

Project Report - Data Analysis and Integration 22/23

a) SQL instructions to create the Data Warehouse

• Create a **database** for the DW:

```
DROP DATABASE IF EXISTS airports_dw;

CREATE DATABASE airports_dw;

USE airports_dw;
```

• Create table for the **airline dimension**:

```
CREATE TABLE dim_airline (
    AIRLINE_ID SMALLINT(6),
    AIRLINE_NAME VARCHAR(255),
    PRIMARY KEY (AIRLINE_ID)
);
```

• Create table for the **airplane dimension**:

```
CREATE TABLE dim_airplane (
AIRPLANE_ID INT(11),
TYPE_ID INT(11),
PRIMARY KEY (AIRPLANE_ID)
);
```

• Create table for the **airport dimension**:

```
CREATE TABLE dim_airport (
   AIRPORT_ID SMALLINT(6),
   AIRPORT_NAME VARCHAR(255),
   CITY VARCHAR(255),
   COUNTRY VARCHAR(255),
   PRIMARY KEY (AIRPORT_ID)
);
```

• Create table for the **time dimension**:

```
CREATE TABLE dim_time (
   TIME_ID DATETIME,
   YEAR_ID INT,
   MONTH_ID INT,
   MONTH_NAME VARCHAR(255),
   DAY_ID INT,
   PRIMARY KEY (TIME_ID)
);
```

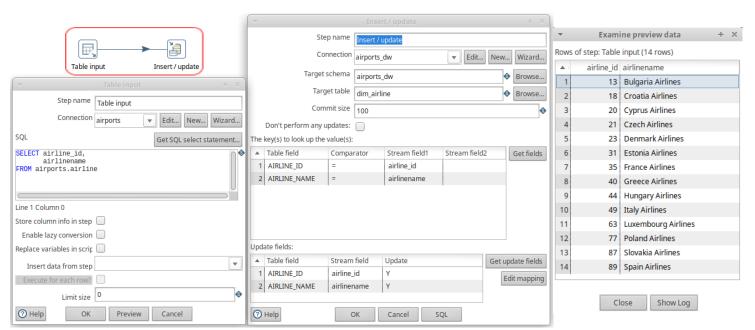


• Create a **fact table** for the **flights**:

```
CREATE TABLE fact_flight (
   FLIGHT_ID INT,
   AIRPORT_ORIG SMALLINT(6),
   AIRPORT_DEST SMALLINT(6),
   DEPARTURE DATETIME,
   ARRIVAL DATETIME,
   AIRLINE_ID SMALLINT(6),
   AIRPLANE ID INT(11),
   N PASSENGERS INT,
   REVENUE DECIMAL(10,2),
   PRIMARY KEY (FLIGHT ID),
   FOREIGN KEY (AIRPORT_ORIG) REFERENCES dim_airport (AIRPORT_ID),
   FOREIGN KEY (AIRPORT_DEST) REFERENCES dim_airport (AIRPORT_ID),
   FOREIGN KEY (DEPARTURE) REFERENCES dim_time (TIME_ID),
   FOREIGN KEY (ARRIVAL) REFERENCES dim_time (TIME_ID),
   FOREIGN KEY (AIRLINE_ID) REFERENCES dim_airline (AIRLINE_ID),
   FOREIGN KEY (AIRPLANE_ID) REFERENCES dim_airplane (AIRPLANE_ID)
```

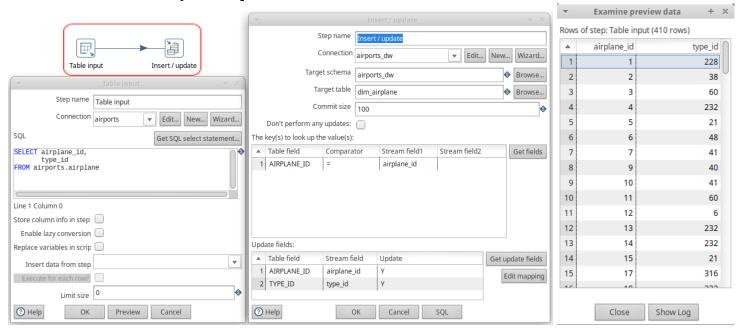
b) PDI transformations

• Populate **airline dimension** table:

