Hive 下载：

在 Docker Desktop 中将 Hive 4.0.0 pull 下来，在终端以管理员身份启动并进入容器

```
PS C:\Users\asus> docker run -d -p 10000:10000 -p 10002:10002 --env SERVICE_NAME=hiveserver2 --name hive apache/hive:4.0.0
14db4c2f7dd590c6c323c5a96f189c8dc5e2e435f6a632de5d5d7c19c9882c81
PS C:\Users\asus> docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                                                                  NAMES
14db4c2f7dd5   apache/hive:4.0.0   "sh -c /entrypoint.sh"   19 seconds ago   Up 18 seconds   0.0.0.0:10000->10000/tcp, 9083/tcp, 0.0.0.0:10002->10002/tcp   hive
PS C:\Users\asus> docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                                                                  NAMES
14db4c2f7dd5   apache/hive:4.0.0   "sh -c /entrypoint.sh"   24 seconds ago   Up 22 seconds   0.0.0.0:10000->10000/tcp, 9083/tcp, 0.0.0.0:10002->10002/tcp   hive
PS C:\Users\asus> docker exec -it hive beeline -u 'jdbc:hive2://localhost:10000/'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000/
Connected to: Apache Hive (version 4.0.0)
Driver: Hive JDBC (version 4.0.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 4.0.0 by Apache Hive
```

数据准备：

将 user_profile_table.csv 和 user_balance_table.csv 复制到 hive 的/tmp 目录下

```
PS C:\Users\asus> docker cp 'C:\Users\asus\Desktop\Financial Big Data\lab3\user_profile_table.csv' hive:/tmp/user_profile_table.csv
Successfully copied 748kB to hive:/tmp/user_profile_table.csv
PS C:\Users\asus> docker cp 'C:\Users\asus\Desktop\Financial Big Data\lab3\user_balance_table.csv' hive:/tmp/user_balance_table.csv
Successfully copied 158MB to hive:/tmp/user_balance_table.csv
```

Task1 数据加载到 Hive 中：

创建两张表格 user_profile_table 和 user_balance_table：

```
CREATE TABLE user_profile_table (
    user_id INT,
    sex INT,
    city INT,
    constellation STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES("skip.header.line.count"="1");
```

```
CREATE TABLE user_balance_table (
    user_id STRING,
    report_date STRING,
    tBalance DOUBLE,
    yBalance DOUBLE,
    total_purchase_amt DOUBLE,
    direct_purchase_amt DOUBLE,
    purchase_bal_amt DOUBLE,
    purchase_bank_amt DOUBLE,
    total_redeem_amt DOUBLE,
    consume_amt DOUBLE,
    transfer_amt DOUBLE,
    tftobal_amt DOUBLE,
    tftocard_amt DOUBLE,
    share_amt DOUBLE,
    category1 DOUBLE,
```

```
    category2 DOUBLE,
    category3 DOUBLE,
    category4 DOUBLE
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES ("skip.header.line.count"="1");
```

