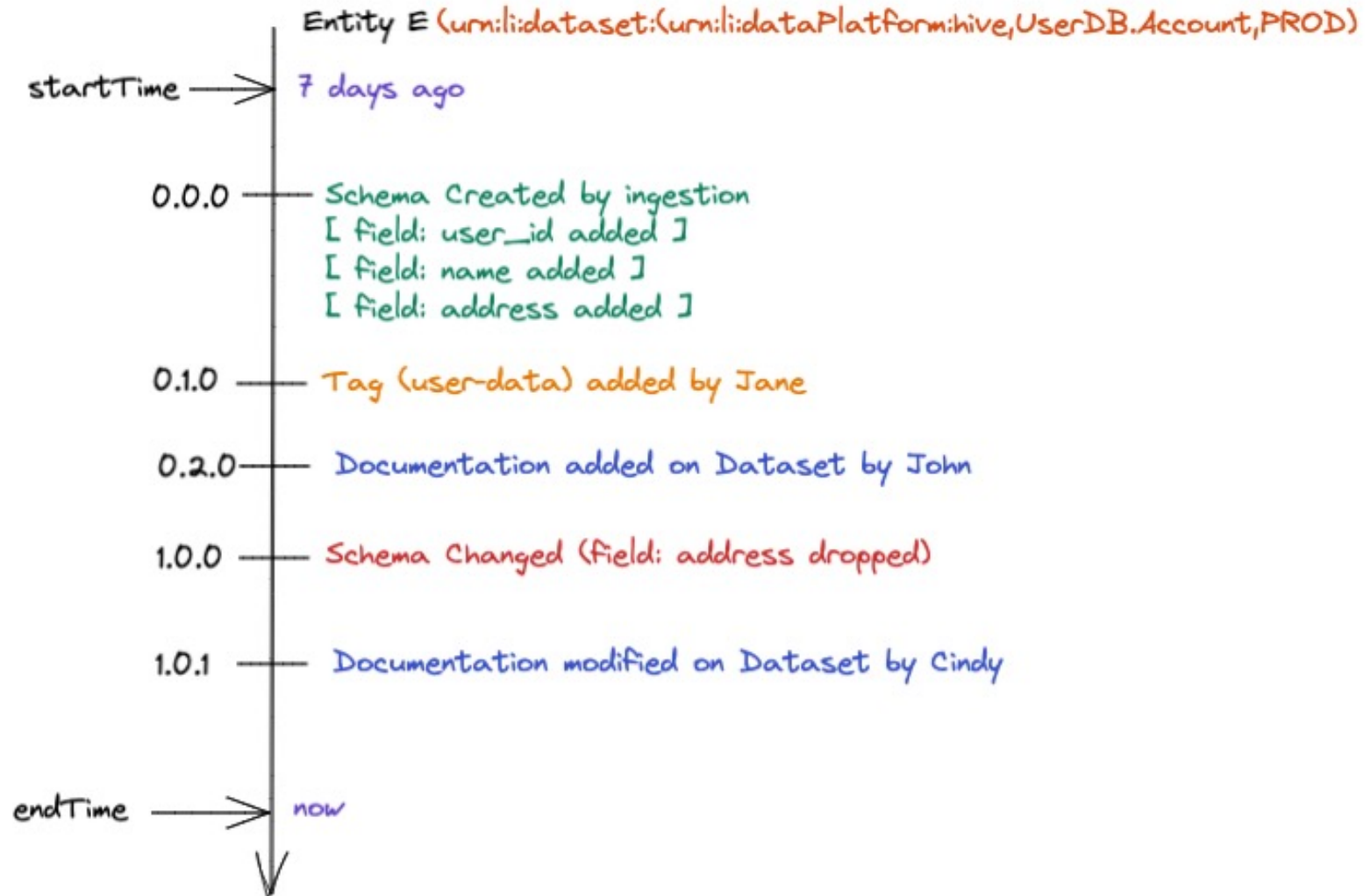
DataHub



Система управления
метаданными
с возможностью
историзации датасетов
и управлением
эволюцией схемы



<https://datahubproject.io/>



Dremio



Инструмент для построения Data Lakehouse

- СУБД
- S3-like хранилища

Поддерживает Apache Iceberg

Фичи:

- Версионирование датасетов
- Ветки для датасетов

<https://docs.dremio.com/>





Архитектурные ПОДХОДЫ К ЭВОЛЮЦИИ

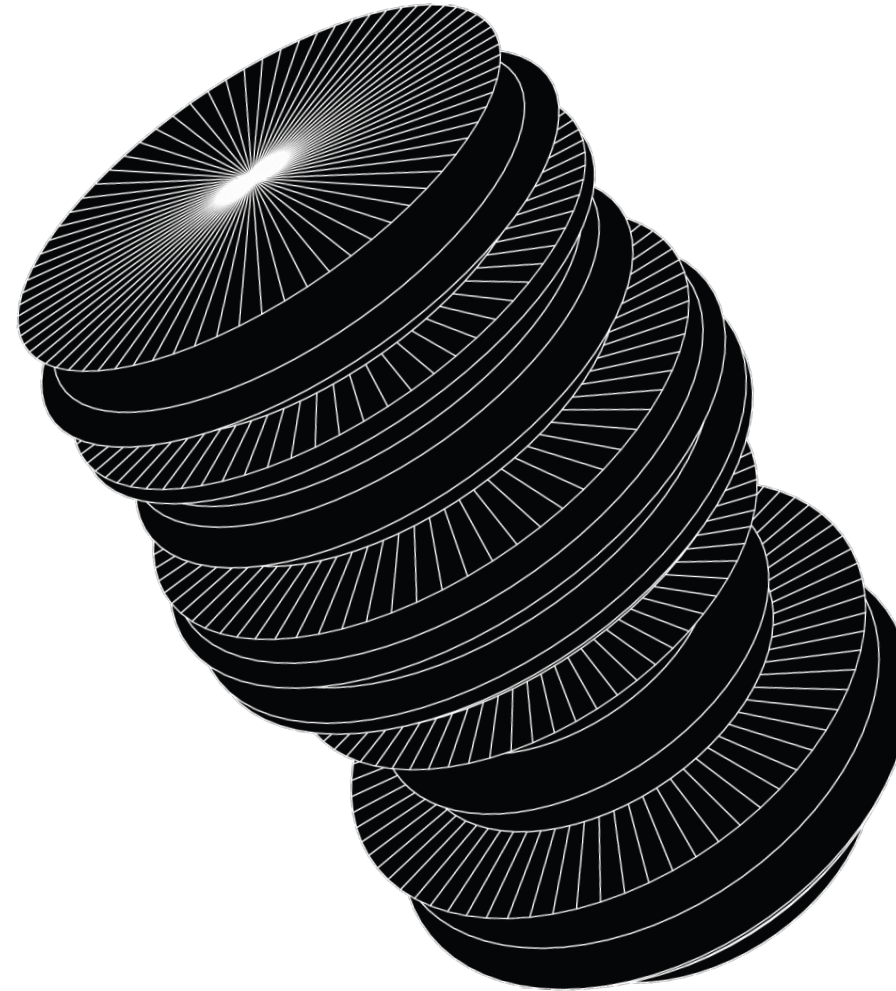


Схема «Звезда»

