

|  |
| --- |
| DWH  WildBerries онлайн-магазин |
| EPBYBI17-WB17 |

| RELATED ARTIFACTS | |
| --- | --- |
| Ref. | Artifact Name |
|  |  |
|  |  |

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

Table of Contents

[WildBerries онлайн-магазин 1](#_Toc500340682)

[1 Описание предметной области 5](#_Toc500340683)

[1.1 Бизнес информация 5](#_Toc500340684)

[1.2 Недостатки существующего управления данными 5](#_Toc500340685)

[1.3 Преимущества внедрения хранилища данных 5](#_Toc500340686)

[2 Бизнес измерения 6](#_Toc500340687)

[2.1 Описание бизнес-процесса 6](#_Toc500340688)

[2.2 Зерно 6](#_Toc500340689)

[2.3 Измерения 6](#_Toc500340690)

[2.4 Факты 9](#_Toc500340691)

[2.5 Источники информации 9](#_Toc500340692)

[3 Логическая схема 10](#_Toc500340693)

[3.1 Фундаментальный слой DWH - схема 3 NF 10](#_Toc500340694)

[3.2 Слой доступа к данным DWH - схема Звезда 17](#_Toc500340695)

[4 Модель потоков данных 18](#_Toc500340696)

[5 Загрузка измерений на уровень 3NF 20](#_Toc500340697)

[5.1 Таблицы. 20](#_Toc500340698)

[5.2 Пакеты для Загрузки Данных 23](#_Toc500340699)

[5.2.1 Implicit cursor 23](#_Toc500340707)

[5.2.2 Explicit cursor 25](#_Toc500340708)

[5.2.3 Using merge 28](#_Toc500340709)

[6 Загрузка фактовых таблиц на 3NF 30](#_Toc500340710)

[6.1 Таблицы фактов. 30](#_Toc500340711)

[6.2 Пакет для генерации и Загрузки Данных 30](#_Toc500340712)

[6.3 Загрузка и перезагрузка данных. 31](#_Toc500340713)

[7 Загрузка измерений из 3нф в STAR схему 34](#_Toc500340714)

[7.1 Создание измерений и их загрузка 34](#_Toc500340715)

[7.2 Загрузка и перезагрузка данных. 36](#_Toc500340716)

[7.2.1 Тестирование перезагрузки данных измерения SCD1 37](#_Toc500340717)

[7.2.2 Тестирование перезагрузки данных измерения SCD2 39](#_Toc500340718)

[8 Загрузка фактовой таблицы в схему STAR 41](#_Toc500340719)

[9 тратегия партиционирования таблицы фактов 44](#_Toc500340720)

[10 Стратегия парaллельной загрузки 45](#_Toc500340721)

[11 Производительность схемы звезда. 48](#_Toc500340722)

[11.1 Запрос в 3NF 48](#_Toc500340723)

[11.2 Запрос в STAR 50](#_Toc500340724)

[11.3 Сравнение и вывод 51](#_Toc500340725)

[12 Макеты отчетов 52](#_Toc500340726)

# Описание предметной области

## Бизнес информация

Хранилище данных будет реализовано для нужд белорусского online-магазина Wildberries. Wildberries – это один из крупнейших интернет-магазинов модной одежды, обуви, аксессуаров, косметических средств, существующий уже более 12 лет. Магазин насчитывает более 7 тысяч брендов на сайте и более 100 тысяч заказов в день. Из года в год он развивается, расширяет географию и улучшает качество обслуживания!

## Недостатки существующего управления данными

С каждым днем количество товаров, отделений магазина и клиентов растет в геометрической прогрессии. Это требует пересмотра обработки данных. На данный момент база данных приложения недостаточно производительна, чтобы отрабатывать сложные запросы, поэтому на получение одного простейшего отчета, с которым хорошо построенное DWH справляется за считанные секунды, уходят часы. В такой ситуации многие запросы управляющего сектора не находят отклика, нет возможности быстрого реагирования на изменение рынка, реактивного изменения стратегий продаж, открытия новых. В результате теряется скорость принятия решений, теряются деньги, растет недоверие к ИТ решениям.

## Преимущества внедрения хранилища данных

Внедрение DWH поможет навести порядок и упростит использование данных для анализа. Сложные отчеты будут быстро формироваться, а, следовательно, не будут нагружать приложения, будут чаще использоваться и потому помогут в развитии бизнеса. Увеличится скорость реагирования на изменения рынка, что приведет к принятию стратегически важных решений, улучшится качество обслуживания, увеличится количество клиентов и прибыли соответственно.

Расширение бизнеса приведет к увеличению количества рабочих мест, как в самом онлайн магазине, так и у поставщиков и производителей приобретаемой продукции.

# Бизнес измерения

## Описание бизнес-процесса

Учет продажи товаров различных марок в интернет-магазине Wildberries с учетом различных показателей.

## Зерно

Зерном будет одна продажа конкретного товара одной торговой марки из одной категории одному клиенту в одном пункте самовывоза в пределах одного заказа.

## Измерения

В данном проекте будет реализовано 5 dimensions. Измерение DIM\_DATES – содержит необходимые данные для дальнейшего анализа в разрезе времени – дата, день недели, название дня недели, день месяца, день года, номер месяца, названия месяца, квартал, полугодие, год. Будет применяться как измерение для дат заказов и дат оплат товаров. Измерение DIM\_Customers\_SCD – dimension, содержащий информацию о клиентах магазина. Измерение является большим и включает более 100 000 строк. Измерение DIM\_Products\_SCD – измерение, которое содержит необходимую информацию о товаре, бренде (производителе) и категории товара. Измерение является иерархическим dimension. Измерение DIM\_PayDeliveries – описание способа оплаты и способа доставки. Измерение является Junk dimension. Измерение DIM\_Points – информация о пунктах самовывоза.

DIM\_Customers\_SCD и DIM\_Products\_SCD – это медленно меняющиеся измерения 2 типа (Slowly Changing Dimensions). DIM\_PayDeliveries и DIM\_Points - измерения 1 типа.

Измерение DIM\_DATES – содержит необходимые данные для дальнейшего анализа в разрезе времени – дата, день недели, название дня недели, день месяца, день года, номер месяца, названия месяца, квартал, полугодие, год. Применяется как измерение для дат заказов товаров.

CREATE TABLE dim\_dates (

full\_date DATE PRIMARY KEY,

day\_week NUMBER(1),

day\_week\_name VARCHAR2(50),

day\_month NUMBER(2),

day\_year NUMBER(3),

month\_num NUMBER(2),

month\_name VARCHAR2(50),

quarter NUMBER(1),

half\_year NUMBER(1),

year NUMBER(4) );

Измерение DIM\_Customers\_SCD – dimension, содержащий информацию о клиентах магазина.

CREATE TABLE dim\_customers\_scd (

customer\_surrid NUMBER(12) PRIMARY KEY,

customer\_srcid NUMBER(12,0) NOT NULL,

customer\_last\_name VARCHAR2(50 CHAR) NOT NULL,

customer\_first\_name VARCHAR2(50 CHAR) NOT NULL,

customer\_middle\_name VARCHAR2(50 CHAR) NOT NULL,

customer\_birthdate DATE,

customer\_gender VARCHAR2(3 CHAR) NOT NULL,

customer\_personal\_discount NUMBER(2,0),

customer\_loc\_srcid NUMBER(6,0) NOT NULL,

customer\_location\_name VARCHAR2(100 CHAR) NOT NULL,

customer\_address VARCHAR2(100 CHAR) NOT NULL,

customer\_loc\_type\_srcid NUMBER(2) NOT NULL,

customer\_loc\_type\_short VARCHAR2(10 CHAR) NOT NULL,

customer\_loc\_type\_full VARCHAR2(25 CHAR) NOT NULL,

customer\_dis\_srcid NUMBER(3,0) NOT NULL,

customer\_district VARCHAR2(100 CHAR) NOT NULL,

customer\_reg\_srcid NUMBER(2,0) NOT NULL,

customer\_region VARCHAR2(100 CHAR) NOT NULL,

start\_dt DATE NOT NULL,

end\_dt DATE NOT NULL,

is\_active VARCHAR2(1 CHAR) NOT NULL,

ta\_insert\_dt DATE NOT NULL,

ta\_update\_dt DATE NOT NULL);

Измерение DIM\_Products\_SCD – измерение, которое содержит информацию о товаре, бренде (производителе) и категории товара.

