

|  |
| --- |
| Анализ продаж  MJX |
| Магазин радиоуправляемых моделей |

| Related Artifacts | |
| --- | --- |
| Ref. | Name |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Abbreviations and Acronyms | |
|  |  |
|  |  |

Contents

[1.1 Business background 4](#_Toc498569157)

[1.2 Problems because of poor data management 4](#_Toc498569158)

[1.3 Benefits from implementing a Data Warehouse 4](#_Toc498569159)

[2.1 Business Process 5](#_Toc498569160)

[2.2 Grain 5](#_Toc498569161)

[2.3 dimensions 5](#_Toc498569162)

[2.4 facts 5](#_Toc498569163)

[2.5 data sources 5](#_Toc498569164)

[3.1 access layer – snowflake 6](#_Toc498569165)

[3.2 fundamential layer - 3 NF model 6](#_Toc498569166)

1. Business Description

## Business background

Хранилище данных будет разрабатываться для магазина радиоуправляемых моделей MJX.

MJX – это магазин качественных и интересных игрушек и моделей ведущих мировых брендов от прямого импортёра в Республику Беларусь ООО «Оушнтрейд». Это относительно молодой (около 4 лет), но быстро развивающийся бизнес по продаже радиоуправляемых моделей, таких как танковые бои, вертолеты, квадрокоптеры с камерой и т.д. Как показывает практика, для большинства людей различного возраста подобные модели интересны, так как могут использоваться в качестве игрушек или становиться объектами хобби, таких как коллекционирование.

## Problems because of poor data management

Первоначальной целью для MJX являлось наращивание прибыли в рекордные сроки и приобретение клиентской базы. Теперь, когда проект стабилен, количество продаж и новых клиентов растут, необходимо зафиксировать данное положение и обеспечить и дальнейший рост вышеперечисленных показателей. Чтобы добиться этого, потребуется улучшить хранение и обработку данных, проведение анализа клиентской базы и продаваемых товаров.

На данный момент используются малоэффективные недорогостоящие решения, приобретенные на старте работы данного проекта, которые неспособны обеспечить быструю обработку данных, что приводит к незавершенности, неточности, несвоевременности работы управляющего персонала.

## Benefits from implementing a Data Warehouse

Построение DWH обеспечит специализированным сотрудникам:

- быстрый доступ к детализированной информации о продуктах

- возможность поддержки данных в актуальном состоянии без снижения производительности

- построение отчетов различной сложности для анализа состояния рынка, продаж и клиентской базы

- централизованное хранение информации

# Dimensions of a Business

## Business Process

Ежедневные продажи радиоуправляемых моделей от различных поставщиков в магазинах MJX.

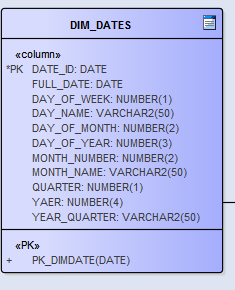
## Grain

Зерном является одна продажа товара от конкретного поставщика в определенный день определенному покупателю в одном из магазинов.

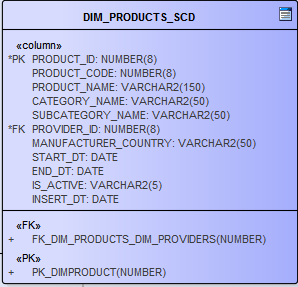
## dimensions

В ходе анализа бизнес-процесса и определения целей построения хранилища данных были определены 8 измерений:

Dim\_Dates – временное измерение, играющее ключевую роль, так как продажи рассматриваются в разрезе реального времени. Минимальная единица измерения – день.

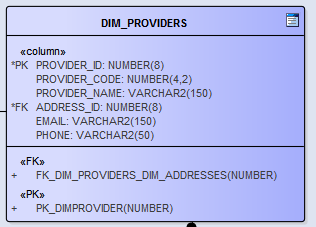


Dim\_Products\_SCD2 – иерархическое измерение, содержащее информацию о товаре: наименование, код продукта, категорию, подкатегорию, поставщика, страну-производителя.

