|  |
| --- |
|  |
| DWH продаж сети АЗС "Беларуснефть" |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| История изменений | | | | | |
| Версия | Изменения | Автор | Дата | Рецензировано | |
| Рецензент | Дата |
| 0.1 | Начальная версия | Olga Hilko | 08-Nov-2017 |  |  |
| 0.2 | Описание бизнес-процесса, контекста, переход на русский язык | Olga Hilko | 08-Nov-2017 |  |  |
| 0.3 | Перевод на русский язык раздела | Olga Hilko | 10-Nov-2017 |  |  |
| 0.4 | Добавление второго зерна (раздел ), описание фактов (раздел ).Разделы , , | Olga Hilko | 15-Nov-2017 |  |  |
| 0.5 | Изменение структуры DWH, описание потока данных, исходных источников и логики загрузки. | Olga Hilko | 02-Dec-2017 |  |  |
|  |  |  |  |  |  |

Содержание

[1. Описание предметной области 4](#_Toc500072557)

[1.1. Миссия проекта / Бизнес-основа 4](#_Toc500072558)

[1.2. Словарь 4](#_Toc500072559)

[1.3. Недостатки существующих решений 4](#_Toc500072560)

[1.4. Преимущества от внедрения разрабоываемого хранилища данных 4](#_Toc500072561)

[2. Бизнес-контекст 4](#_Toc500072562)

[2.1. Описание бизнес-процесса 4](#_Toc500072563)

[2.2. Выбор зерна 5](#_Toc500072564)

[2.3. Выделение контекста 5](#_Toc500072565)

[2.4. Источники данных 5](#_Toc500072566)

[2.5. Описание факта 7](#_Toc500072567)

[3. Логическая схема 7](#_Toc500072568)

[3.1. Схема слоя BL\_3NF 7](#_Toc500072569)

[3.2. Схема слоя BL\_DM 8](#_Toc500072570)

[4. Поток загрузки данных (Data Flow) 8](#_Toc500072571)

[5. Стратегия партиционирования фактовой таблицы 8](#_Toc500072572)

[6. Отличительные черты 9](#_Toc500072573)

[7. Отчеты 9](#_Toc500072574)

[7.1. BL\_3NF 9](#_Toc500072575)

[7.2. DM\_BL 10](#_Toc500072576)

# Описание предметной области

## Миссия проекта / Бизнес-основа

Миссией проекта является проектирование и разработка промышленной базы данных для хранения и последующей обработки информации о произведенных платежах за реализуемое топливо и сопутствующие товары в сети АЗС «Беларуснефть».

## Словарь

|  |  |  |  |
| --- | --- | --- | --- |
| **Термин** | **Перевод** | **Сокра-щение** | **Description** |
| Хранилище данных | Data Warehouse | DWH (ХД) | Разрабатываемое хранилище данных на с использованием СУБД Oraсle 12с |
| Автозаправочная станция | Petrol Station / Station | АЗС | Точка продаж топлива и/или сопутствующих товаров |
| Валюта | Currency |  | Валюта, в которой была произведена оплата |
| Покупатель | Customer |  | Физическое или юридическое лицо, оплатившее отпущенные товары или топливо |
| Товар | Product |  | Сопутствующий товар, приобретаемый покупателем на АЗС |
| Топливо | Fuel |  | Тип топлива, отпускаемого на АЗС |
| Прейскурант | PriceList |  | Исторические данные о цене товаров и/или топлива, действовавшей в течение определенного периода. |

## Недостатки существующих решений

Каждая заправочная станция имеет свою собственную базу данных по продажам, что вызывает некоторые проблемы:

– низкая скорость обмена данными между отдельными заправочными станциями, головным офисом и бухгалтерией,

– трудности при сборе информации для ежемесячных, квартальных и годовых отчетов.

## Преимущества от внедрения разрабоываемого хранилища данных

Новая структура DWH позволит:

– сделать обмен данными между подразделениями и управляющими отделами более быстрым, надежным и гибким,

– консодилировать исходные данные из информаионных систем отдельных АЗС в одном хранилище,

– повысить производительность информационных систем, предоставляющих информацию для лиц, принимающих решения,

– ускорить обработку данных,

– повысить качество

Также внедрение DWH будет способствовать:

– улучшению качества обслуживания клиентов.

# Бизнес-контекст

## Описание бизнес-процесса

Сохраняемая в проектируемом хранилище информация описывает процесс приобретения и оплаты топлива и сопутствующих товаров на заправочных станциях "Беларуснефть". Основной задачей проекта является структуризация и консолидация данных о проведенных платежах.

