Setup Files

Data Check

```
[hdfs@big-nn01 /]$ cd /training/analyst/data/Sample\ Examination/
```

Import Test Files

```
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -mkdir /user/exam
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem1 /user/exam/Problem1
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem2 /user/exam/Problem2
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem3 /user/exam/Problem3
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem4 /user/exam/Problem4
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem5 /user/exam/Problem5
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem6 /user/exam/Problem6
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem7 /user/exam/Problem7
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem8 /user/exam/Problem8
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem9 /user/exam/Problem9
[hdfs@big-nn01 Sample Examination]$ hdfs dfs -put Problem10 /user/exam/Problem10
```

Problem 1

Instructions

Write a query to compare each active account's balance to the average balance of all active accounts of the same type.

Data Description

The account data exists in the metastore account table in the problem1 database. The amount column gives the account balance. The type column gives the type of account and the status column gives the status of the account.

Output Requirements

1. Write the report query in the local file: /home/training/problem1/solution.sql
2. Executing the solution.sql script from the hive command-line tools should generate the report output
3. Only include accounts for which status is Active
4. Create a difference column holding the difference between the account balance (amount) and the average balance for accounts of the same type to the right of all the other columns
5. The query should return the following columns from the account table: id, type, status, amount, as well as the difference calculated

---

- Create Database

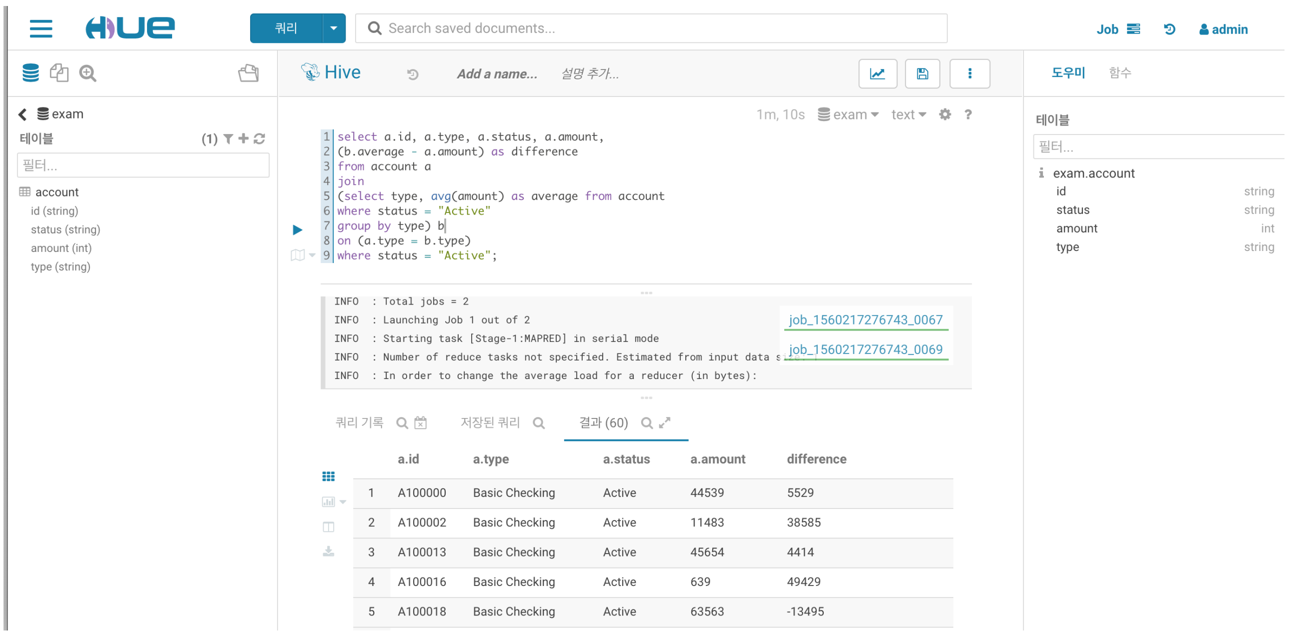
```
create database exam;
usee exam;
```

- Make Table in Hive

```
create external table account (
id STRING, status STRING, amount INT, type STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/exam/Problem1'
```

- Excute Query

```
select a.id, a.type, a.status, a.amount,
(b.average - a.amount) as difference
from account a
join
(select type, avg(amount) as average from account where status = "Active"
group by type) b
on (a.type = b.type)
where status = "Active";
```



```sql
select a.id, a.type, a.status, a.amount,
(b.average - a.amount) as difference
from account a
join
(select type, avg(amount) as average from account
where status = "Active"
group by type) b
on (a.type = b.type)
where status = "Active";
```

```
INFO  : Total jobs = 2
INFO  : Launching Job 1 out of 2
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Number of reduce tasks not specified. Estimated from input data s
INFO  : In order to change the average load for a reducer (in bytes):
```

	a.id	a.type	a.status	a.amount	difference
1	A100000	Basic Checking	Active	44539	5529
2	A100002	Basic Checking	Active	11483	38585
3	A100013	Basic Checking	Active	45654	4414
4	A100016	Basic Checking	Active	639	49429
5	A100018	Basic Checking	Active	63563	-13495

Problem 2

Instructions

Create an employee table in the metastore that contains the employee records stored in HDFS.

Data Description

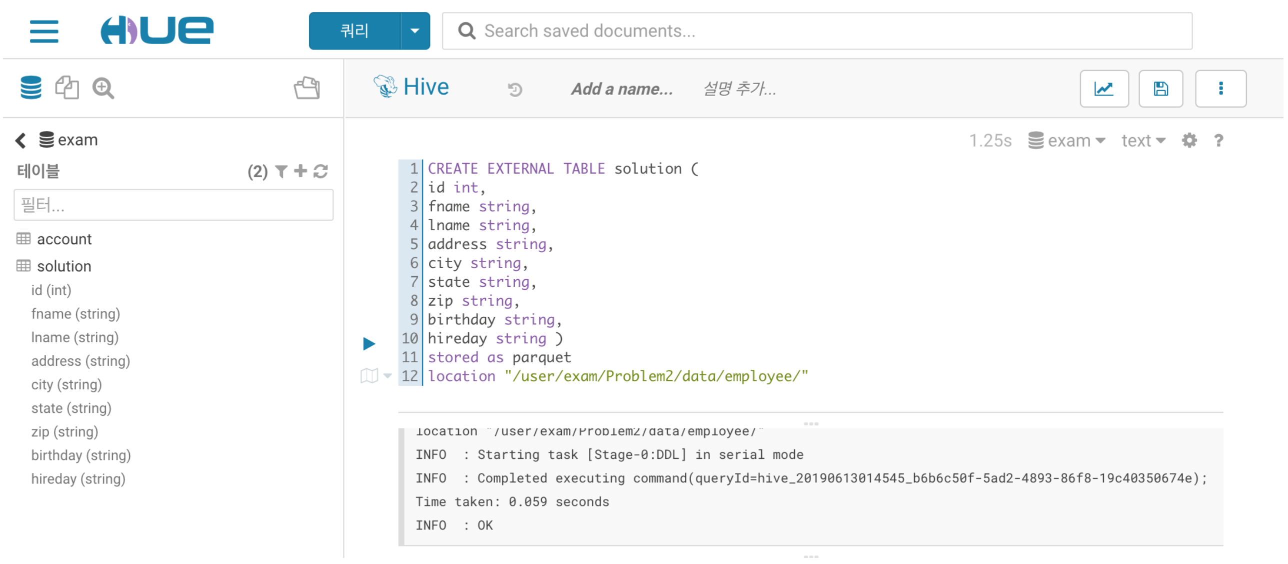
All of the employee records are stored in the /user/training/problem2/data/employee/ HDFS directory in Parquet file format. The files contain the following columns and types:

Output Requirements

1. The table you create should be named solution and stored in the problem2 database
2. The table must point to the existing data in HDFS

- Make Table

> CREATE EXTERNAL TABLE solution (
>
> id int, fname string, lname string, address string, city string, state string, zip string, birthday string, hireday string)
>
> stored as parquet
>
> location "/user/exam/Problem2/data/employee/"



```
CREATE EXTERNAL TABLE solution (
id int,
fname string,
lname string,
address string,
city string,
state string,
zip string,
birthday string,
hireday string )
stored as parquet
location "/user/exam/Problem2/data/employee/"
```

```
location "/user/exam/Problem2/data/employee/"
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190613014545_b6b6c50f-5ad2-4893-86f8-19c40350674e);
Time taken: 0.059 seconds
INFO  : OK
```

Problem 3

Instructions

Generate a table that contains all customers who have negative account balances.

