|  |
| --- |
| EPAM Systems, RD Dep. |
| Partitioning |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | Valeryia\_Lupanava | 13-NOV-2017 |  |  |

Содержание

[1. Практическое задание 3](#_Toc498389563)

[1.1. Партиционирование по диапазону (Partition by Range) 3](#_Toc498389564)

[1.2. Партиционирование по хэш-функции (Partition by Hash) 8](#_Toc498389565)

[1.3. Партиционирование по списку (Partition by List) 11](#_Toc498389566)

[1.4. Слияние партиций (Coalescing Partition) 15](#_Toc498389567)

[1.5. Удаление партиции (Dropping Partition) 16](#_Toc498389568)

[1.6. Объединение партиций (Merging Partition) 16](#_Toc498389569)

[1.7. Перемещение партиций (Moving Partition) 17](#_Toc498389570)

[1.8. Разделение партиции (Splitting Partition) 18](#_Toc498389571)

[1.9. Удаление данных из партиции (Truncating Partition) 19](#_Toc498389572)

[1.10. Анализ партиций (Partition Analysis) 21](#_Toc498389573)

[2. Аналитическое задание 22](#_Toc498389574)

[2.1. Партиционирование 22](#_Toc498389575)

[2.2. Бизнес-анализ 23](#_Toc498389576)

[2.2.1. Анализ продаж 24](#_Toc498389577)

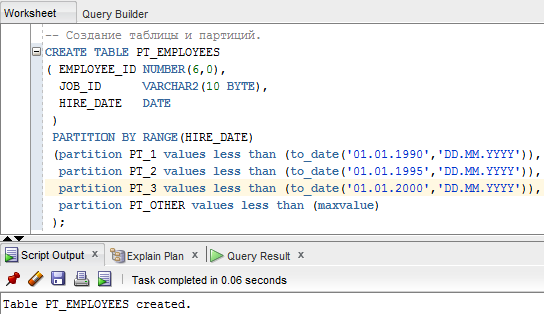
[2.2.2. Анализ чеков 25](#_Toc498389578)

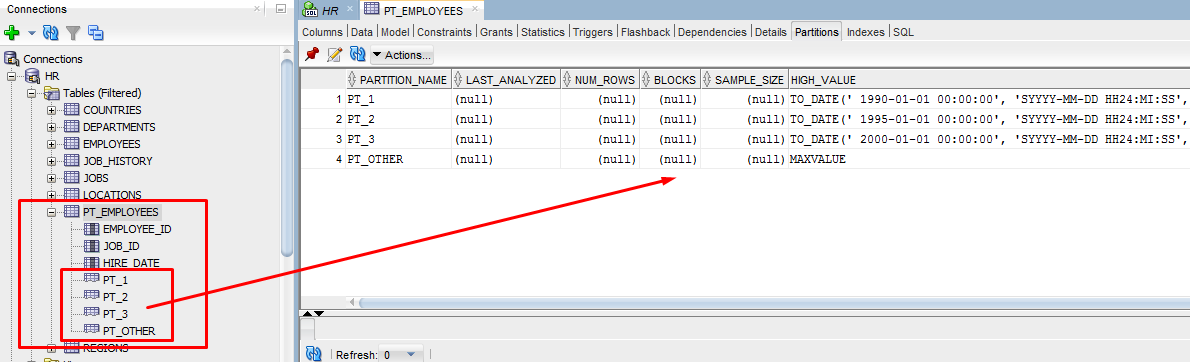
[2.2.3. Анализ нагрузки 26](#_Toc498389579)

# Практическое задание

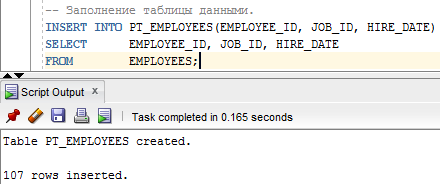
## Партиционирование по диапазону (Partition by Range)

* Создание партиции по диапазону значений.

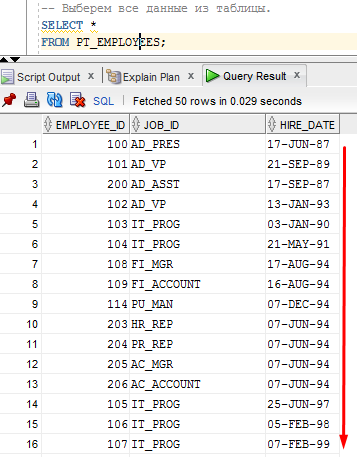




* Заполняем таблицу данными.

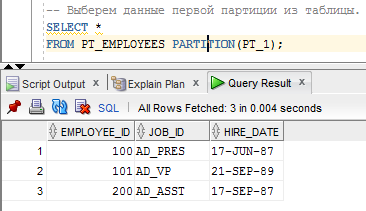


* Выберем данные из таблицы для проверки.
  + Все данные.