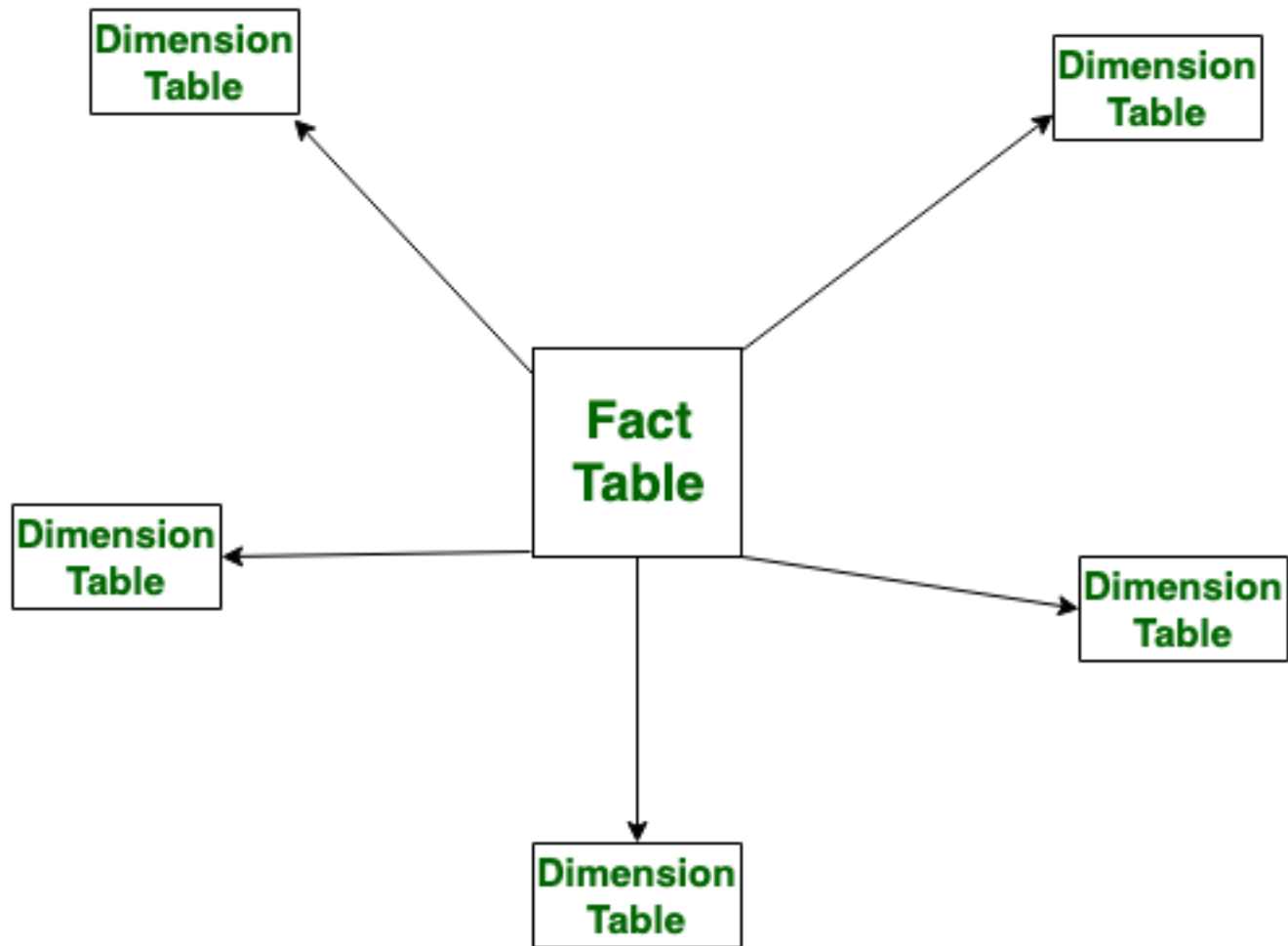
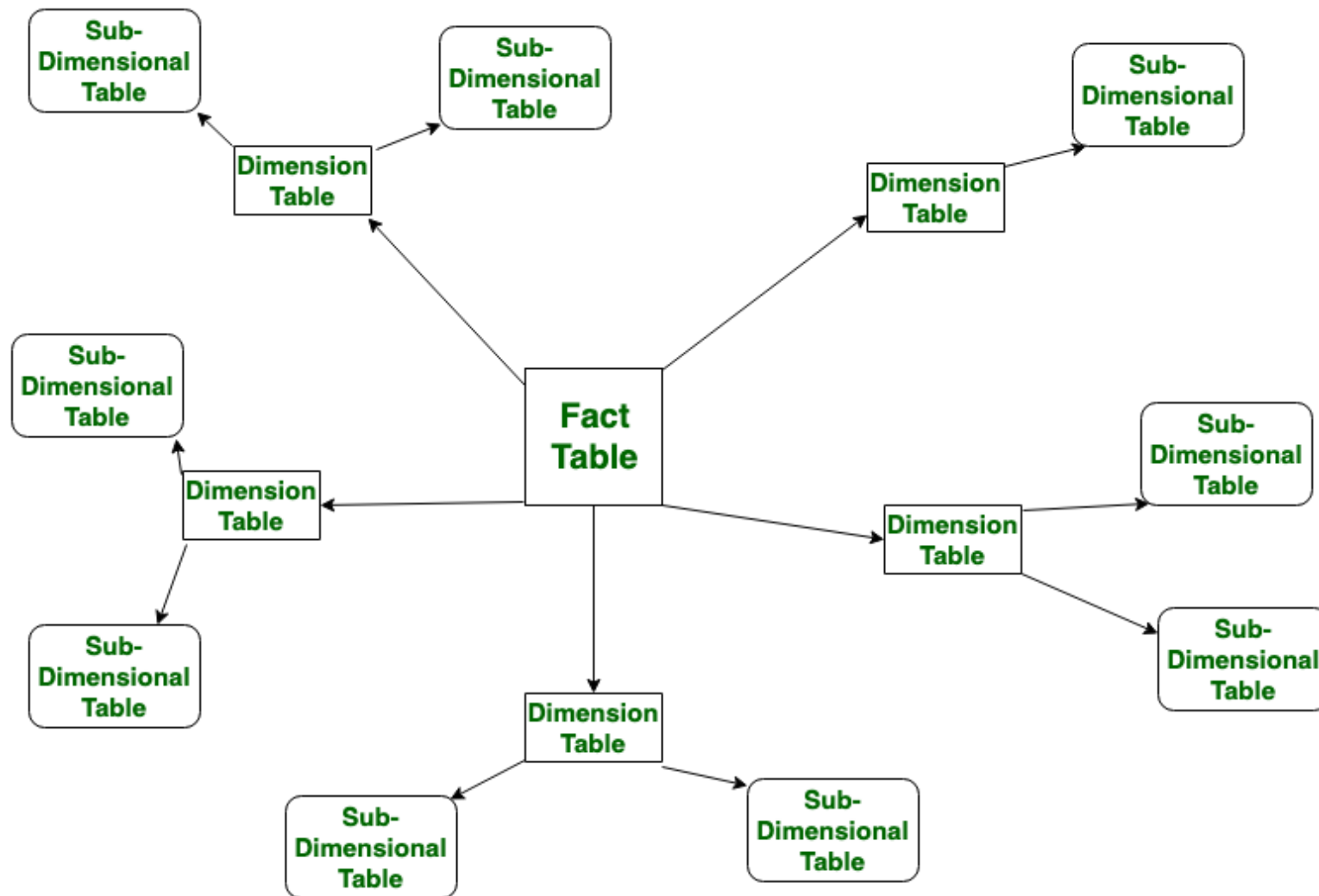
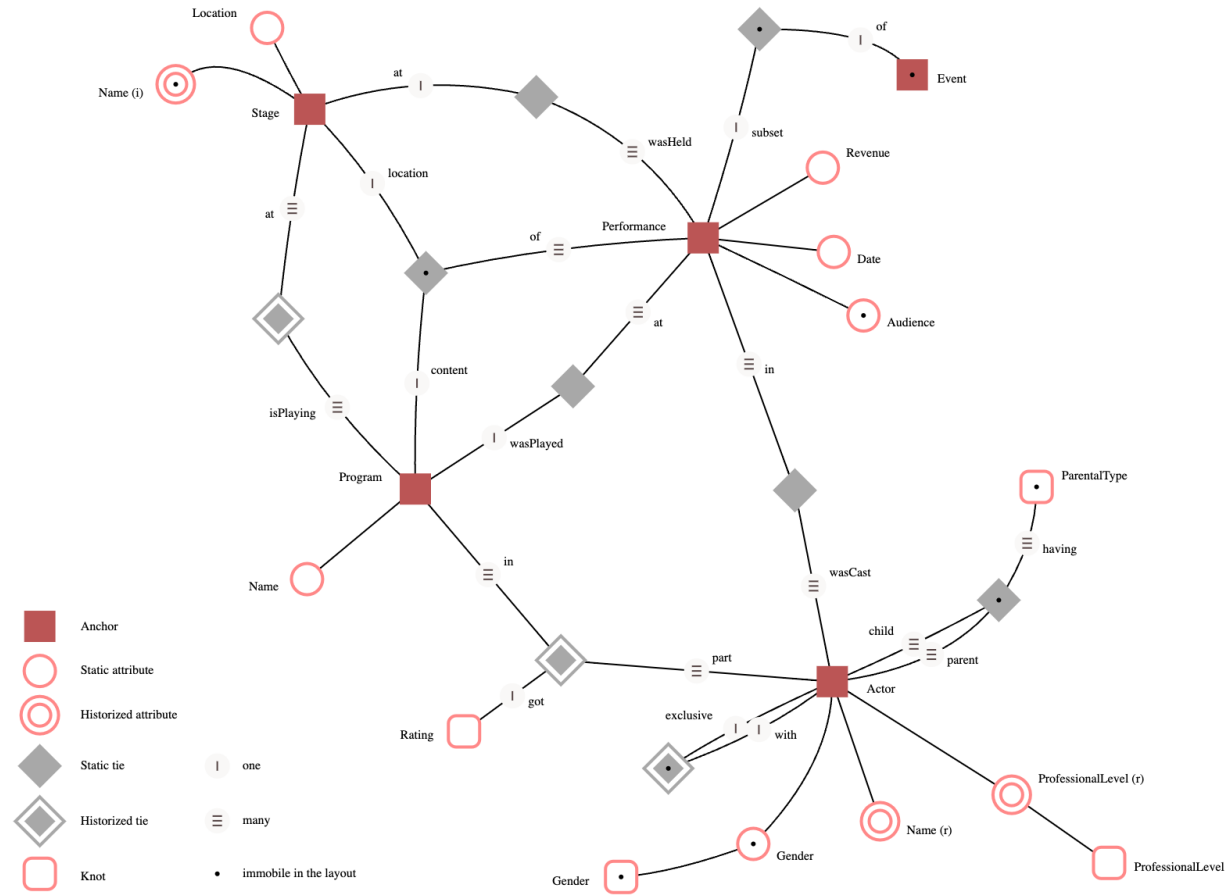