Data Description

The customer records are stored in the customer table in the problem3 database. The account records are stored in the account table in the problem3 database. The account records contain a field called amount that may be negative. The custid field is a foreign key into the customer table.

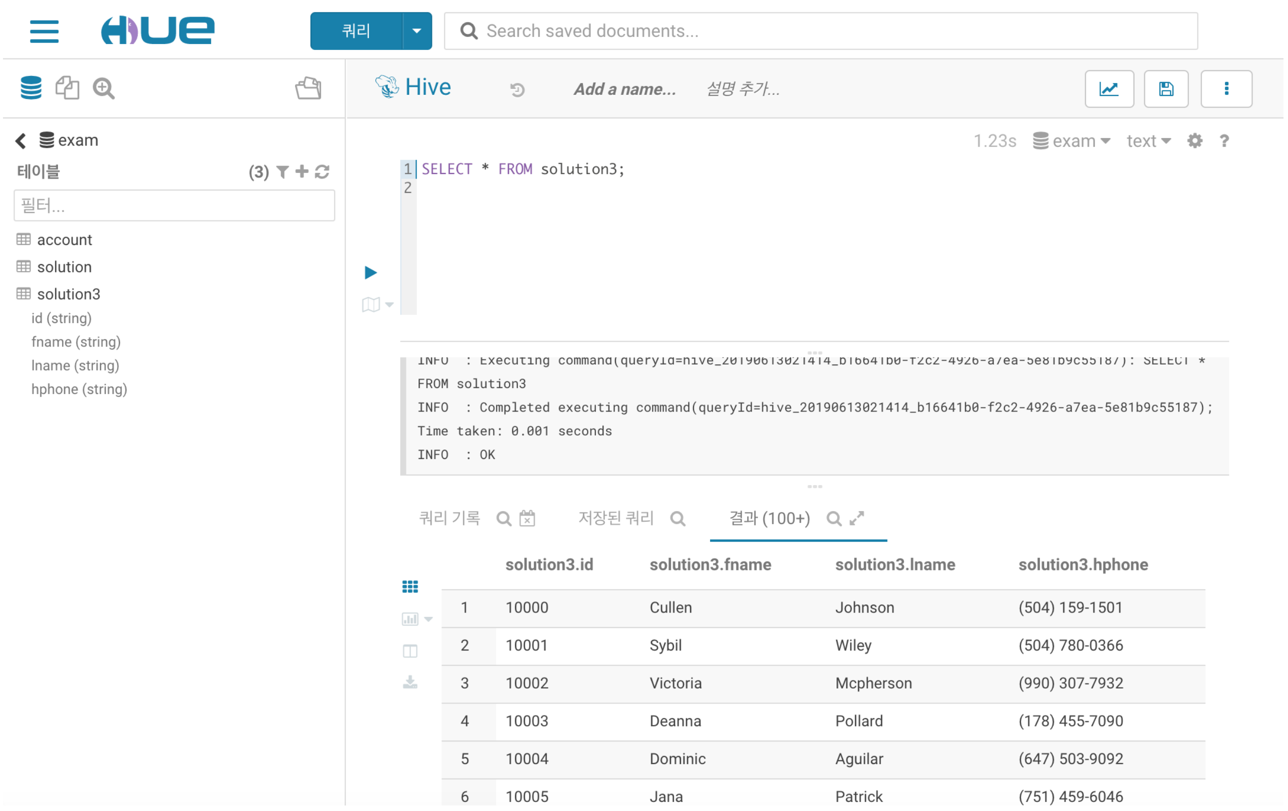
Output Requirements

1. Create a metastore table named solution stored in the problem3 database that contains the customer records that match the criteria
2. Each solution record should contain the Customer ID, first name, last name, and home phone of the customer
3. Maintain the same column names and datatypes in the new table as the fields from the customer table

---

- Make Table

> create external table solution3
> (id STRING, fname string, lname string, hphone string)
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> LOCATION '/user/exam/Problem3/customer'



```
SELECT * FROM solution3;
```

```
INFO  : Executing command(queryId=hive_20190613021414_b16641b0-f2c2-4926-a7ea-5e81b9c55187): SELECT *
FROM solution3
INFO  : Completed executing command(queryId=hive_20190613021414_b16641b0-f2c2-4926-a7ea-5e81b9c55187);
Time taken: 0.001 seconds
INFO  : OK
```

	solution3.id	solution3.fname	solution3.lname	solution3.hphone
1	10000	Cullen	Johnson	(504) 159-1501
2	10001	Sybil	Wiley	(504) 780-0366
3	10002	Victoria	Mcpherson	(990) 307-7932
4	10003	Deanna	Pollard	(178) 455-7090
5	10004	Dominic	Aguilar	(647) 503-9092
6	10005	Jana	Patrick	(751) 459-6046

Problem 4

Instructions

LoudAcre Mobile has merged with another company located in California. Each company has a list of customers in different formats. Combine the two customer lists into a single dataset using an identical schema.

Data Description

The original customer data exists in the HDFS directory /user/training/problem4/data/employee1/. It contains expanded, nine-digit zip codes. The new files are in the HDFS directory /user/training/problem4/data/employee2/. It contains last names before first names, both using all capital letters.

Output Requirements

1. Combine these files into a single tab-delimited dataset and stored in the HDFS directory /user/training/problem4/solution/
2. Each record should be in the following format:
3. Only include customers whose state is ‘CA’

Excute on PIG

> employee1 = LOAD '/user/training/problem4/data/employee1' AS (customerID:Int, fname:chararray, lname:chararray, address:chararray, city:chararray, state:chararray, zip: chararray);

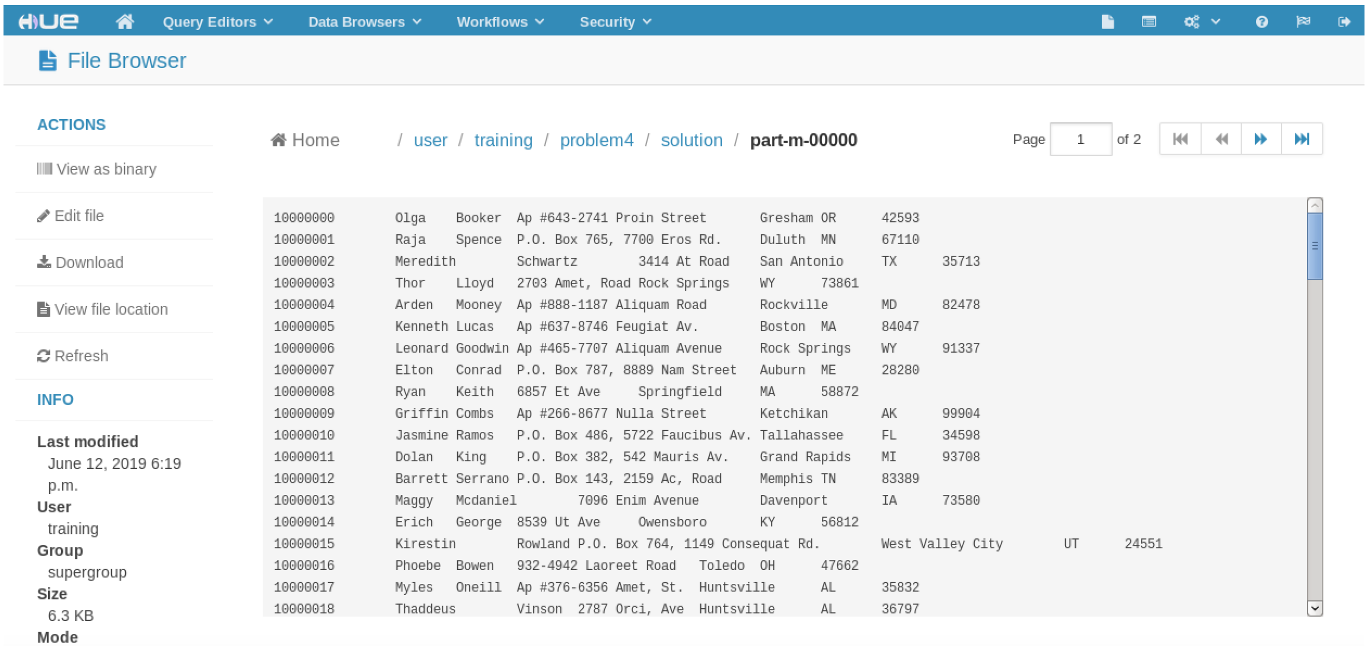
> e1 = FOREACH employee1 GENERATE customerID, fname, lname, address, city, state, SUBSTRING(zip,0,5);