Платеж (Payment) определенного типа (Payment Type) совершается клиентом (Customer) за один видов топлива (Fuel Type) на определенной заправочной станции (Station). Оплата производится в белорусских рублях. Отгрузка (отпуск) товаров и топлива осуществляется в момент оплаты, характеризуемый заданной датой (Date).

Клиент может быть как физическим лицом, так и юридическим и относится к одному из типов (Customer Type).

## Выбор зерна

Так как основным бизнес-процессом является прием платежа и сохранение информации о нем, то в роли базового зерна выступает факт оплаты отпущенного топлива на определенную дату на определенной заправочной станции.

## Выделение контекста

Дальнейшее использование данных о проведенных платежах предполагает их анализ в следующих разрезах:

– временном,

– по покупателям и их типам,

– по заправочным станциям,

– по типам топлива,

– по типам платежа.

Измерение **Station (STD1)**:

– ID (суррогатный первичный ключ),

– номер,

– адрес,

– географические координаты,

– номер(а) автодорог(и),

– перечень реализуемых видов топлива,

– флаг Current (является ли значение текущим),

– дата появления информации в системе,

– дата изменения (для текущих значений с флагом Current = true дата равна 01-01-9999).

Измерение **Customer (STD1)**:

– ID (суррогатный первичный ключ)

– тип (физическое или юридическое лицо):

для физического лица: имя, фамилия, отчество, дата рождения, пол, транспортное средство (модель, год выпуска),

для юридического лица: наименование юридического лица, УНП, наименование и код подразделения налоговой инспекции,

– адрес (страна, индекс, область/регион, город, улица, дом/строение/корпус, квартира/офис),

– дата появления информации в системе (только дата начала, или две даты + флаг Current).

Измерение **Date**:

– значение даты (естественный первичный ключ),

– год, квартал (номер в году, номер вмете с годом), месяц (номер и название на разных языках), номер недели в году, день в месяце,

– день недели (номер и название на разных языках).

Измерение **Payment Type (STD2)**:

– ID (суррогатный первичный ключ),

– код вида платежа (натуральный ключ),

– название вида платежа (полное и сокращенное).

Измерение **Fuel Type (STD2)**:

– ID (суррогатный первичный ключ),

– код вида топлива (натуральный ключ),

– название вида топлива (полное и сокращенное).

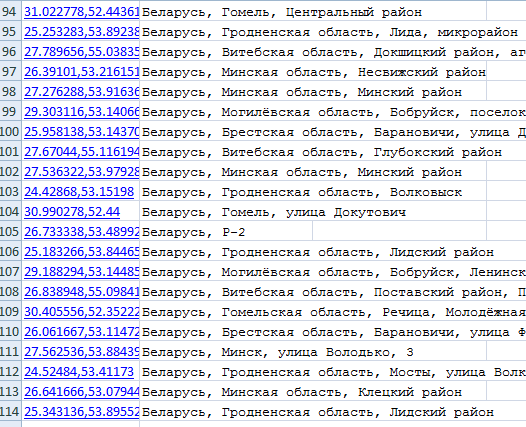
## Источники данных

|  |  |  |
| --- | --- | --- |
| **Сущность** | **Тип источника** | **Исходное расположение** |
| **Petrol Station / Station/AZS** | Excel | http://www.belorusneft.by/beloil-map/?lang=ru |
| **Customer** | Excel | Персональные личные данные сгенерированы по запросу (100 000 комбинаций), данные об организациях и ИП взяты с сайта Министерства по налогам и сборам |
| **SOATO** | Excel | Территориальное деление РБ, https://www.bygeo.ru/materialy/naselenie-belarusi/421-naselennye-punkty-belarusi-odnim-spiskom-skachat-v-formate-excel.html |
| **Fuel\_Type** | Excel | Только 7 видов, заполнялся вручную |
| **Payment\_Type** | Excel | Только 3 вида, заполнялся вручную |
| **Address** | Excel | Был сформирован по географическим координатам АЗС при помощи сервиса Яндекс (по примерам https://geocode-maps.yandex.ru/1.x/?geocode=27.563888,53.887222, см. рисунок ниже) |

Географические данные (Гео-данные из классификатора SOATO) фильтровались по дате обновления, проводилась фильтрация по уникальным кодам районов (GNI), для отдельных районов проводилась группировка с объединением (Минский р-н, Гомель). Для административных единиц республиканского и областного подчинения были введены фэйковые области и регионы с отрицательными кодами для сохранения связи с соответствующей областью.