CREATE TABLE dim\_products\_scd (

product\_surrid NUMBER(8) PRIMARY KEY,

product\_srcid NUMBER(8) NOT NULL,

product\_desc VARCHAR2(50) NOT NULL,

product\_vendorcode VARCHAR2(8 CHAR) NOT NULL,

product\_color VARCHAR2(25) NOT NULL,

product\_brand VARCHAR2(100) NOT NULL,

product\_type VARCHAR2(100) NOT NULL,

product\_subcategory VARCHAR2(150) NOT NULL,

product\_category VARCHAR2(50) NOT NULL,

start\_dt DATE NOT NULL,

end\_dt DATE NOT NULL,

is\_active VARCHAR2(1 CHAR) NOT NULL,

ta\_insert\_dt DATE NOT NULL,

ta\_update\_dt DATE NOT NULL);

Измерение DIM\_PayDeliveries – описание способа оплаты и способа доставки. Измерение является Junk dimension.

CREATE TABLE dim\_paydeliveries (

paydelivery\_surrid NUMBER(2) PRIMARY KEY,

delivery\_srcid NUMBER(8) NOT NULL,

delivery\_desc VARCHAR2(50) NOT NULL,

payoption\_srcid NUMBER(3) NOT NULL,

payoption\_desc VARCHAR2(50) NOT NULL,

ta\_insert\_dt DATE NOT NULL,

ta\_update\_dt DATE NOT NULL );

Измерение DIM\_ Pickuppoints – информация о пунктах самовывоза.

CREATE TABLE dim\_pickuppoints (

pickuppoint\_surrid NUMBER(3) PRIMARY KEY,

pickuppoint\_srcid NUMBER(3) NOT NULL,

address VARCHAR2(250 BYTE) NOT NULL,

loc\_srcid NUMBER(6,0) NOT NULL,

location\_name VARCHAR2(100 CHAR) NOT NULL,

loc\_type\_srcid NUMBER(2) NOT NULL,

loc\_type\_short VARCHAR2(10 CHAR) NOT NULL,

loc\_type\_full VARCHAR2(25 CHAR) NOT NULL,

dis\_srcid NUMBER(3,0) NOT NULL,

district VARCHAR2(100 CHAR) NOT NULL,

reg\_srcid NUMBER(2,0) NOT NULL,

region VARCHAR2(100 CHAR) NOT NULL,

ta\_insert\_dt DATE NOT NULL,

ta\_update\_dt DATE NOT NULL

);

## Факты

Таблица фактов будет содержать количество проданных единиц товара, сумму без скидки, сумму предоставленной скидки и итоговую сумму реализации за месяц в разрезе товара, бренда, категории товара, клиента, способа доставки и оплаты. Содержит дегенеративное измерение – Order\_ID.

CREATE TABLE fct\_salesitems

(

dd\_order\_srcid NUMBER(15),

dim\_order\_date DATE,

dim\_customer\_surrid NUMBER(12,0),

dim\_product\_surrid NUMBER(10),

dim\_paydelivery\_surrid NUMBER(2,0),

dim\_pickuppoint\_surrid NUMBER(3,0),

fct\_quantity NUMBER(20),

fct\_item\_sum NUMBER(35,2),

fct\_discount\_sum NUMBER(35,2),

fct\_total\_sum NUMBER(35,2),

insert\_dt DATE );

## Источники информации

Основная информация о товарах, категориях, брендах, способах доставки, оплаты и пунктах самовывоза взята на официальном сайте: <https://www.wildberries.by/>

<https://www.wildberries.by/services/sposoby-oplaty>

<https://www.wildberries.by/services/besplatnaya-dostavka#pickupPoints>

<https://www.wildberries.by/catalog/0/brand.aspx>

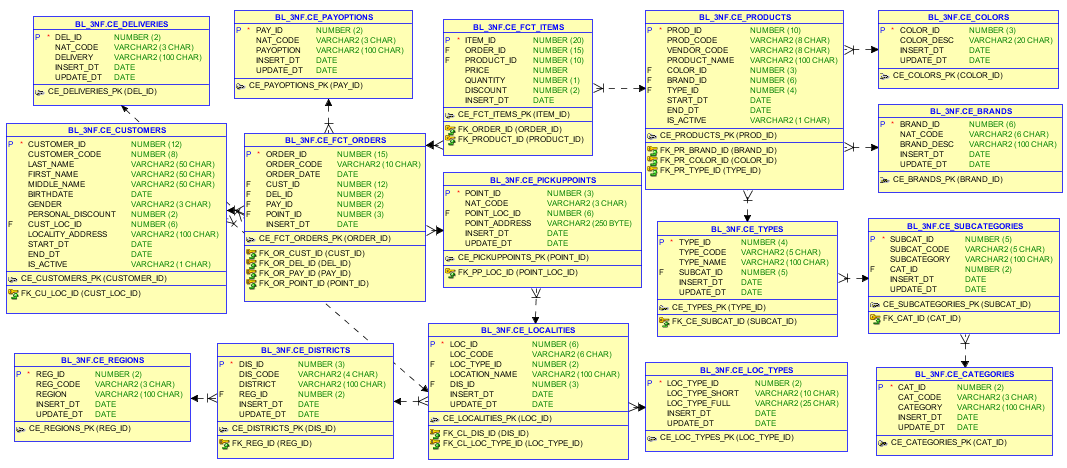
Населенные пункты РБ - <https://www.bygeo.ru/materialy/naselenie-belarusi/>

Названия улиц - <https://ato.by/streets>.

Какая-то часть данных будет взята с сайтов, какая-то сгенерирована c помощью других интернет сайтов, например http://www.fakenamegenerator.com, также будет применена генерация с помощью PLSQL и поиск, копирование и вставка данных вручную.

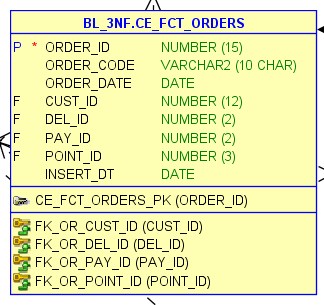
# Логическая схема

## Фундаментальный слой DWH - схема 3 NF



Одна из основных таблиц CE\_FCT\_ORDERS содержит в себе информацию о заказе, его номер, дата заказа, идентификатор клиента, совершившего данный заказ, идентификатор типа оплаты, идентификатор типа доставки и идентификатор пункта самовывоза.

Один заказ Order\_Code может быть сделан одним клиентом Cust\_Id в один календарный день Order\_Date. Он может быть оплачен с помощью одного из вариантов Pay\_Id и может быть доставлен курьером, почтой или получен в пункте самовывоза Del\_Id , в третьем случае указывается какой именно пункт Point\_Id. Также фиксируются дата заказа Order\_Date.



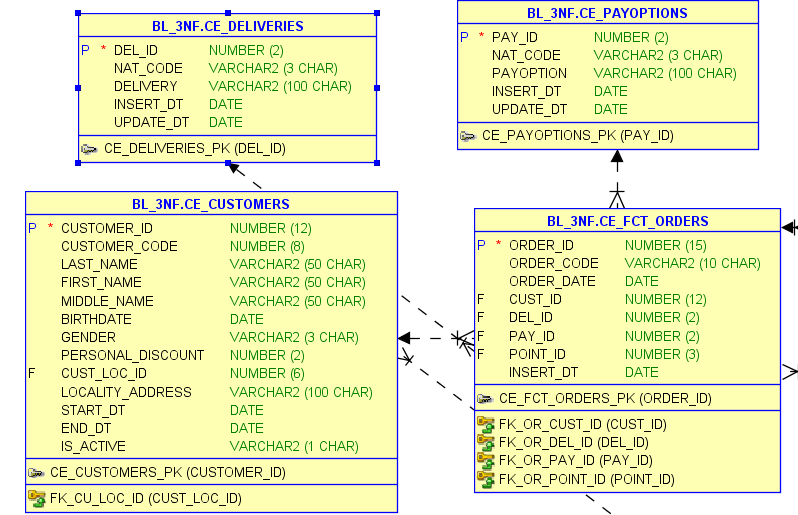
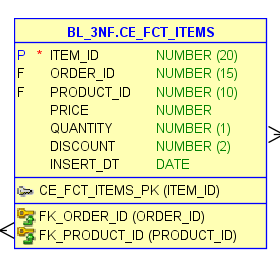


