**FIT 5137 Semester 2 2018**

**Group Assignment**

**Student:**

**Xiaofen Pan**

**Sanan Ibrahimov**



**ASSIGNMENT COVER SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student’s name** | Xiaofen Pan | Sanan Ibrahimov | |
| **ID number** | 27592987  28237927 | **Phone** |  |
| **Unit name** | Database analysis and processing | **Unit code** | FIT 5137 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Title of assignment** | **Assignment S2 2018** | | |
| **Lecturer/tutor** | **Assoc. Prof. David Taniar / Zhou Shao** | | |
| **Is this an authorised group assignment?**  **Yes  No**  If this submission is a group assignment, each student must attach their own signed cover sheet to the assignment. | | | |
| **Has any part of this assignment been previously submitted as part of another unit/course?**  **Yes**  **No** | | | |
| **Tutorial/laboratory day & time** | | Wednesday 18:00 – 20:00 | |
| **Due date** 17 October 2018 | | | **Date submitted** 14 October 2018 |

All work must be submitted by the due date. If an extension of time to submit work is required, a [Special Consideration Application (In-semester Assessment Task)](http://www.monash.edu.au/connect/assets/docs/forms/in-semester.pdf) must be submitted.

**Has an extension been approved? Yes No If yes, please give the new submission date ….…/..…./…….**

Please note that it is your responsibility to retain copies of your assessments.

|  |
| --- |
| ***Intentional plagiarism or collusion amounts to cheating under Part 7 of the*** [***Monash University (Council) Regulations***](http://monash.edu/legal/legislation/current-statute-regulations-and-related-resolutions/)***.***  **Plagiarism**: means taking and using another person’s ideas or manner of expressing them and passing them off as one’s own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published or unpublished works).  **Collusion**: means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.  Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Chief Examiner, who may disallow the work concerned by prohibiting assessment or refer the matter to the Associate Dean Teaching and Learning. |

|  |
| --- |
| **Student Statement:**   * I have read the University’s [Student Academic Integrity Policy](http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-policy.html) and the University’s [Student Academic Integrity: Managing Plagiarism and Collusion Procedures](http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-managing-plagiarism-collusion-procedures.html). * I understand the consequences of engaging in plagiarism and collusion as described in [Part 7 of the Monash University (Council) Regulations](http://monash.edu/legal/legislation/current-statute-regulations-and-related-resolutions/). * I have taken proper care of safeguarding this work and made all reasonable effort to ensure it could not be copied. * I acknowledge that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:  1. provide to another member of faculty; and/or 2. submit it to a plagiarism checking service; and/or 3. submit it to a plagiarism checking service which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.  * I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.   *Signature.........* *..................... ...................................................... Date……14 October 2018………………………………* |
| **Privacy Statement**  The information on this form is collected for the primary purpose of assessing your assignment. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: [privacyofficer@adm.monash.edu.au](mailto:privacyofficer@adm.monash.edu.au) |

**Contribution Declaration Form**

**(to be completed by all team members)**

**Please fill in the form with the contribution from each student towards the assignment.**

**1 NAME AND CONTRIBUTION DETAILS**

|  |  |  |
| --- | --- | --- |
| Student ID | Student Name | Contribution Percentage |
| 27592987 | Xiaofen Pan | 50% |
| 28237927 | Sanan Ibrahimov | 50% |

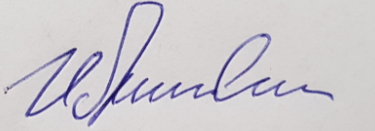
**Lists of parts: everything was implemented in collaboration, where both contributors worked on each task together.**

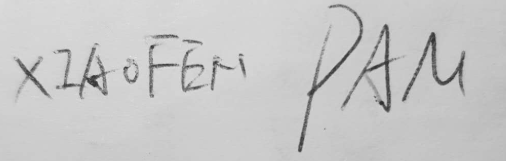
**2 DECLARATION**

**We declare that:**

* The information we have supplied in or with this form is complete and correct.
* We understand that the information we have provided in this form will be used for individual assessment of the assignment.

