|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра прикладной математики (ПМ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Системы управления данными»

**Практическое занятие № 4**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИМБО-02-22, Ким Кирилл Сергеевич* | (подпись) | |
| Преподаватель | *Алексеева Екатерина Сергеевна, преподаватель* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_202\_\_г. | |  | |

Москва 2025 г.

1. Практическая работа №4
   1. Цель работы

В этой лабораторной работе вы создадите таблицу для данных об автомобиле, разделенную по номеру детали. Вы уже создали данные с помощью службы pig в предыдущей лабораторной работе. Мы можем запросить данные об автомобиле в той же таблице, но различать их по номерам деталей. Это делается путем создания раздела для каждой записи.

## Создание и загрузка статического раздела в таблице

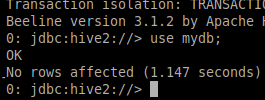
В окне терминала,

### Запустите Beeline.

$ beeline –u jdbc:hive2://

### Используйте базу данный mydb для лабораторной.

0: jdbc:hive2://> use mydb;



### Убедитесь, что обрабатываемые данные находятся в следующем HDFS.

$ hdfs dfs –ls –R /mywarehouse/car[12]

###### 텍스트, 키보드, 닫기이(가) 표시된 사진 자동 생성된 설명

### Создайте таблицу с названием cars с помощью следующей схемы:

|  |  |
| --- | --- |
| Column Name | Column Type |
| Code | Int |
| brand | string |
| l\_date | string |
| model | string |
| country | string |
| price | int |

#### Столбец раздела – part\_num (тип tinyint), разделитель поля – ‘\t’

#### Расположение таблицы – /mywarehouse/car с управляемым типом.﻿

0: jdbc:hive2://> ﻿create table if not exists cars (

code int,

brand string,

model string,

r\_date string,

country string,

price int

)

PARTITIONED BY (part\_num tinyint)

row format delimited

fields terminated by ‘\t’

location ‘/mywarehouse/cars’;

###### 

###### 

#### Измените таблицу cars добавив два раздела. Один раздел part 1, а другой part 2.

﻿0: jdbc:hive2://> alter table cars add partition (part\_num=1);

﻿0: jdbc:hive2://> alter table cars add partition (part\_num=2);

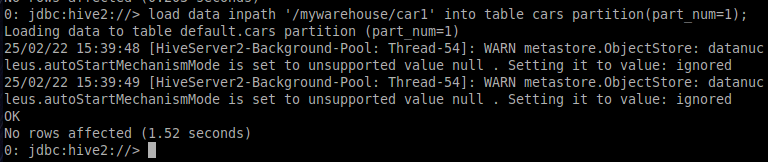
###### 

###### 

#### Загрузите данные из /mywarehouse/car1 в раздел part 1, и загрузите данные из /mywarehouse/car2 в раздел part 2.

﻿0: jdbc:hive2://> load data inpath ‘/mywarehouse/car1’ into table cars partition(part\_num=1);

Когда команда load выполнена, файл содержащийся в car1 перемещен в cars/part\_num=1 folder.



#### Проверьте папку cars.

###### 

#### Выведите данные cars с помощью оператора select

0: jdbc:hive2://> ﻿select code, brand, model, r\_date, price from cars;

###### 텍스트이(가) 표시된 사진 자동 생성된 설명

Примечание: Сообщение предупреждения можно проигнорировать.

#### Переместите данные в car2 в part\_num=2 с помощью hdfs mv вместо команды load и проверьте, чтобы данные выглядели также как в 1.4.6.

###### 

0: jdbc:hive2://> select code, brand, model, r\_date, price from cars;

###### 

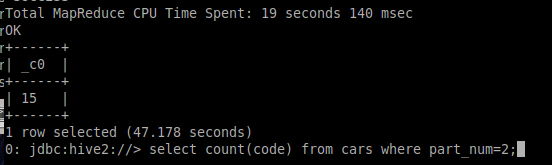
Примечание: Мы проверили все данные, включая part\_num=2.

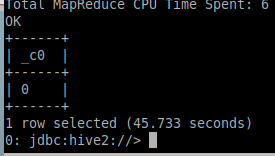
#### Убедитесь, что данные для обоих номеров деталей были правильно загружены, подсчитав записи для каждого:

﻿0: jdbc:hive2://> select count(code) from cars where part\_num=1;

﻿0: jdbc:hive2://> select count(code) from cars where part\_num=2;

Вопрос: Каково количество записей в каждом разделе?



19 (18)

## Работа с таблицей в hue.

Попрактикуемся в выполнении запросов в Hue.

### Запустите HUE:

$ sudo service hue start

### 

### В поле firefox url, введите <http://localhost:8888>. Id – student, password – student. Нажмите “Sign in” для входа в hue.