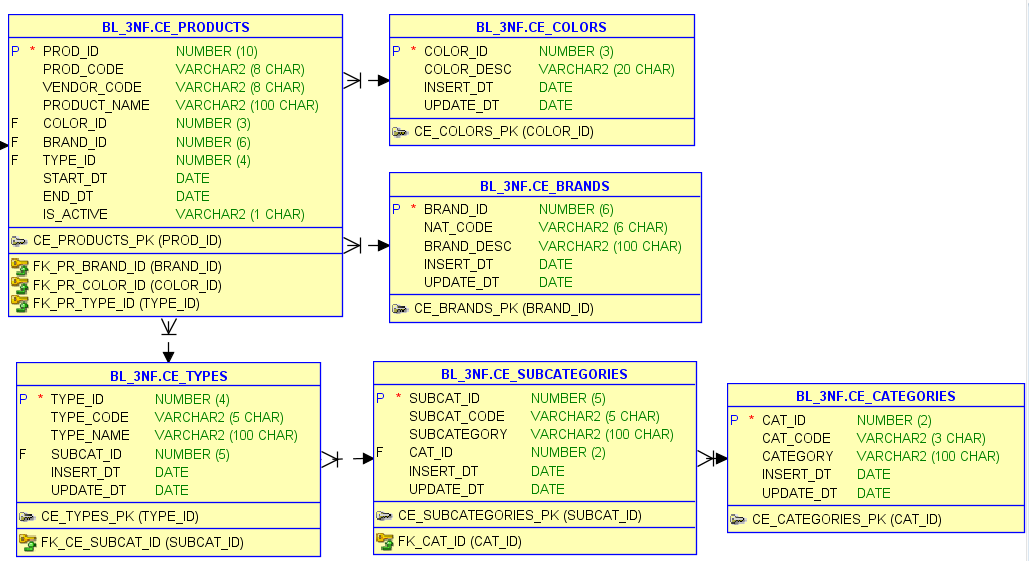
Таблица CE\_SCT\_ITEMS содержит в себе информацию об идентификаторе заказа, идентификатор товара, а также цену товара, его количество, процент скидки (суммируются скидка клиента и разовые скидки).

Один пункт заказа Item\_Num относится только к одному заказу Order\_Id. Один заказ может содержать N пунктов. Пункт относится исключительно к одному товару Product\_Id.

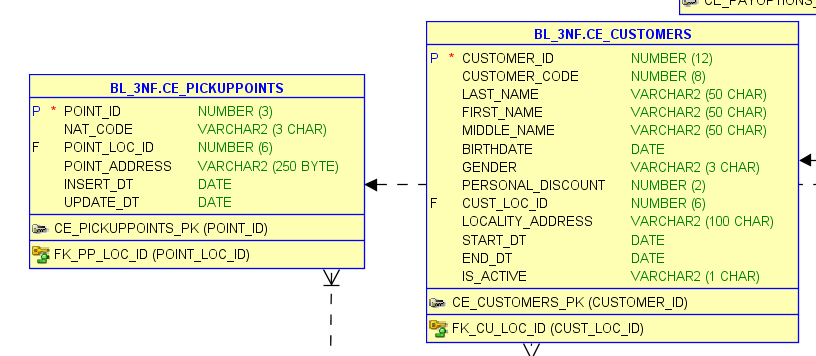


Пример данных для ветки товаров:

* Таблица CE\_Products – Product\_Name: Платье Bracegirdle, Product Code: 2120550 🡪
* Таблица CE\_Types – Type Desc: [Платья-миди](https://www.wildberries.by/catalog/zhenshchinam/odezhda/platya-midi) 🡪
* Таблица CE\_Brands – Brand Desc: ONLY 🡪
* Таблица CE\_Subcategories: Subcategory Desc - Платья 🡪
* Таблица CE\_Categories: Category Desc - Женская одежда

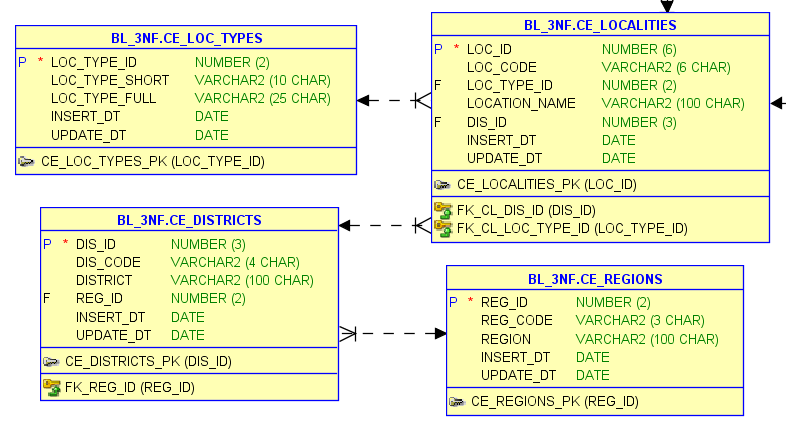


Пункты самовывоза и клиенты имеют point\_loc\_id и cust\_loc\_id, который связывает их с таблицей CE\_locations. Пункт самовывоза указывается в любом заказе, независимо от метода его оплаты и доставки клиенту, также нету зависимости локации клиента при покупки им товара и пунктом самовывоза для этого товара.



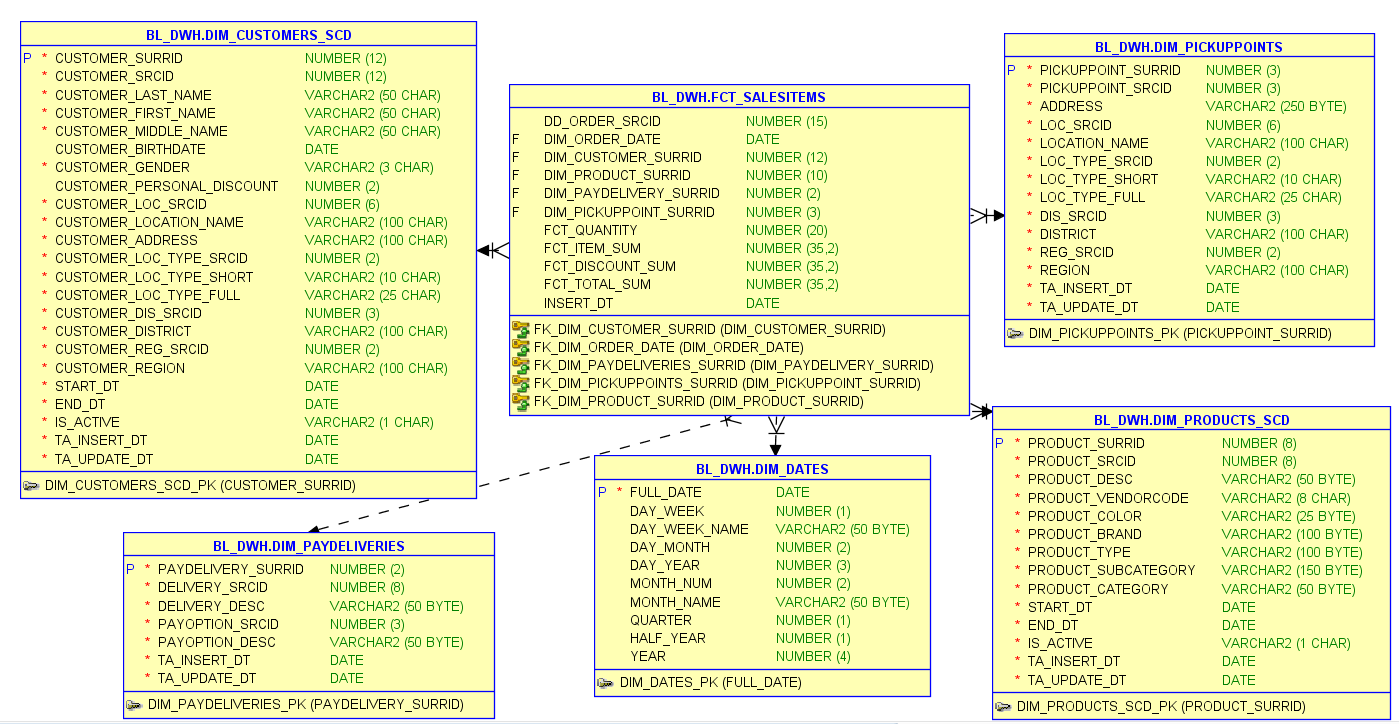
Поскольку доставка производится по Беларуси, то это отражается в формировании ветки локаций. Пример:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Location\_Code | Location\_Desc | District\_Desc | Region\_Desc | Type\_desc\_short |
| 6240833091 | Нарочь | Мядельский | Минская область | аг. |



## Слой доступа к данным DWH - схема Звезда

Все связи на слое доступа к данным между измерениями и таблицей фактов реализованы как один ко многим.