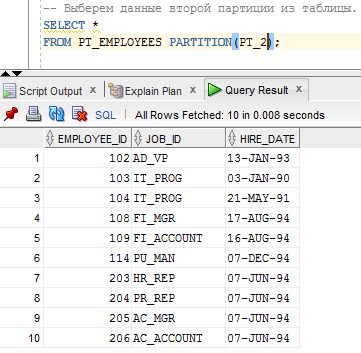
**Из выполнения видно, что все года учавствуют в выборке.**

* + Первая партиция – до 1990 года.

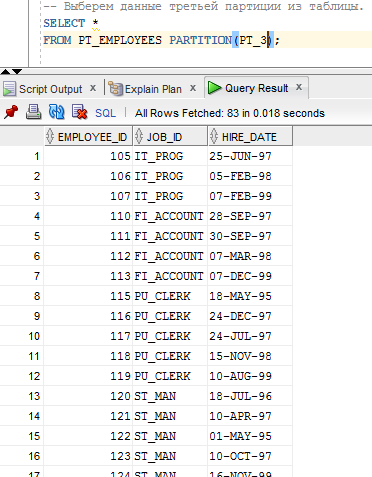


**В результате выполнения БД вывела только строки с датой до 1990 года.**

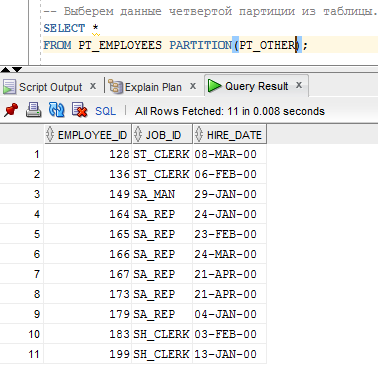
* + Вторая партиция – до 1995 года.



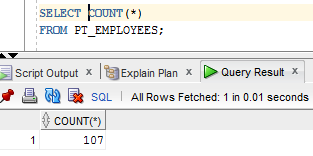
* + Третья партиция – до 2000 года.



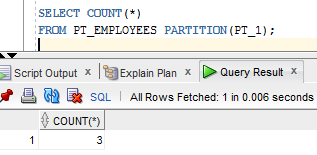
* + Четвертая партиция – после 2000 года.



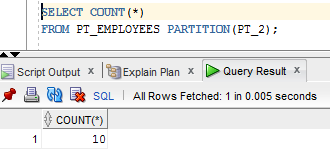
* + Проверим правильность, просуммировав количество полученных значений с количеством значений в таблице.
    - Все данные.



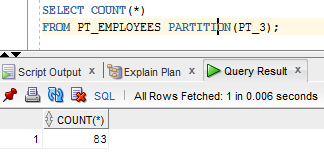
* + - Данные по партициям в порядке возрастания номера партиции.
    - Партиция с данными до 1990 года.



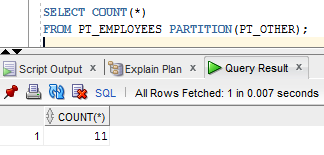
* + - Партиция с данными до 1995 года.



* + - Партиция с данными до 2000 года.



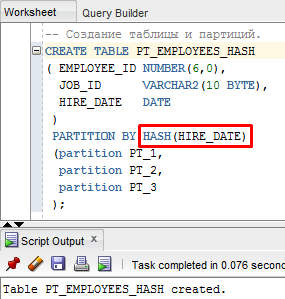
* + - Партиция с данными после 2000 года.

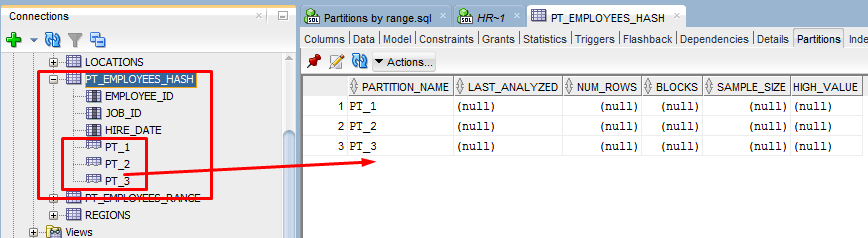


**Можно посчитать полученные результаты: 3 + 10 + 83 + 11 = 107. Все верно**.

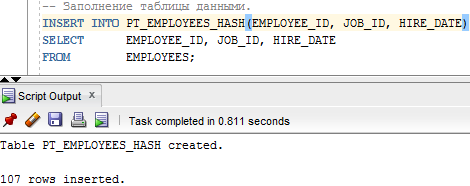
## Партиционирование по хэш-функции (Partition by Hash)

* Создадим таблицу с фрагментацией по хэш-функции.

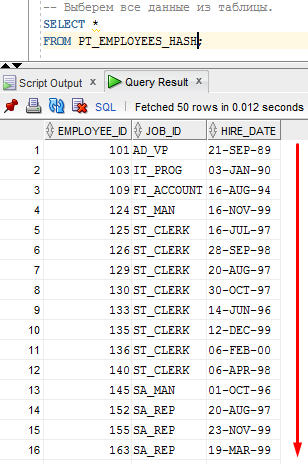
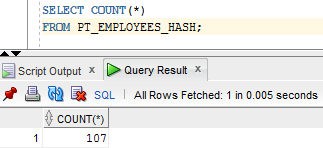




* Заполним данными.