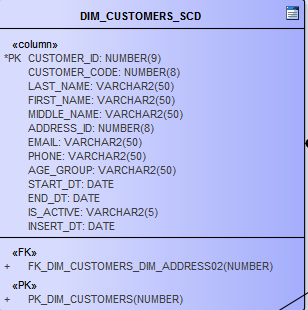


Dim\_Providers – измерение, содержащее информацию о поставщике конкретного товара: код, наименование, адрес, контактную информацию.

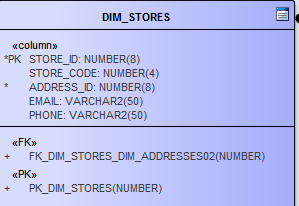
В данном бизнес-процессе играет ключевую роль, при этом в случае возникновения каких-либо проблем всегда можно отказаться от услуг поставщика, поэтому нам и нужна детальная информация о поставщиках.



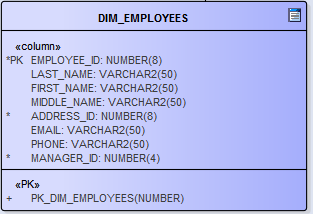
Dim\_Customers\_SCD – измерение, содержащее в себе информацию о покупателях: код, ФИО, адрес, контактную информацию, возрастную группу.



Dim\_Stores – измерение, содержащее информацию о магазинах: код, адрес, контактную информацию.

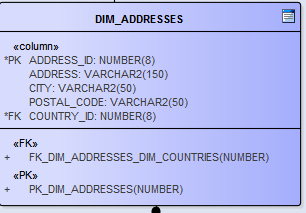


Dim\_Employees – сотрудники магазинов, с ролями обычный сотрудник и менеджер, и их контактная информация.

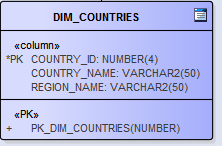


Группа географических измерений: адреса и страны. Данные из этого измерения будут использоваться измерениями Dim\_Customers, Dim\_Employees, Dim\_Stores, Dim\_Providers.

Dim\_Addresses – включает в себя адрес, город, код города и ссылку на измерение страны.

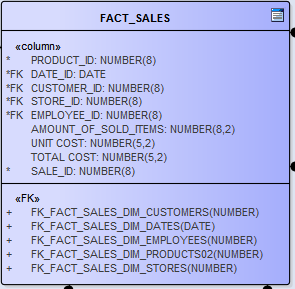


Dim\_Countries – страна + регион.



## facts

Таблица фактов будет содержать количество проданных единиц товара, цену одного товара, оплаченную сумму за количество товаров в разрезе дня, категории и подкатегории товара, возрастной группы покупателя. Содержит дегенеративное измерение – SALE\_ID.



## data sources

Данные о радиоуправляемых моделях, а именно категории, подкатегории, цены, магазины на территории РБ, способы оплаты и доставки выложены на сайте <https://mjx.by/>

Такая информация как покупатели, сотрудники будут генерироваться с помощью онлайн-инструмента <http://www.fakenamegenerator.com/advanced.php>

Данные о товарах, категориях, подкатегориях, страны-производители и поставщики выгружены из базы 1С.

Количество различных поставщиков и стран-производителей маленькое, поэтому эта информация будет дополнена.

Временные данные так же будут сгенерированы. Это ежедневные продажи в различных точках в период с 2013 (2013 год – год основания проекта).