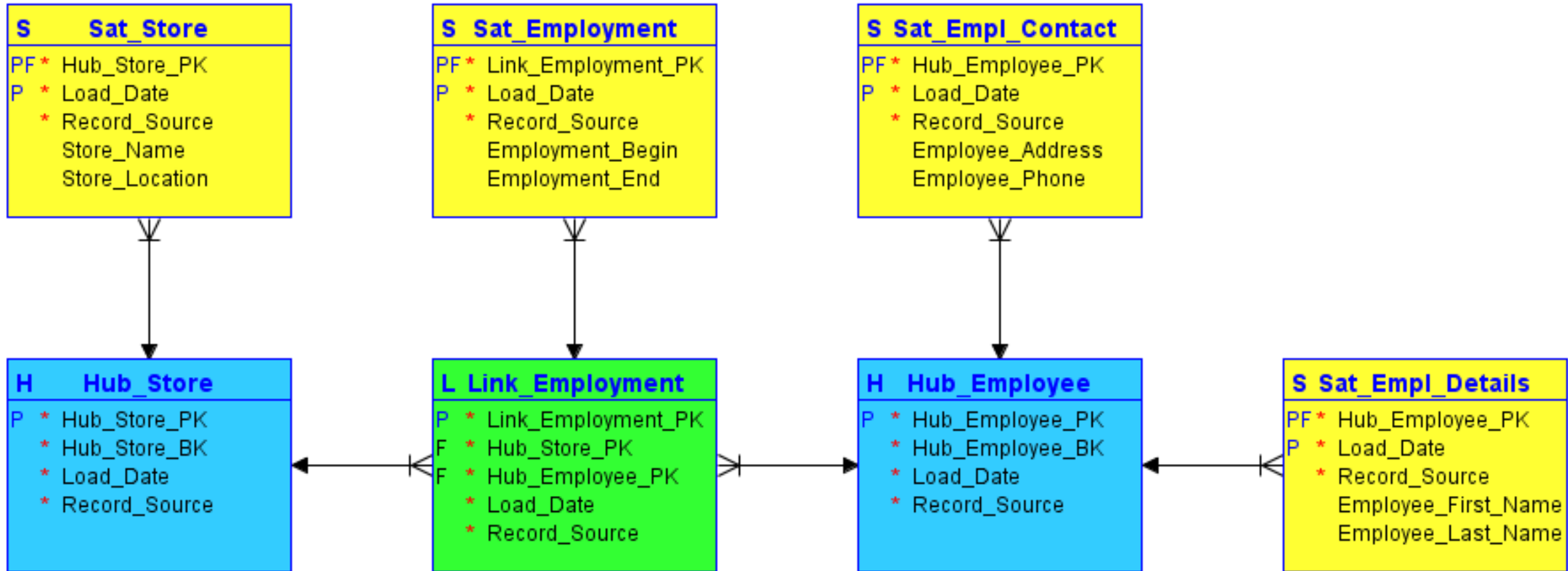
Схема «Снежинка»

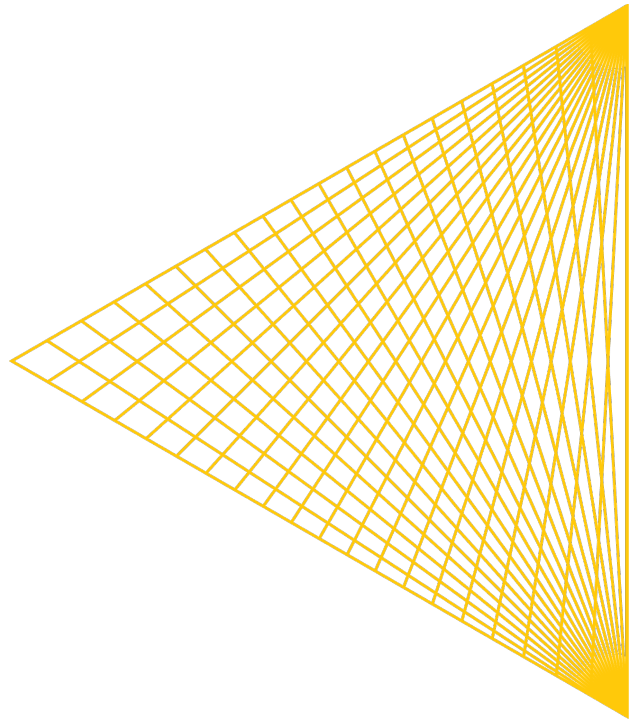


Якорная модель = Anchor model



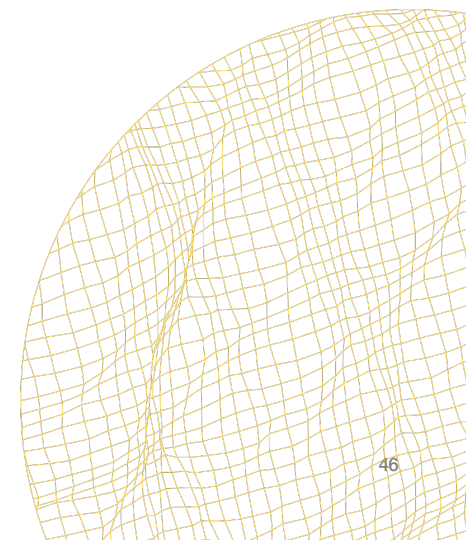
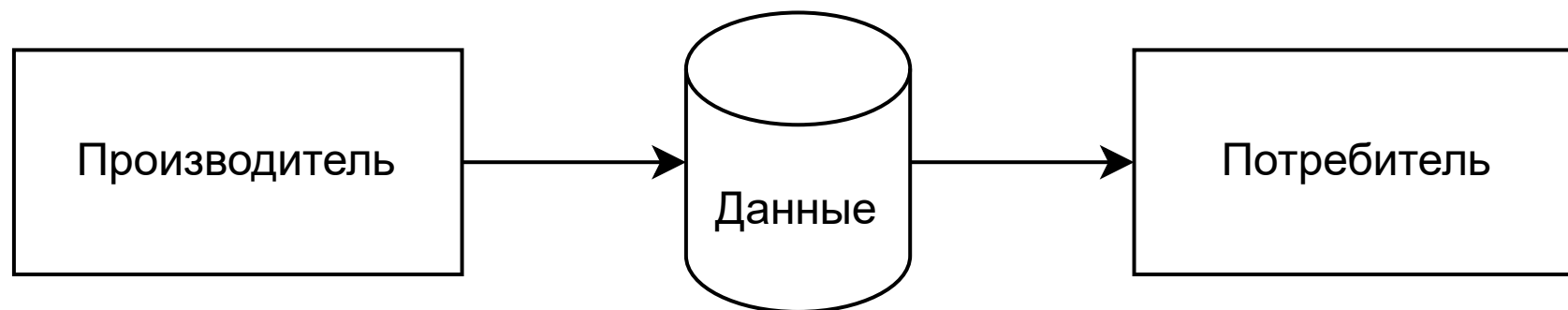
Data Vault



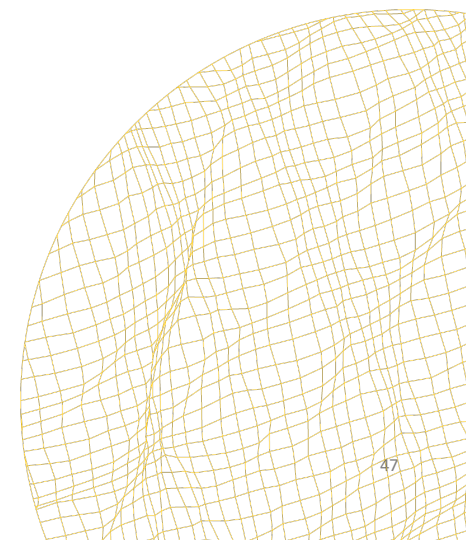
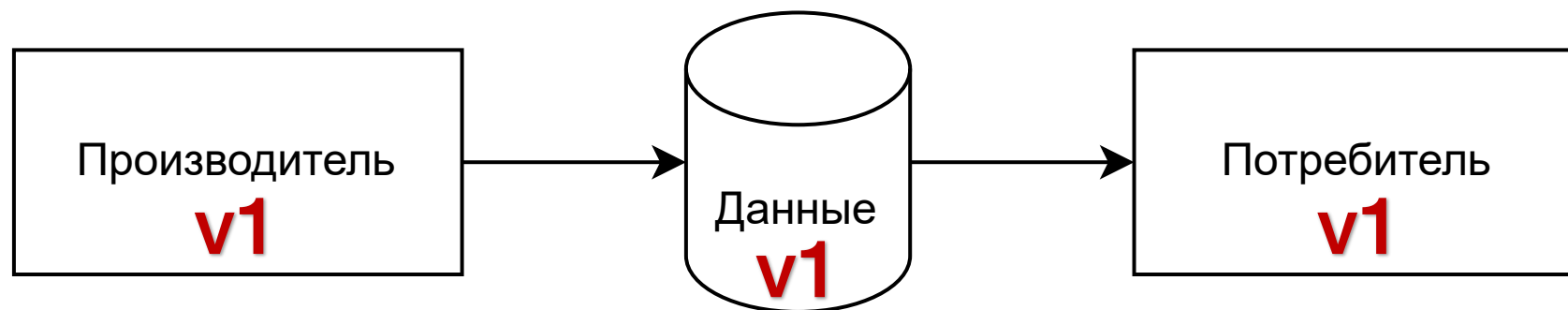