> employee2 = LOAD '/user/training/problem4/data/employee2' USING PigStorage(',') AS (customerID:Int, junk:Int, lname:chararray, fname:chararray, address:chararray, city:chararray, state:chararray, zip: chararray);

> e2 = FOREACH employee2 GENERATE customerID, UCFIRST(LOWER(fname)), UCFIRST(LOWER(lname)), address, city, state, zip;

> both = UNION e1 , e2;

> STORE both INTO '/user/training/problem4/solution/'



Last modified June 12, 2019 6:19 p.m.
User training
Group supergroup
Size 6.3 KB
Mode

Home / user / training / problem4 / solution / part-m-00000

Page 1 of 2

```
10000000        Olga    Booker  Ap #643-2741 Proin Street       Gresham OR      42593
10000001        Raja    Spence  P.O. Box 765, 7700 Eros Rd.     Duluth  MN      67110
10000002        Meredith        Schwartz        3414 At Road    San Antonio     TX      35713
10000003        Thor    Lloyd   2703 Amet, Road Rock Springs    WY      73861
10000004        Arden   Mooney  Ap #888-1187 Aliquam Road       Rockville       MD      82478
10000005        Kenneth Lucas   Ap #637-8746 Feugiat Av.        Boston  MA      84047
10000006        Leonard Goodwin Ap #465-7707 Aliquam Avenue     Rock Springs    WY      91337
10000007        Elton   Conrad  P.O. Box 787, 8889 Nam Street   Auburn  ME      28280
10000008        Ryan    Keith   6857 Et Ave     Springfield     MA      58872
10000009        Griffin Combs   Ap #266-8677 Nulla Street       Ketchikan       AK      99904
10000010        Jasmine Ramos   P.O. Box 486, 5722 Faucibus Av. Tallahassee     FL      34598
10000011        Dolan   King    P.O. Box 382, 542 Mauris Av.    Grand Rapids    MI      93708
10000012        Barrett Serrano P.O. Box 143, 2159 Ac, Road     Memphis TN      83389
10000013        Maggy   Mcdaniel        7096 Enim Avenue        Davenport       IA      73580
10000014        Erich   George  8539 Ut Ave     Owensboro       KY      56812
10000015        Kirestin        Rowland P.O. Box 764, 1149 Consequat Rd.        West Valley City        UT      24551
10000016        Phoebe  Bowen   932-4942 Laoreet Road   Toledo  OH      47662
10000017        Myles   Oneill  Ap #376-6356 Amet, St.  Huntsville      AL      35832
10000018        Thaddeus        Vinson  2787 Orci, Ave  Huntsville      AL      36797
```

Problem 5

Instructions

The bank is making a Facebook group for the Palo Alto, CA branch. Generate a script that outputs the customers and employees who live in Palo Alto, CA.

Data Description

The employee records are stored in the employee metastore table in the problem5 database, The customer records are stored in the customer metastore table in the problem5 database.

Output Requirements

1. Write the report query in the local file /home/training/problem5/solution.sql
2. Executing the solution.sql script from the hive command-line tools should generate the report output
3. The report should contain first name, last name, city, and state with a tab as the delimiter between fields
4. Only output records that have a city value of ‘Palo Alto’ and state value of ‘CA’
5. Duplicate records (if any) should be included

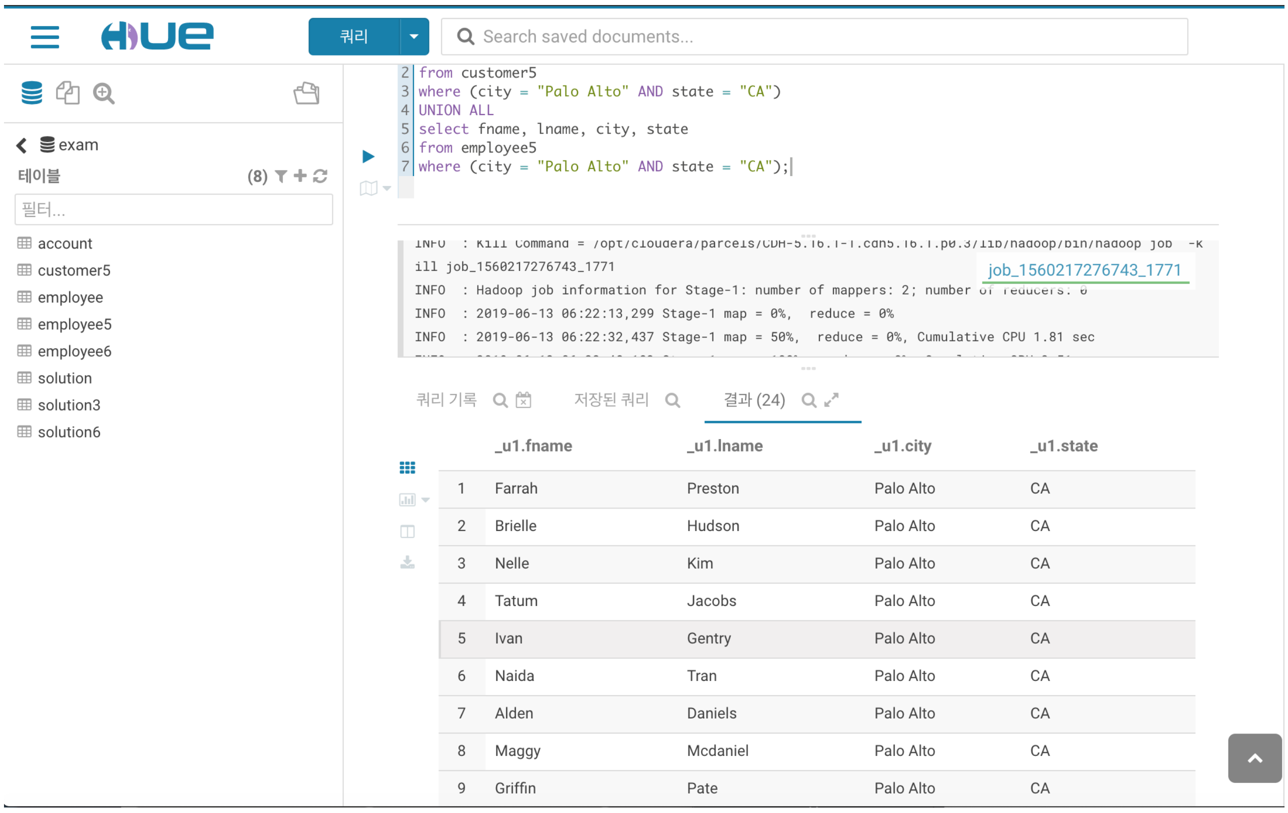
- Make Table

> create external table customer5 (id int, fname string, lname string, address string, city string, state string, zip string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem5/customer";

> create external table employee5 (id int, fname string, lname string, address string, city string, state string, zip string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem5/employee";

- Excute Query

> select fname, lname, city, state from customer5 where (city = "Palo Alto" AND state = "CA") UNION ALL select fname, lname, city, state from employee5 where (city = "Palo Alto" AND state = "CA");



```
from customer5
where (city = "Palo Alto" AND state = "CA")
UNION ALL
select fname, lname, city, state
from employee5
where (city = "Palo Alto" AND state = "CA");
```

```
INFO  : Kill Command = /opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/lib/hadoop/bin/hadoop job  -k
ill job_1560217276743_1771
INFO  : Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 0
INFO  : 2019-06-13 06:22:13,299 Stage-1 map = 0%,  reduce = 0%
INFO  : 2019-06-13 06:22:32,437 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 1.81 sec
```

	_u1.fname	_u1.lname	_u1.city	_u1.state
1	Farrah	Preston	Palo Alto	CA
2	Brielle	Hudson	Palo Alto	CA
3	Nelle	Kim	Palo Alto	CA
4	Tatum	Jacobs	Palo Alto	CA
5	Ivan	Gentry	Palo Alto	CA
6	Naida	Tran	Palo Alto	CA
7	Alden	Daniels	Palo Alto	CA
8	Maggy	Mcdaniel	Palo Alto	CA
9	Griffin	Pate	Palo Alto	CA

Problem 6

Instructions

There are privacy concerns about the employee data that is stored on the cluster. Your task is to remove any age information from the employee data by creating a new table for the data analysts to query against.