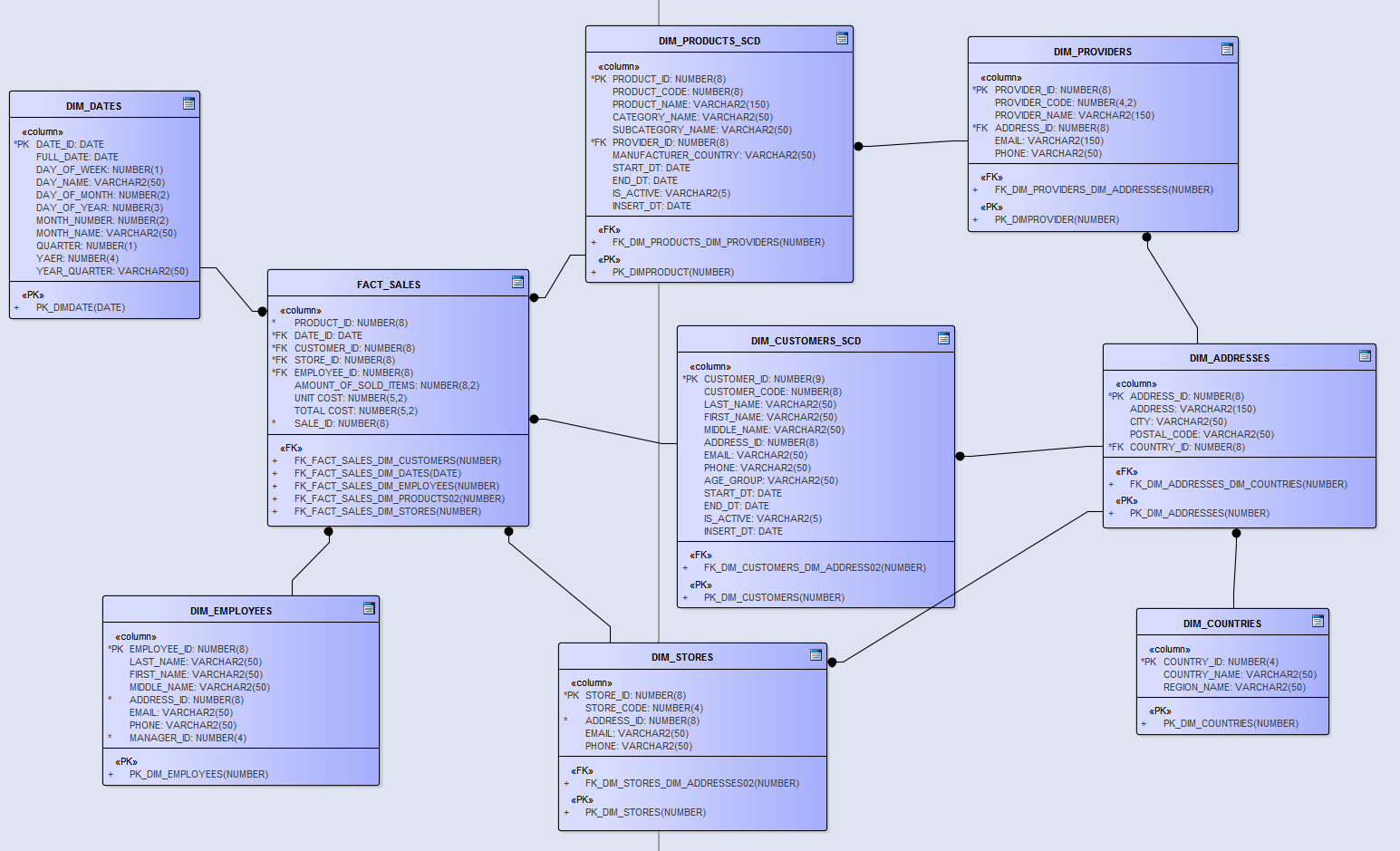
Целями анализа данного бизнеса является скорее не проанализировать рост прибыли, а проанализировать клиентскую базу - сами покупатели с их возрастной категорией и товары, которые чаще всего покупают.