**А что, если ничего не
помогает?
Как делать
несовместимое
изменение?**

Алгоритмы миграции

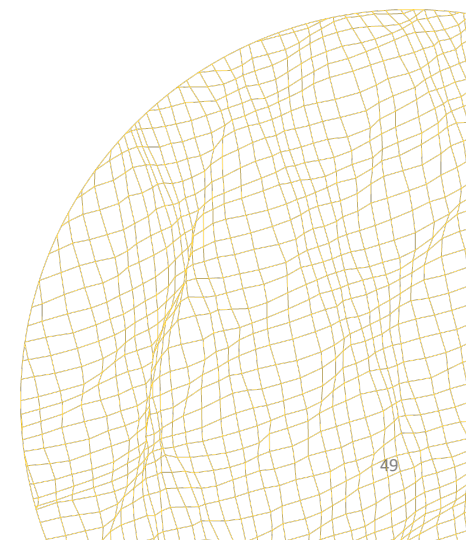
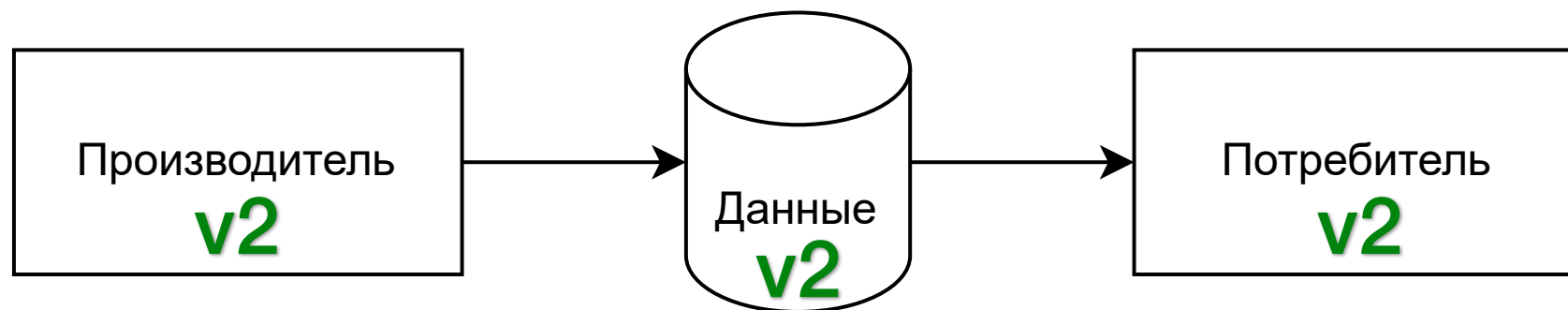


Алгоритмы миграции

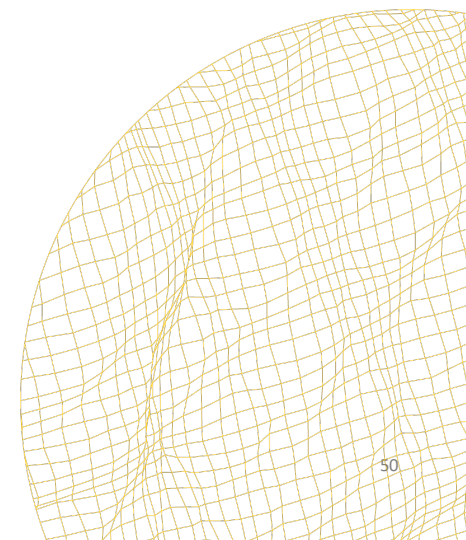
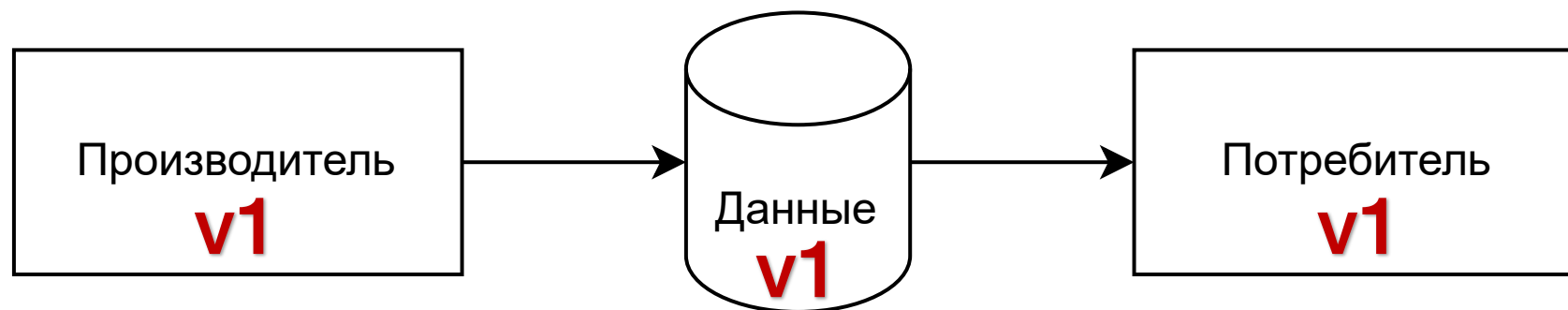




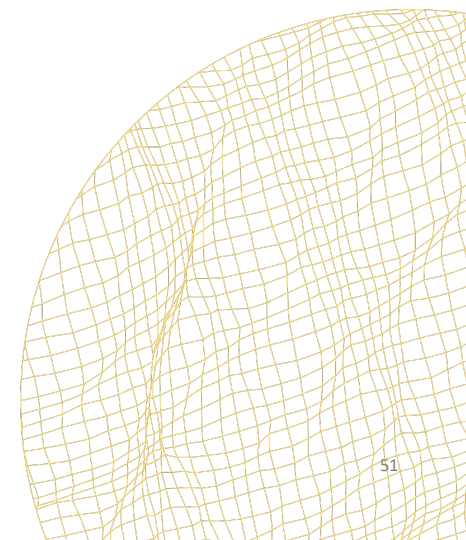
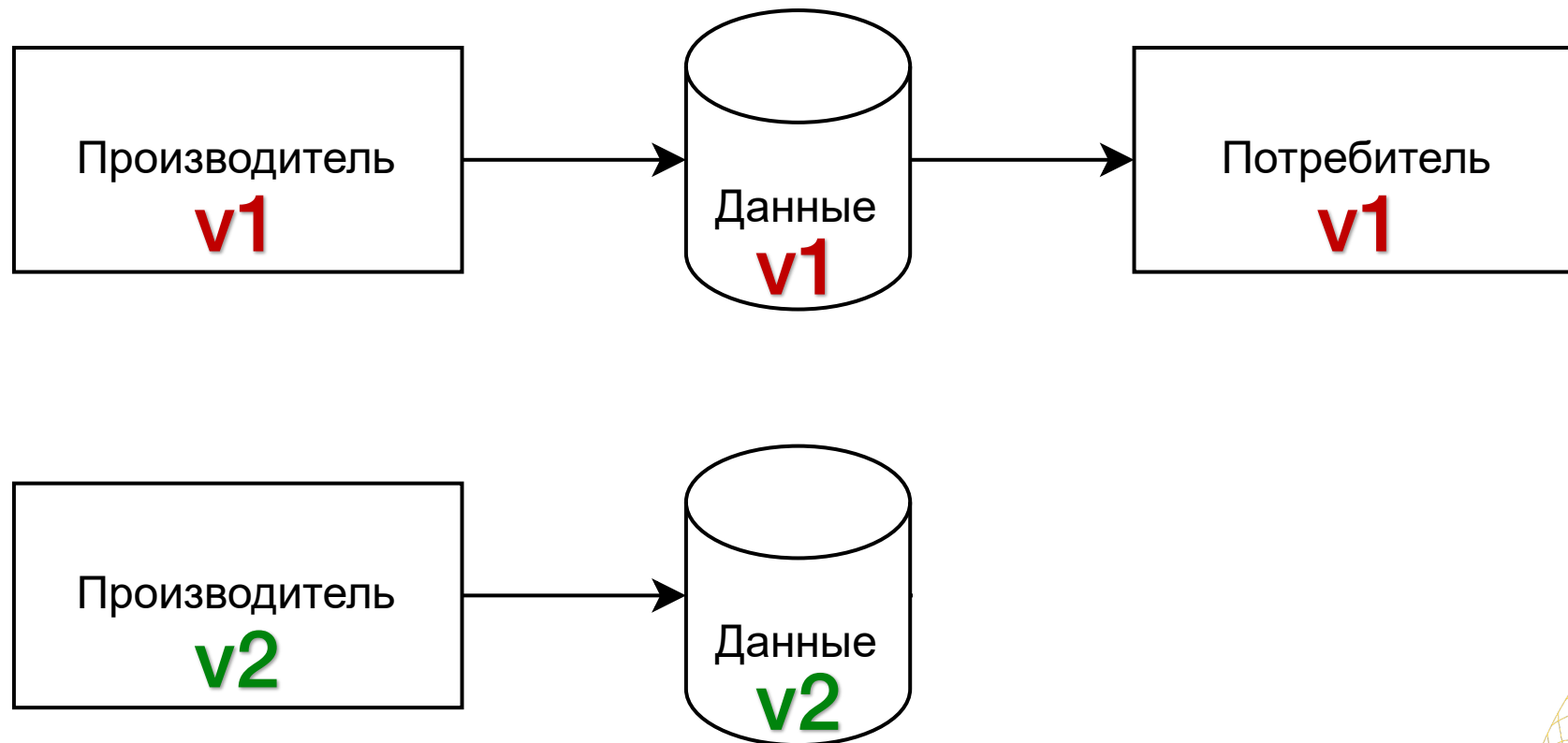
Алгоритмы миграции