Data Description

All of the employee records are stored in the employee metastore table in the problem6 database.

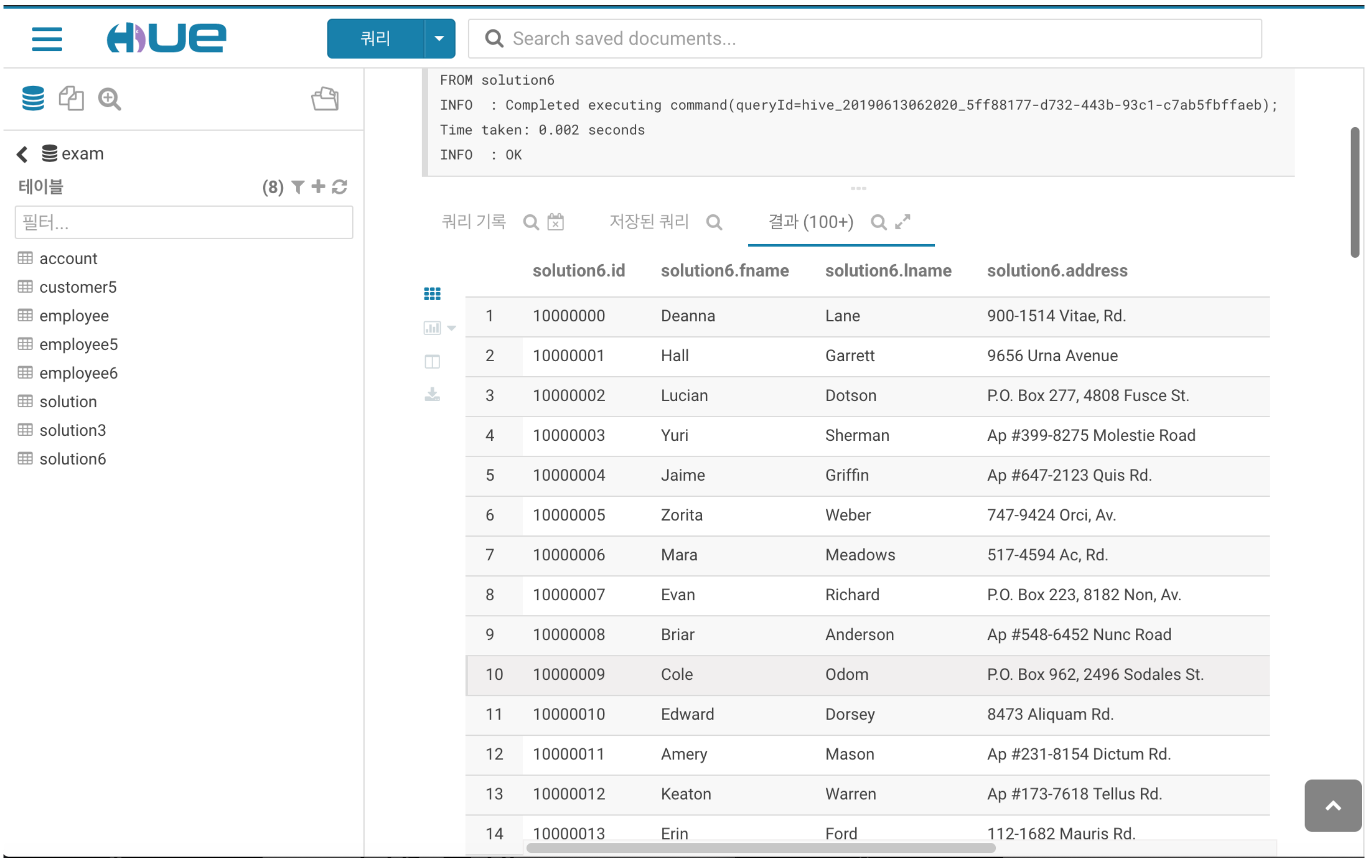
Output Requirements

1. Create a new table named solution stored in the problem6 database with the same file format as the employee table
2. Maintain the same column names and datatypes in the new table
3. The birthday field in the solution table should be truncated to only contain month/day instead of the current month/day/year data that is in the employee table

---

> create external table employee6 (id int, fname string, lname string, address string, city string, state string, zip string, birthday string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem6/employee";

> create table solution6 as select id, fname, lname, address, city, state, zip, substr(birthday,1,5) from employee6;



```
FROM solution6
INFO  : Completed executing command(queryId=hive_20190613062020_5ff88177-d732-443b-93c1-c7ab5fbffaeb);
Time taken: 0.002 seconds
INFO  : OK
```

	solution6.id	solution6.fname	solution6.lname	solution6.address
1	10000000	Deanna	Lane	900-1514 Vitae, Rd.
2	10000001	Hall	Garrett	9656 Urna Avenue
3	10000002	Lucian	Dotson	P.O. Box 277, 4808 Fusce St.
4	10000003	Yuri	Sherman	Ap #399-8275 Molestie Road
5	10000004	Jaime	Griffin	Ap #647-2123 Quis Rd.
6	10000005	Zorita	Weber	747-9424 Orci, Av.
7	10000006	Mara	Meadows	517-4594 Ac, Rd.
8	10000007	Evan	Richard	P.O. Box 223, 8182 Non, Av.
9	10000008	Briar	Anderson	Ap #548-6452 Nunc Road
10	10000009	Cole	Odom	P.O. Box 962, 2496 Sodales St.
11	10000010	Edward	Dorsey	8473 Aliquam Rd.
12	10000011	Amery	Mason	Ap #231-8154 Dictum Rd.
13	10000012	Keaton	Warren	Ap #173-7618 Tellus Rd.
14	10000013	Erin	Ford	112-1682 Mauris Rd.

Problem 7

Instructions

Generate a report that contains all of the Seattle employee names in sorted order.

Data Description

The employee records are stored in the employee table in the problem7 database.