# Logical Scheme

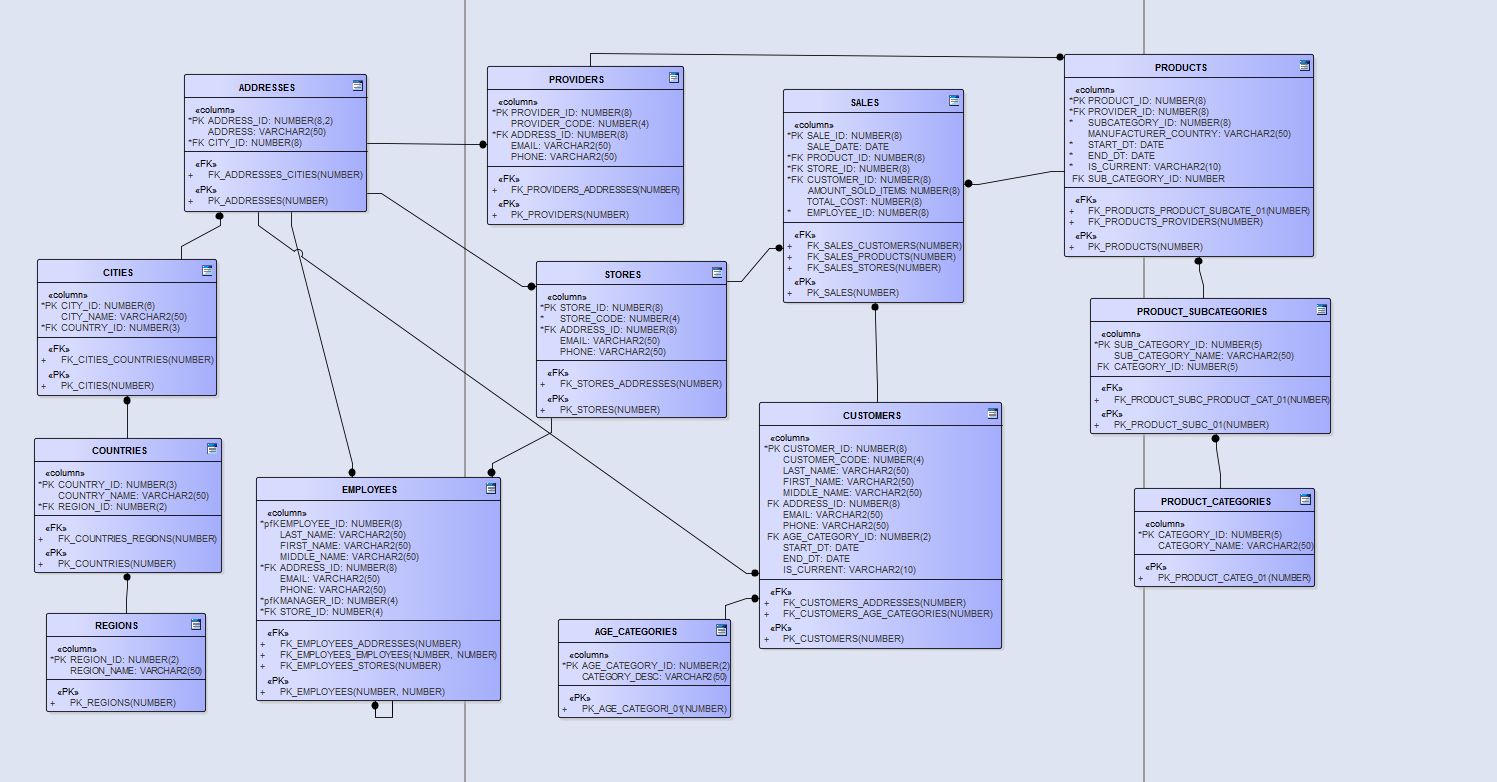
## access layer – snowflake

Все связи на Access layer между измерениями и таблицей фактов реализованы как один ко многим.