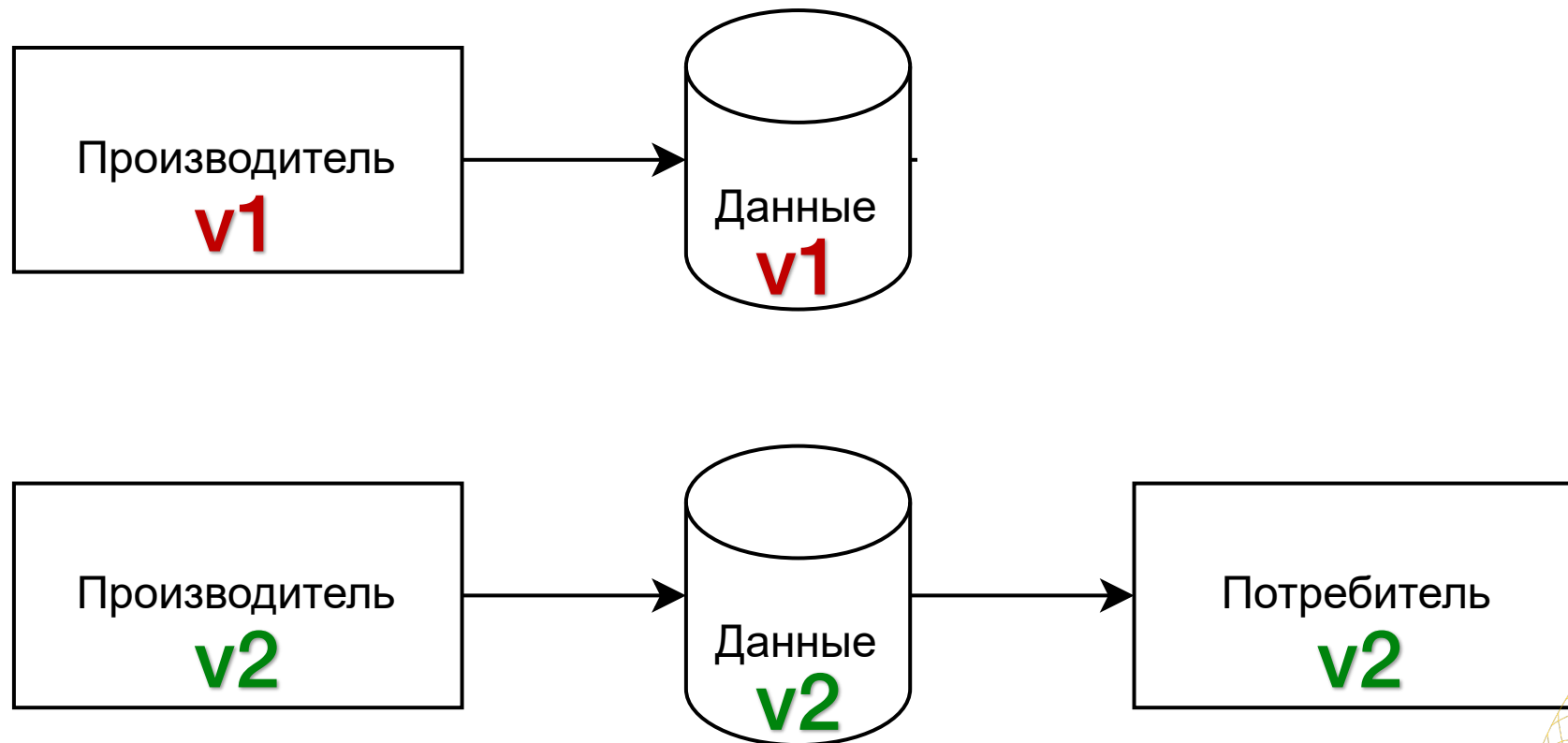
Алгоритмы миграции



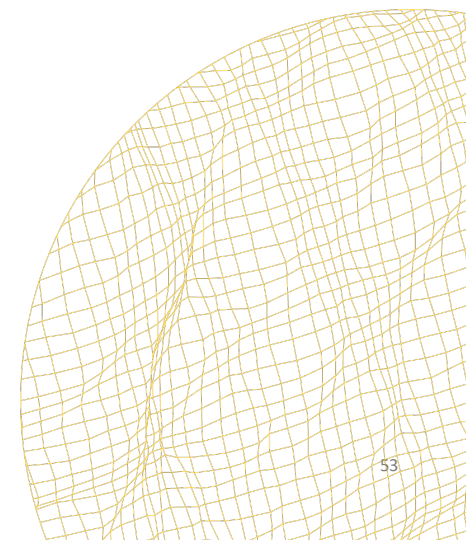
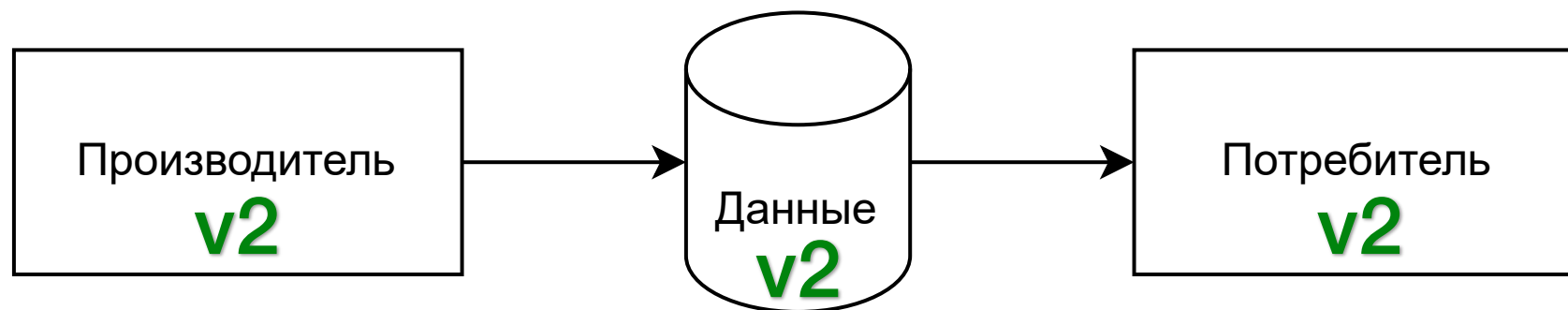
Алгоритмы миграции

