### 1.звезда Sell out

**1. Что такое схема "звезда" и зачем она нужна**

Схема "звезда" состоит из:

* **Таблицы фактов (Fact table)**: хранит числовые данные (например, количество продаж, суммы) и ссылки на таблицы измерений.
* **Таблиц измерений (Dimension tables)**: хранят описательные данные (например, информацию о клиентах, датах, городах).

В твоем случае таблица Fact\_sales — это центральная таблица фактов, а Dim\_Date, Dim\_Clients, Dim\_Partners, Dim\_Products и Dim\_City — это таблицы измерений.

Цель схемы "звезда" — организовать данные так, чтобы можно было легко анализировать, например, продажи по городам, клиентам или датам.

**2. Анализ исходной таблицы**

Твоя исходная таблица содержала следующие столбцы:

* Partner\_ID, Partner\_name
* Sold\_date
* Sale\_city
* Sale\_city\_code
* Reference\_code
* Sold\_QTY
* Purchase\_price
* Client\_INN
* Client\_city
* Client\_city\_code
* Client\_name
* Client\_category

Эти данные нужно разделить на факты (числовые показатели) и измерения (описательные данные).

**3. Определение таблицы фактов (Fact\_sales)**

Таблица фактов хранит количественные данные и ключи для связи с таблицами измерений. В твоей схеме это Fact\_sales. Давай посмотрим, какие поля сюда попали:

* **transaction\_id**: это первичный ключ (PK), который, скорее всего, добавили для уникальной идентификации каждой продажи. В исходной таблице его не было, но в хранилищах данных часто добавляют такие идентификаторы.
* **Partner\_id**: взяли из столбца Partner\_ID — это внешний ключ (FK), связывающий с таблицей Dim\_Partners.
* **Reference\_id**: взяли из столбца Reference\_code — это внешний ключ, связывающий с таблицей Dim\_Products.
* **date\_id**: взяли из Sold\_date, но преобразовали. В исходной таблице Sold\_date был в формате "дата и время" (например, 08.09.21 12:49). Для схемы "звезда" дату обычно связывают с таблицей Dim\_Date, где хранятся детали о датах.
* **client\_id**: в исходной таблице его нет напрямую, но, скорее всего, его создали на основе Client\_INN, чтобы идентифицировать клиента. Это внешний ключ, связывающий с таблицей Dim\_Clients.
* **city\_id**: взяли из Client\_city\_code или Sale\_city\_code (в зависимости от контекста). Это внешний ключ, связывающий с таблицей Dim\_City.
* **Sold\_QTY**: взяли напрямую из исходной таблицы — это количественный показатель (факт).
* **Purchase\_price**: тоже взяли напрямую — это еще один количественный показатель.

Итак, Fact\_sales хранит числовые данные (Sold\_QTY, Purchase\_price) и ключи для связи с измерениями (Partner\_id, Reference\_id, date\_id, client\_id, city\_id).

**4. Создание таблиц измерений (Dimension tables)**

Теперь разберем, как создали таблицы измерений, которые окружают Fact\_sales.

**Dim\_Date (из Sold\_date)**

В исходной таблице был столбец Sold\_date (например, 08.09.21 12:49). Вместо хранения полной даты в таблице фактов, ее вынесли в отдельную таблицу Dim\_Date, чтобы можно было анализировать продажи по дням, месяцам, годам и т.д. Поля в Dim\_Date:

* **date\_id**: первичный ключ, который ссылается на дату (например, 20210908 для 08.09.21).
* **day**: день из даты (например, 8).
* **month**: месяц (например, 9).
* **year**: год (например, 2021).
* **quarter**: квартал (например, 3-й квартал для сентября).
* **date\_of\_week**: день недели (например, среда для 08.09.21).

Эти поля извлекли из Sold\_date, чтобы упростить аналитику, например, "продажи по месяцам".

**Dim\_Partners (из Partner\_ID и Partner\_name)**

Эта таблица хранит информацию о партнерах:

* **Partner\_id**: взяли из столбца Partner\_ID (например, 1000576).
* **partner\_name**: взяли из Partner\_name (например, Казарм).
* **start\_date, end\_date, is\_current**: этих полей в исходной таблице не было, но их добавили для поддержки истории изменений (например, если партнер больше не активен). Это стандартная практика в хранилищах данных.

