Latvijas Universitāte

Datorikas fakultāte

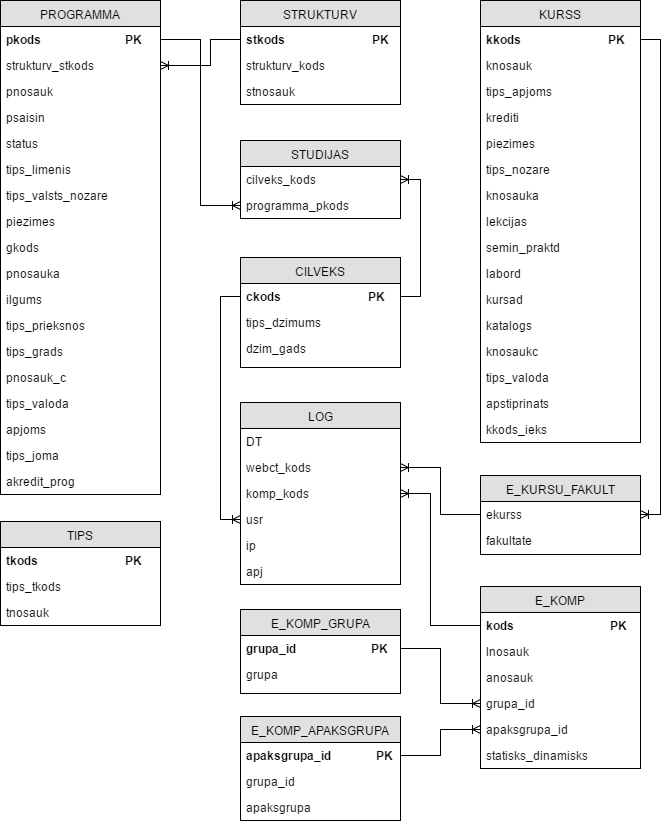
**2. praktiskais darbs**

Kursā „DatZ6054: Datu noliktavu izvēlētas nodaļas”

**Autors:** Dāvis Sporāns DS16042

# Avots

Datu avotam par pamatu ņemti dati no log tabulas, kas atlasīti pēc datuma: 16.,17. un 27. Septembris. Dati eksportēti no datu avota shēmas (1. att.) un saglabāti CSV failā.



. . att. Datu avota relāciju shēma

# Vides apraksts

Praktiskā darba pildīšanai izmantota virtuālā vide uz kuras sagatavota Hadoop vide izmantojot Cloudera sagatavi, kas pieejama Cloudera mājaslapā. Par virtuālās vides dzinēju tika izmanto Oracle VirtualBox risinājumu.

# Map-reduce funkcijas

Lai apstrādātu eksortētos CSV failus, tos pārvieto uz virtuālās vides OS un no tās ar ‘put’ komandu pārvieto uz Hadoop failu sistēmas (HDFS).

Praktiskā darba uzdevums ir noskaidrot cik katram lietotājam katrā no eksportētajiem datumiem ir log ierakstu, jeb hitu skaits. Risinājuma tiek izmantota map-reduce funkcijas, ar kuru palīdzību prasīto informāciju var noskaidrot.

#! /usr/bin/env python

**import** sys

#dt webct\_kods komp\_kods usr ip apj

**for** place**,** line **in** enumerate**(**sys**.**stdin**):**

**if** place **!=** 0**:**

line **=** line**.**strip**()**

unpacked **=** line**.**split**(**"\t"**)**

dt**,**webct\_kods**,**komp\_kods**,**usr**,**ip**,**apj **=** line**.**split**(**"\t"**)**

results **=** **[**usr**,**dt**,** "1"**]**

**print(**"\t"**.**join**(**results**))**

Mapper fukcija Python programmēšanas valodā

#! /usr/bin/env python

**import** sys

last\_key **=** **None**

key\_count **=** 0

**for** line **in** sys**.**stdin**:**

line **=** line**.**strip**()**

usr**,** dt**,** count **=** line**.**split**(**"\t"**)**

key **=** usr**+**"\t"**+**dt

count **=** int**(**count**)**

# if this is the first iteration

**if** **not** last\_key**:**

last\_key **=** key

# if they're the same, log it

**if** key **==** last\_key**:**

key\_count **+=** count

**else:**

result **=** **[**last\_key**,** key\_count**]**

**print(**"\t"**.**join**(**str**(**v**)** **for** v **in** result**))**

last\_key **=** key

key\_count **=** 1

# this is to catch the final value that we output

**print(**"\t"**.**join**(**str**(**v**)** **for** v **in** **[**last\_key**,** key\_count**]))**

Reducer funkcija Python programmēšanas valodā.

# Rezultāts un tā pārbaude.

Lai izsauktu map-reduce funkcijas izpilda Hadoop komandu, kas norāda uz datu avotu, map un reduce funkciju atrašanās vietu un ceļu uz rezultātu mapi. Jāatzīmē, ka datu ielādes un rezultātu izvades ceļiem jāatrodas HDFS, bet map un reduce failiem – lokāli.

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \

-input /user/cloudera/2pd/data/ \

-output /user/cloudera/2pd/output\_new \

-mapper /home/cloudera/Documents/2\_prakt\_darbs/mapper.py \

-reducer /home/cloudera/Documents/2\_prakt\_darbs/reducer.py

Hadoop map-reduce komanda

Pēc komandas izpildes tiek iegūts rezultāta fails (pielikumā). Lai pārliecinātos, ka rezultāti ir koretki, tos salīdzina ar datu avotu.

Pirmie 10 rezultāti pēc map-reduce funkcijas:

100282 17-SEP-2006 7

100282 27-SEP-2006 16

100548 27-SEP-2006 115

101010 27-SEP-2006 26

101936 16-SEP-2006 89

101936 17-SEP-2006 4

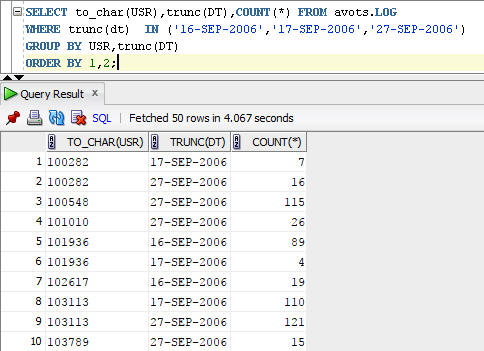
102617 16-SEP-2006 19

103113 27-SEP-2006 121

103113 17-SEP-2006 110

103789 27-SEP-2006 15

Dati no datu avota:



Rezultāti sakrīt, tāpēc var secināt, ka map-reduce funkcija strādā korekti.