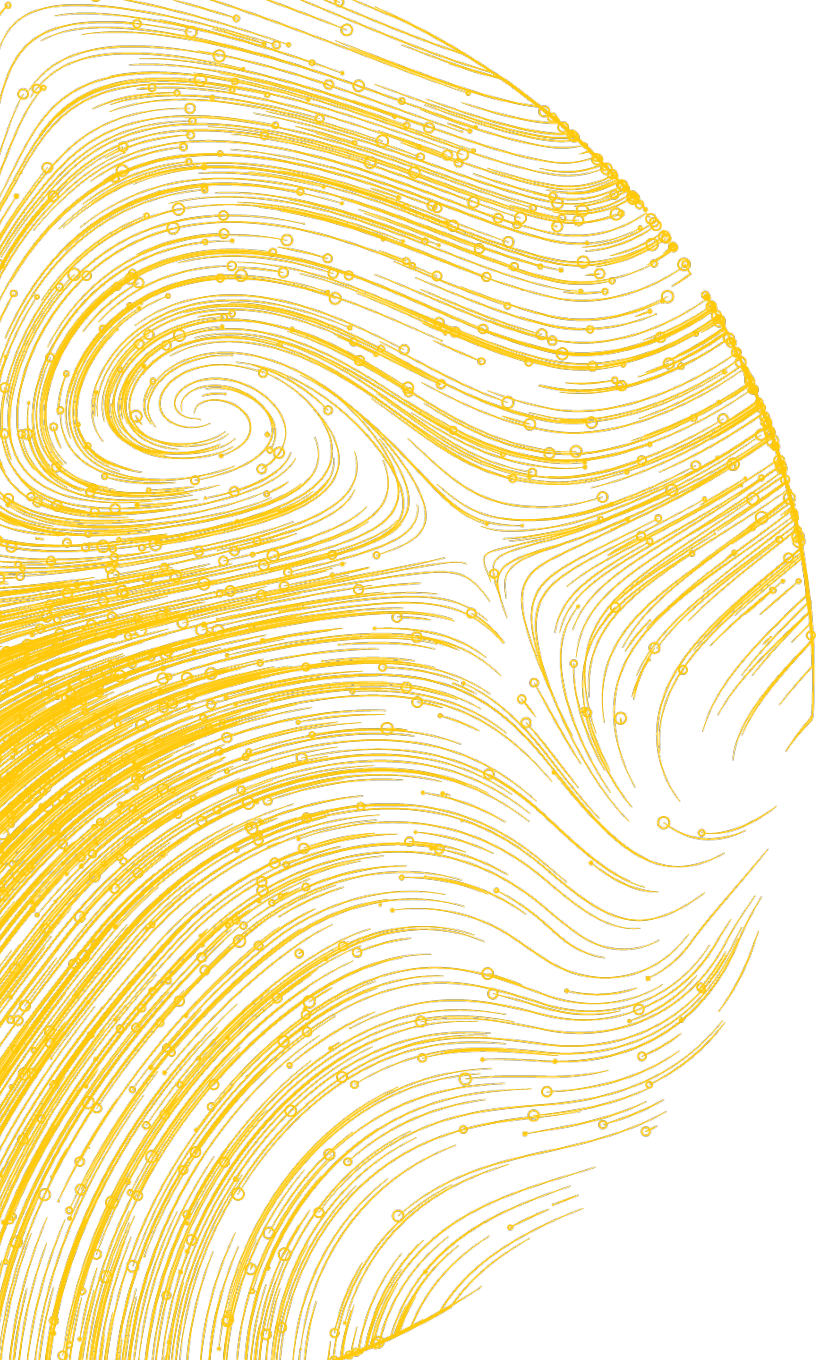


Алгоритмы миграции



Алгоритмы миграции



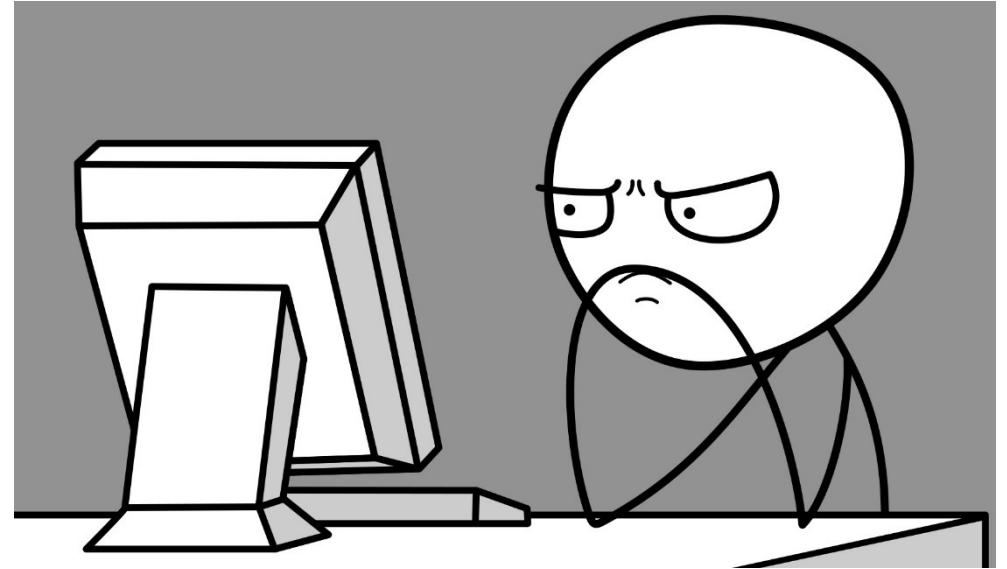


Мониторинг изменений источника. Data Quality

Что и как мониторить?

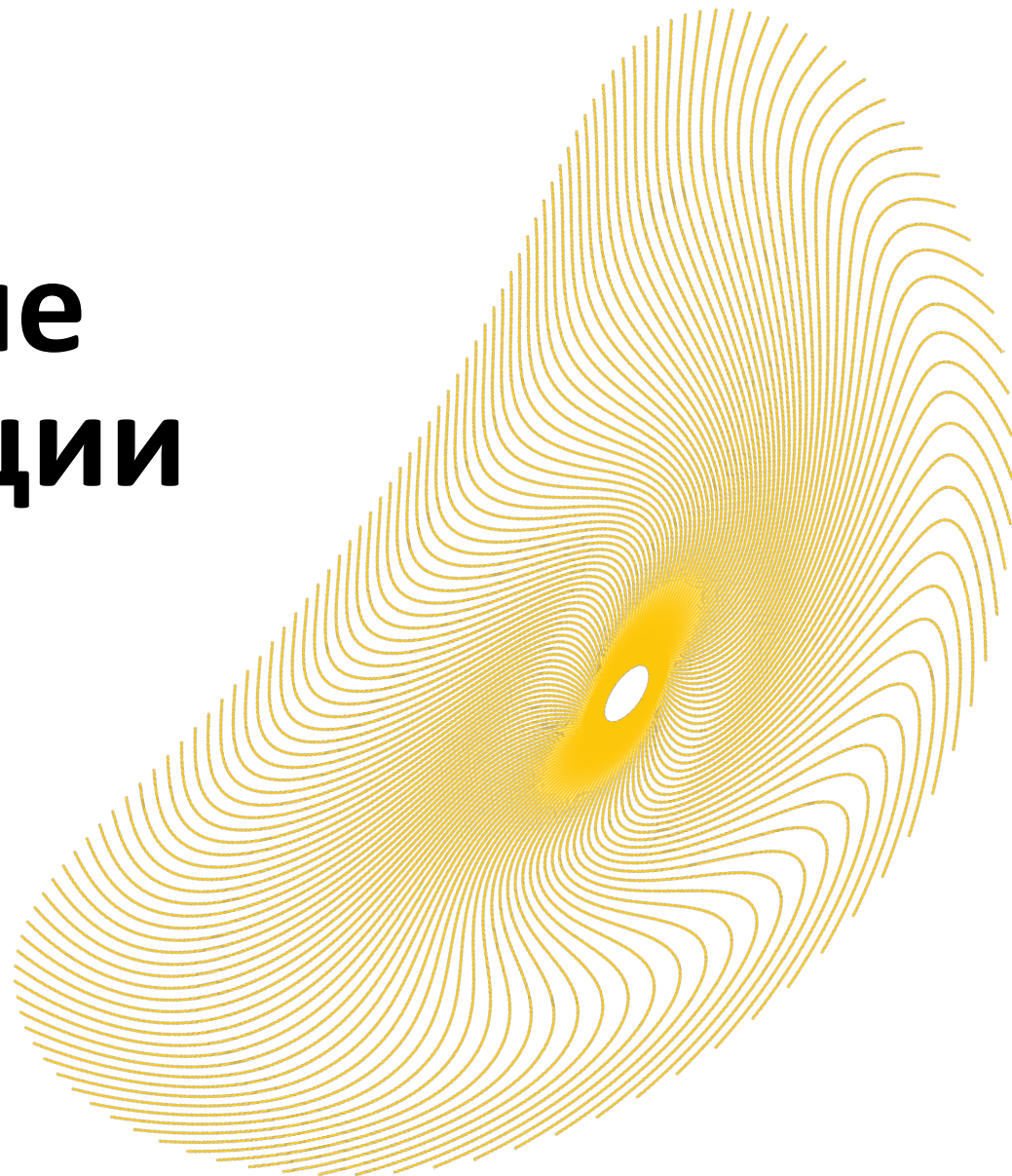


- Изменения схемы
- Новые колонки
- Данные
- Наличие данных в колонках (% null значений)
- Количество данных (+- n%)
- Бизнес-показатели по данным



Каждый триггер – повод к анализу источника данных и обсуждению

Административные подходы к эволюции данных. КУС. Интерфейсные соглашения





Development **Interface Agreement** is a document that defines interfaces and documents between customer and supplier