**Dim\_Products (из Reference\_code)**

Эта таблица хранит информацию о продуктах:

* **Reference\_id**: взяли из Reference\_code (например, FERT-1233).
* **price**: взяли из Purchase\_price, но в исходной таблице цена указана в Fact\_sales. Здесь, скорее всего, хранят "базовую" цену продукта, хотя в исходной таблице это не очевидно.
* **start\_date, end\_date, is\_current**: добавили для истории изменений (например, если цена продукта менялась).

**Dim\_Clients (из Client\_INN, Client\_name, Client\_category, Client\_city, Client\_city\_code)**

Эта таблица хранит данные о клиентах:

* **client\_id**: создали на основе Client\_INN (например, 1234695212 становится client\_id).
* **client\_name**: взяли из Client\_name (например, ООО Велук).
* **inn\_client**: взяли из Client\_INN.
* **client\_category**: взяли из Client\_category (например, Ритейл).
* **client\_city\_code, client\_city**: взяли из Client\_city\_code и Client\_city (например, 99 и Москва).
* **start\_date, end\_date, is\_current**: добавили для истории изменений.

**Dim\_City (из Sale\_city, Sale\_city\_code, Client\_city, Client\_city\_code)**

Эта таблица хранит информацию о городах:

* **city\_id**: создали на основе Sale\_city\_code или Client\_city\_code (например, 66 для Екатеринбурга).
* **sale\_city\_code**: взяли из Sale\_city\_code.
* **city\_name**: взяли из Sale\_city или Client\_city (например, Екатеринбург).
* **start\_date, end\_date, is\_current**: добавили для истории изменений.

**5. Как связали таблицы**

Каждая таблица измерений связана с Fact\_sales через внешние ключи:

* Fact\_sales.Partner\_id → Dim\_Partners.Partner\_id
* Fact\_sales.Reference\_id → Dim\_Products.Reference\_id
* Fact\_sales.date\_id → Dim\_Date.date\_id
* Fact\_sales.client\_id → Dim\_Clients.client\_id
* Fact\_sales.city\_id → Dim\_City.city\_id

Эти связи позволяют, например, объединить данные о продажах с информацией о клиентах или датах.

**6. Пример, как это работает**

Допустим, ты хочешь узнать, сколько было продано в Москве в сентябре 2021 года. Схема "звезда" позволяет:

1. Найти date\_id для сентября 2021 в Dim\_Date.
2. Найти city\_id для Москвы в Dim\_City.
3. В Fact\_sales выбрать все записи, где date\_id и city\_id совпадают, и суммировать Sold\_QTY.

**Итог**

Схему "звезда" построили так:

1. Выделили числовые данные (Sold\_QTY, Purchase\_price) в таблицу Fact\_sales.
2. Описательные данные (даты, клиенты, партнеры, города, продукты) вынесли в таблицы измерений.
3. Создали связи через ключи (Partner\_id, date\_id и т.д.).
4. Добавили поля вроде start\_date, end\_date для поддержки истории изменений.

2DWH table

Давай разберем, как из таблиц Order\_Header, Order\_Lines и Customers, которые ты показал в первом изображении, построили схему, представленную во втором изображении. Схема — это модель данных, где таблицы разделены на измерения (dim\_) и факты (Fact\_), что типично для хранилищ данных в формате "звезда" (star schema). Я объясню, как это сделали, шаг за шагом.

**Шаг 1: Определение измерений (Dimensions)**

Измерения — это справочные таблицы, которые описывают "кто", "где", "когда" и "что". Они содержат описательные данные, которые редко меняются. Давай посмотрим, как из исходных таблиц выделили измерения.