Для связи адреса с кодами районов из Гео-данных при помощи сервиса Яндекс обратным геокодированием по географическим координатам были получены в виде JSON/XML-объектов адресные данные для каждой АЗС различной иерархии, что также представляло сложность для обработки. Выделялся конечный населенный пункт и связывался (если это было возможно) с локацией из Гео-данных, по которым определялся район. Если местом расположения АЗС являлась дорога, такая АЗС связывалась с фейковыми областями и регионами (в зависимости от подчинения: республиканского или районного)

При помощи указанного алгоритма для 536 АЗС из 545 имеющихся регион был определен.



Для генерации покупателей использовался сторонний сервис, формирующий используемого в качестве бизнес-ключа national ID, который был негарантированно уникальным. Для отсечения повторяющихся значений использовалась ранжирующая функция Dence\_rank с выбором первого по дате рождения.

В качестве бизнес-ключа для АЗС использовалось сочетание геокоординат, так как номер АЗС не являлся уникальным, а пользовательские данные неприкосновенны и изменять исходный источник нельзя.

## Описание факта

Так как основным бизнес-процессом является прием платежа и сохранение информации о нем, то в роли базового зерна выступает факт оплаты ()Payment) определенного количества приобретенного топлива на определенную дату.

В фактовую таблицу **Payments** включаются внешние ключи, соответствующие связанным измерениям :

– время платежа,

– тип платежа,

– тип приобретенного топлива,

– покупатель (если известен),

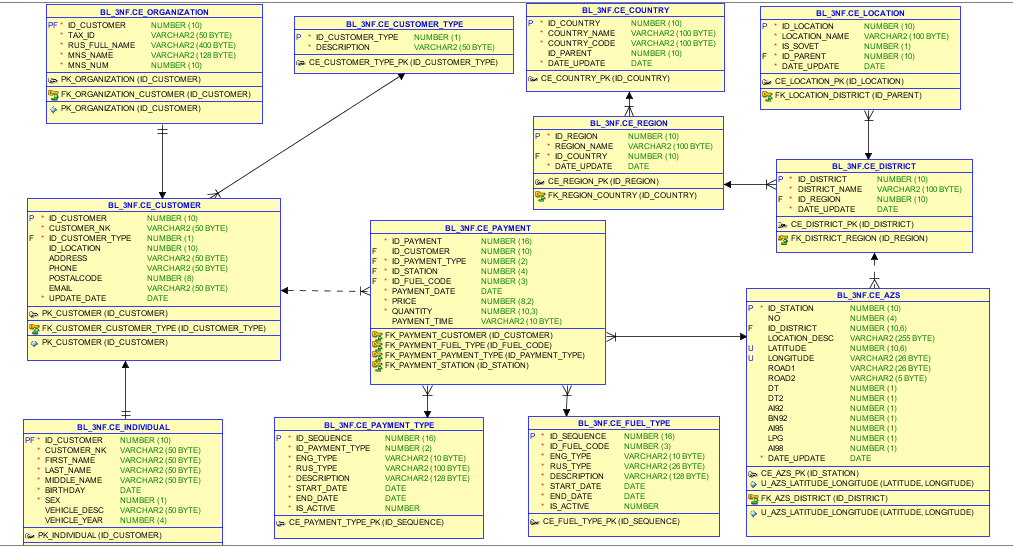
– заправочная станция.