**3 SIGNATURE**





**Signatures**

DayMonth Year

14 / Oct/2018

**Date**

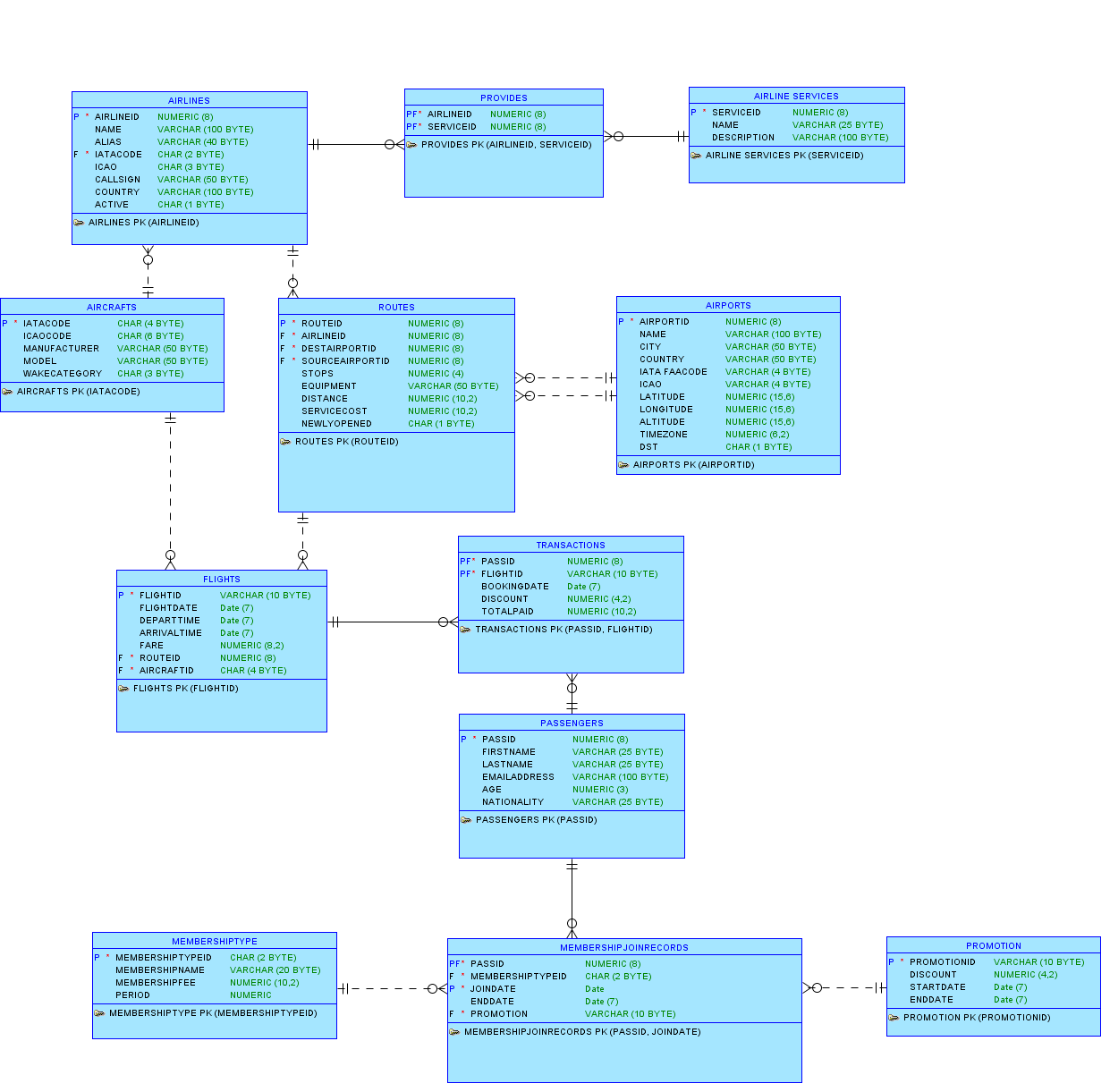
**Oracle Account:**

**ID: S27592987**

**Password: student**

**C.1**

**(a) ERD**



**b) Data Cleaning**

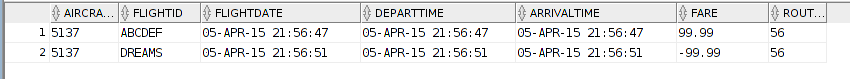
**1)--Is FLIGHTS table have any illegal entries that is not in Aircraft table?**

Select \* from FLIGHTS

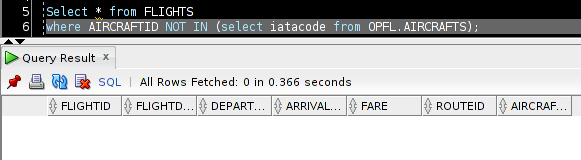
where AIRCRAFTID NOT IN (select iatacode from OPFL.AIRCRAFTS);

--Yes

--Two flights were accomplished by plane AIRCRAFTID 5137 that are not in aircrafts table

Before:

After:

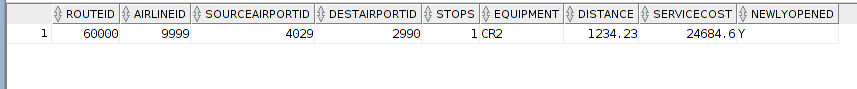


**2)--is there any routes that has airlineid entry not existing in airlines table**

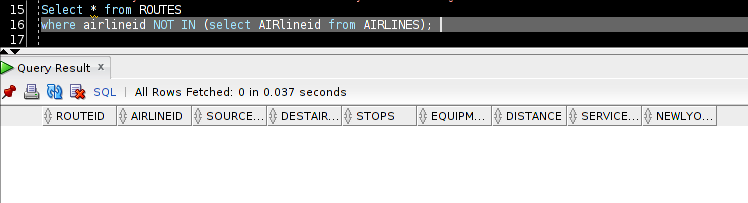
Select \* from ROUTES

where airlineid NOT IN (select AIRlineid from AIRLINES);

Before:



Initial number of airlines was 532. One extra than on requirement paper. One illegal entry (with airlineid 9999) was found and deleted.

After:

**3) --Check if there is any route with distance less than 0.**

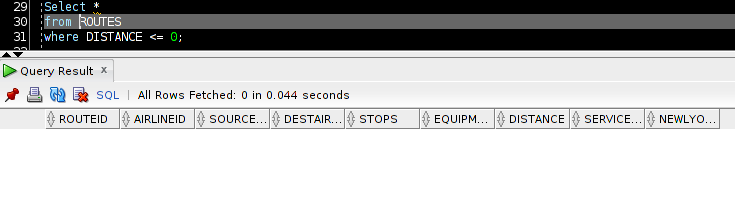
Select \* from ROUTES where DISTANCE <= 0;

Before:



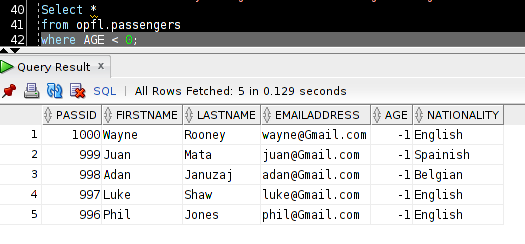
RouteID "-1" was having distance "-1000" and was deleted.

After:



**4)--find if there are any illegal entries with negative passenger age**

Select \* from passengers where AGE < 0;

Before:

After:

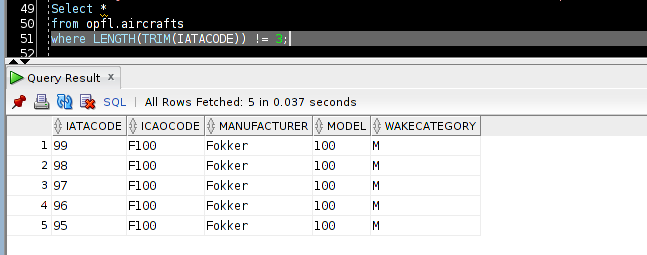
delete from passengers where AGE < 0;

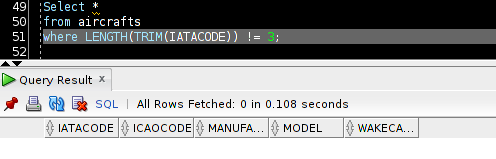


**5)--deleting aircrafts with Erroneous IATACODE that does not match with requirements of 3 symbol length**

Select \* from aircrafts where LENGTH(TRIM(IATACODE)) != 3;

Before:

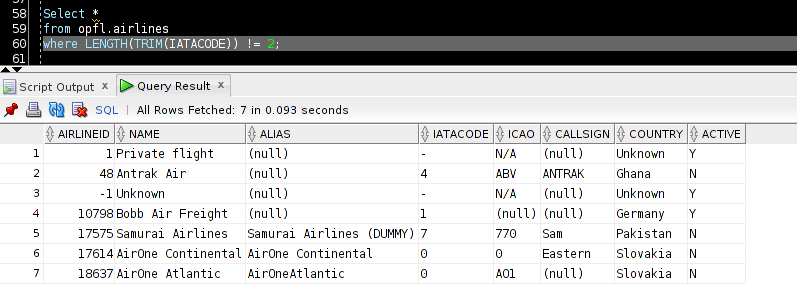


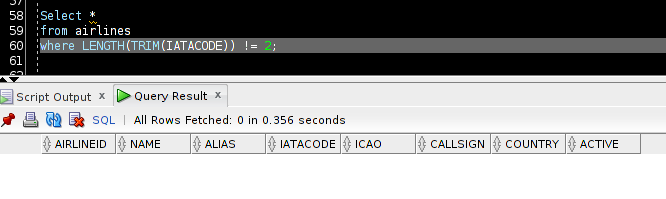
After:

Delete from aircrafts where LENGTH(TRIM(IATACODE)) != 3;

**6) Finding all airlines with wrong IATA code ( unequal to 2 chars)**

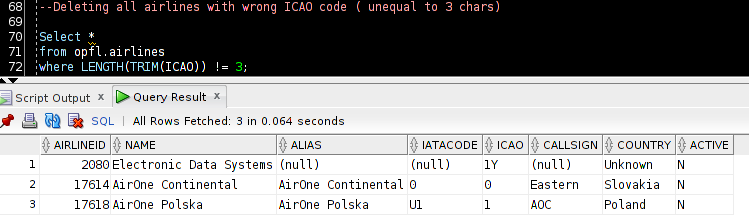
Before:

Select \* from airlines where LENGTH(TRIM(IATACODE)) != 2;

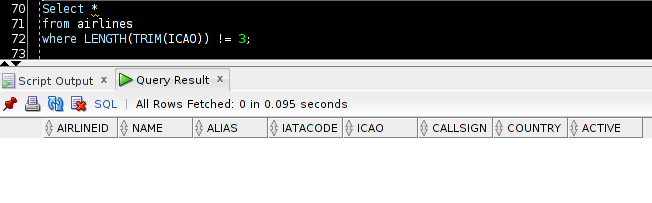
After:

**7)--Finding entries with ICAO code that does not match 4 char length**

Before:

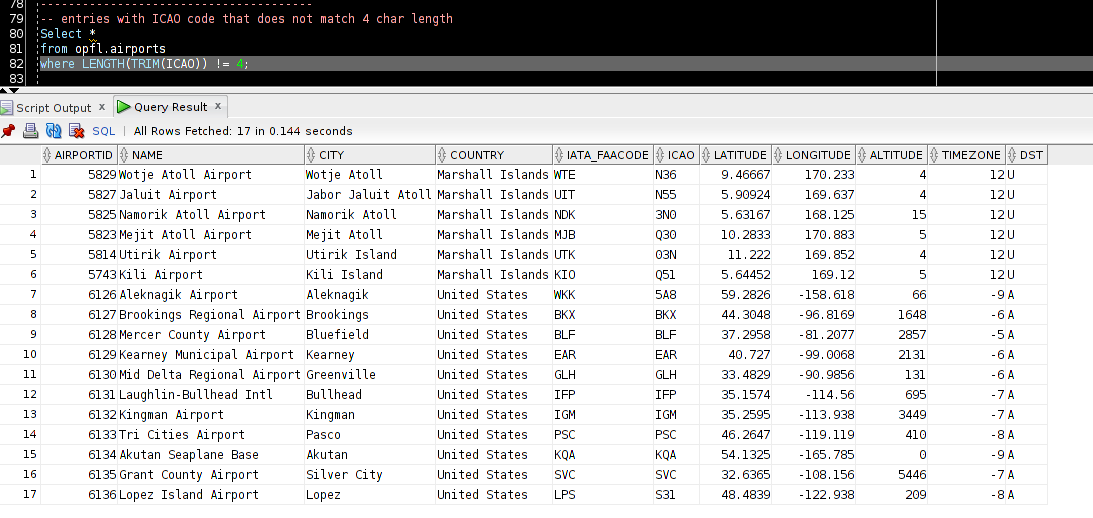


After:



**8)--Finding entries with ICAO code that does not match 4 char length**

before:

Select \* from airports where LENGTH(TRIM(ICAO)) != 4;

After:

Delete

from airports

where LENGTH(TRIM(ICAO)) != 4;



d) For Our snowflake Schema no Hierarchy structure was chosen in order to simplify the schema and create less number of tables. This will subsequently lead to less number of join processing when producing reports from star schema. Also considering that creating hierarchy would require normalised tables versus to un-normalized in Non-hierarchy model and final report from requirements paper did not include drill-down style reports we decided to keep it simple as per requirements.

e) We used SCD type 4 to reflect the discounts history as we noticed that despite that price of membership is stable, however discounts history is changing over time. For the purposes of producing the required reports we needed to keep all the history of discounts over time. Just keeping last and previous was not enough as changes were frequent enough. So the choice was SCD type 4 as the one that reflects all the history of changes.

f) The difference between version 1 and 2 is the aggregation level. First we created the star schema with high aggregation. Then for the purposes of lower level we added Passenger Desc Dim to reduce the level of aggregation of Transaction and membership sales fact tables. For the same purposes Route fact table on Level 0 connects directly to Airport dimensions instead of City. This provides the minimum level of aggregation.