* **dim\_customer (Клиенты)**  
  Таблица Customers из исходных данных содержит информацию о клиентах: Customer\_Id, Customer\_Name, Email, Customer\_Category. Эти данные идеально подходят для измерения, так как описывают клиентов.  
  В схеме добавили поля start\_date, end\_date и is\_current, которые обычно используются для отслеживания изменений данных во времени (например, если клиент сменил email). Это стандартная практика в хранилищах данных для поддержки истории изменений, хотя в исходной таблице таких полей не было.
* **dim\_product (Продукты)**  
  В таблице Order\_Lines есть поле Product\_Id, но нет описания продукта. В схеме появилась таблица dim\_product с полями product\_id, product\_desc (описание), price, start\_date, end\_date, is\_current.  
  Поле Price из Order\_Lines перенесли сюда, предположив, что цена продукта фиксирована и не зависит от заказа. Это упрощает анализ, так как цена теперь связана с продуктом, а не с каждой строкой заказа. Поля start\_date, end\_date, is\_current добавлены для истории изменений (например, если цена продукта меняется со временем).
* **dim\_city (Города)**  
  В Order\_Lines есть поля Sales\_City и Dest\_City (город продажи и город назначения). Эти города выделили в отдельную таблицу dim\_city с полями city\_id, city\_name, start\_date, end\_date, is\_current.  
  Каждому городу присвоили уникальный city\_id, а названия городов (Sales\_City и Dest\_City) перенесли в поле city\_name. Теперь вместо текстовых названий в основной таблице будут использоваться идентификаторы городов (city\_id), что уменьшает дублирование данных.
* **dim\_calendar (Календарь)**  
  В Order\_Header есть поле Order\_Date. Из него создали таблицу dim\_calendar с полями order\_date\_id, date, day\_of\_week, month, quarter, year.  
  Эта таблица позволяет анализировать заказы по дням, месяцам, кварталам и годам. Например, дата 01.01.2019 из Order\_Header превращается в запись в dim\_calendar, где указаны день недели, месяц (январь), квартал (1-й) и год (2019). Это удобно для аналитики, например, чтобы понять, в какие месяцы больше заказов.
* **dim\_order\_header (Заголовки заказов)**  
  Таблица Order\_Header превратилась в dim\_order\_header с полями header\_id, line\_id, data\_created.  
  Здесь header\_id — это тот же Header\_Id из исходной таблицы, а line\_id добавлен, чтобы связать заголовок с конкретной строкой заказа. Поле data\_created соответствует Order\_Date, но переименовано для ясности. Эта таблица стала измерением, так как заголовки заказов — это справочные данные.

**Шаг 2: Создание таблицы фактов (Fact\_order)**

Факты — это данные, которые отражают события (например, продажи, заказы) и содержат числовые показатели (метрики). Таблица Fact\_order — это основа схемы, где собраны все ключевые данные о заказах.

* **Поля таблицы Fact\_order:**
  + order\_id: Уникальный идентификатор для каждой строки заказа (взято из Line\_Id в Order\_Lines).
  + product\_id: Ссылка на продукт из dim\_product (взято из Product\_Id в Order\_Lines).
  + customer\_id: Ссылка на клиента из dim\_customer (взято из Customer\_Id в Order\_Lines).
  + sales\_city\_id: Ссылка на город продажи из dim\_city (взято из Sales\_City).
  + dest\_city\_id: Ссылка на город назначения из dim\_city (взято из Dest\_City).
  + header\_id: Ссылка на заголовок заказа из dim\_order\_header (взято из Header\_Id).
  + order\_date\_id: Ссылка на дату из dim\_calendar (взято из Order\_Date в Order\_Header).
  + price: Цена (перенесена в dim\_product, но может быть продублирована здесь для удобства).
  + order\_num: Номер заказа из Order\_Header.
  + data\_created: Дата создания заказа (из Order\_Date).
  + qty: Количество из Order\_Lines.
* **Как это сделали:**  
  Таблица Order\_Lines стала основой для Fact\_order, так как она содержит данные о конкретных строках заказов (событиях). Каждую запись из Order\_Lines связали с соответствующими измерениями:
  + Header\_Id связывает с dim\_order\_header.
  + Customer\_Id связывает с dim\_customer.
  + Product\_Id связывает с dim\_product.
  + Sales\_City и Dest\_City заменили на sales\_city\_id и dest\_city\_id из dim\_city.
  + Order\_Date из Order\_Header заменили на order\_date\_id из dim\_calendar.  
    Поля qty и order\_num оставили как есть, так как это метрики, которые нужны для анализа (например, сколько единиц товара продано).