###### 

### Нажмите на Tables в левой части панели инструментов.

### Перейдите в базу данных mydb и откройте там таблицу cars:

###### 

### Запустите выполненную в beeline команду в редакторе hue и проверьте результат.

#### Для этого перейдите на страницу редактора выбрав соответствующий значок на панели слева:

#### 

#### Выполните запрос: select code, brand, model, r\_date, price from mydb.cars;

###### 

#### Теперь выполните: select count(code) from mydb.cars where part\_num=1;

###### 텍스트이(가) 표시된 사진 자동 생성된 설명

#### select count(code) as CNT from mydb.cars where part\_num=2;

###### 

#### Вы можете проверить информацию о схеме и содержимое выборки данных, нажав кнопку “i” рядом с таблицей cars в левом меню.

###### 테이블이(가) 표시된 사진 자동 생성된 설명

### Создайте запросы по cars для получения следующей информации.

#### Всю информацию о машинах модели TESLA.

#### 

#### Код, модель и марка всех машин в cars.

#### 

#### Названия марок в алфавитном порядке.

#### 

#### Марка, модель, дата выпуска и цена, упорядоченные по цене, по убыванию.

#### 

#### То же, что и выше, но в порядке возрастания.

#### 

#### Марки машин с ценой менее $2000.00.

#### 

#### Марка машин с ценой от $3000.00 до $4000.00.

#### 

#### Всею информацию о машинах с названиями моделей MINI, BWM, Porsche.

#### 

#### Назовите марку с самой низкой ценой.

#### 

#### Назовите вторую по дороговизне марку.

#### 

#### Сколько уникальных марок (brand) представлено в таблице?

#### 

#### Какова средняя цена машин по маркам (игнорируйте записи, в которых цена равна NULL)?

#### 

# По работе со сложными типами данных

В данной лабораторной работе вы самостоятельно выполните операции со сложными типами данных и операцию объединения. Работа со всеми таблицами должна производиться через Hive. Используйте Hue или beeline на ваше усмотрение.

## Создайте таблицу continents, в которой будет храниться информация о континентах.

### Создайте столбец id установите его в качестве уникального ключа

### Создайте continent\_name чтобы сохранить названия континентов

### 

### Добавьте следующие континенты в указанном порядке: Africa Asia Europe North America South America

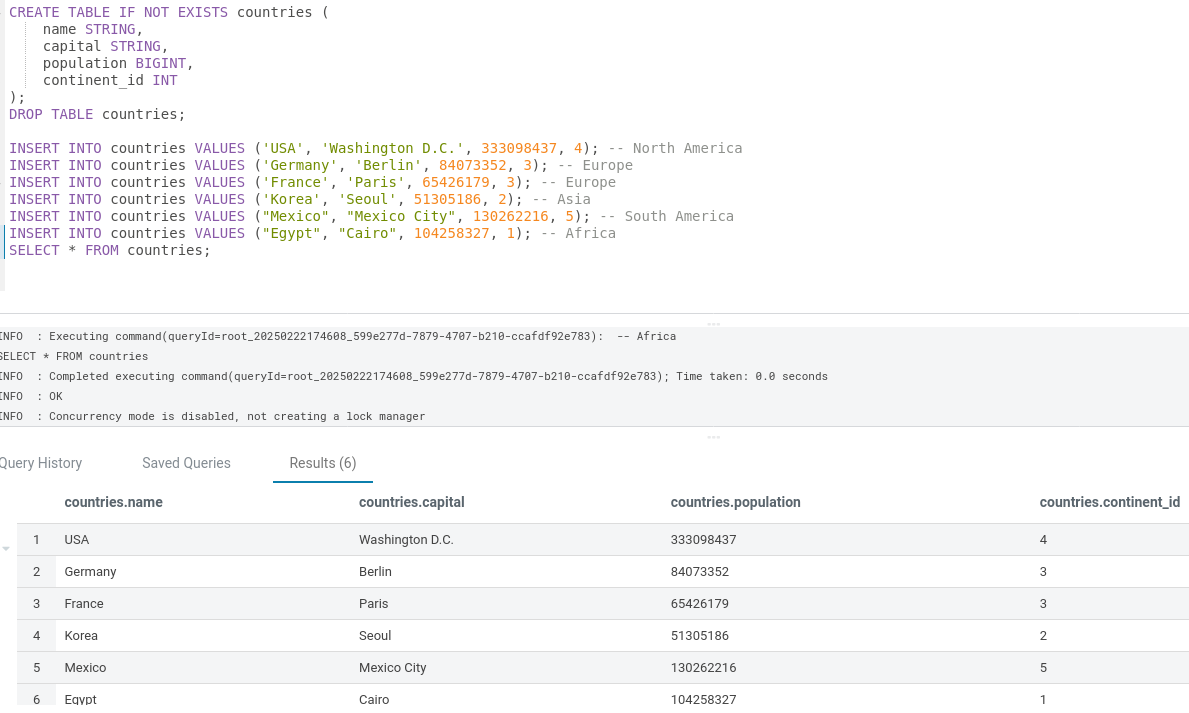
### 

## Создайте таблицу countries.

#### Таблица должна иметь следующие столбцы: name (название страны), capital (название столицы), population (население), continent\_id (id континента, где страна находится).