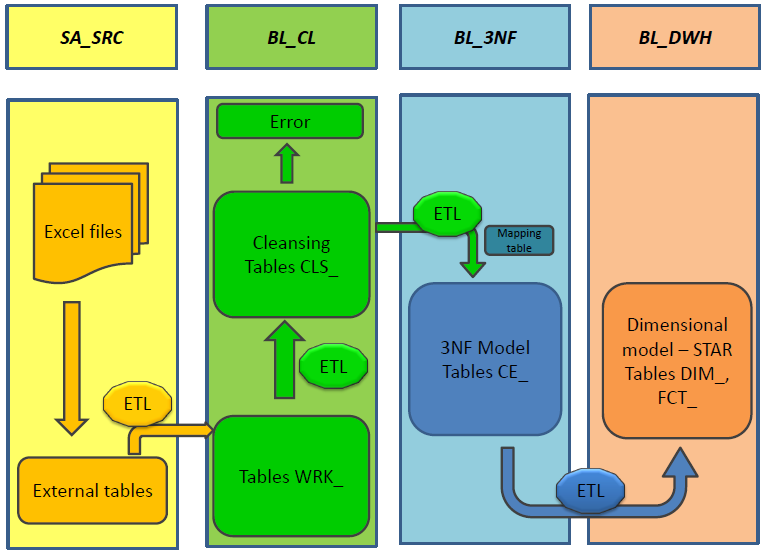
# Модель потоков данных

Хранилище состоит из 5 слоев:

* sa\_src – source layer. Это слой на котором находятся внешние источники информации в виде Excel файлов и external tables.
* bl\_cl – это слой временных данных, который состоит из двух основных частей – область Work и область Cleansing. В таблицы с префиксом wrk\_ данные заливаются напрямую из external tables без применения каких-либо преобразований. Далее данные очищаются, валидируются, фильтруются. Перед каждым новой загрузкой старые данные из этих таблиц будут удаляться, а загрузка будет производиться в очищенные таблицы.
* bl\_3nf – это слой постоянного хранения данных, куда загружаются данные из таблиц cls\_ в третью нормальную форму. Здесь же производится маппинг данных согласно специальной таблице. При этом устраняется аномальность, избыточность и оптимизируется производительность доступа к данным.
* bl\_cl\_dwh – это слой временных данных, между слоем 3NF и DM, которая состоит из двух основных частей – область Work и область Cleansing. Перед каждым новой загрузкой старые данные из этих таблиц будут удаляться, а загрузка будет производиться в очищенные таблицы. Так же этот слой может отсутсвовать, тогжа данные из 3NF сразу загружаются в схему bl\_dwh.
* bl\_dwh – это слой постоянного хранения данных, куда загружаются данные из bl\_3nf после применения агрегирования. В таблицах dim\_ и fct\_ хранятся денормализованные данные, которые позволяют значительно ускорить доступ к данным. В качестве dimensional модели была выбрана схема Star. Причиной выбора была ее простота и малое количество таблиц в сравнении со схемой Snowflake.

Все этапы загрузок и изменения данных производятся с помощью ETL. В нашем случае это будут процедуры PL/SQL.

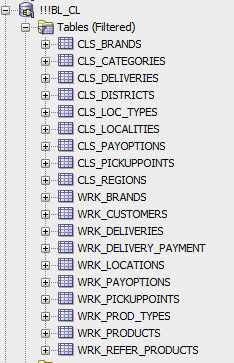
Картинка изображения процесса схематичная для общего случая.

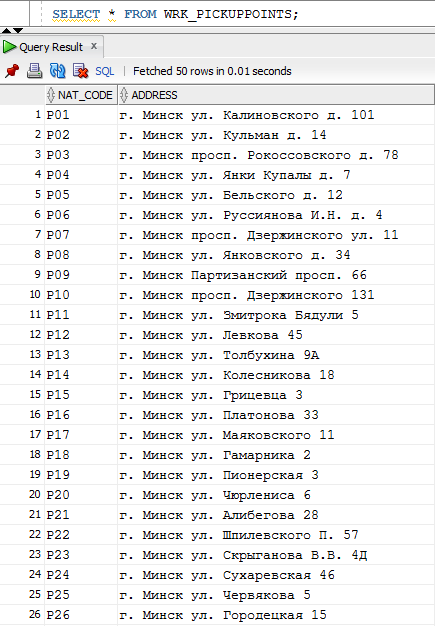
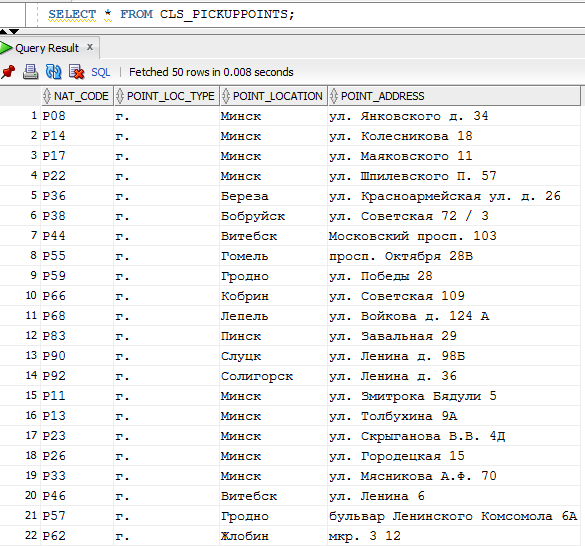


# Загрузка измерений на уровень 3NF

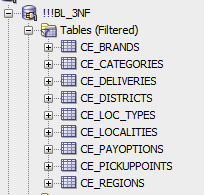
## Таблицы.

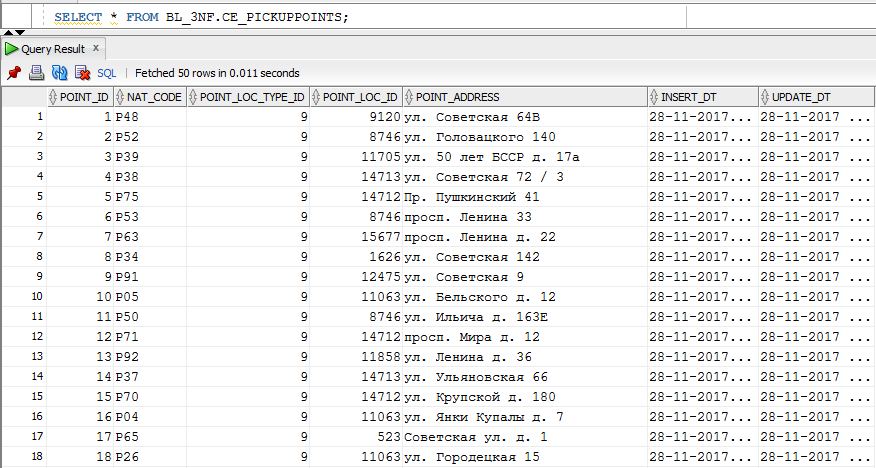
Таблицы WRK слоя BL\_CL содержат данные загруженные из экстернал таблиц без применения трансформации. Таблицы CLS слоя BL\_CL имеют структуру типа 3НФ, но без использования ограничений целостности. Данные трансформируются при переходе из WRK в CLS. При загрузке данных я использовала явные и неявные курсоры.



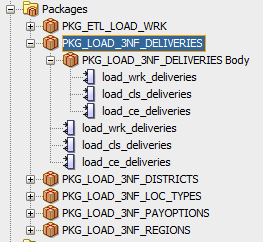
Таблицы CE слоя 3НФ загружаются из CLS с помощью Merge и ид генерируются с использованием sequence.





## Пакеты для Загрузки Данных

Загрузка каждой сущности производится в отдельном пакете. Пакет состоит из спецификации и тела, и в большинстве своем содержит три процедуры, каждая из которых производит загрузку в wrk, cls и ce таблицы соответственно. Например:



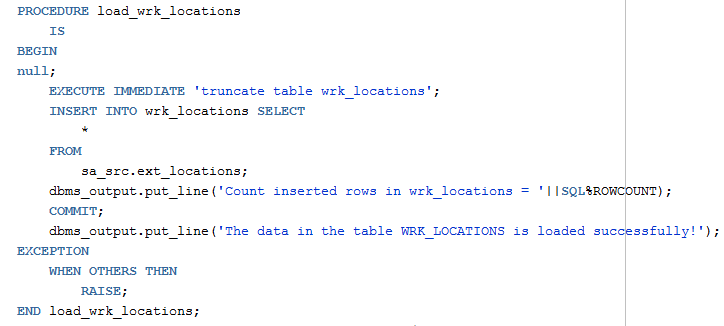
Процедуры загрузки данных возвращают один и тот же результат, при одинаковых входных данных.