В качестве мер используется уплаченная сумма и количество отпущенного топлива,

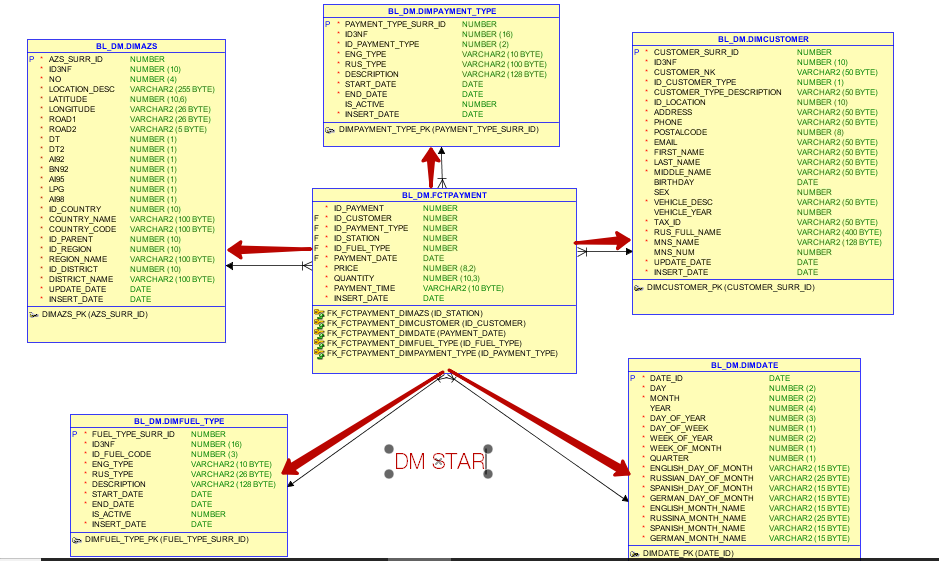
Дополнительны атрибутом, по которому не будет проводиться анализ, но данные которого необходимо сохранить, является время проведения оплаты.

# Логическая схема

## Схема слоя BL\_3NF



## Схема слоя BL\_DM



# Поток загрузки данных (Data Flow)



# Стратегия партиционирования фактовой таблицы

Партиционирование фактовой таблицы целесообразно производить по дате оплаты, причем:

– в первой партиции целесообразно размещать данные по продажам за все прошедшие годы, кроме текущего календарного и прошлого (на настоящим момент – до 01-01-2016)

– во второй – данные за прошедший год (с 01-01-2016 по 01-01-2017),

– в третьей – данные за текущий год (с 01-01-2017 по sysdate),

– в четвертой – данные будущих периодов (заготовка для переходного периода при наступлении нового календарного года).

# Отличительные черты

* Использование фреэймворка для повторяющихся операций в различных схемах (подсчет количества строк в целевой таблице, вывод сообщения о статусе проведенной операции, удаление таблиц и sequence, наделение правами
* Логирование ошибок для загрузки данных по странам, регионам, районам и локациям (некорректные данные, не удовлетворяющие условиям соответствующих таблиц записываются в специализированные файлы на диске
* Подсчет количества вставляемых строк для практически каждой таблицы CL\_, CE\_ и DM (рассчитывалось путем разности количества строк в целевой таблице до и после вставки, для чего использовались функции собственного фреймворка)
* Использование sql, dynamic sql, pl/sql практически во всех пакетах.
* Использование по два DIMENTION типов STD2 и STD1.

# Отчеты

## BL\_3NF

select ct.DESCRIPTION "Тип покупателя",

case when r.Region\_name is null then 'Республика Беларусь' else r.Region\_name end "Pегион",

ft.RUS\_TYPE "Тип топлива", trunc(sum(p.price)) "Выручка"

from BL\_3NF.CE\_PAYMENT p

left join BL\_3NF.CE\_PAYMENT\_Type pt on pt.ID\_SEQUENCE= p.ID\_PAyment\_type

left join BL\_3NF.CE\_CUSTOMER c ON c.ID\_CUSTOMER=p.ID\_CUSTOMER

left JOIN BL\_3NF.CE\_CUSTOMER\_TYPE ct on ct.ID\_CUSTOMER\_TYPE=c.ID\_CUSTOMER\_TYPE

left join BL\_3NF.CE\_FUEL\_Type ft on ft.ID\_SEQUENCE=p.ID\_FUEL\_CODE

left join BL\_3NF.CE\_AZS a on a.ID\_STATION=p.ID\_STATION

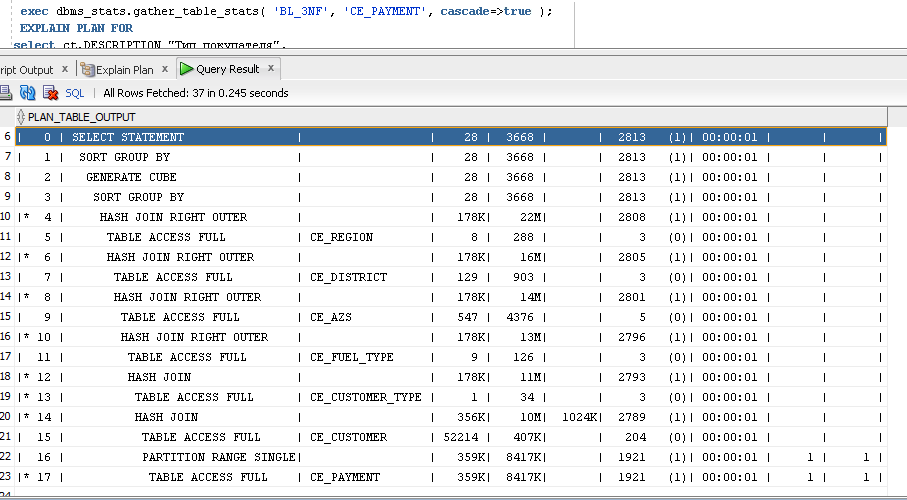
left join BL\_3NF.CE\_DISTRICT d on d.ID\_DISTRICT = a.ID\_DISTRICT left JOIN BL\_3NF.CE\_REGION r ON r.id\_region = d.id\_region