**Шаг 3: Построение связей (Foreign Keys, FK)**

На схеме видно, что Fact\_order связана с измерениями через ключи:

* product\_id ссылается на dim\_product.
* customer\_id ссылается на dim\_customer.
* sales\_city\_id и dest\_city\_id ссылаются на dim\_city.
* header\_id ссылается на dim\_order\_header.
* order\_date\_id ссылается на dim\_calendar.  
  Эти связи позволяют объединять данные из измерений с фактами для анализа. Например, если нужно узнать, сколько товаров продал конкретный клиент в определенном городе, мы берем customer\_id, sales\_city\_id и qty из Fact\_order и присоединяем данные из dim\_customer и dim\_city.

**Почему выбрали схему "звезда"?**

Схема "звезда" удобна для аналитики:

* В центре — таблица фактов (Fact\_order), которая хранит события (заказы) и числовые данные (количество, цена).
* Вокруг — измерения (dim\_), которые описывают события (кто, что, где, когда).  
  Такой подход упрощает запросы: меньше соединений (joins) и быстрее анализ. Например, чтобы посчитать продажи по месяцам, достаточно соединить Fact\_order с dim\_calendar и сгруппировать по month.

**Пример: Как данные из таблиц попали в схему**

Возьмем строку из Order\_Lines:  
Header\_Id = 1, Line\_Id = 1, Product\_Id = 456, Sales\_City = Таганрог, Dest\_City = Ростов, Qty = 4, Price = 100, Customer\_Id = 1.

* В Fact\_order это станет:
  + order\_id = 1 (из Line\_Id).
  + product\_id = 456 (ссылка на dim\_product).
  + customer\_id = 1 (ссылка на dim\_customer).
  + sales\_city\_id и dest\_city\_id — идентификаторы для Таганрога и Ростова из dim\_city.
  + header\_id = 1 (ссылка на dim\_order\_header).
  + order\_date\_id — идентификатор для 01.01.2019 из dim\_calendar.
  + qty = 4, price = 100, order\_num = 10 (из Order\_Header).
* В dim\_customer: customer\_id = 1, customer\_name = Иван Иванов, email = ivanov@email.com.
* В dim\_product: product\_id = 456, product\_desc = Резинка, price = 100.
* В dim\_calendar: order\_date\_id для 01.01.2019 с month = январь, year = 2019.

**Итог**

Схему построили, разделив данные на измерения и факты:

* Измерения (dim\_) создали для клиентов, продуктов, городов, дат и заголовков заказов.
* Факты (Fact\_order) собрали из строк заказов, связав их с измерениями через идентификаторы.
* Добавили поля вроде start\_date, end\_date, is\_current для поддержки истории изменений.  
  Это позволяет легко анализировать данные, например, сколько товаров продано в Ростове в январе 2019 года.

Если нужно что-то уточнить или посчитать на основе этой схемы, дай знать!

Хорошо, давай разберем все еще подробнее, чтобы все стало на свои места. Я объясню, почему выбрали именно такую схему, откуда брали данные и как их преобразовали из твоих исходных таблиц (Order\_Header, Order\_Lines, Customers) в модель "звезда" с таблицами Fact\_order и измерениями (dim\_customer, dim\_product, dim\_city, dim\_calendar, dim\_order\_header). Пойдем шаг за шагом, с примерами и логикой.

### Почему выбрали схему "звезда"?

Схема "звезда" — это популярный подход в хранилищах данных, где есть одна центральная таблица фактов (здесь Fact\_order), которая содержит события (заказы), и несколько таблиц измерений (dim\_), которые описывают эти события. Почему это сделали?

* **Эффективность анализа**: В исходных таблицах данные смешаны (например, Order\_Lines содержит и продукты, и города, и клиентов). Чтобы быстро считать, например, общую выручку по городам или клиентам, нужно много соединений (joins) между таблицами. Схема "звезда" упрощает это, вынося описательные данные в отдельные таблицы и связывая их через идентификаторы.
* **Уменьшение дублирования**: В Order\_Lines город Ростов повторяется несколько раз. В dim\_city он будет один раз с уникальным city\_id, что экономит место и упрощает обновления (например, если название города изменится).
* **Поддержка аналитики**: Добавление полей вроде day\_of\_week или quarter в dim\_calendar позволяет анализировать тренды (например, больше ли заказов по пятницам).
* **История изменений**: Поля start\_date, end\_date, is\_current добавлены, чтобы отслеживать, как данные менялись со временем (например, если клиент сменил категорию или цена продукта выросла).

Теперь давай разберем, откуда что взяли и как преобразовали.