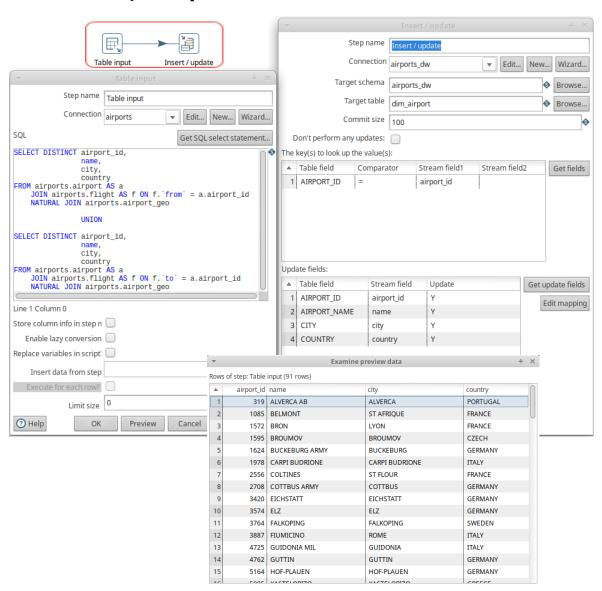




• Populate airplane dimension table:



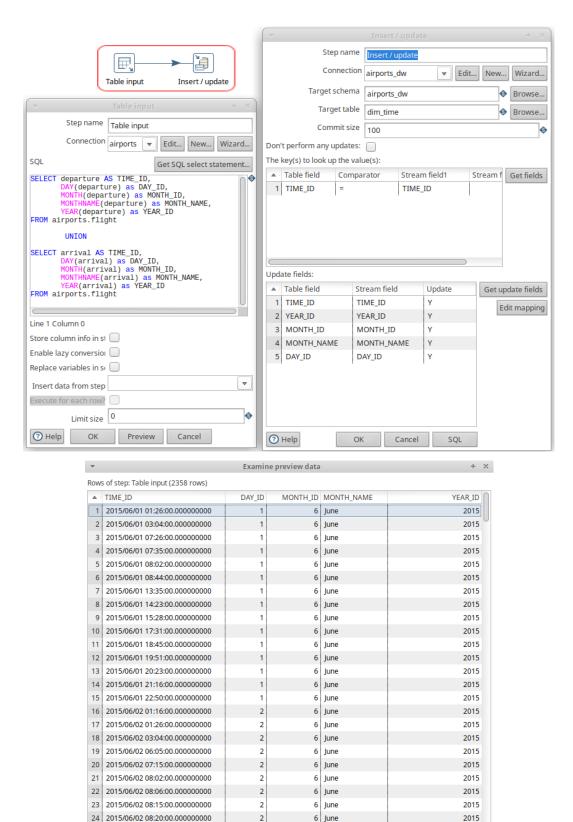
• Populate airport dimension table:





• Populate **time dimension** table:

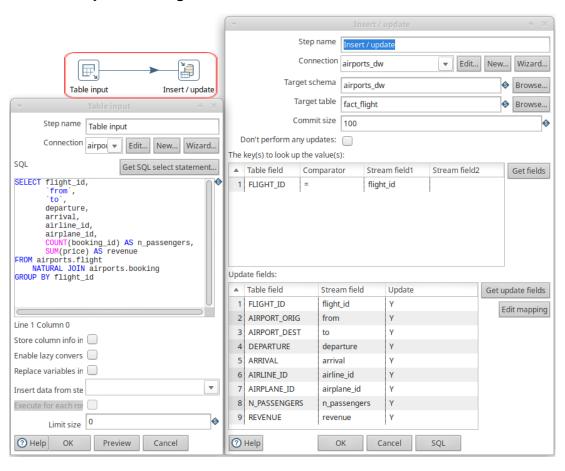
2015/06/02 08:44:00.000000000

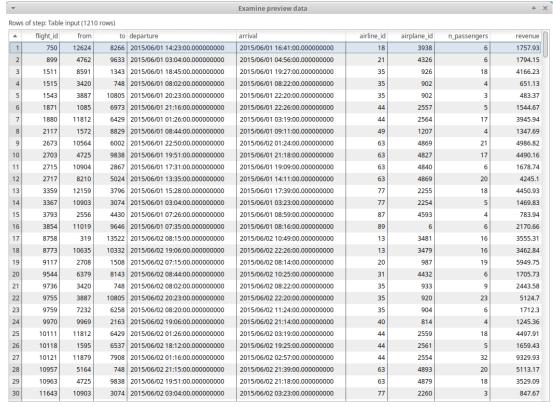


6 June



Populate fact_flight table







c) XML Data Cube

```
<Schema name="airports_dw">
 <Cube name="Flights" visible="true" cache="true" enabled="true">
   <Table name="fact_flight">
   </Table>
   <Dimension type="StandardDimension" visible="true" foreignKey="AIRLINE_ID" highCardinality="false" name="Airline">
     <Hierarchy name="Airline Hierarchy" visible="true" hasAll="true" allMemberName="All Airlines" primaryKey="AIRLINE_ID">
       <Table name="dim_airline">
       </Table>
      <Level name="Airline Name" visible="true" column="AIRLINE NAME" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Tevel>
     </Hierarchy>
   <Dimension type="StandardDimension" visible="true" foreignKey="AIRPLANE ID" highCardinality="false" name="Airplane">
     <Hierarchy name="Airplane Hierarchy" visible="true" hasAll="true" allMemberName="All Airplanes" primaryKey="AIRPLANE_ID">
      <Table name="dim airplane">
      <Level name="Airplane Type" visible="true" column="TYPE ID" type="Integer" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
     </Hierarchy>
   <Table name="dim_time">
       </Table>
       <Level name="Year" visible="true" column="YEAR_ID" type="Integer" uniqueMembers="false" levelType="TimeYears" hideMemberIf="Never">
       </Level>
       </Level>
       <Level name="Day" visible="true" column="DAY ID" type="Integer" uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">
       </Level>
     </Hierarchy
   </Dimension>
   .

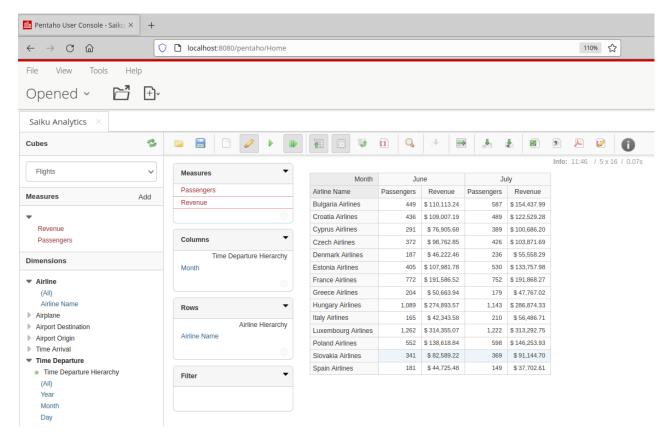
«Dimension type="TimeDimension" visible="true" foreignKey="DEPARTURE" highCardinality="false" name="Time Departure">
     <Hierarchy name="Time Departure Hierarchy" visible="true" hasAll="true" allMemberName="All Time_Departure" primaryKey="TIME_ID">
      <Table name="dim time">
       </Table>
       .
- «Level name="Year" visible="true" column="YEAR ID" type="Integer" uniqueMembers="false" levelType="TimeYears" hideMemberIf="Never">
      </Tevel>
       <Level name="Month" visible="true" column="MONTH NAME" ordinalColumn="MONTH ID" type="String" uniqueMembers="false" levelType="TimeMonths" hideMemberIf="Never">
      <Level name="Day" visible="true" column="DAY_ID" type="Integer" uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">
     </Hierarchy>
   <Table name="dim_airport">
       <Level name="Country" visible="true" column="COUNTRY" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
       <Level name="City" visible="true" column="CITY" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
      <Level name="Airport Name" visible="true" column="Airport NAME" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
       </r>
     </Hierarchy>
   </Dimension>
   <Dimension type="StandardDimension" visible="true" foreignKey="AIRPORT ORIG" highCardinality="false" name="Airport Origin">
     <Hierarchy name="Airport Origin Hierarchy" visible="true" hasAll="true" allMemberName="All Airport Origin" primaryKey="AIRPORT ID">
      <Table name="dim_airport">
       </Table>
       <Level name="Country" visible="true" column="COUNTRY" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
       <Level name="City" visible="true" column="CITY" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
      Clevel name="Airport Name" visible="true" column="AIRPORT NAME" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
     </Hierarchy>
   <Measure name="Revenue" column="REVENUE" datatype="Numerio" formatString="$ #,###.00" aggregator="sum" visible="true">
   </Measure>
   <Measure name="Passengers" column="N_PASSENGERS" datatype="Integer" formatString="#,####" aggregator="sum" visible="true">
   </Measure>
</Schema>
```