Для данного хранилища была выбрана схема снежинка так как здесь важно учитывать привязку измерения поставщиков(Dim\_Providers) к измерению продуктов(Dim\_Products), а так же необходимость владеть географической информацией для измерений покупателей(Dim\_Customers), сотрудников(Dim\_Employees), магазинов(Dim\_Stores), поставщиков(Dim\_Providers).

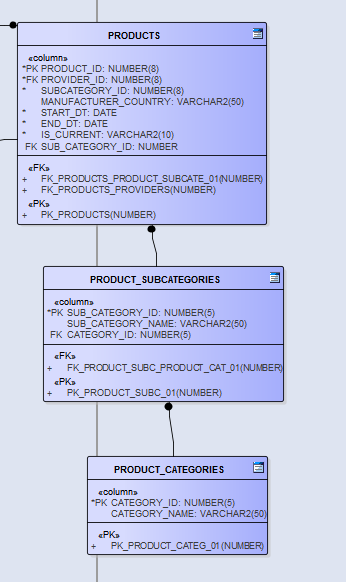
Так как проект только развивается и данных в нем еще не очень много, то схема снежинка не сильно повлияет на производительность.

## fundamential layer - 3 NF model



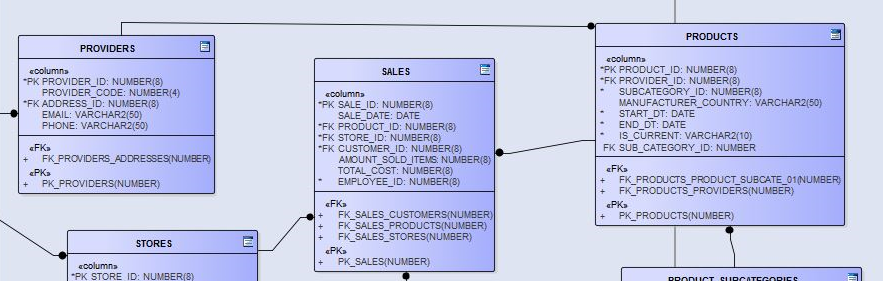
Ключевые моменты:

- У товара есть подкатегория и категория. Связь один ко многим – у одной подкатегории может быть много товаров, у одной категории может быть много подкатегорий.

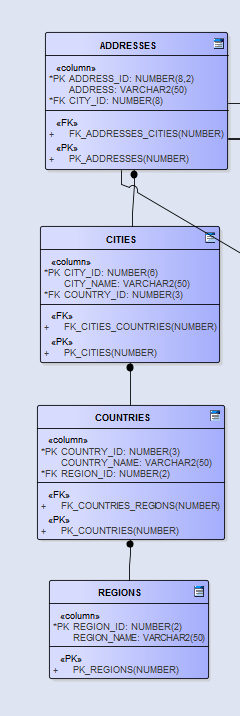


- Один товар поставляется одним поставщиком. От одного поставщика приходит много товаров. Связь один ко многим.

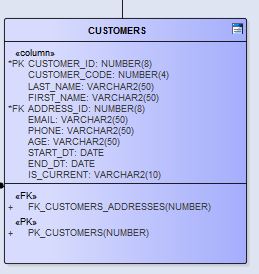
- Таблица Sales содержит информацию о том, когда была совершена продажа, кому, где, что было продано и сколько было заплачено денег. Sales является фактовой таблицей на уровне ST.