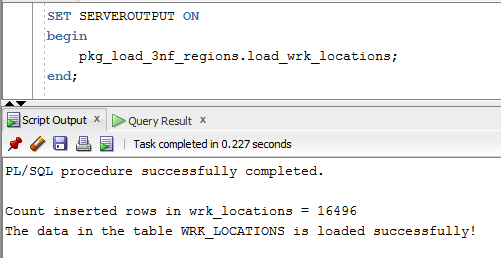
### Implicit cursor

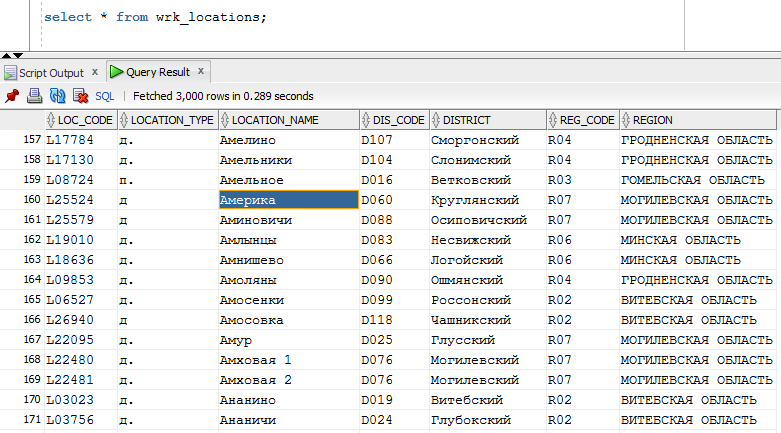
Примерами неявных курсоров являются INSERT, UPDATE, или SELECT INTO. Оператор DML, который находится в секции выполнения или исключения блока PL/SQL, являются неявными курсорами. Неявные курсоры не обязательно должны быть DECLAREd, OPENed, FETCHed или CLOSEd. Неявные атрибуты курсора указываются с помощью курсора SQL, таким является, например, SQL%ROWCOUNT, использующийся для подсчет записей в курсоре.

Пример неявного курсора:



Выполнение:

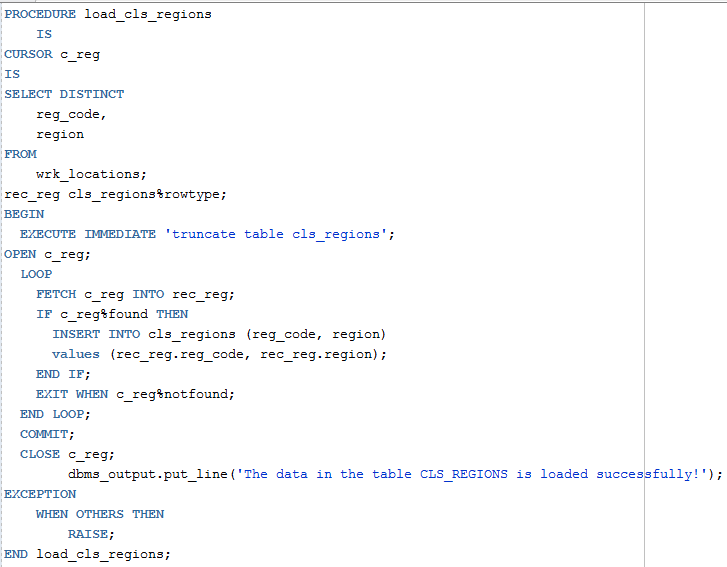




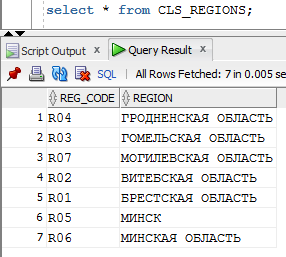
### Explicit cursor

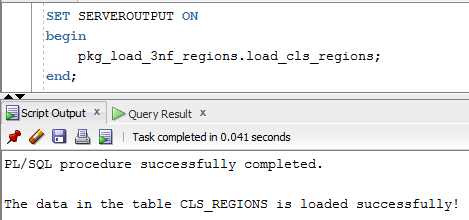
Явный курсор - это курсор, который нужно создать явно, а затем управлять им. Нужно объявить и определить явный курсор, указать ему имя и связать его с запросом. Курсоры бывают с параметрами и без.

Пример явного курсора без параметра:

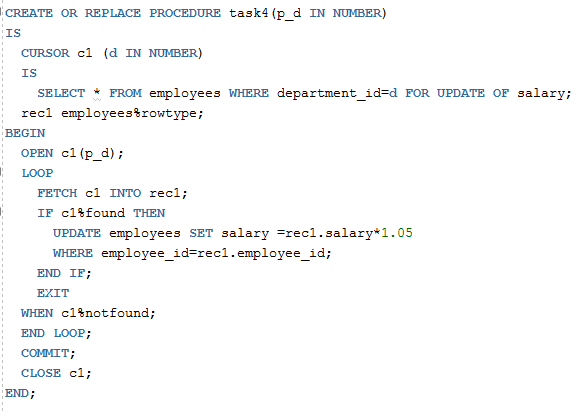


Выполнение:





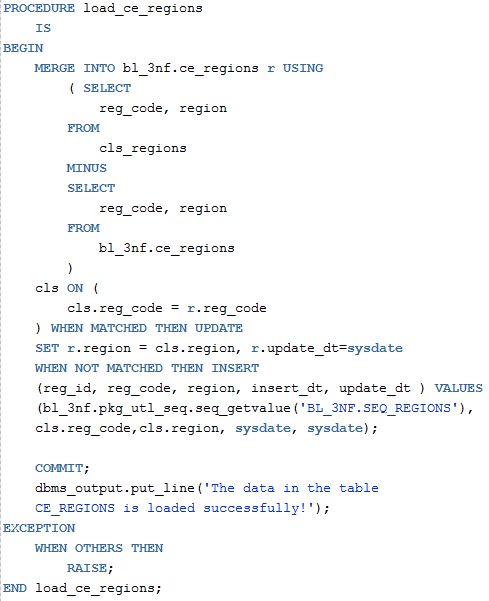
В exit task я не использовала курсоры с параметром, но делала их в classwork. Пример явного курсора с параметром:



### Using merge

Merge хорош, когда нужно при определенных совпадениях сделать update данных, а в противном случае INSERT или DELETE. В нашем случае подходит для заполнения таблиц CE на слое 3НФ.

Пример использования:



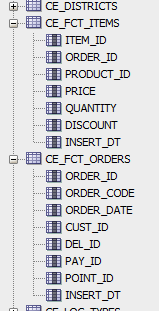
# Загрузка фактовых таблиц на 3NF

## Таблицы фактов.

В DWH на уровне 3NF реализованы две таблицы фактов.

Первая CE\_FCT\_ORDERS содержит информацию в разрезе заказа – клиент, дата заказа, тип доставки, оплаты, пункт самовывоза.

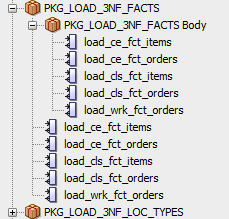
Вторая CE\_FCT\_ITEMS содержит информацию в разрезе одного пункта из заказа – ид заказа, которому он принадлежит, проданный товар, его цена, количество и процент скидки на него в момент продажи.



## Пакет для генерации и Загрузки Данных

У меня загрузка фактовых таблиц производится в одном пакете. Пакет состоит из спецификации и тела, и содержит 5 процедур, каждая из которых производит загрузку в wrk, cls и ce таблицы соответственно.

Пакет:

****

Все данные генерируются с помощью SQL и PLSQL. В частности для генерации данных я использовала функции value и normal пакета dbms\_random. Это позволило неравномерно сгенерировать заказы в разрезе клиентов, продуктов и других показателей. При загрузке данных я использовала явные и неявные курсоры, а также bull collect.