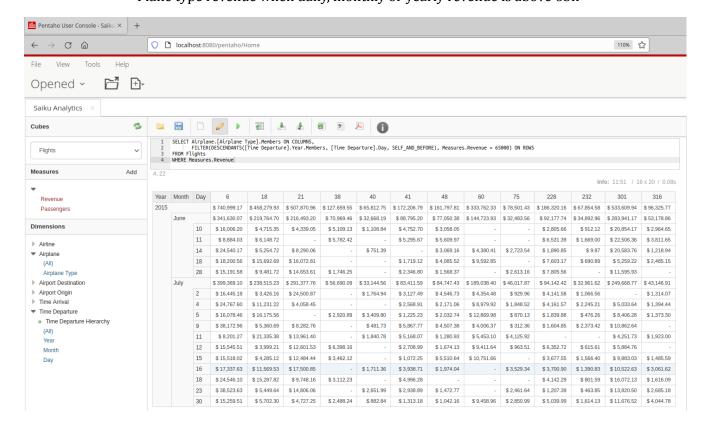
d) & e) Pentaho Server Queries

--- airports.sql ---

• Passengers and revenue by airline and month

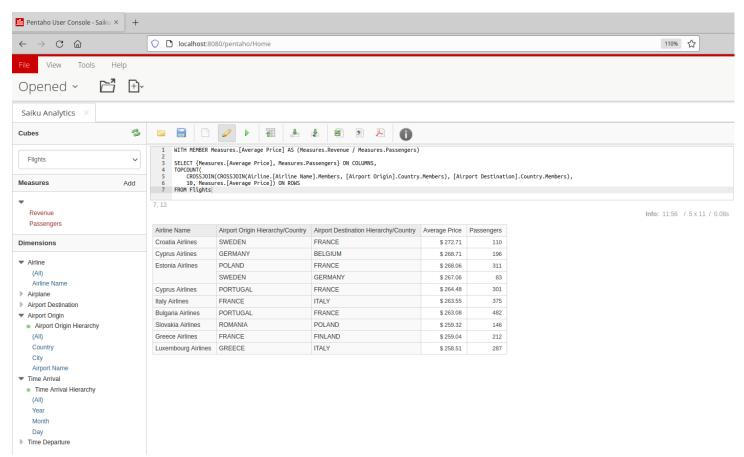


• Plane type revenue when daily, monthly or yearly revenue is above 65k





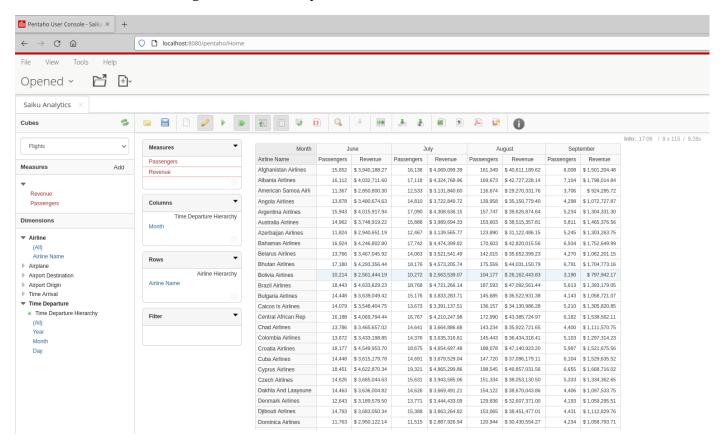
• Top 10 routes with the highest average price per person and their information (airline, origin and destination countries, average price and total of passengers)