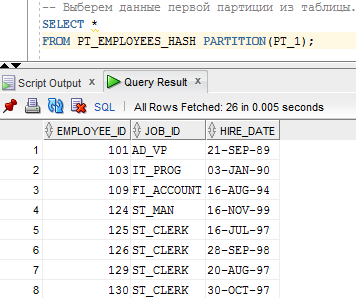
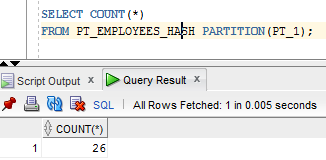


* Выберем данные из таблицы для проверки.
  + Все данные.

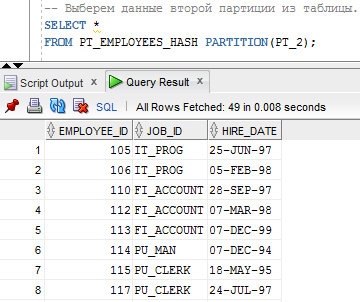
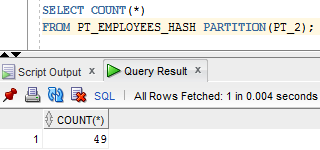


**Из выполнения видно, что все данные учавствуют в выборке.**

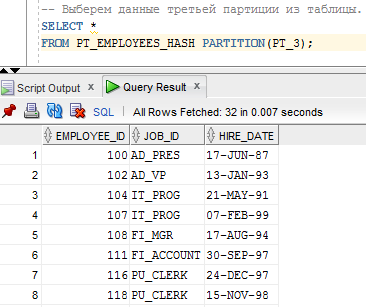
* + Первая партиция.

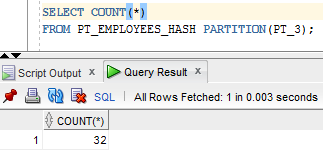


* + Вторая партиция.



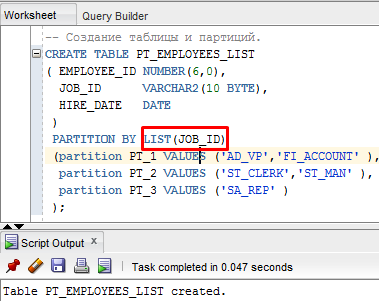
* + Третья партиция.

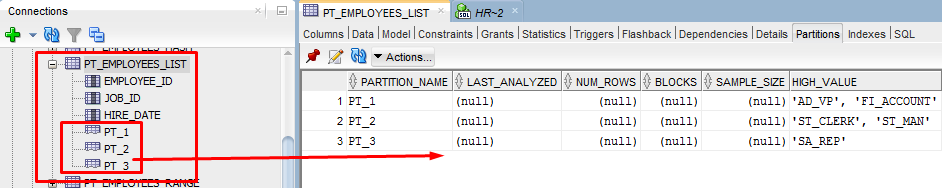


* + 
  + Проверим правильность, просуммировав количество полученных значений с количеством значений в таблице. **Можно посчитать полученные результаты: 26 + 49 + 32 = 107. Все верно.**

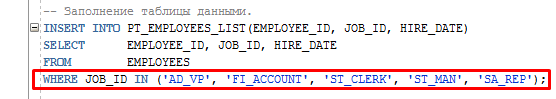
## Партиционирование по списку (Partition by List)

* Создадим таблицу с использованием списка значений.

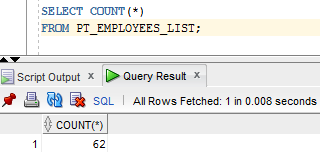
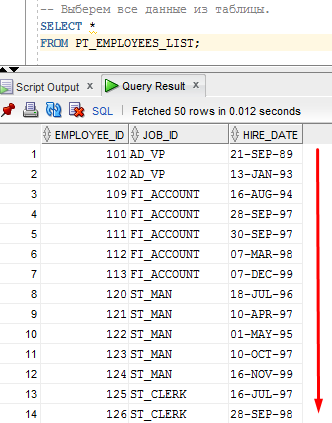




* Заполним данными. **При заполнении необходимо указывать, какие значения для заполнения партиций БД необходимо использовать при вставке, поскольку при создании партиции было явно указано, какие значения какой партиции принадлежат.**

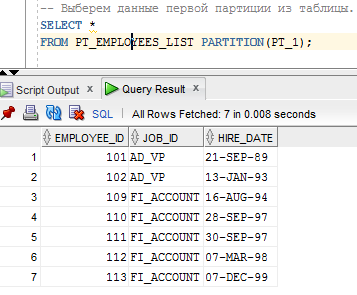
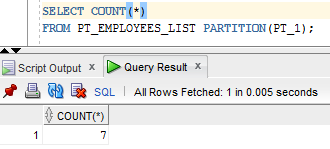


* Выберем данные из таблицы для проверки.
  + Все данные.