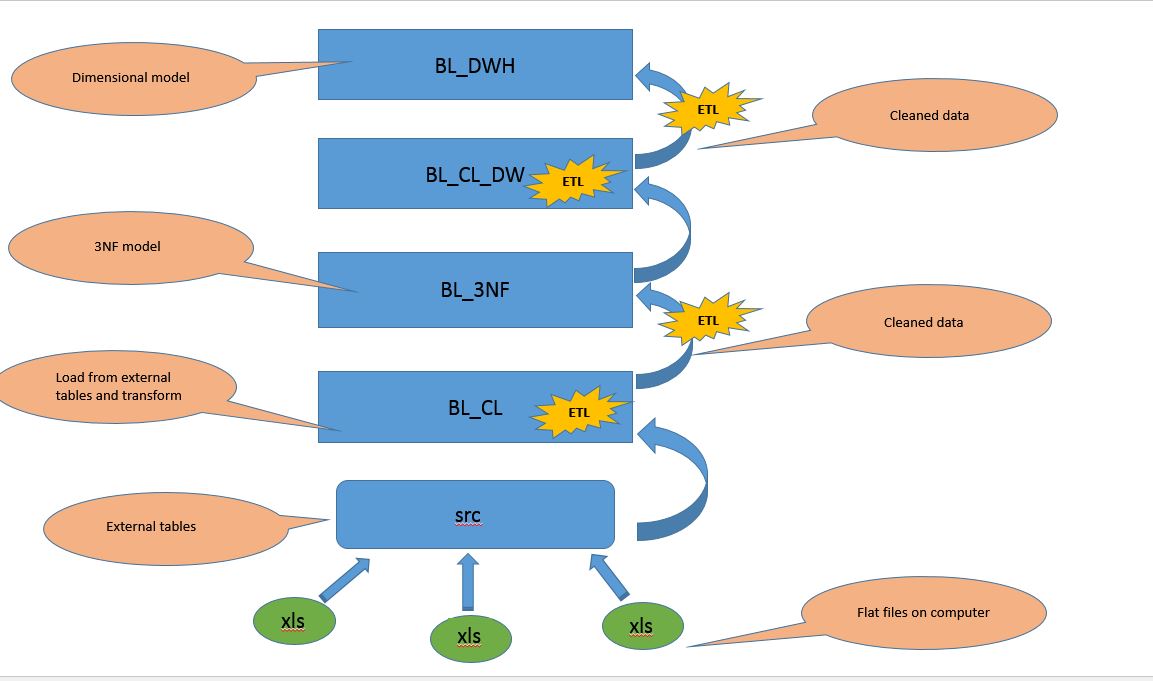
- Географические данные



Измерение Customers так же является SCD2 и содержит в себе натуральный ключ, фио, адрес, контактную информацию.



# Data Flow



## SA\_SRC

На этом этапе все источники сохраняются в виде внешних таблиц (таблицы, доступныу только для чтения, метаданные которых хранятся в базе данных, а данные – вне базы данных) добавляются в базу данных.

## Staging layer

### BL\_WRK

На этом этапе все данные из внешних таблиц загружаются в таблицы бд: это делается для увеличения производительности, так как считывание из бд быстрее, чем считывание с диска.

Таким образом, структуры таблиц sa\_src и bl\_wrk совпадают.

### BL\_CL

На этом этапе все данные очищаются,фильтруются. Именно на этом этапе осуществляется маппинг данных из разных таблиц.

Структура bl\_cl таблиц такая же как у bl\_3NF.

### BL\_3NF

На этом этапе уже полностью очищенные и подготовленные данные раскидываются по таблицам в третьей нормальной форме для их дальнейшей загрузки в DWH. Здесь уже появляются все необходимые ограничение (PK, FK) для поддержания ссылочной целостности.