Output Requirements

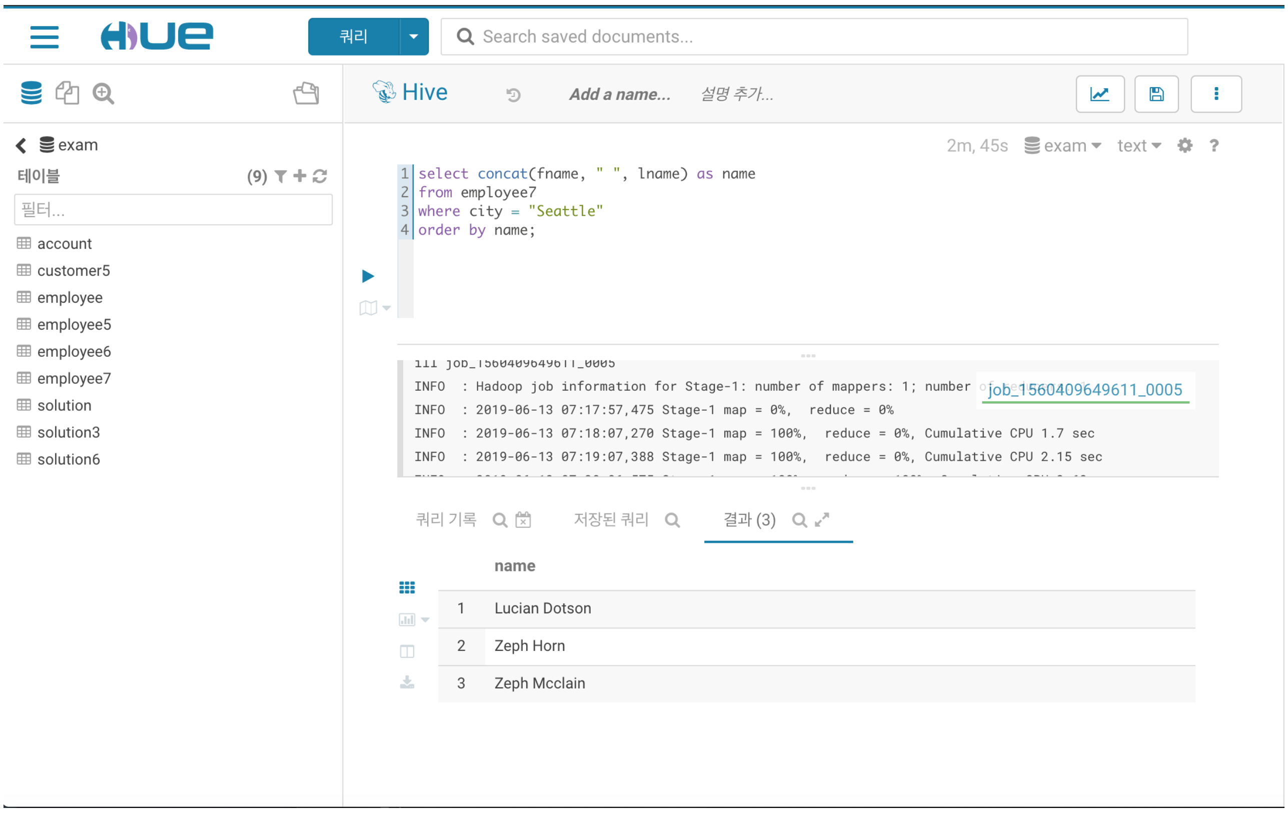
1. Write the report query in the local file /home/training/problem7/solution.sql
2. Executing the solution.sql script from the hive command-line tools should generate the report output
3. The employee names should be printed out as the first name, a space, and the last name
4. The output should be sorted by first name and then by last name and should only contain employees whose city is 'Seattle'
5. Duplicate names should be included (if any)

- Make Table

> create external table employee7 (id int, fname string, lname string, address string, city string, state string, zip string, birthday string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem7/employee";

- Excute Query

> select concat(fname, " ", lname) as name from employee7 where city = "Seattle" order by name;



```
select concat(fname, " ", lname) as name
from employee7
where city = "Seattle"
order by name;
```

```
INFO  : Hadoop job information for Stage-1: number of mappers: 1; number
INFO  : 2019-06-13 07:17:57,475 Stage-1 map = 0%,  reduce = 0%
INFO  : 2019-06-13 07:18:07,270 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.7 sec
INFO  : 2019-06-13 07:19:07,388 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.15 sec
```

	name
1	Lucian Dotson
2	Zeph Horn
3	Zeph Mcclain

Problem 8

Instructions

Use Sqoop to export customer data from HDFS into a MySQL database table. Place the data in the solution table in MySQL, which has been created and is currently empty.

Data Description

The data files are in the HDFS directory /user/training/problem8/data/customer/. MySQL database information:

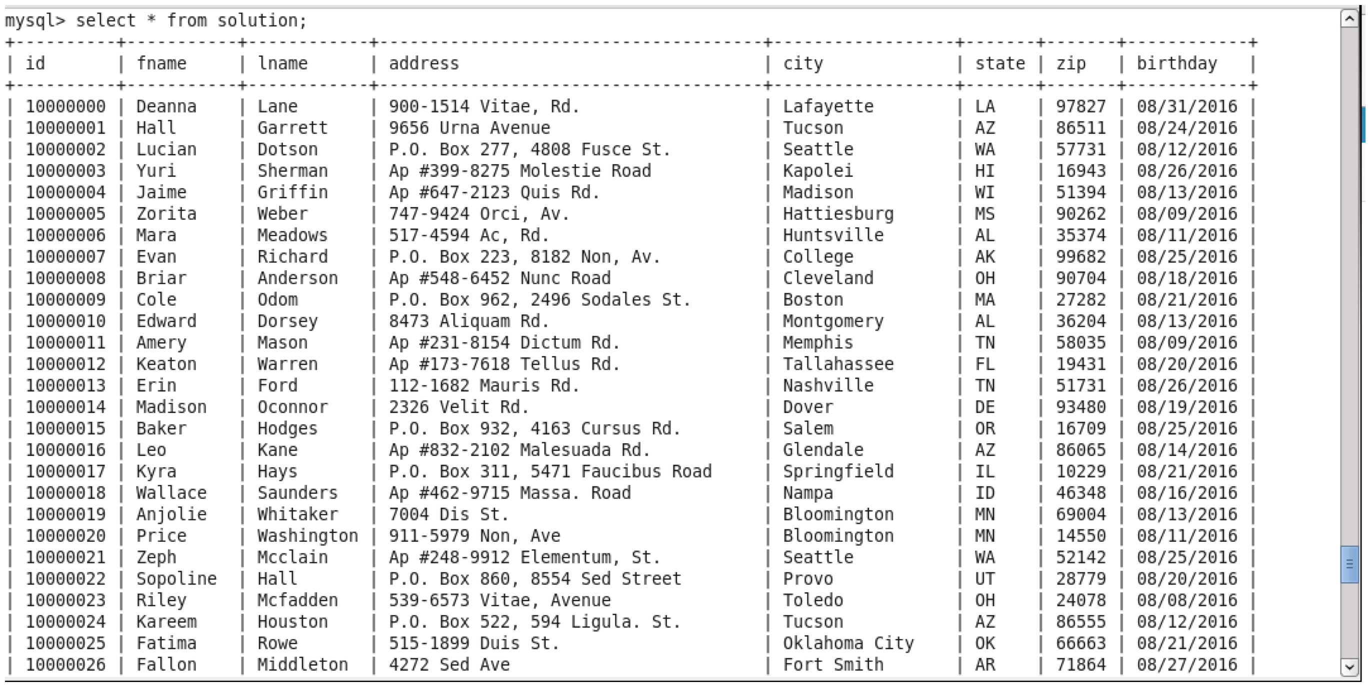
1. Installation: localhost
2. Database name: problem8 | Table name: solution
3. Username: cloudera
4. Password: cloudera

Output Requirements

1. Export all of the customer data from HDFS into the MySQL solution table
2. The solution table already exists in the MySQL database but currently has no rows