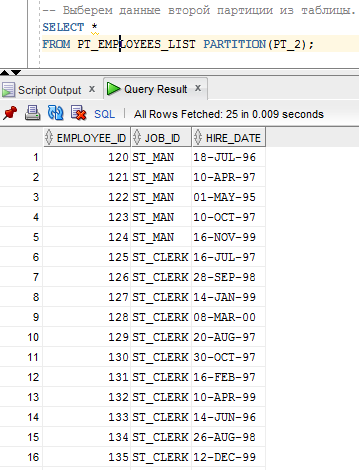
 

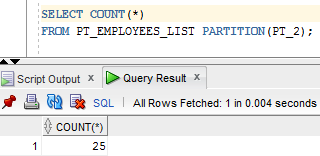
**Из выполнения видно, что все данные учавствуют в выборке.**

* + Первая партиция.

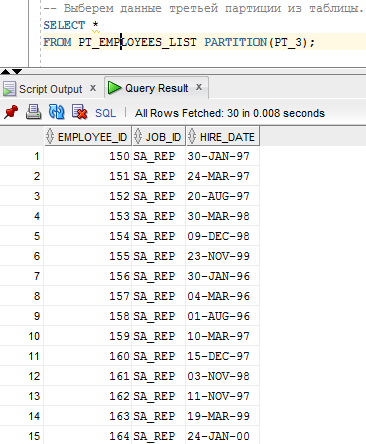
 

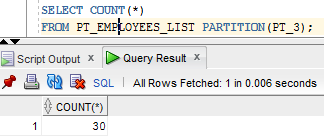
* + Вторая партиция.





* + Третья партиция.





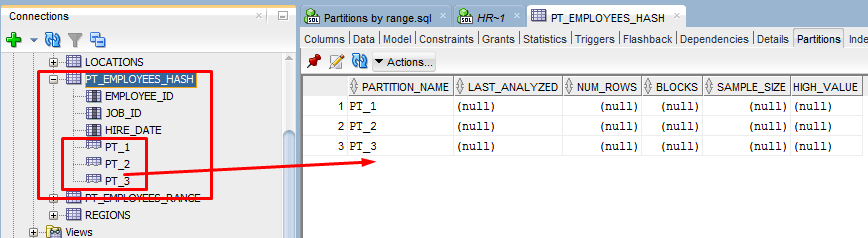
* + Проверим правильность, просуммировав количество полученных значений с количеством значений в таблице. **Можно посчитать полученные результаты: 7 + 25 + 30 = 62. Все верно.**

**Выводы:** применяя партиционирование в больших таблицах с несколькими миллионами записей, всегда есть возможность освободить часть дискового пространства и перенести неиспользуемые данные на архивный носитель.

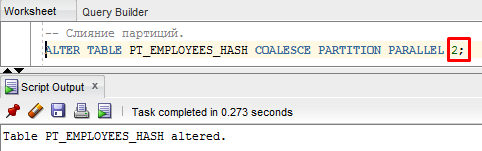
**LIST PARTITION** удобен, если нужно выделить для партиции конкретные значения. **RANGE PARTITION** удобен для работы с диапазонами данных. **HASH PARTITION** означает равномерное распределение строк таблицы по назначенному табличному пространству в зависимости от значения поля партиционирования, которое в данном случае будет хэшироваться. Этот вид удобно применять для строк, у которых распределение значений поля партиционирования неравномерно или плохо группируется.

## Слияние партиций (Coalescing Partition)

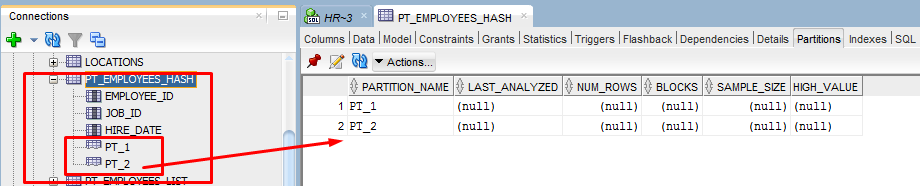
* До применения команды таблица содержит три партиции.



* Выполняем слияние партиций. Итоговое количество партиций должно быть равно двум, как мы указали.



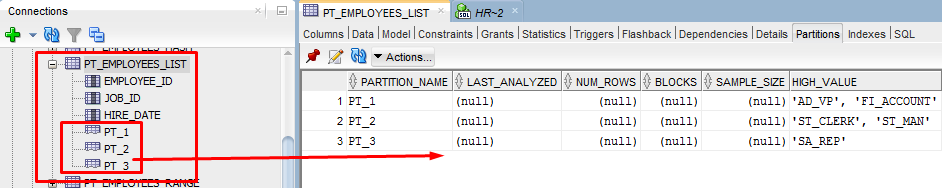
* После применения команды таблица содержит две партиции.