之后将数据导入创建的表格：

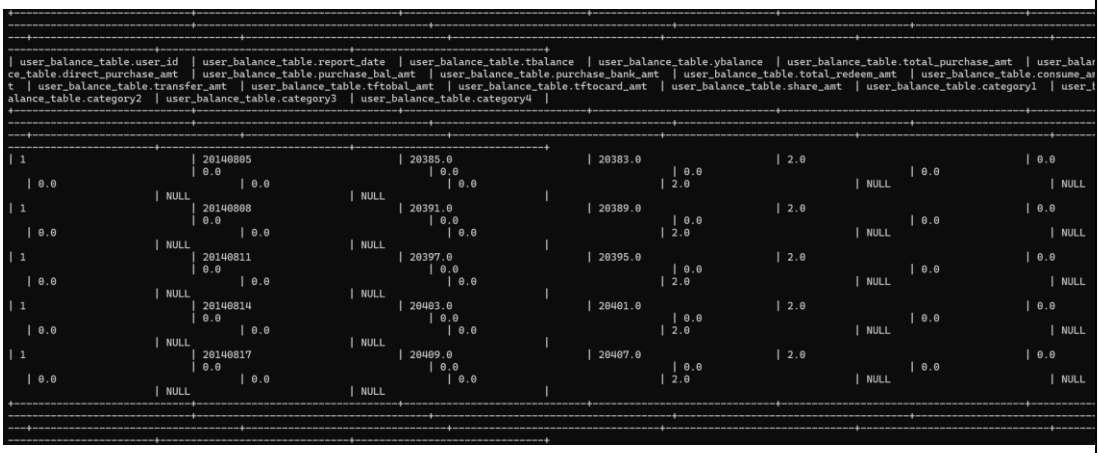
```
LOAD DATA LOCAL INPATH '/tmp/user-profile_table.csv' INTO TABLE user_profile_table;
LOAD DATA LOCAL INPATH '/tmp/user-balance_table.csv' INTO TABLE user_balance_table;
```

进行数据查询：

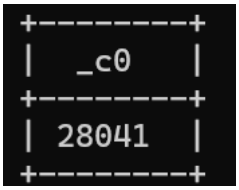
```
SELECT * FROM user_profile_table LIMIT 5;
```

user_profile_table.user_id	user_profile_table.sex	user_profile_table.city	user_profile_table.constellation
2	1	6411949	狮子座
12	1	6412149	摩羯座
22	1	6411949	双子座
23	1	6411949	双鱼座
25	1	6481949	双鱼座

```
SELECT * FROM user_balance_table LIMIT 5;
```

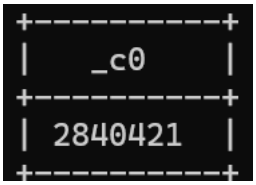


```
SELECT COUNT(*) FROM user_profile_table;
```



_c0
28041

```
SELECT COUNT(*) FROM user_balance_table;
```



_c0
2840421

Task2 基本数据查询：

1. 查询星座用户数量

```
SELECT
    constellation,
    COUNT (*) AS user_count
FROM
    user_profile_table
GROUP BY
    constellation
ORDER BY
    ser_count DESC;
```

constellation	user_count
天秤座	2910
天蝎座	2640
处女座	2497
狮子座	2387
射手座	2336
水瓶座	2336
摩羯座	2280
双鱼座	2265
巨蟹座	2202
金牛座	2108
双子座	2088
白羊座	1992

2. 查询特定日期的资金流入和流出情况

```
CREATE TABLE daily_flow_table AS
SELECT
    report_date,
    SUM(total_purchase_amt) AS total_inflow,
    SUM(total_redeem_amt) AS total_outflow
FROM
    user_balance_table
GROUP BY
    report_date;
SELECT * FROM daily_flow_table LIMIT 20;
```

daily_flow_table.report_date	daily_flow_table.total_inflow	daily_flow_table.total_outflow
20130701	3.2488348E7	5525022.0
20130702	2.903739E7	2554548.0
20130703	2.727077E7	5953867.0
20130704	1.8321185E7	6410729.0
20130705	1.1648749E7	2763587.0
20130706	3.6751272E7	1616635.0
20130707	8962232.0	3982735.0
20130708	5.7258266E7	8347729.0
20130709	2.6798941E7	3473059.0
20130710	3.0696506E7	2597169.0
20130711	4.4075197E7	3508800.0
20130712	3.4183904E7	8492573.0
20130713	1.5164717E7	3482829.0
20130714	2.2615303E7	2784107.0
20130715	4.8128555E7	1.3107943E7
20130716	5.0622847E7	1.1864981E7
20130717	2.9015682E7	1.0911513E7
20130718	2.4234505E7	1.1765356E7
20130719	3.3680124E7	9244769.0
20130720	2.0439079E7	4601143.0