---

> sqoop export
> --connect jdbc:mysql://localhost/problem8
> --username cloudera
> --password cloudera
> --table solution
> --input-fields-terminated-by '\t'
> --export-dir hdfs://localhost:8020/user/training/problem8/data/customer



```
mysql> select * from solution;
+----------+----------+------------+-----------------------------------------+-------------------+-------+-------+------------+
| id       | fname    | lname      | address                                 | city              | state | zip   | birthday   |
+----------+----------+------------+-----------------------------------------+-------------------+-------+-------+------------+
| 10000000 | Deanna   | Lane       | 900-1514 Vitae, Rd.                     | Lafayette         | LA    | 97827 | 08/31/2016 |
| 10000001 | Hall     | Garrett    | 9656 Urna Avenue                        | Tucson            | AZ    | 86511 | 08/24/2016 |
| 10000002 | Lucian   | Dotson     | P.O. Box 277, 4808 Fusce St.            | Seattle           | WA    | 57731 | 08/12/2016 |
| 10000003 | Yuri     | Sherman    | Ap #399-8275 Molestie Road              | Kapolei           | HI    | 16943 | 08/26/2016 |
| 10000004 | Jaime    | Griffin    | Ap #647-2123 Quis Rd.                   | Madison           | WI    | 51394 | 08/13/2016 |
| 10000005 | Zorita   | Weber      | 747-9424 Orci, Av.                      | Hattiesburg       | MS    | 90262 | 08/09/2016 |
| 10000006 | Mara     | Meadows    | 517-4594 Ac, Rd.                        | Huntsville        | AL    | 35374 | 08/11/2016 |
| 10000007 | Evan     | Richard    | P.O. Box 223, 8182 Non, Av.             | College           | AK    | 99682 | 08/25/2016 |
| 10000008 | Briar    | Anderson   | Ap #548-6452 Nunc Road                  | Cleveland         | OH    | 90704 | 08/18/2016 |
| 10000009 | Cole     | Odom       | P.O. Box 962, 2496 Sodales St.          | Boston            | MA    | 27282 | 08/21/2016 |
| 10000010 | Edward   | Dorsey     | 8473 Aliquam Rd.                        | Montgomery        | AL    | 36204 | 08/13/2016 |
| 10000011 | Amery    | Mason      | Ap #231-8154 Dictum Rd.                 | Memphis           | TN    | 58035 | 08/09/2016 |
| 10000012 | Keaton   | Warren     | Ap #173-7618 Tellus Rd.                 | Tallahassee       | FL    | 19431 | 08/20/2016 |
| 10000013 | Erin     | Ford       | 112-1682 Mauris Rd.                     | Nashville         | TN    | 51731 | 08/26/2016 |
| 10000014 | Madison  | Oconnor    | 2326 Velit Rd.                          | Dover             | DE    | 93480 | 08/19/2016 |
| 10000015 | Baker    | Hodges     | P.O. Box 932, 4163 Cursus Rd.           | Salem             | OR    | 16709 | 08/25/2016 |
| 10000016 | Leo      | Kane       | Ap #832-2102 Malesuada Rd.              | Glendale          | AZ    | 86065 | 08/14/2016 |
| 10000017 | Kyra     | Hays       | P.O. Box 311, 5471 Faucibus Road        | Springfield       | IL    | 10229 | 08/21/2016 |
| 10000018 | Wallace  | Saunders   | Ap #462-9715 Massa. Road                | Nampa             | ID    | 46348 | 08/16/2016 |
| 10000019 | Anjolie  | Whitaker   | 7004 Dis St.                            | Bloomington       | MN    | 69004 | 08/13/2016 |
| 10000020 | Price    | Washington | 911-5979 Non, Ave                       | Bloomington       | MN    | 14550 | 08/11/2016 |
| 10000021 | Zeph     | Mcclain    | Ap #248-9912 Elementum, St.             | Seattle           | WA    | 52142 | 08/25/2016 |
| 10000022 | Sopoline | Hall       | P.O. Box 860, 8554 Sed Street           | Provo             | UT    | 28779 | 08/20/2016 |
| 10000023 | Riley    | Mcfadden   | 539-6573 Vitae, Avenue                  | Toledo            | OH    | 24078 | 08/08/2016 |
| 10000024 | Kareem   | Houston    | P.O. Box 522, 594 Ligula. St.           | Tucson            | AZ    | 86555 | 08/12/2016 |
| 10000025 | Fatima   | Rowe       | 515-1899 Duis St.                       | Oklahoma City     | OK    | 66663 | 08/21/2016 |
| 10000026 | Fallon   | Middleton  | 4272 Sed Ave                            | Fort Smith        | AR    | 71864 | 08/27/2016 |
```

Problem 9

Instructions

Your company is being acquired by another company. To prepare for this acquisition, update the customer records to guarantee there will be no duplicate IDs with their existing customer IDs.

Data Description

The customer records are stored in the customer table in the problem9 database. The id column is a unique identifier for that record.

Output Requirements

1. Create a new table named solution stored in the problem9 database
2. Maintain the same column names and datatypes as the customer table, except store the id as a string
3. The solution table should have all of the data from the customer table, with the addition of the letter 'A' to the existing id values to make them unique

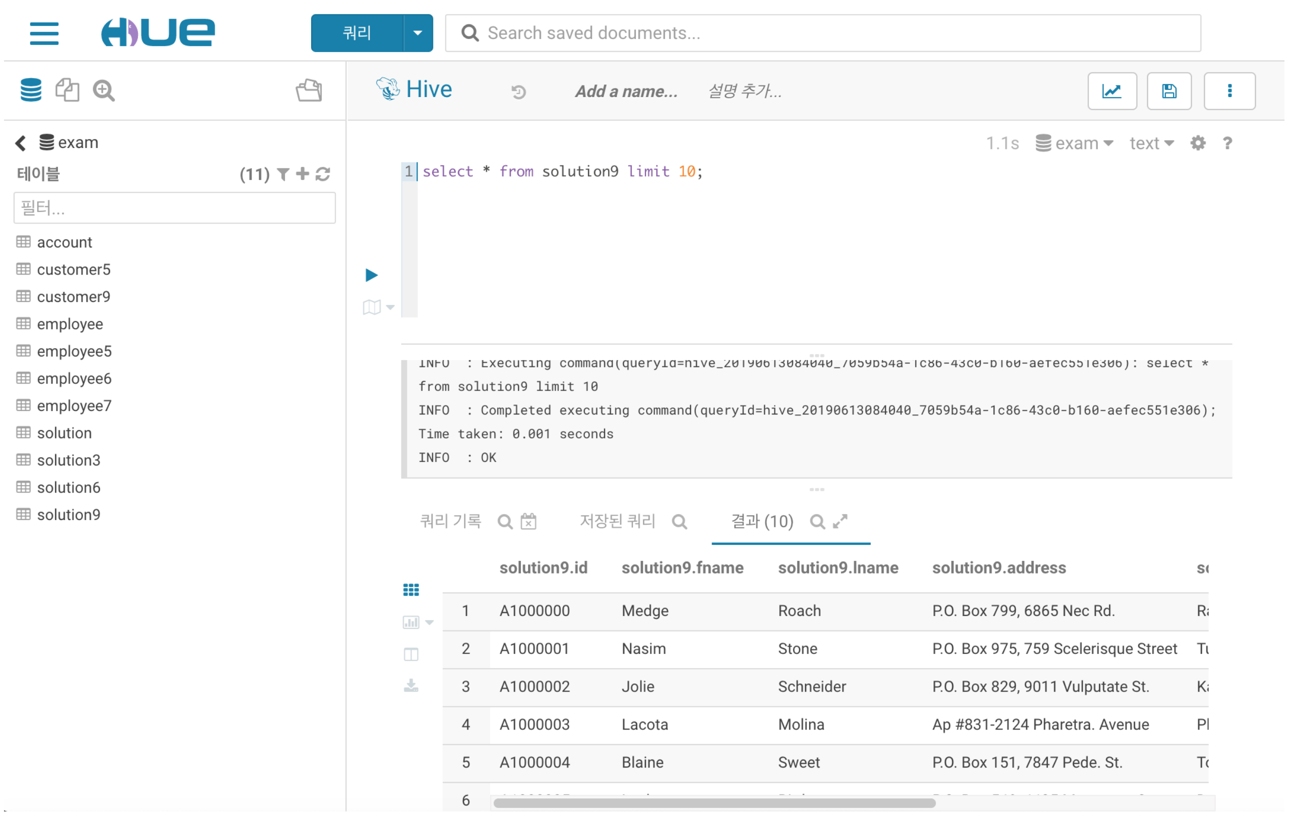
---

- Make Table

> create external table customer9 (id int, fname string, lname string, address string, city string, state string, zip string, birthday string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem9/customer";

- Excute Query

> create table solution9 as select concat("A", cast(id as string)) as id, fname, lname, address, city, state, zip, birthday from customer9;



```
select * from solution9 limit 10;
```

```
INFO  : Executing command(queryId=hive_20190613084040_7059b54a-1c86-43c0-b160-aefec551e306): select *
from solution9 limit 10
INFO  : Completed executing command(queryId=hive_20190613084040_7059b54a-1c86-43c0-b160-aefec551e306);
Time taken: 0.001 seconds
INFO  : OK
```

	solution9.id	solution9.fname	solution9.lname	solution9.address	s
1	A1000000	Medge	Roach	P.O. Box 799, 6865 Nec Rd.	Ra
2	A1000001	Nasim	Stone	P.O. Box 975, 759 Scelerisque Street	Tu
3	A1000002	Jolie	Schneider	P.O. Box 829, 9011 Vulputate St.	Ka
4	A1000003	Lacota	Molina	Ap #831-2124 Pharetra. Avenue	Pl
5	A1000004	Blaine	Sweet	P.O. Box 151, 7847 Pede. St.	To
6					

Problem 10

Instructions

Your boss needs specialized reports using the billing data and is constantly asking for help to write SQL queries. Create a database view in the metastore so that your boss has customer and billing data joined.