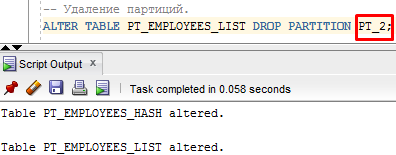
* Применение: **COALESCE PARTITION** используется в хэш-партиционировании для изменения количества партиций, т.е. разделения общего количества значение на меньшее или большее количество партиций.

## Удаление партиции (Dropping Partition)

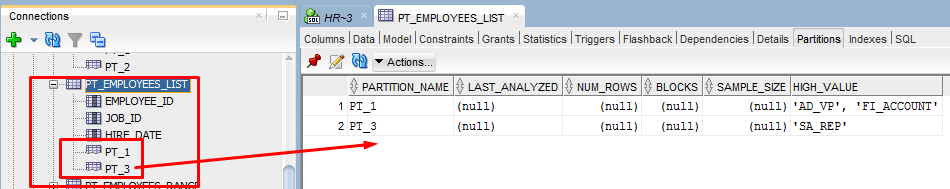
* До применения команды таблица содержит три партиции.



* Применим команду удаления партиций.

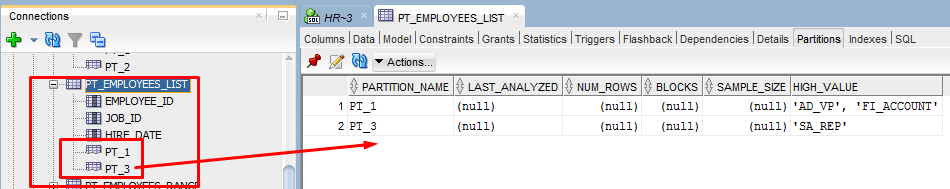


* В результате получим две партиции. Значения PT\_2 не разделяются меджу другими партициями, как в случае слияния, а просто удаляются.

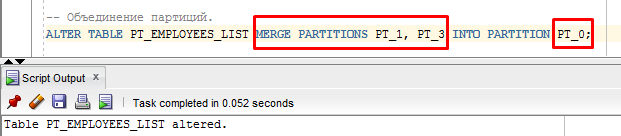


## Объединение партиций (Merging Partition)

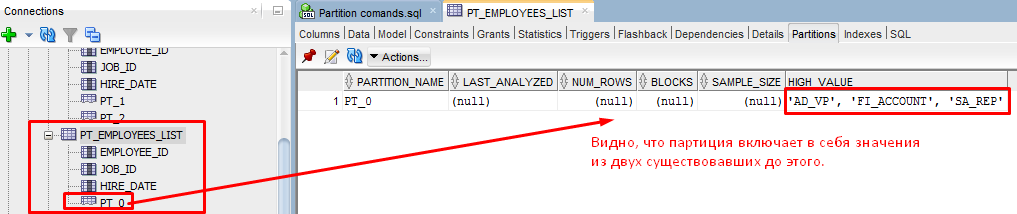
* До применения команды таблица содержит две партиции.



* Применим команду объединения партиций – **MERGE**. При применении этой команды указанные в команде партиции объединяются в одну новую и автоматически удаляются. В синтаксисе можно также не просто перечислить партиции, а указать диапазон: **MERGE PARTITIONS PT\_1 TO PT\_3.**

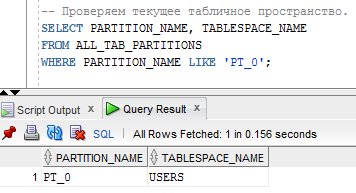


* В результате получим одну партицию PT\_0.

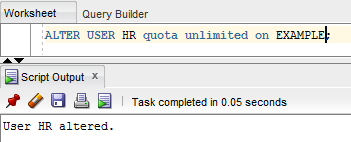


## Перемещение партиций (Moving Partition)

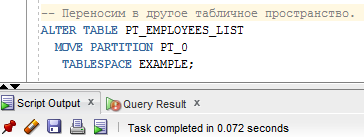
* До применения команды партиция находится в табличном пространстве USERS.



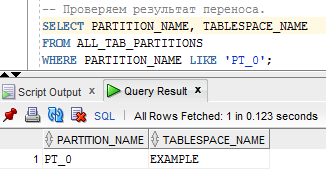
* Дадим пользователю грант на другое табличное пространство, в которое будет осуществлен перенос партиции.



* Перенесем партицию PT\_0 в табличное пространство EXAMPLE.

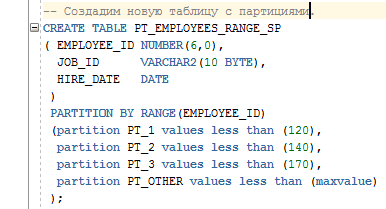


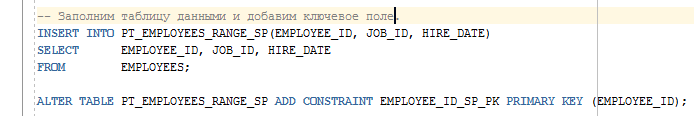
* Проверяем результат.