where p.PAYMENT\_DATE between to\_date('01/01/2005','mm/dd/yyyy') and to\_date('01/01/2015','mm/dd/yyyy')

and ct.DESCRIPTION='Юридические лица'

GROUP BY cube(ct.DESCRIPTION, ft.RUS\_TYPE , case when r.Region\_name is null then 'Республика Беларусь' else r.Region\_name end)

order by 1,2,3

;

## DM\_BL

select

c.CUSTOMER\_TYPE\_DESCRIPTION "Тип покупателя", a.Region\_name "Назв. региона",

ft.RUS\_TYPE "Тип топлива", trunc(sum(p.price)) "Выручка"

from BL\_DM.FCTPAYMENT p

inner join BL\_DM.DIMPAYMENT\_Type pt on pt.PAYMENT\_TYPE\_SURR\_ID=p.ID\_PAYMENT\_TYPE

inner join BL\_DM.DIMCUSTOMER c ON c.CUSTOMER\_SURR\_ID=p.ID\_CUSTOMER

inner join BL\_DM.DIMFUEL\_Type ft on ft.FUEL\_TYPE\_SURR\_ID=p.ID\_FUEL\_TYPE

inner join BL\_DM.DIMAZS a on a.AZS\_SURR\_ID=p.ID\_STATION

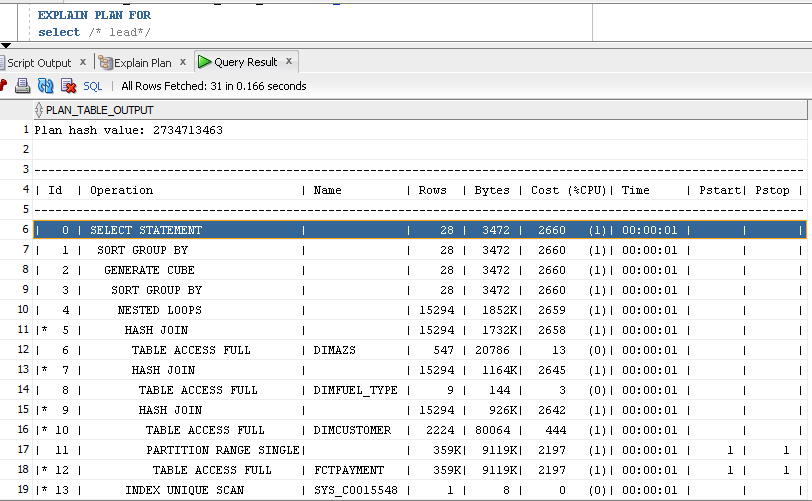
inner join BL\_DM.DIMDATE d on trunc(p.PAYMENT\_DATE)=d.DATE\_ID

where p.PAYMENT\_DATE between to\_date('01/01/2005','mm/dd/yyyy') and to\_date('01/01/2015','mm/dd/yyyy')

and c.CUSTOMER\_TYPE\_DESCRIPTION='Юридические лица'

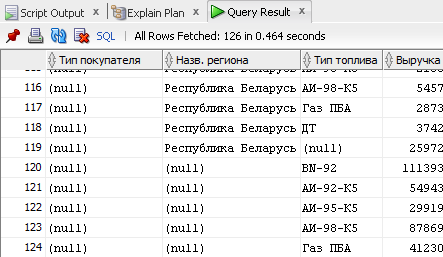
GROUP BY cube(c.CUSTOMER\_TYPE\_DESCRIPTION, ft.RUS\_TYPE ,a.Region\_name)

order by 1,2,3;



Cost: DM 2660 против 3NF 2813; и время выполнения DM 0.464сек против 3NF 0,521 сек. на фактовой таблице в 1 млн записей.

DM



3NF