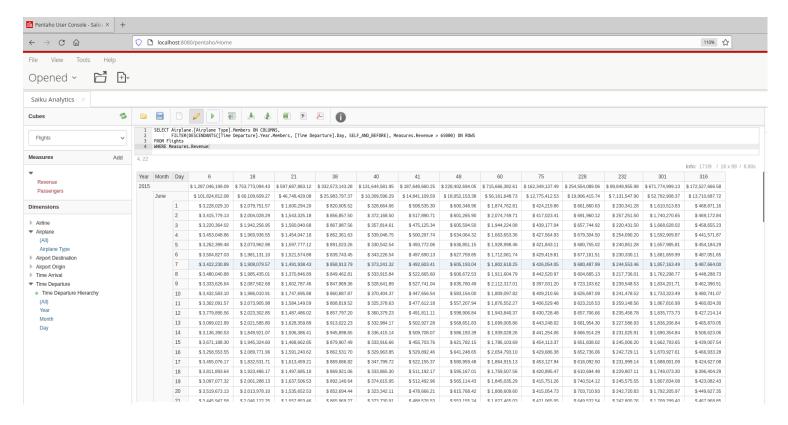


--- airports-large.sql ---

• Passengers and revenue by airline and month



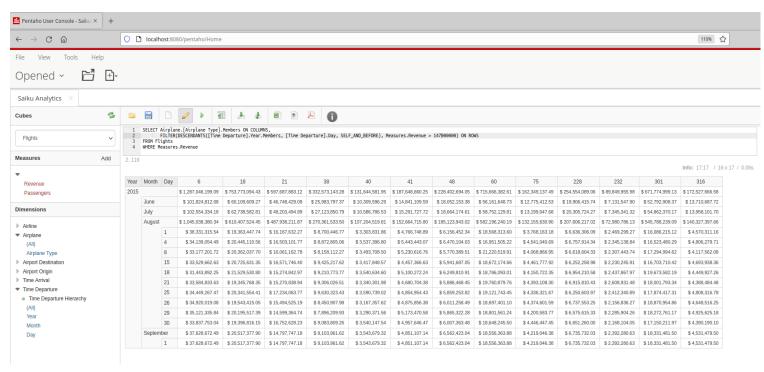
• Plane type revenue when daily, monthly or yearly revenue is above 65k



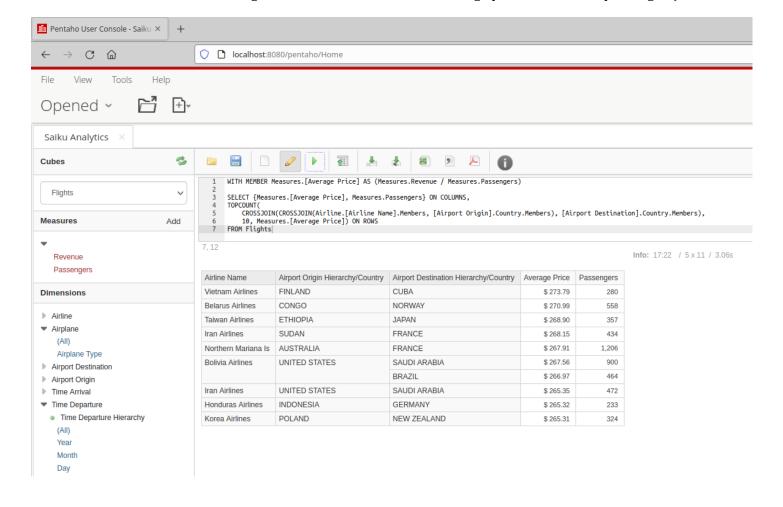


<u>Comment:</u> We noticed that 65k was a negligible value for airports-large so we decided to test the query with a much higher value that made sense (147M)

• Plane type revenue when daily, monthly or yearly revenue is above 147M



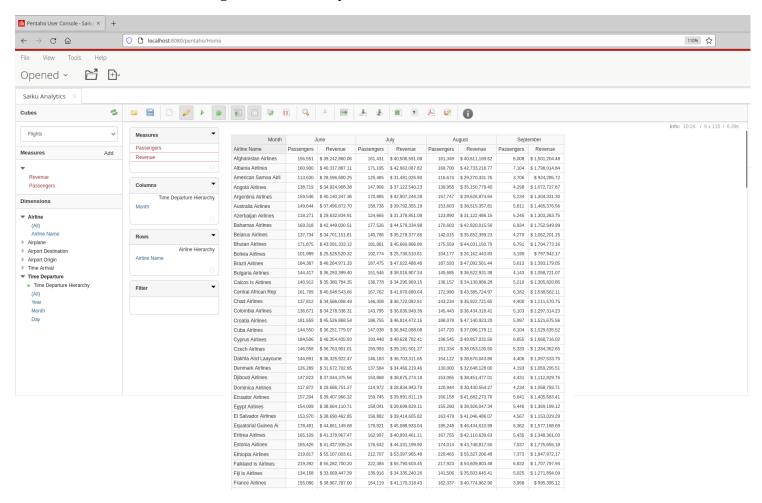
• Top 10 routes with the highest average price per person and their information (airline, origin and destination countries, average price and total of passengers)