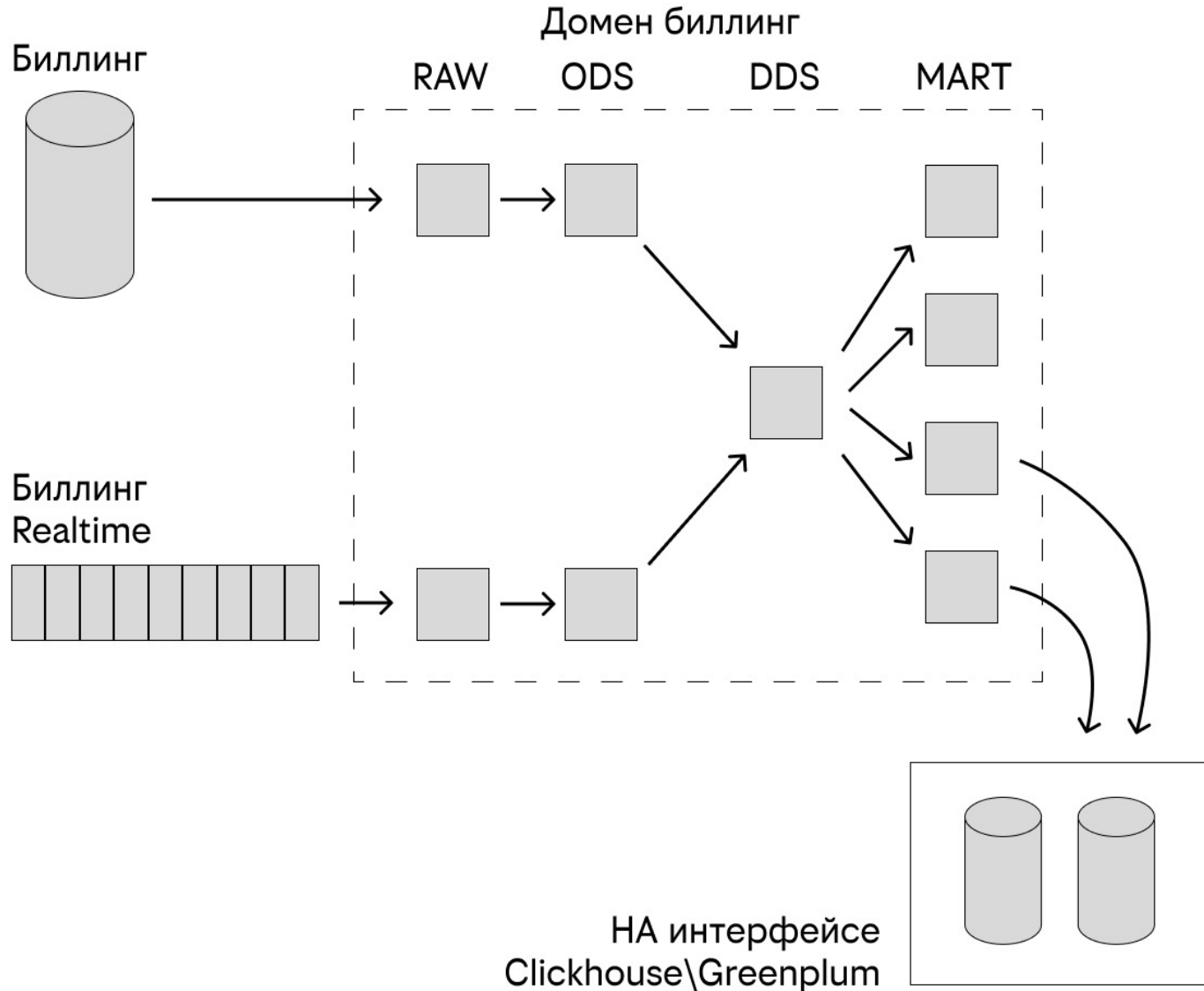
Интерфейсное соглашение – это документ, определяющий интерфейсы и договорённости между клиентом и поставщиком

<http://www.interfacemanagement.org/>

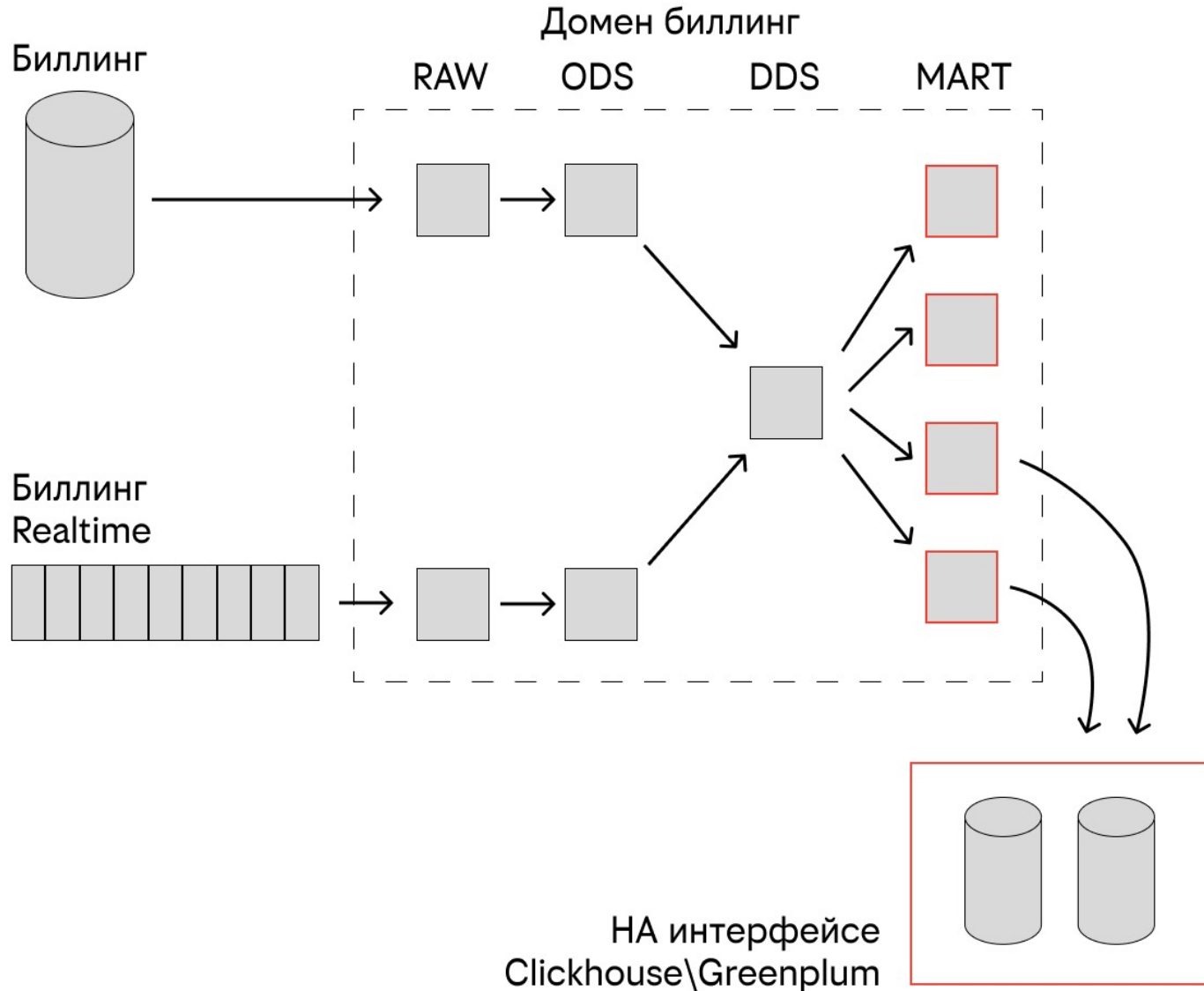
https://en.wikipedia.org/wiki/Design_by_contract



Что включить в интерфейсные соглашения?



Что включить в интерфейсные соглашения?