### 1. Измерение dim\_customer (Клиенты)

**Откуда взяли:**

* Исходная таблица Customers содержит:
  + Customer\_Id (например, 1)
  + Customer\_Name (например, Иван Иванов)
  + Email (например, [ivanov@email.com](mailto:ivanov@email.com))
  + Customer\_Category (например, НК Сотрудник)

**Почему так сделали:**

* Эти данные описывают клиентов — это классическое измерение. Они не меняются часто, и их удобно использовать для фильтрации (например, показать заказы только для "НК Сотрудник").
* Добавили start\_date, end\_date, is\_current, чтобы учитывать историю. Например, если клиент сменил email или категорию, можно сохранить старую запись с end\_date и добавить новую с start\_date.

**Пример преобразования:**

* Строка Customer\_Id = 1, Customer\_Name = Иван Иванов, Email = ivanov@email.com, Customer\_Category = НК Сотрудник стала:
  + customer\_id = 1
  + customer\_name = Иван Иванов
  + email = ivanov@email.com
  + customer\_category = НК Сотрудник
  + start\_date = (например, 01.01.2019)
  + end\_date = NULL (если актуально)
  + is\_current = 1 (если актуально)

### 2. Измерение dim\_product (Продукты)

**Откуда взяли:**

* В Order\_Lines есть поле Product\_Id (например, 456), но нет описания продукта или полной информации о цене. Поле Price (например, 100) тоже из Order\_Lines.
* В реальной базе данных обычно есть отдельная таблица продуктов, но здесь ее нет, поэтому предположили, что данные о продукте (описание, цена) можно вынести.

**Почему так сделали:**

* Цена в Order\_Lines зависит от Product\_Id, но дублируется для каждой строки заказа. Лучше хранить цену в dim\_product, чтобы она была фиксированной для каждого продукта, а не менялась в каждом заказе.
* Добавили product\_desc (например, "Резинка") как вымышленное поле, так как в исходных данных его нет — это стандартная практика для измерений.
* Поля start\_date, end\_date, is\_current добавлены на случай, если цена или описание продукта меняются (например, с 01.01.2019 цена 100, а с 01.06.2019 — 120).

**Пример преобразования:**

* Для Product\_Id = 456, Price = 100 из Order\_Lines:
  + product\_id = 456
  + product\_desc = Резинка (добавлено предположительно)
  + price = 100
  + start\_date = 01.01.2019
  + end\_date = NULL
  + is\_current = 1

### 3. Измерение dim\_city (Города)

**Откуда взяли:**

* В Order\_Lines есть Sales\_City (например, Таганрог) и Dest\_City (например, Ростов). Эти названия городов повторяются.

**Почему так сделали:**

* Города — это справочные данные. Вместо хранения текстовых названий в каждой строке Order\_Lines, создали таблицу dim\_city с уникальным city\_id для каждого города (например, Таганрог = 1, Ростов = 2).
* Добавили start\_date, end\_date, is\_current на случай, если названия городов или их статус меняются (например, если город переименуют).

**Пример преобразования:**

* Из Sales\_City = Таганрог и Dest\_City = Ростов:
  + city\_id = 1, city\_name = Таганрог, start\_date = 01.01.2019, end\_date = NULL, is\_current = 1
  + city\_id = 2, city\_name = Ростов, start\_date = 01.01.2019, end\_date = NULL, is\_current = 1

### 4. Измерение dim\_calendar (Календарь)

**Откуда взяли:**

* В Order\_Header есть поле Order\_Date (например, 01.01.2019).

**Почему так сделали:**

* Дата — важный аспект для анализа (по месяцам, годам, дням недели). Вместо хранения одной даты в Order\_Header, создали таблицу dim\_calendar, где каждая дата разбита на детали: day\_of\_week (например, пятница), month (январь), quarter (1), year (2019).
* Это позволяет легко считать, например, сколько заказов было в первом квартале.

**Пример преобразования:**

* Для Order\_Date = 01.01.2019:
  + order\_date\_id = 1
  + date = 01.01.2019
  + day\_of\_week = пятница
  + month = январь
  + quarter = 1
  + year = 2019

### 5. Измерение dim\_order\_header (Заголовки заказов)

**Откуда взяли:**

* Таблица Order\_Header содержит Header\_Id, Order\_Date, Order\_Num.