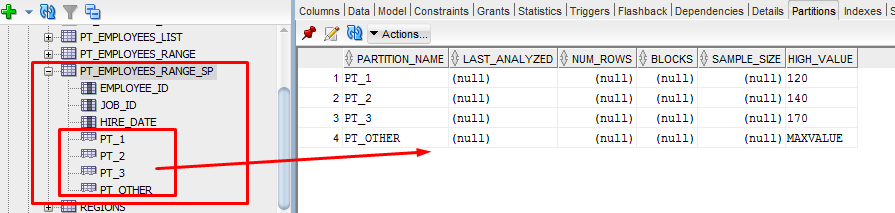
## Разделение партиции (Splitting Partition)

* Создадим новую таблицу с партициями. Партиции сделаем по полю с первичным ключом EMPLOYEE\_ID.

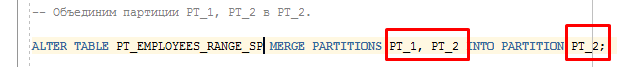


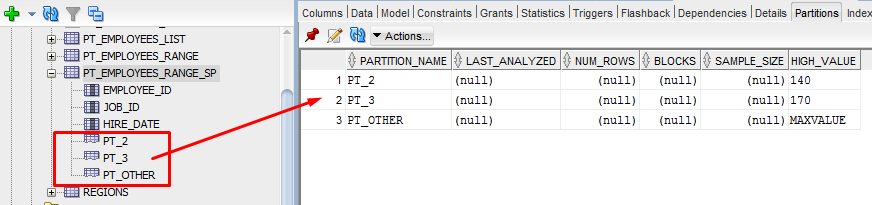


* Проверим полученный результат.

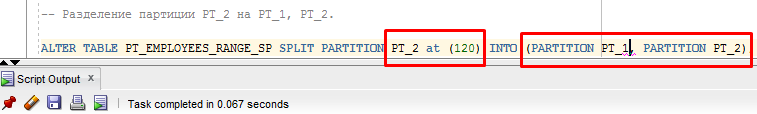


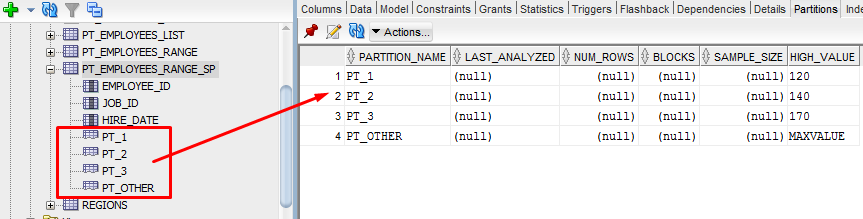
* Теперь объединим две партиции PT\_1 и PT\_2 в PT\_2.





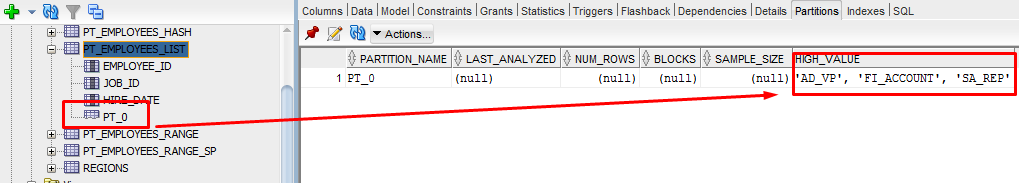
* Теперь проведем обратное действие и разделим полученную партицию PT\_2 на две: PT\_1 и PT\_2.



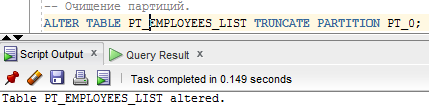


## Удаление данных из партиции (Truncating Partition)

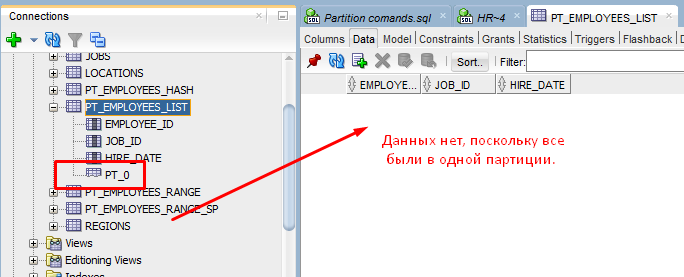
* До применения команды таблица содержит одну партицию с некоторыми значениями.



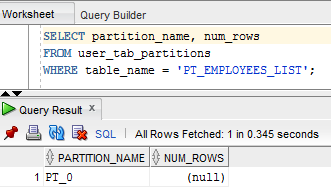
* Применим команду TRUNCATE.



* В результате получим данные без данных партиции PT\_0.