## Aggregation layer

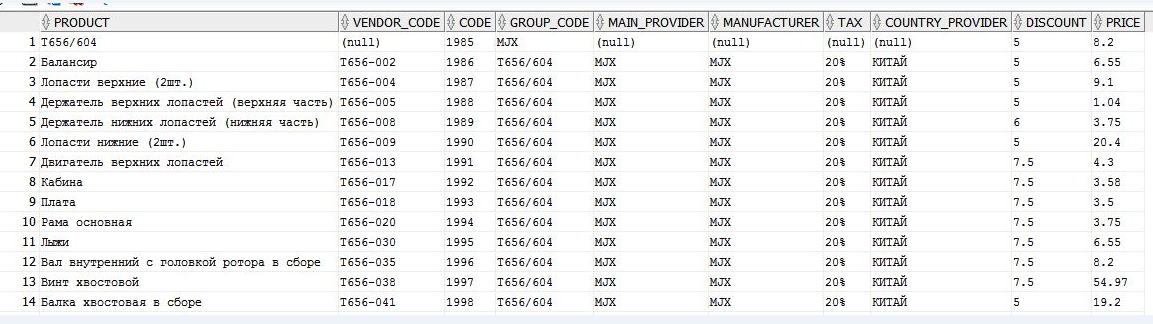
### BL\_CL\_DM

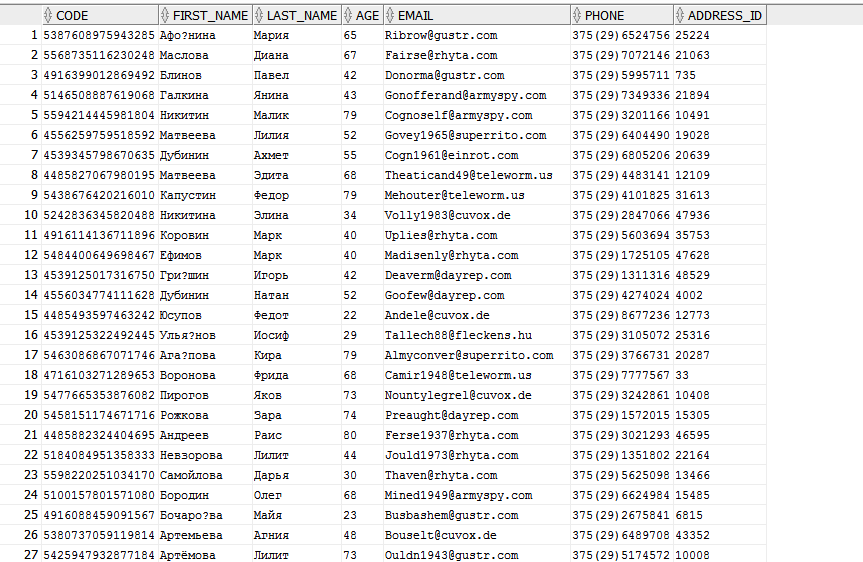
На этом этапе происходят все необходимые джоины, маппинг всех FK на значения из DWH

### BL\_DM

Непосредственно наше хранилище данных.

# SOURCE DESCRIPTION





# Fact table

# CLS-tables

# 3NF-tables

1. Перенос в 3НФ таблицы осуществлялся по одному и тому же прицнипу для всех:

Использовалась операция MERGE, где

* в условии USING c помощью оператора MINUS я выбирала записи, которых еще нет в 3НФ таблице
* в условии ON я испольовалcя бизнес-ключ
* Когда записи совпадали (WHEN MATCHED) по бизнес-ключу, обновлялись все колонки (кроме бизнес-ключа)
* Когда записи не совпадали (WHEN NOT MATCHED), новая запись вставлялась + id с помощью последовательности.

Все 3НФ таблицы типа SCD\_1, то есть имеют дату вставки и дату изменеий (sysdate по дефолту).

1. Для фактовой таблицы использовался просто INSERT, так как в ней ничего не может изменяться.

# partitionin Strategy

Для ускорения доступа к данным и скорости выполнения запросов будет использоваться Composite Partitioning:

- Первый этап - Range Partitioning для таблицы фактов Fact\_Sales по годам.

- Второй этап – List Partitioning для возрастной категории покупателей или же по категориям/подкатегориям.

Партицирование будет осуществляться по полю Full\_Date\_DT по годам:

* Партиция с данными с 2013 по 2017 года.
* Отдельные партиции с данными за каждый год с 2013 по 2020.
* Партиция со всеми остальными годами (с 2020 по 2045).

Внутри также будет партицирование (composite partitioning) по возрастным категориям, в которых находятся магазины.

Будет использоваться Range-List partition.