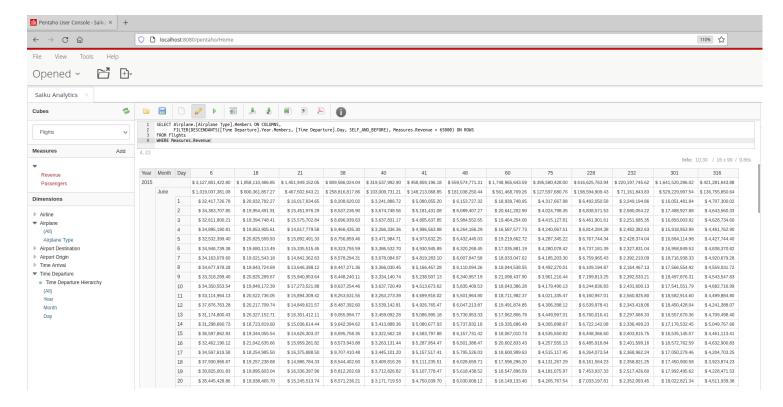


--- airports-large-extra.sql ---

• Passengers and revenue by airline and month

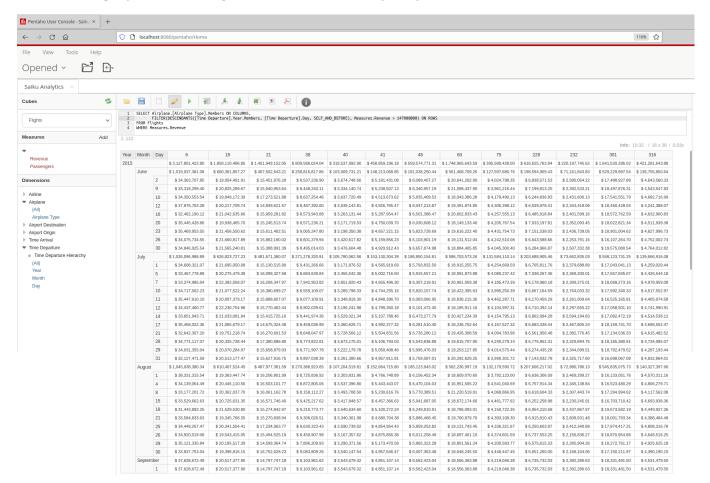


• Plane type revenue when daily, monthly or yearly revenue is above 65k

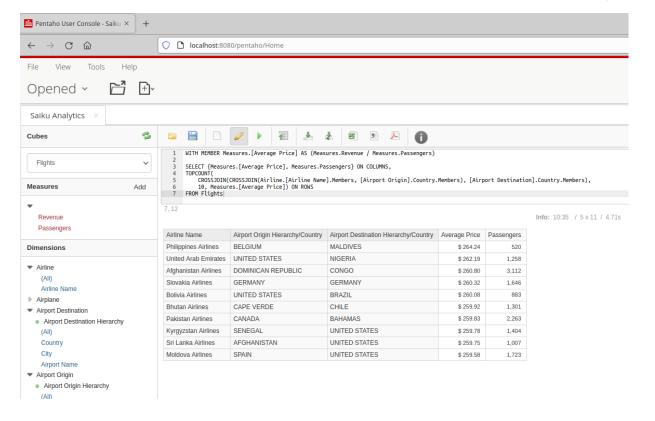




<u>Comment:</u> Once again, we noticed that 65k was a negligible value for airports-large-extra so we decided to test the query with a much higher value that made sense (147M)



• Top 10 routes with the highest average price per person and their information (airline, origin and destination countries, average price and total of passengers)





Final Observations:

- We decided to create only one table for airports and time since we tried to minimize the data redundancy and optimize the Insert/Update operations.
- We noticed that if we took advantage of SQL statements in the Table Inputs (paragraph b.), the performance was enhanced quite significantly, especially on the larger databases.
 After trying different methods, the usage of too many PDI steps proved to negatively affect the efficiency.
- The larger databases were slower to load. The airports-large-extra took an average time of 38 minutes and 25 seconds to populate the Data Warehouse! The queries on both these larger Data Warehouse were also significantly slower, as shown in the screenshots.
- In the queries, we can observe that:
 - As expected, the revenue and numbers of passengers by airline and month change significantly with the size of the dataset (the bigger databases store more flights)
 - The revenue threshold that we established for the smaller dataset needed to be changed for the larger ones in order for the query to have meaning. Larger datasets have increased revenue.
 - The average price of the most expensive routes does not vary a lot while the total number of passengers/customers do.