**Почему так сделали:**

* Заголовки заказов — это справочные данные, которые связывают строки заказа (Order\_Lines) с общей информацией. line\_id добавили, чтобы каждая строка заказа имела уникальный идентификатор в рамках заголовка.
* data\_created — это переименованное Order\_Date для ясности.
* Это измерение нужно, чтобы хранить метаданные заказа (номер, дата) отдельно от деталей.

**Пример преобразования:**

* Для Header\_Id = 1, Order\_Date = 01.01.2019, Order\_Num = 10:
  + header\_id = 1
  + line\_id = 1 (для первой строки)
  + data\_created = 01.01.2019

### 6. Таблица фактов Fact\_order (Факты о заказах)

**Откуда взяли:**

* Основу взяли из Order\_Lines, добавив данные из Order\_Header и связав с измерениями.
  + Line\_Id → order\_id
  + Product\_Id → product\_id
  + Customer\_Id → customer\_id
  + Sales\_City → sales\_city\_id (из dim\_city)
  + Dest\_City → dest\_city\_id (из dim\_city)
  + Header\_Id → header\_id (из dim\_order\_header)
  + Order\_Date → order\_date\_id (из dim\_calendar)
  + Qty → qty
  + Price → price (хотя его перенесли в dim\_product)
  + Order\_Num → order\_num (из Order\_Header)

**Почему так сделали:**

* Order\_Lines содержит строки заказов — это события, которые нужно анализировать (сколько продано, за какую цену).
* Вместо текстовых полей (Sales\_City, Order\_Date) ввели идентификаторы (sales\_city\_id, order\_date\_id), чтобы связать с измерениями.
* qty и price — это метрики, которые можно суммировать или усреднять (например, общая выручка = qty \* price).

**Пример преобразования:**

* Строка из Order\_Lines: Header\_Id = 1, Line\_Id = 1, Product\_Id = 456, Sales\_City = Таганрог, Dest\_City = Ростов, Qty = 4, Price = 100, Customer\_Id = 1:
  + order\_id = 1
  + product\_id = 456
  + customer\_id = 1
  + sales\_city\_id = 1 (Таганрог)
  + dest\_city\_id = 2 (Ростов)
  + header\_id = 1
  + order\_date\_id = 1 (01.01.2019)
  + price = 100
  + order\_num = 10
  + data\_created = 01.01.2019
  + qty = 4

### Логика связей

* Fact\_order связана с измерениями через внешние ключи (FK):
  + product\_id → dim\_product
  + customer\_id → dim\_customer
  + sales\_city\_id и dest\_city\_id → dim\_city
  + header\_id → dim\_order\_header
  + order\_date\_id → dim\_calendar
* Это позволяет, например, запросить: "Сколько товаров продал Иван Иванов в Ростове в январе 2019?" — соединив Fact\_order с dim\_customer, dim\_city и dim\_calendar.

### Почему добавили дополнительные поля?

* **start\_date, end\_date, is\_current**: Чтобы хранить историю изменений. Например, если клиент сменил категорию, старая запись получает end\_date, а новая — start\_date.
* **day\_of\_week, month, quarter, year**: Для удобства анализа по времени, что невозможно сделать напрямую из Order\_Date.

### Проверка на примере

Возьмем заказ с Header\_Id = 1:

* Order\_Header: Header\_Id = 1, Order\_Date = 01.01.2019, Order\_Num = 10
* Order\_Lines: Header\_Id = 1, Line\_Id = 1, Product\_Id = 456, Sales\_City = Таганрог, Dest\_City = Ростов, Qty = 4, Price = 100, Customer\_Id = 1
* В схеме:
  + dim\_order\_header: header\_id = 1, line\_id = 1, data\_created = 01.01.2019
  + dim\_customer: customer\_id = 1, customer\_name = Иван Иванов
  + dim\_product: product\_id = 456, product\_desc = Резинка, price = 100
  + dim\_city: city\_id = 1 (Таганрог), city\_id = 2 (Ростов)
  + dim\_calendar: order\_date\_id = 1, date = 01.01.2019, month = январь
  + Fact\_order: order\_id = 1, product\_id = 456, customer\_id = 1, sales\_city\_id = 1, dest\_city\_id = 2, header\_id = 1, order\_date\_id = 1, qty = 4