Data Description

The customer data exists in the customer metasore table in the problem10 database. The billing data exists in the billing metastore table in the problem10 database. The id column is a foreign key into which customer has the charge.

Output Requirements

1. Create a new view named solution stored in the problem10 database
2. The new view should maintain the same column datatypes as the source tables
3. Do not return customer records that have no billing data, or billing records that have no matching customer
4. The new view should contain the following columns
5. The billdate column should only contain the date field and no time information

---

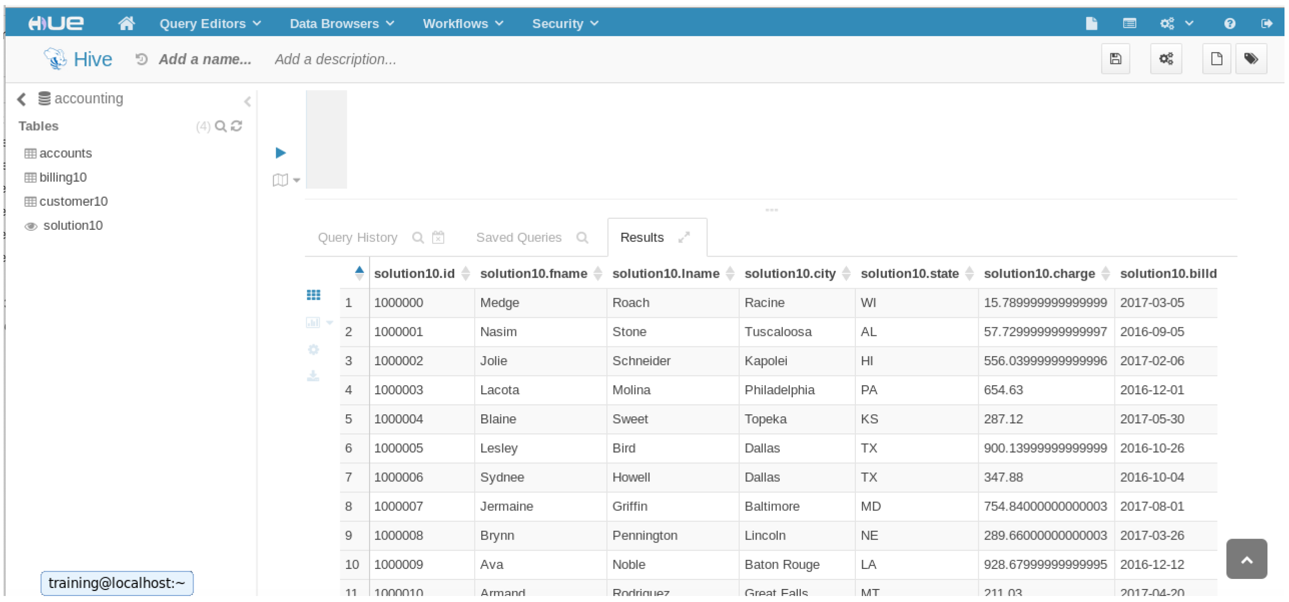
- Make Table

> create external table customer10 (id int, fname string, lname string, address string, city string, state string, zip string, birthday string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem9/customer";

> create external table billing10 (id int, charge double, tstamp string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION "/user/exam/Problem10/billing";

- Make View

> create view solution as select c.id as id, c.fname as fname, c.lname as lname, c.city as city, c.state as state, b.charge as charge, SUBSTR(b.tstamp,0,10) as billdate from customer c join billing b on (c.id = b.id);



	solution10.id	solution10.fname	solution10.lname	solution10.city	solution10.state	solution10.charge	solution10.billd
1	1000000	Medge	Roach	Racine	WI	15.789999999999999	2017-03-05
2	1000001	Nasim	Stone	Tuscaloosa	AL	57.729999999999997	2016-09-05
3	1000002	Jolie	Schneider	Kapolei	HI	556.03999999999996	2017-02-06
4	1000003	Lacota	Molina	Philadelphia	PA	654.63	2016-12-01
5	1000004	Blaine	Sweet	Topeka	KS	287.12	2017-05-30
6	1000005	Lesley	Bird	Dallas	TX	900.13999999999999	2016-10-26
7	1000006	Sydnee	Howell	Dallas	TX	347.88	2016-10-04
8	1000007	Jermaine	Griffin	Baltimore	MD	754.84000000000003	2017-08-01
9	1000008	Brynn	Pennington	Lincoln	NE	289.66000000000003	2017-03-26
10	1000009	Ava	Noble	Baton Rouge	LA	928.67999999999995	2016-12-12
11	1000010	Armand	Rodriguez	Great Falls	MT	211.03	2017-04-20

Problem 11

Instructions

Several analysis questions are described below and you will need to write the SQL code to answer them. You can use whichever tool you prefer – Impala or Hive – using whichever method you like best, including shell, script, or the Hue Query Editor, to run your queries.

Data Description

Using default database from the metastore for these queries.

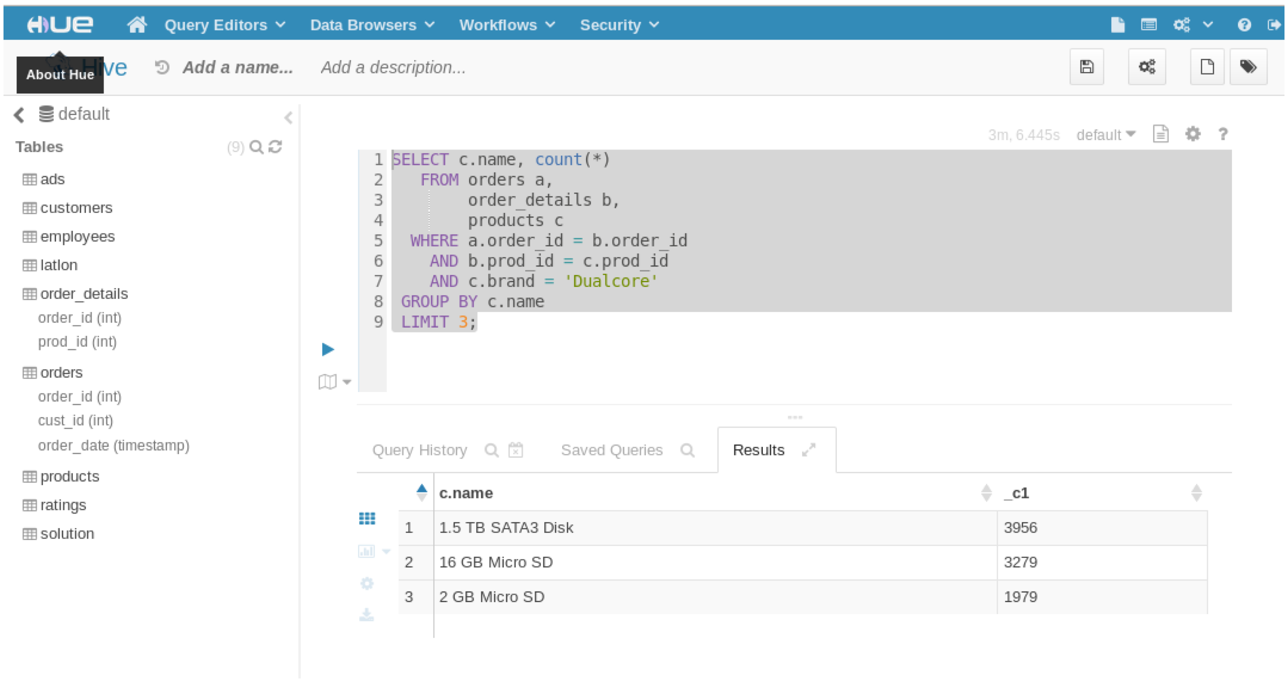
Output Requirements

1. Write the report query in the local file /home/training/problem11/solution.sql
2. Executing the solution.sql script from the hive command-line tools should generate

the report output

- a. Which top three products has Dualcore sold more of than any other? Hint: Remember that if you use a GROUP BY clause, you must group by all fields listed in the SELECT clause that are not part of an aggregate function.