Про данные



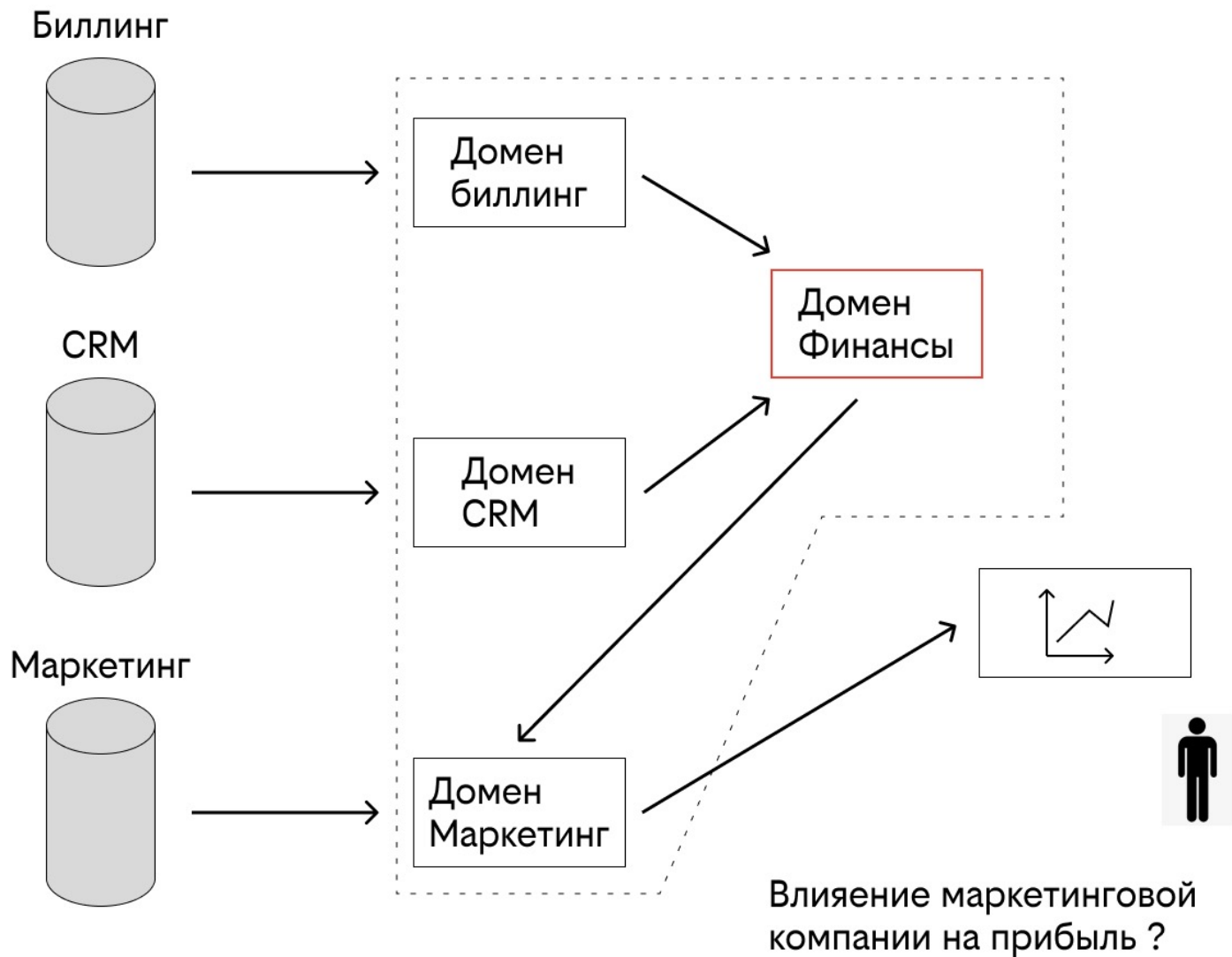
Название таблицы

Описание бизнес-смысла таблицы

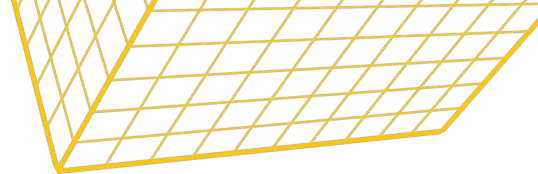
Имя колонки 1	Имя колонки 2	...	Имя колонки N
Тип колонки 1	Тип колонки 2		Тип колонки N
Описание колонки 1	Описание колонки 2		Описание колонки N



Про домен



Про SLA



Частота обновления данных

Раз в час

Раз в день

Раз в месяц

Глубина хранения

Один слепок

Год

10 лет

Задержка обновления данных

Отстаём от источника на 3 мин

Отстаём от источника на день

Время обновления данных

Раз в день до 9 утра

Раз в месяц до начала 2 рабочего дня

Семантика доставки

at-most-once

at-least-once

exactly-once

Регламент доступности

Есть период недоступности

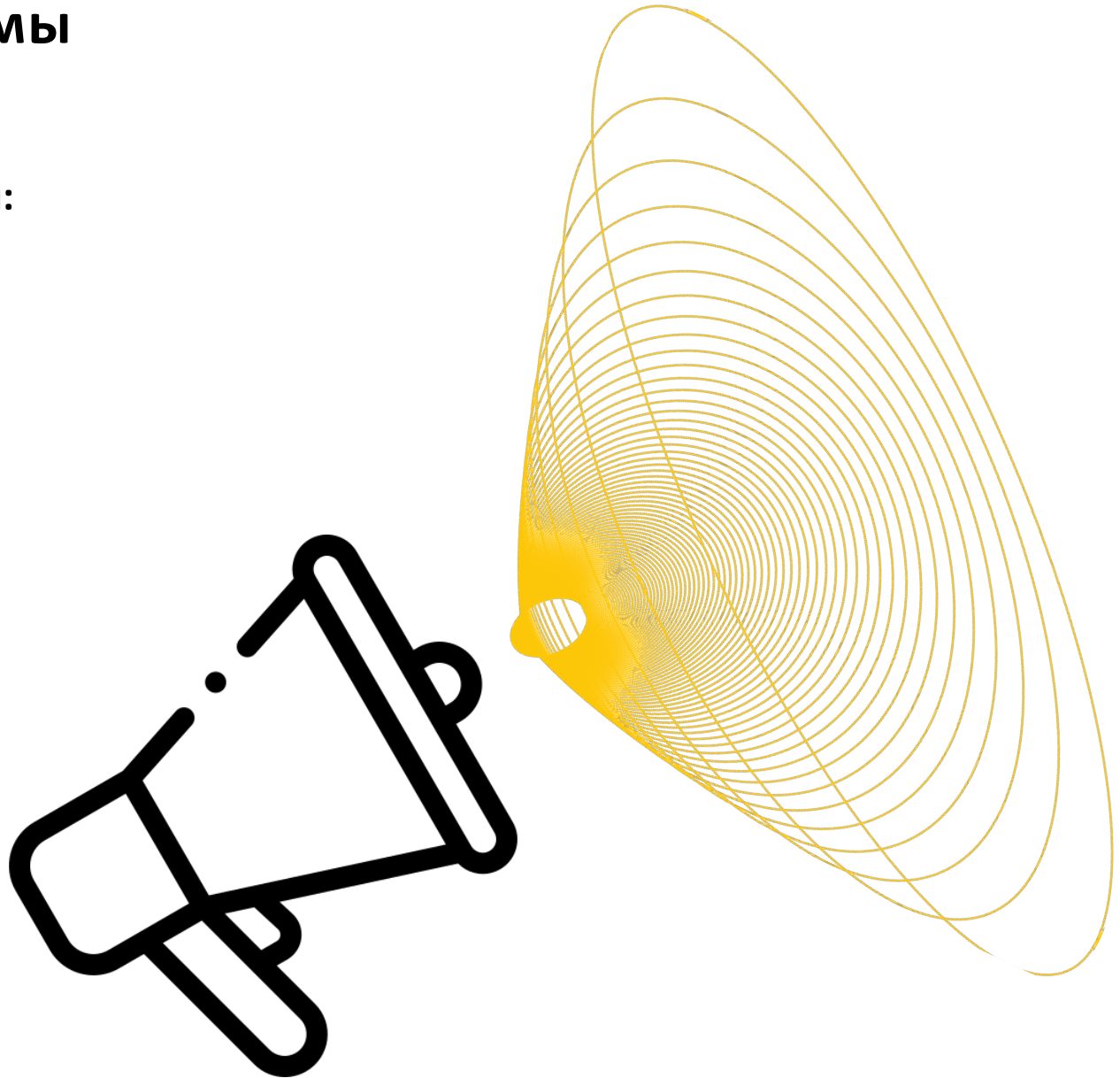
Данные доступны всегда

Про изменения и проблемы



Детально описать внесения изменений:

- Оповещение о намерениях
- Время ожидания
- Тестовые стенды
- Процесс миграции
- По каким каналам будет оповещение пользователей об инцидентах
- По каким каналам принимается информация о проблемах с данными



Про стороны

Информация о производителе данных

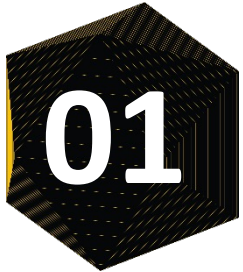
- Ответственные лица
- Возможности эскалации

Информация о потребителе данных

- Ответственные лица
- Возможности эскалации
- Цель использования данных
- Критичность сервиса



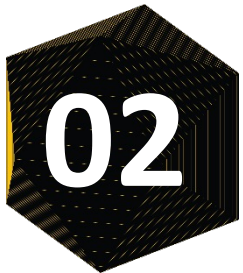
Итоги



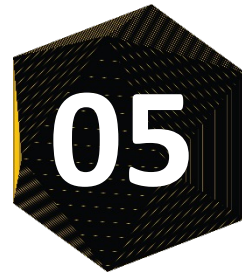
Проблема эволюции схемы данных существует



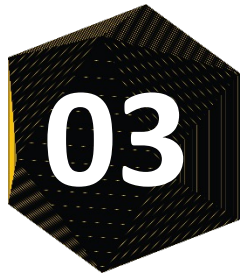
Мониторить можно не только себя, но и источник



Популярные форматы неплохо справляются с эволюцией, но есть ограничения



Интерфейсные соглашения, SLA и другие страшные слова на самом деле не страшные, а полезные



Есть инструменты для версионирования датасетов и валидации схемы данных



**Любые источники
меняются.**

**Подумайте об этом до того,
как всё сломается!**



mblagov@beeline.ru
mikhail.blagov@gmail.com
@mblagov_beeline