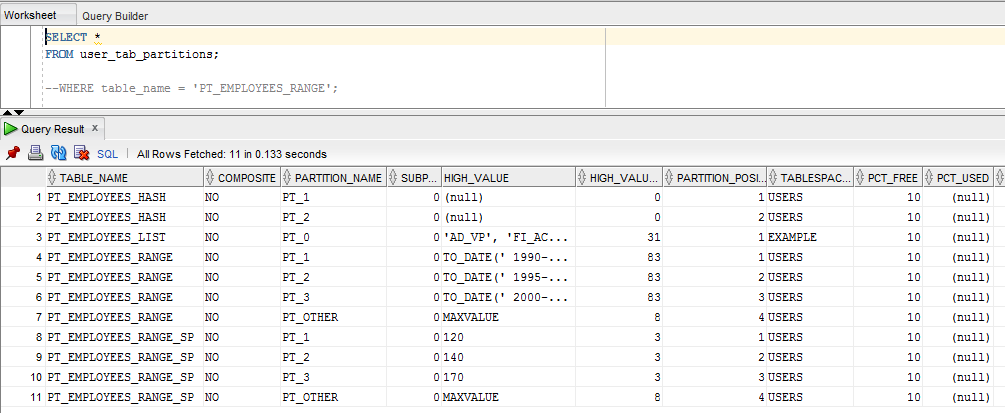


* Также можно сделать проверку с помощью команды, описанной ниже.



## Анализ партиций (Partition Analysis)

* Также можно в таблице **USER\_TAB\_PARTITIONS** можно посмотреть аналитику по всем партициям.



**Скрипты по выполненным выше командам вложены в папку.**

# Аналитическое задание

## Партиционирование

Для определения таблицы с композитным секционированием используется комбинация двух методов распределения данных. Сначала таблица секционируется одним методом распределения данных. Затем каждая секция подразделяется на подсекции с использованием другого метода распределения данных. Например, если для таблицы задано композитное секционирование range-list (диапазон-список), то такая таблица сначала секционируется по диапазонам ключей, а затем каждая секция делится на подсекции по спискам, указанным во второй партиции. Секции таблицы с композитным секционированием существуют только как метаданные и не предусматривают фактического хранения данных: подсекции конкретной секции таблицы или индекса с композитным секционированием представляют собой физические сегменты базы данных, где хранятся данные секции.

Среди доступных методов партиционирования в работе будет применен вышеописанный **RANGE-LIST** (диапазон –список) для таблицы фактов **FACT\_RETAIL\_SALES**.

Пример скрипта:

**CREATE TABLE fact\_retail\_sales (**

**sale\_id NUMBER(10) NOT NULL,**

**receipt\_id NUMBER(10) NOT NULL,**

**date\_id NUMBER(10) NOT NULL,**

**product\_details\_id NUMBER(10) NOT NULL,**

**employee\_id NUMBER(10) NOT NULL,**

**customer\_id NUMBER(10),**

**store\_id NUMBER(10) NOT NULL,**

**payment\_method\_id NUMBER(10) NOT NULL,**

**promotion\_id NUMBER(10),**

**currency\_id NUMBER(10) NOT NULL,**

**sale\_sum NUMBER(10) NOT NULL,**

**sale\_amount NUMBER(10) NOT NULL**

**)**

**PARTITION BY RANGE (date\_id) INTERVAL (NUMTOYMINTERVAL(3,'MONTH'))**

**SUBPARTITION BY LIST (store\_id) SUBPARTITION TEMPLATE**

**( SUBPARTITION local VALUES('...'),**

**SUBPARTITION Russia VALUES ('...'),**

**SUBPARTITION Europe VALUES ('...'),**

**SUBPARTITION Asia VALUES ('I')**

**)**

**(PARTITION half\_year\_2017 VALUES LESS THAN (TO\_DATE('01-JAN-2017','dd-MON-yyyy'))**

**)**

**PARALLEL;**

Данный метод партиционирования наиболее удобен для данной таблицы фактов, поскольку она будет разделена на пиртиции по декадам, а каждая декада, в свою очередь будет содержать партиции с магазинами в регионах, где ведутся продажи. Т.к. сеть продаж достаточно обширная, гранулярность в партициях до уровня магазинов нецелесообразна. Для бизнеса будет гораздо удобнее отслеживание продаж в рамках конкретного региона.

Для таблицы фактов **FACT\_PRODUCT\_BALANCES** будет применена аналогичная стратегия партиционирования, поскольку в данной таблице будут анализироваться товарные запасы.

## Бизнес-анализ

В работе будут исследоваться два бизнес-процесса. Первый из них – месячные продажи магазинов в рамках регионов, стран и городов, а также продажи по клиентам, второй – месячные товарные запасы магазинов в различных географических разрезах и по менеджерам сети, для выявления наиболее ценных кадров.

В качестве отчетов предлагается три документа:

* **Анализ продаж.**

Данная таблица будет предназначена для предоставления общей информации о функционирующих магазинах, продажах, количестве товаров в наличии, а также о количестве чеков. Низший уровень гранулярности будет представлен регионами. Затем по технологии DRILL-DOWN можно будет углубить детализацию аналитики до стран и городов. Самый высший уровень иерархии будет предстанен магазинами с конкретным адресом в городе.