  > SELECT c.name, count(*) FROM orders a, order_details b, products c WHERE a.order_id = b.order_id AND b.prod_id = c.prod_id AND c.brand = 'Dualcore' GROUP BY c.name LIMIT 3;

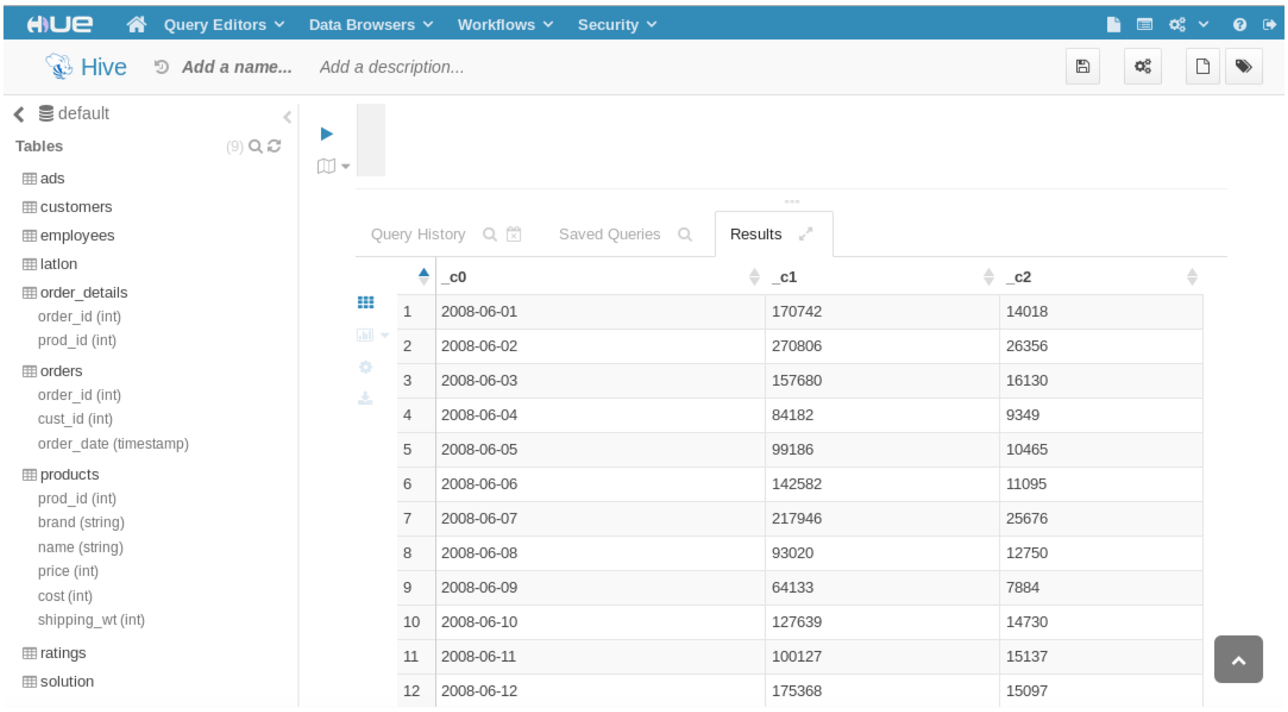


```
SELECT c.name, count(*)
  FROM orders a,
       order_details b,
       products c
 WHERE a.order_id = b.order_id
   AND b.prod_id = c.prod_id
   AND c.brand = 'Dualcore'
 GROUP BY c.name
 LIMIT 3;
```

	c.name	_c1
1	1.5 TB SATA3 Disk	3956
2	16 GB Micro SD	3279
3	2 GB Micro SD	1979

- b. Calculating Revenue and Profit – write a query to show Dualcore's revenue (total price of products sold) and profit (price minus cost) by date. Hint: The order_date column in the orders table is of type TIMESTAMP. Use the function to_date to get just the date portion of the value.

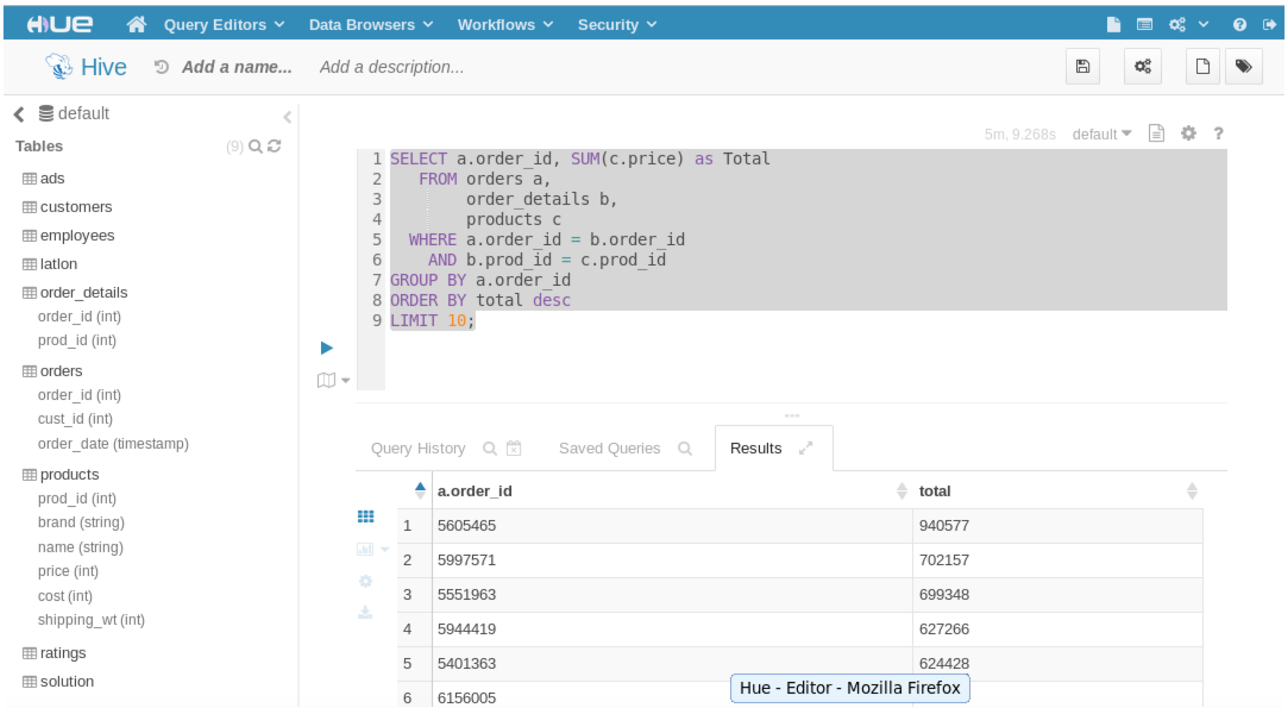
> SELECT to_date(a.order_date), sum(c.price), sum(c.price - c.cost)
> FROM orders a, order_details b, products c WHERE a.order_id = b.order_id AND b.prod_id = c.prod_id AND c.brand = 'Dualcore' GROUP BY to_date(a.order_date);



	_c0	_c1	_c2
1	2008-06-01	170742	14018
2	2008-06-02	270806	26356
3	2008-06-03	157680	16130
4	2008-06-04	84182	9349
5	2008-06-05	99186	10465
6	2008-06-06	142582	11095
7	2008-06-07	217946	25676
8	2008-06-08	93020	12750
9	2008-06-09	64133	7884
10	2008-06-10	127639	14730
11	2008-06-11	100127	15137
12	2008-06-12	175368	15097

- c. Calculating the order Total – Which ten orders had the highest total dollar amounts?

> SELECT a.order_id, SUM(c.price) as Total FROM orders a, order_details b, products c WHERE a.order_id = b.order_id AND b.prod_id = c.prod_id GROUP BY a.order_id ORDER BY total desc LIMIT 10;



```sql
SELECT a.order_id, SUM(c.price) as Total
  FROM orders a,
       order_details b,
       products c
 WHERE a.order_id = b.order_id
   AND b.prod_id = c.prod_id
GROUP BY a.order_id
ORDER BY total desc
LIMIT 10;
```

	a.order_id	total
1	5605465	940577
2	5997571	702157
3	5551963	699348
4	5944419	627266
5	5401363	624428
6	6156005	