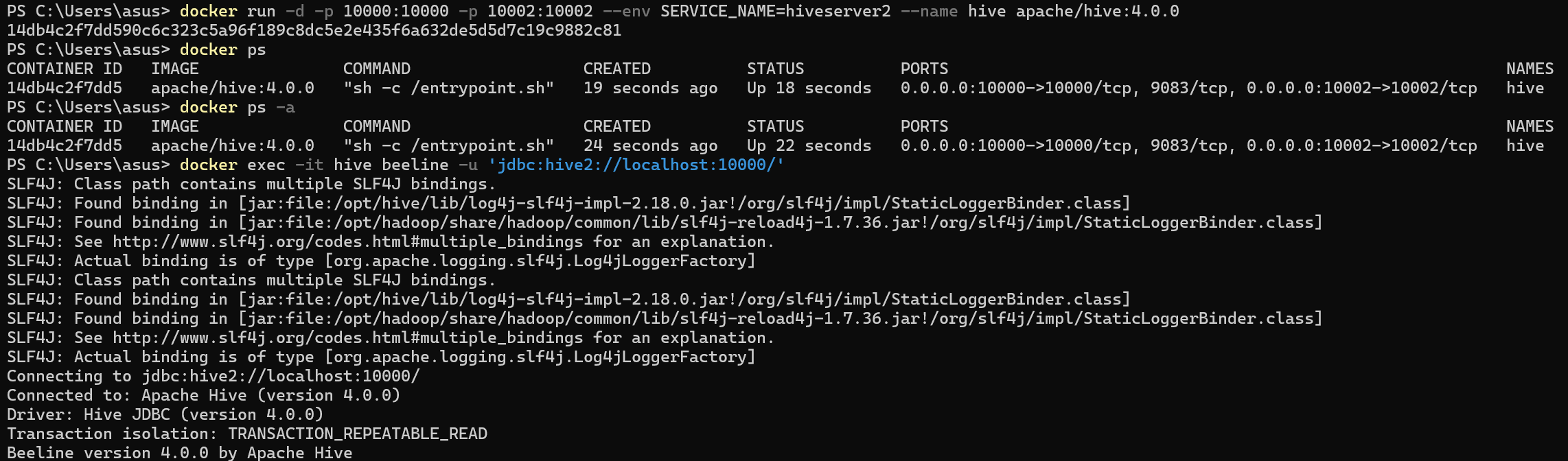
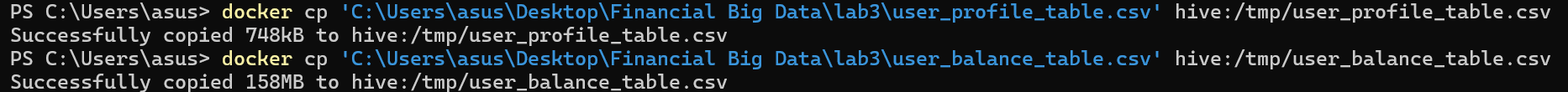
**Hive下载：**

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



**数据准备：**

将user\_profile\_table.csv和user\_balance\_table.csv复制到hive的/tmp目录下



**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; |
| SELECT \* FROM user\_balance\_table LIMIT 5; |
| SELECT COUNT(\*) FROM user\_profile\_table; |
| SELECT COUNT(\*) FROM user\_balance\_table; |

**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; |

1. 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; |

**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; |

1. 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; |

**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; |

1. 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; |