## Загрузка и перезагрузка данных.

Процедуры генерации и загрузки данных для фактовых таблиц 3NF слоя:

pkg\_load\_3nf\_facts\_def.sql

pkg\_load\_3nf\_facts\_impl.sql

Факты генерируются случайно, поэтому каждый раз набор данных разныЙ. Но при одинаковых входных данных, процедуры загрузки данных возвращают один и тот же результат. Загрузка и перезагрузка данных осуществляется с помощью скрипта load\_fct\_tables:

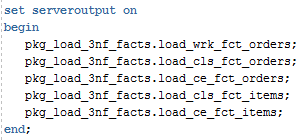
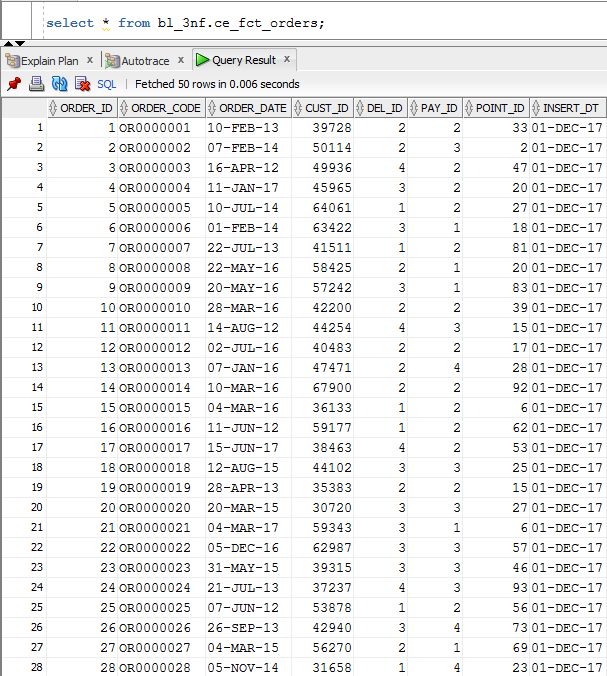


Таблица фактов с заказами:



Всего заказов:

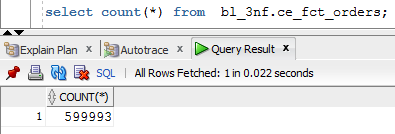
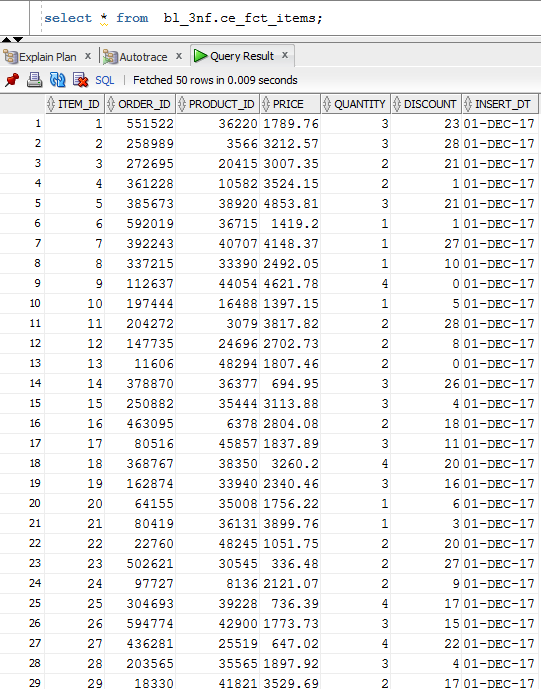
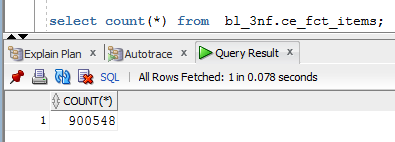


Таблица фактов с подпунктами заказов:

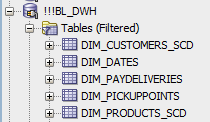


Всего подпунктов заказов:



# Загрузка измерений из 3нф в STAR схему

В DWH на уровне DM реализовано 5 таблиц измерений: даты, продукты, пункты доставки, клиенты и общий дим для способов доставки и оплаты. Два из них SCD 2 типа.

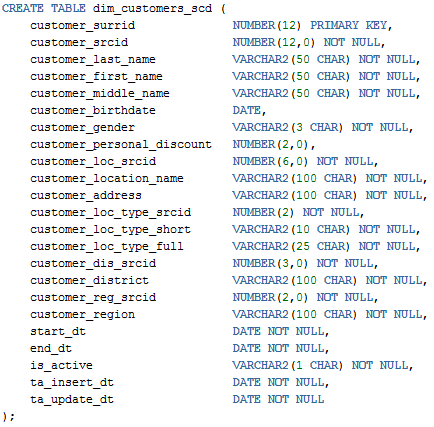


## Создание измерений и их загрузка

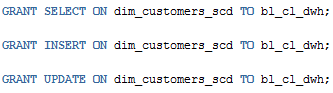
Каждое измерение, кроме даты, находятся в схеме BL\_DWH и реализуются с помощью 3 скриптов: создание таблицы, создание сиквенса и права для других схем. Также используется промежуточные таблицы cls2\_ схемы BL\_CL\_DWH, они практически полностью повторяют структуру dim\_ таблиц, не имеют ключей и используется для промежуточной заливки данных.

Пример на измерении клиентов.

Создание измерения dim\_:



Необходимые гранты на таблицу dim\_customers\_scd:



Создание сиквенса и грант на него для пользователя, осуществяющего загрузку данных:



Загрузка данных в измерения происходит с помощью процедур пакетов. Пример будет располагаться в домашней папке, остальные находятся в папке проекта:

pkg\_load\_dim\_customers\_def.sql

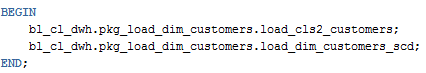
pkg\_load\_dim\_customers\_impl.sql

При загрузке данных использовались такие конструкции PLSQL как Bulk collect, Forall и For Loop.

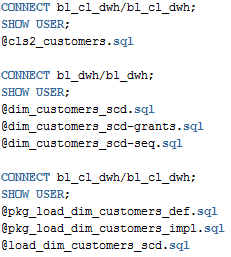
Bulk collect – это select, который извлекает несколько строк с одной выборкой, улучшая скорость извлечения данных. Forall используется для INSERT, UPDATE или DELETE для быстрого изменения нескольких строк данных. Их можно объединить, тогда результат будет заполнен итерациями выражения FORALL. Цикл FOR LOOP в Oracle позволяет выполнить код повторно в течение фиксированного количества раз.

## Загрузка и перезагрузка данных.

Загрузка и перезагрузка данных осуществляется с помощью скриптов, которые вызывают процедуры пакетов. Для каждого измерения используется отдельный пакет.

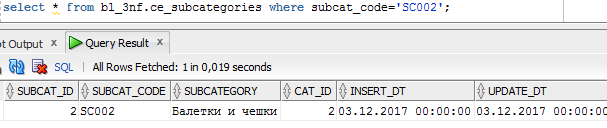


Полное пересоздание объектов и загрузка данных производится с помощью батча:

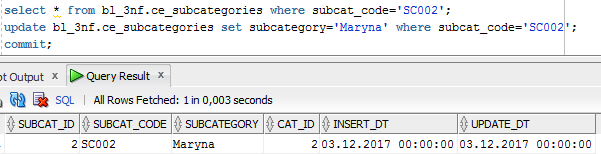


### Тестирование перезагрузки данных измерения SCD1

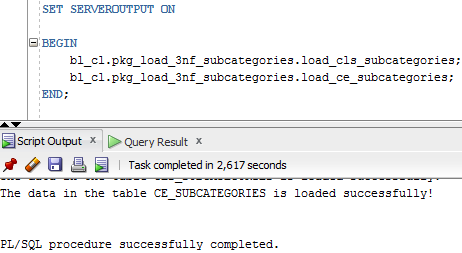
Функционирование системы приведу на одном из измерений SCD01 Подкатегории товаров. Данные на момент начала тестирования:



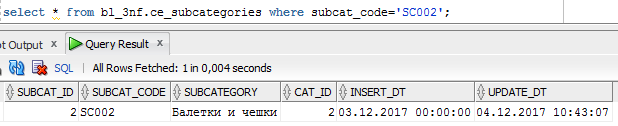
Чтобы не изменять сорс, я изменю название категории в самой таблице, как будто эти данные были изначально залиты в систему:



Далее я запускаю процедуру загрузки подкатегорий:



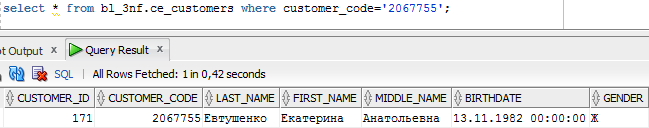
И проверяю полученный результат:

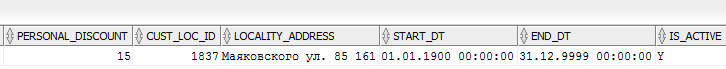


Название подкатегории изменилось на новое, а также была изменена дата обносления записи update\_dt.

