**Jupyter http://178.170.197.61:21000**

**HUE http://37.230.196.24:8889**

**Cloudera Manager http://37.230.196.24:7180**

**YARN http://37.230.196.24:8088/**

**YARN History http://37.230.196.24:19888/**

**Задание 1.** Загрузите файл Salary\_Data\_Based\_country\_and\_race.csv в новую папку в своей HDFS директории. <https://disk.yandex.ru/d/T6r30VlBIKtqcA>.

Создайте базу данных со своими ФИ Ivanov\_ivan. Все задания выполняйте в своей БД.

Создайте таблицу salaries и external таблицу salaries\_ext . Проверьте типы таблиц. Удалите salaries\_ext и посчитайте количество записей в salaries, должно получиться 6704.

CREATE DATABASE ivanov;

SHOW DATABASES;

**USE ivanov;**

**CREATE TABLE salaries (**

**id int,**

**age int,**

**gender STRING, education\_level STRING, job\_title STRING, years\_experience int,**

**salary int, country STRING, race STRING**

**)**

**ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**

**LOCATION '/user/std03/Salary';**

**CREATE external TABLE salaries\_ext (**

**id int,**

**age int,**

**gender STRING, education\_level STRING, job\_title STRING, years\_experience int,**

**salary int, country STRING, race STRING**

**)**

**ROW FORMAT DELIMITED FIELDS TERMINATED BY ','**

**LOCATION '/user/std03/Salary\_ext';**

**USE ivanov;**

**show tables;**

**USE ivanov\_ivan;**

**load data inpath '/user/std03/Salary\_Data\_Based\_country\_and\_race.csv' overwrite into table salaries;**

load data inpath '/user/std03/Salary\_Data\_Based\_country\_and\_race.csv' overwrite into table salaries\_ext;

**describe extended salaries;**

**describe extended salaries\_ext;**

**describe formatted salaries;**

**describe formatted salaries\_ext;**

**drop table salaries\_ext;**

**SELECT count(\*) FROM salaries;**

**Задание 2**. Выведите напротив каждой записи таблицы salaries сумму зарплат по всем записям, сумму зарплат по стране и накопительную сумму зарплат, отсортированных по id.

**select \*,**

**sum(salary) over() as sum\_all,**

**sum(salary) over(partition by country) as sum\_country,**

**sum(salary) over(order by id) as CumSrome**

**from salaries**

**order by id;**

**Задание 3.** Создайте таблицы gender\_type, education\_level\_type, job\_title\_type, country\_type, race\_type.

Поля: id, gender / education\_level\_type/ job\_title / country\_type / race\_type.

Создайте таблицу salaries\_normalized, в которой вместо соответствующих значений будут id.

Например, было gender – стало gender\_id. Значение gender сможем получить из таблицы gender\_type.

**CREATE table gender\_type as**

**SELECT**

**row\_number() OVER() as id,**

**g.gender**

**from**

**(SELECT DISTINCT gender FROM salaries) as g;**

**CREATE table salaries\_normalized as**

**WITH gender\_id as**

**(**

**SELECT s.id as id, g.id as gender\_id from salaries as s**

**FULL JOIN gender\_type as g**

**on s.gender = g.gender**

**order by id),**

**country\_id as**

**(**

**SELECT s.id as id, g.id as country\_id from salaries as s**

**FULL JOIN country\_type as g**

**on s.country = g.country**

**order by id),**

**education\_level\_id as**

**(**

**SELECT s.id as id, g.id as education\_level\_id from salaries as s**

**FULL JOIN education\_level\_type as g**

**on s.education\_level = g.education\_level**

**order by id),**

**job\_title\_id as**

**(**

**SELECT s.id as id, g.id as job\_title\_id from salaries as s**

**FULL JOIN job\_title\_type as g**

**on s.job\_title = g.job\_title**

**order by id),**

**race\_id as**

**(**

**SELECT s.id as id, g.id as race\_id from salaries as s**

**FULL JOIN race\_type as g**

**on s.race = g.race**

**order by id)**

**SELECT ge.id, ge.gender\_id, c.country\_id, e.education\_level\_id, j.job\_title\_id, r.race\_id FROM gender\_id ge**

**left JOIN country\_id c on ge.id = c.id**

**left JOIN education\_level\_id e on ge.id = e.id**

**left JOIN job\_title\_id j on ge.id = j.id**

**LEFT JOIN race\_id r on ge.id = r.id;**

**Задание 4.** Создайте таблицу transactions\_sample

размером 10 тысяч записей на основе dwh.transactions.

Создайте столбцы tr\_day и tr\_time на основе tr\_datetime и установите подходящие для них типы данных.

**CREATE table transactions\_sample as**

**SELECT \*,**

**cast(reverse((substring(reverse(tr\_datetime),10))) as int) as tr\_day,**

**substring(tr\_datetime,-8) as tr\_time**

**from dwh.transactions**

**order by rand()**

**limit 10000**

**Задание 5.** Сделайте cross join таблиц dwh.transactions и dwh.transactions\_test. Найдите свой application в YARN, удалите его.

SELECT \* from dwh.transactions

CROSS JOIN dwh.transactions\_test

yarn application -list -appStates RUNNING

yarn application -kill application\_1688483399738\_1529