Task3 数据聚合分析：

1. 按星座统计总购买量和赎回量

```
SELECT
    up.constellation,
    SUM(ub.total_purchase_amt) AS total_purchase,
    SUM(ub.total_redeem_amt) AS total_redeem_amt
FROM
    user_profile_table up
JOIN
    user_balance_table ub
ON
    up.user_id = ub.user_id
GROUP BY
    up.constellation;
```

up.constellation	total_purchase	total_redeem_amt
双子座	6.925452079E9	5.348014629E9
双鱼座	6.926003786E9	5.361595639E9
处女座	6.745578008E9	5.293924154E9
天秤座	9.88203188E9	7.989306193E9
天蝎座	1.0046042993E10	7.958903337E9
射手座	8.833197253E9	6.88952704E9
巨蟹座	7.359871686E9	5.992055191E9
摩羯座	7.418393917E9	5.724242577E9
水瓶座	8.096474293E9	6.492539527E9
狮子座	7.069134334E9	5.5723382E9
白羊座	6.655483821E9	5.012246637E9
金牛座	6.633419406E9	5.083604446E9

2. 按城市统计 2014 年 3月1日的平均余额

```
SELECT
up.city,
AVG(ub.tBalance) AS avg_bal
FROM
user_profile_table up
JOIN
user_balance_table ub
ON
up.user_id = ub.user_id
WHERE
ub.report_date = '20140301'
GROUP BY
up.city
ORDER BY
avg_bal DESC
LIMIT 10;
```

up.city	avg_bal
6281949	2795923.837298216
6301949	2650775.0664451825
6081949	2643912.7566638007
6481949	2087617.2136986302
6411949	1929838.5617977527
6412149	1896363.471625767
6581949	1526555.5551020408

Task4 复杂查询与分析：

1. 活跃用户分析

```
SELECT COUNT(DISTINCT user_id) AS total_active_usr
FROM (
    SELECT
        user_id,
        COUNT(DISTINCT report_date) AS total_active_day
    FROM
        user_balance_table
    WHERE
        report_date BETWEEN '20140801' AND '20140831'
    GROUP BY
        user_id
    HAVING
        total_active_day >= 5
) AS temp;
```

```
+------------------+
| total_active_usr |
+------------------+
| 12767            |
+------------------+
```

2. 统计每个城市总流量前 3 高的用户

```
WITH monthly_user_flow AS (
    SELECT
        user_id,
        SUM(total_purchase_amt + total_redeem_amt) AS flow_amount
    FROM
        user_balance_table
    WHERE
        report_date BETWEEN '20140801' AND '20140831'
    GROUP BY
        user_id
 ),
 user_ranking AS (
    SELECT
        profile.city,
        flow.user_id,
        flow.flow_amount,
        ROW_NUMBER() OVER (PARTITION BY profile.city ORDER BY flow.flow_amount
DESC) AS user_rank
    FROM
        monthly_user_flow flow
    JOIN
```

```
            user_profile_table profile
        ON
            flow.user_id = profile.user_id
 )
SELECT
        city,
        user_id,
        flow_amount
FROM
        user_ranking
WHERE
        user_rank <= 3
ORDER BY
        city,
        user_rank;
```

city	user_id	flow_amount
6081949	27235	1.0847568E8
6081949	27746	7.6065458E7
6081949	18945	5.5304049E7
6281949	15118	1.49311909E8
6281949	11397	1.24293438E8
6281949	25814	1.04428054E8
6301949	2429	1.09171121E8
6301949	26825	9.537403E7
6301949	10932	7.4016744E7
6411949	662	7.5162566E7
6411949	21030	4.9933641E7
6411949	16769	4.9383506E7
6412149	22585	2.00516731E8
6412149	14472	1.3826279E8
6412149	25147	7.0594902E7
6481949	12026	5.1161825E7
6481949	670	4.9626204E7
6481949	14877	3.4488733E7
6581949	9494	3.8854436E7
6581949	26876	2.3449539E7
6581949	21761	2.113644E7