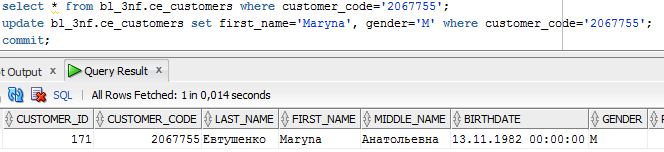
### Тестирование перезагрузки данных измерения SCD2

Функционирование системы приведу на одном из измерений SCD02 Клиенты. Данные на момент начала тестирования:



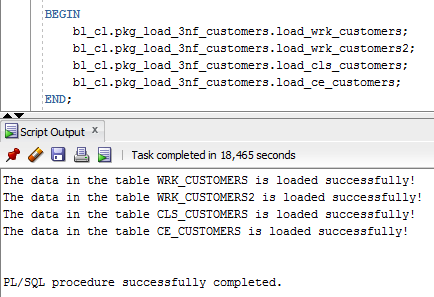


Чтобы не изменять сорс, я изменю атрибуты клиента в самой таблице, как будто эти данные были изначально залиты в систему:

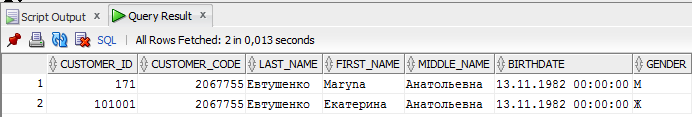


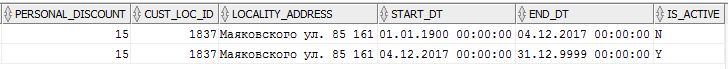


Далее я запускаю процедуру загрузки клиентов:



И проверяю полученный результат:

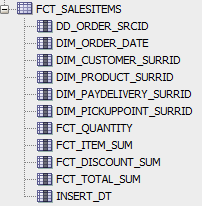




Данные были перезагружены, старая запись была обновлена и стала неактивной, о чем говорят значения полей end\_dt и is\_active, а новая была вставлена

# Загрузка фактовой таблицы в схему STAR

В DWH на уровне DM реализована одна таблица фактов. Она содержит агрегаты количества, суммы товара, суммы скидки и суммы оплаченного клиентом в разрезе клиента, продукта, способа оплаты и дсотавки, пункта самовывоза и даты заказа.

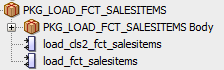


Поскольку все пакеты с процедурами обработки данных находятся в схеме bl\_cl\_dwh нужно дать гранты на возможность оперирования данными таблицы:



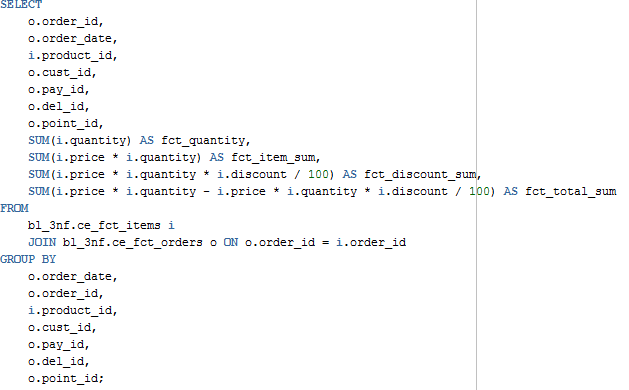
Загрузка фактовой таблицы производится в пакете . Пакет состоит из спецификации и тела, и содержит 5 процедур, каждая из которых производит загрузку в wrk, cls и ce таблицы соответственно.

Пакет:

****

Все данные загружаются с помощью PLSQL. При загрузке данных используются явные курсоры, а также BULL COLLECT, FORALL и FOR LOOP.

Запрос для подсчитывания данных на основе фактовых таблиц в 3NF.

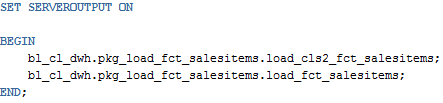


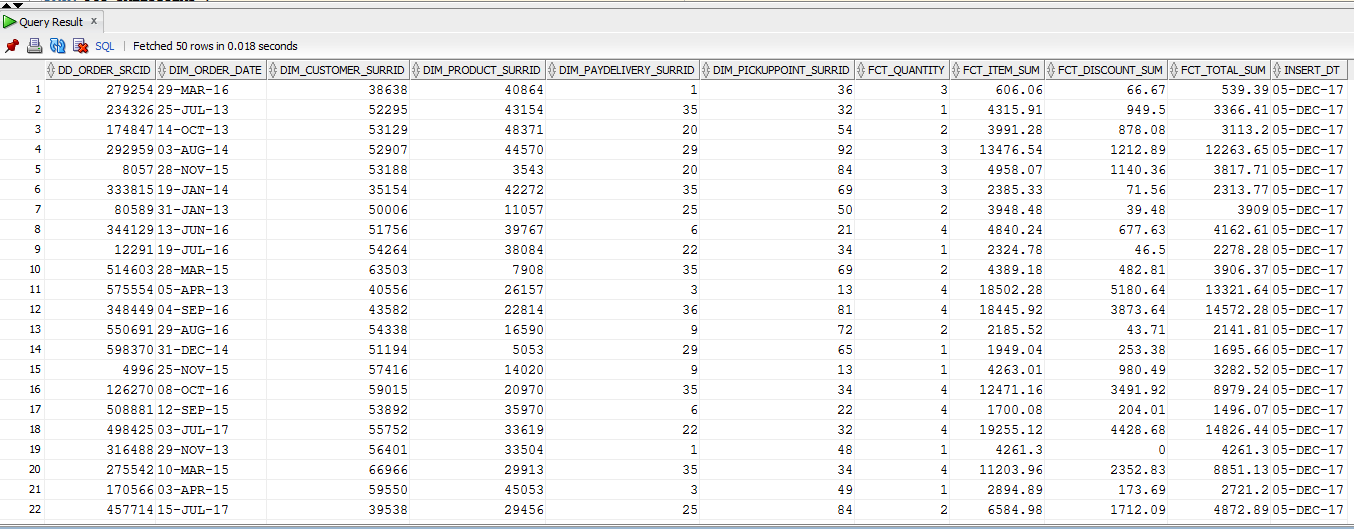
Процедуры генерации и загрузки данных для фактовой таблицы Star схемы находятся в пакете pkg\_load\_fct\_salesitems и для примера располагаются домашней папке:

pkg\_load\_fct\_salesitems\_def.sql

pkg\_load\_fct\_salesitems\_impl.sql

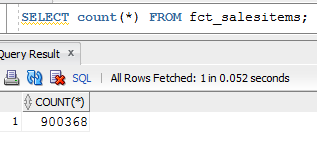
Загрузка данных осуществляется с помощью скрипта load\_fct\_tables:





Общее количество строк меньше, чем в таблицах фактов 3NF, это обусловлено применением агегирования данных.

Всего заказов:



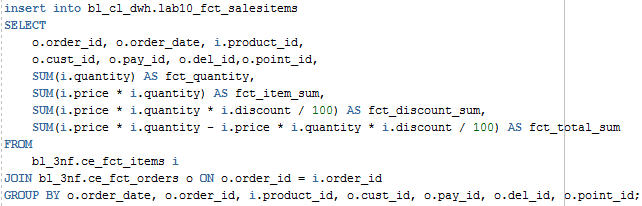
# тратегия партиционирования таблицы фактов

Для ускорения доступа к данным и скорости выполнения запросов используется Partitioning для таблицы фактов FCT\_SalesItems. Таблица разделяется методом Range на партиции по годам продаж.