#### Добавьте следующую информацию в таблицу countries (в столбце continent\_id должны быть id из таблицы continent):

|  |  |  |  |
| --- | --- | --- | --- |
| name | capital | population | continent\_id |
| USA | Washington D.C. | 333,098,437 | соответствует North America |
| Germany | Berlin | 84,073,352 | соответствует Europe |
| France | Paris | 65,426,179 | соответствует Europe |
| Korea | Seoul | 51,305,186 | соответствует Asia |
| Mexico | Mexico City | 130,262,216 | соответствует South America |
| Egypt | Cairo | 104,258,327 | соответствует Africa |



Представим, что наша база данных используется для хранения информации о заказах клиентов в ресторане, имеющем филиалы по всей стране. Создадим необходимые таблицы для хранения такой информации.

## Создайте таблицу customers (информация о клиентах)

### Должна иметь уникальный ключ (id)

### Имя покупателя

### Почта покупателя

## Создайте таблицу orders (информация о заказах)

### Должна иметь уникальный ключ

### Необходимо связать с таблицей customers, используя внешний ключ (ссылка на покупателя, совершившего заказ)

### Необходимо связать с таблицей countries, используя внешний ключ (ссылка на страну, где был совершен заказ)

### Столбец order\_status (строка, статус заказа)

## Создайте таблицу products (информация о товарах в ассортименте нашего ресторана. Если таблица с таким именем уже была в базе данных, удалите ее)

### Должна иметь первичный ключ (id)

### Столбец name (строка, название товара)

### Столбец cost (целое число, цена товара в центах)

### 

### 

## Создайте таблицу order\_items (информация о заказанных товарах)

### Должна иметь уникальный ключ (id)

### Необходимо связать с таблицей orders, используя внешний ключ (ссылка на заказ, к которому относятся заказанные товары)

### Необходимо связать с таблицей products для каждого заказанного товара

### 

### 

# Сложные запросы

## В этой части вам предстоит написать сложные запросы к нескольким таблицам для получения сводной информации их них. В отчет вставьте сами запросы. По желанию можете заполнить все таблицы набором произвольных данных для демонстрации результата выполнения запросов.

## Создавайте сложные запросы, объединяя несколько таблиц (используйте join)

### Создайте запросы к объединенным таблицам countries и continent, чтобы вернуть следующую информацию:

#### Список названий стран и соответствующего континента

#### 

#### Список всех стран в Азии, с названием их столиц

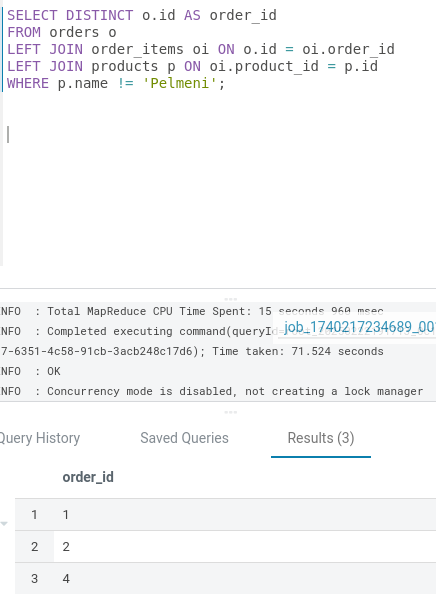
#### 

### Создайте запросы к таблицам customers, orders, order\_items и products, чтобы вернуть следующую информацию:

#### Все товары, которые заказывал заданный клиент с номерами заказов, в которых товар был заказан.

#### 

#### Любой заказ, в который не входит товар “pelmeni”. Отобразите id данного заказа.



#### Имя клиентов, которые ни разу не заказывал сэндвичи.

#### 

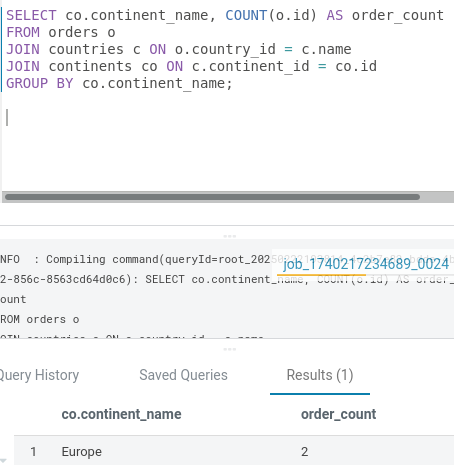
#### Общая стоимость всех заказов заданного клиента в наших ресторанах.



#### Для каждого товара – общее количество раз, когда он был заказан.

#### 

#### Количество заказов по континентам.



#### Страна с самым большим количеством заказов в наших ресторанах.

#### 