В таблице будет представлено описание следующих показателей: оборот за предыдущий период - количество месяцев предыдущего периода можно будет определить, указав соответствующее значение в поле "Период сравнения", текущий период можно будет определить в поле "Период анализа". Показатель процент выполнения покажет, какой процент выручки прошлого периода достигнут за текущий период. Показатель количество товаров определяет товарные остаки, а показатель количество чеков демонстрирует, какое количество продаж было совершено за анализируемый период.

* **Анализ чеков.**

Данная вкладка позволит пользователю посмотреть основные бизнес-показатели по метрикам, выбранным в ListBox. В каждом ListBox возможен выбор нескольких значений одновременно. Период анализа можно определить по аналогии с предыдущей вкладкой. Выбор месяца и года предоставляется с помощью ListBox.

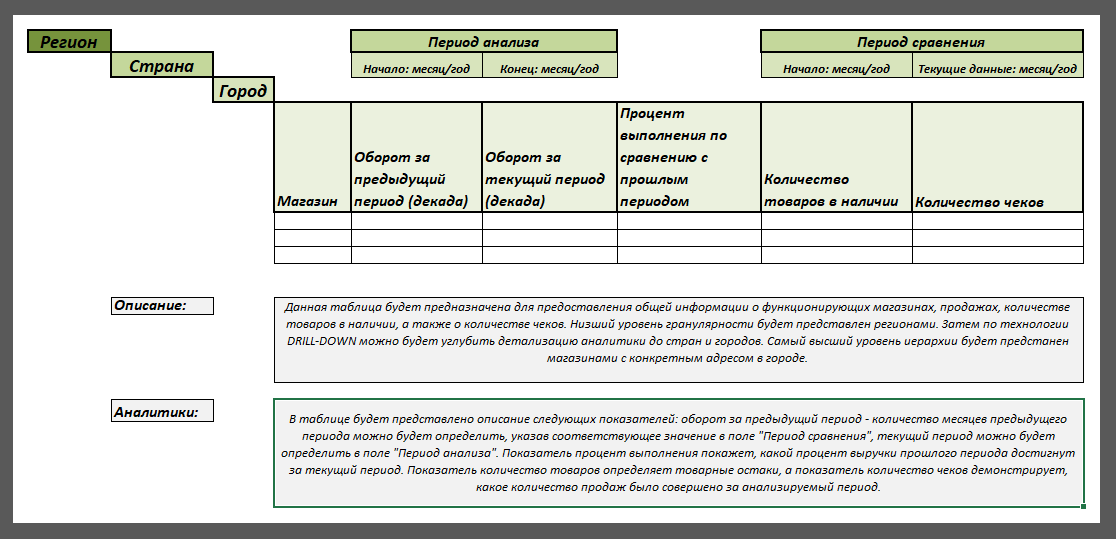
* **Анализ нагрузки.**

Данный анализ пирменяется для исследования наплыва покупателей в рабочие дни и рабочие часы выбранного для анализа дня. Этот анализ позволяет увидеть наиболее благоприятные для проведения акций рабочие периоды, а также время, когда необходимо присутсвие большего количества персонала на местах.

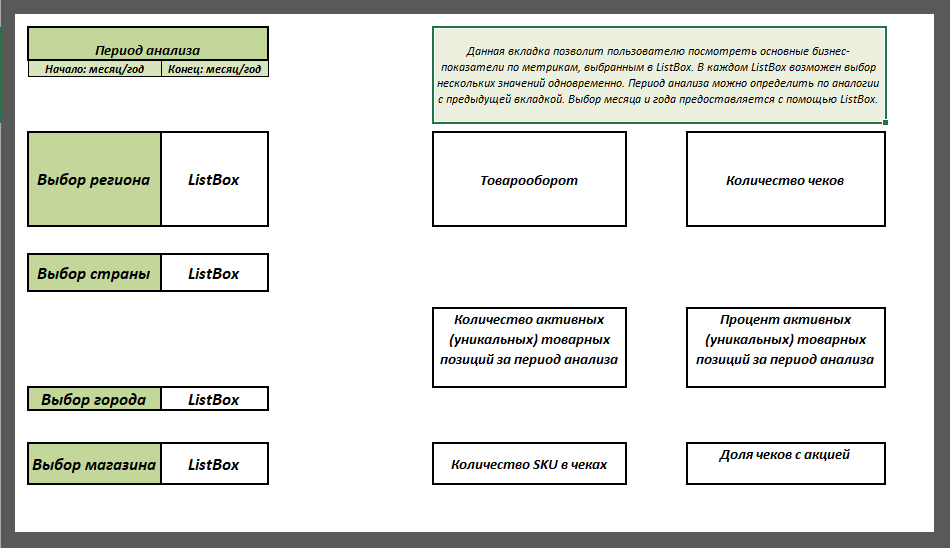
* **Сгенерированные данные.**

Эта вкладка содержит сгенерированные таблицы для построения графиков в отчете «Анализ нагрузки».

### Анализ продаж



### Анализ чеков



### Анализ нагрузки