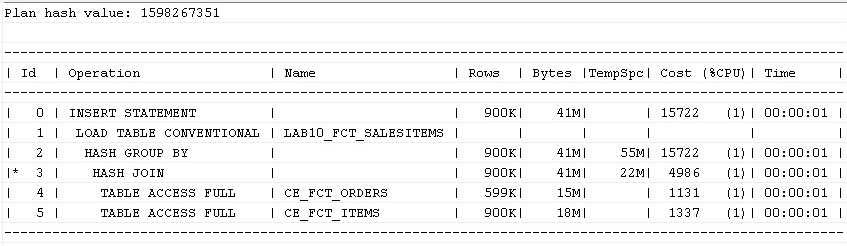


# Стратегия парaллельной загрузки

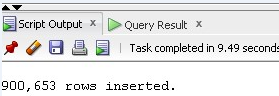
Чтобы выбрать стратегию распараллеливания загрузки данных нужно рассмотреть вариант вставки данных в фактовую таблицу с хинтом и без него. Вначале рассмотрим вставку данных без учета распараллеливания:



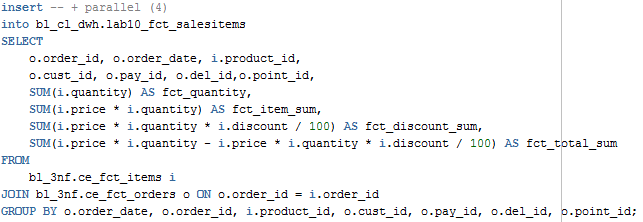
План выглядит следующим образом:



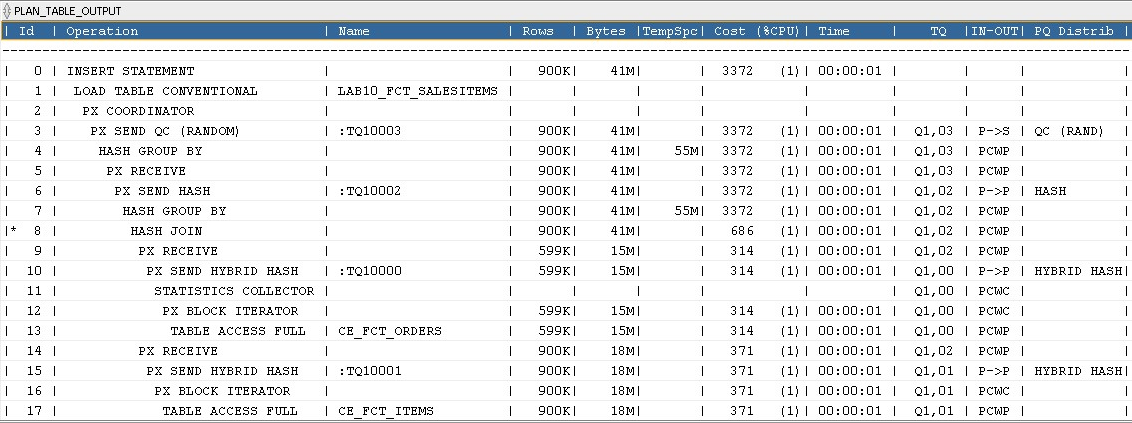
Вставка данных происходит за 9,49 секунд.



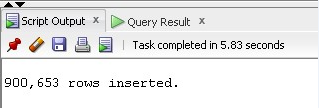
Далее делаем такие же действия, но уже с использованием parallel хинта:



План выполнения:



Время вставки такого же количества записей было выполнено за 5,83 секунды.



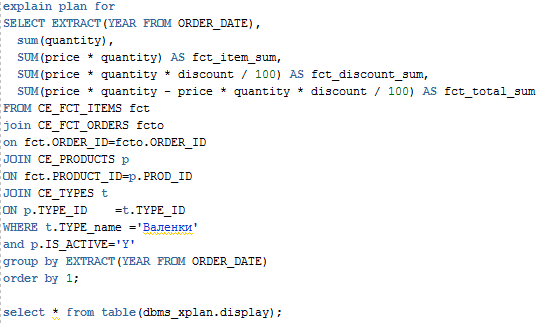
Таким образом, применение хинта parallel привело к уменьшению времени вставки в 1,8 раза, и уменьшению cost почти в 4 раза. Из этого можно сделать вывод, что использование распараллеливания выгодно в нашем случае.

# Производительность схемы звезда.

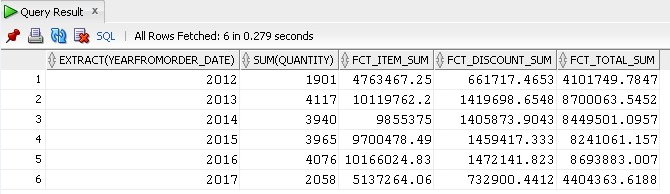
Для определения перфоманса построенной STAR схемы по сравнению со схемой 3NF используем анализ продажи валенок с 2012 по 2017 год. Построим запрос в каждой схеме и проаанлизируем полученные результаты.

## Запрос в 3NF

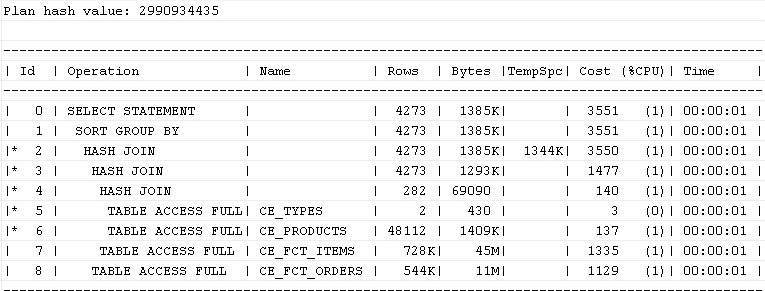
Для получения необходимых данных нам нужно сделать 4 Join таблиц.



Результат был выполнен за 0,279 секунд:

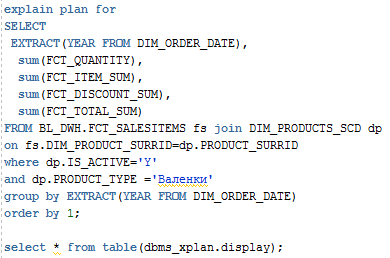


План выполнения:

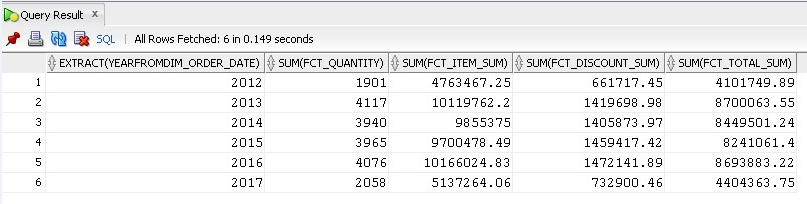


## Запрос в STAR

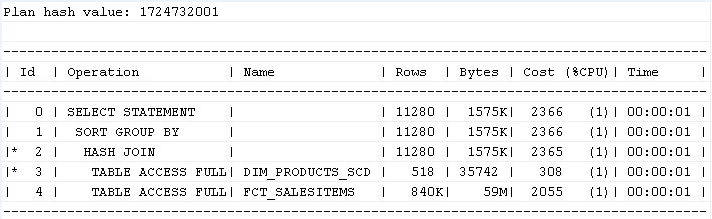
Для получение того же результата в схеме Star нам нужно сделать 1 Join.



Результат был получен за 0,149 секунд



План выполнения:



## Сравнение и вывод

Несмотря на то, что в запросе схемы Star пришлось вытянуть более 11 280 строк в сравнении c 4 273 строками в схеме 3NF, скорость выполнения в звезде увеличилась в 2 раза. Это связано прежде всего с тем, что данные в Star схеме уже сагрегированы в определенных разрезах, и, хотя и применяется суммирование, но строк для агрегации стновится значительно меньше.

Из этого можно сделать вывод, что в схеме звезда запросы выполняются намного быстрее, а следовательно основная цель DWH достигнута.

# Макеты отчетов

С помощью данного DWH можно анализировать ежемесячные, ежеквартальные и годовые продажи товаров по подкатегориям и категориям, в разрезе клиентов или пунктов самовывоза, по регионам и другим доступным критериям. Ниже приведены примеры возможных вариантов отчетов.

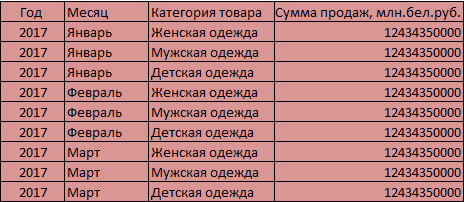
Отчет о ежемесячных продажах товаров из категории Женская одежда за 2017 год.



Отчет о продажах мужской обуви за январь и февраль 2017 года в разрезе областей.



Отчет о совокупных продажах товаров по основным категориям в разрезе месяцев за 1 квартал 2017 года.



| REVISION HISTORY | | | | | |
| --- | --- | --- | --- | --- | --- |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.1. | Начальная версия 2.0 | Марина Глазунова | 17-ноя-2017 |  |  |
| 1.2 | Исправление согласно новым разработкам | Марина Глазунова | 06-дек-2017 |  |  |