**C.2**

**a) SQL statements for Version\_1**

-- Route Fact table

create table route\_fact

as select s.city as Source\_CityID, d.city as Dest\_CityID, p.airlineid,

count(p.routeid) as total\_number\_of\_routes,

sum(p.distance) as total\_route\_distance,

sum(p.servicecost) as total\_service\_cost

from Routes p, airports s, airports d

where p.sourceairportid = s.airportid and p.destairportid = d.airportid

group by s.city, d.city, p.airlineid;

--Create Source\_City\_DIM

create table Source\_City\_DIM as select City,Country,TimeZone, DST

from airports;

--Create Destination\_City\_DIM

create table Destination\_City\_DIM as select City,Country,TimeZone, DST

from airports;

--Create Source\_Country\_DIM

create table Source\_Country\_DIM as select Country,TimeZone, DST

from airports;

--Create Destination\_Country\_DIM

create table Destination\_Country\_DIM as select Country,TimeZone, DST

from airports;

----------------------------------------------

--Creating Airline\_DIM

Create Table Airline\_DIM As

Select

AL.AirlineID,

AL.Name,

1.0/count(PR.ServiceID) As Service\_Weight,

LISTAGG (PR.ServiceID, '\_') Within Group

(Order By PR.ServiceID) As StoreGroupList

From opfl.airlines AL, opfl.provides PR

Where AL.AirlineID = PR.AirlineID

Group By AL.AirlineID, AL.Name;

-------------------------------------------------

--Creating Airline Service Bridge and Service DIM

create table Airline\_Service\_Bridge

as select \* from opfl.provides;

-------------------------------------------------

create table Service\_DIM

as select \* from opfl.Airline\_Services;

-------------------------------------------------

----------------------------------------------

-----source airport DIM

create table source\_airport\_dim as

select \* from airports where airportid in (select sourceairportid from routes);

ALTER TABLE source\_airport\_dim

RENAME COLUMN airportid TO Sourceairportid;

ALTER TABLE source\_airport\_dim DROP COLUMN country;

select \* from source\_airport\_dim;

----destination airport DIM

create table destination\_airport\_dim as

select \* from airports where airportid in (select destairportid from routes);

ALTER TABLE destination\_airport\_dim

RENAME COLUMN airportid TO Destinationairportid;

ALTER TABLE destination\_airport\_dim DROP COLUMN country;

select \* from destination\_airport\_dim;

----flight type dim

create table flight\_type\_dim

(FlightTypeID varchar2(10),

TypeName varchar2(50));

insert into flight\_type\_dim values('F1','Domestic');

insert into flight\_type\_dim values('F2','International');

------------------------------------------------------------------

--Creating Other DIM tables

------------------------------------------------------------------

--create Flight\_class\_DIM table

CREATE TABLE Flight\_Class\_DIM

( class\_id varchar2(3) NOT NULL,

class\_name varchar2(30) NOT NULL,

CONSTRAINT classid\_pk PRIMARY KEY (class\_id));

--insert values

INSERT ALL

INTO Flight\_Class\_DIM (class\_id, class\_name) VALUES ('C1', 'First Class')

INTO Flight\_Class\_DIM (class\_id, class\_name) VALUES ('C2', 'Business Class')

INTO Flight\_Class\_DIM (class\_id, class\_name) VALUES ('C3', 'Economy Class')

SELECT \* FROM dual;

--Create table FlightDate\_DIM

create table Flight\_Date\_DIM as

select distinct to\_char(Flightdate, 'DDMMYY') as TimeID,

to\_char(Flightdate, 'DD') as Flight\_Day,

to\_char(Flightdate, 'Month') as Flight\_Month,

to\_char(Flightdate, 'YYYY') as Flight\_Year

from Flights;

----------------------------------------------------------------

-- Create Temp Transaction Fact

create table temp\_Trans\_fact

as select r.sourceairportid,

r.destairportid,

to\_char(fl.flightdate,'DDMMYY') as flightdateid,

tr.totalpaid,

fl.fare,

a1.country as sourceCountry,

ap.country as destCountry,

al.airlineid,

tr.passid,

tr.flightid,

p.age

from routes r, flights fl, transactions tr, airports a1, airports ap, airlines al, passengers p

where r.sourceairportid = a1.airportid

and r.destairportid = ap.airportid

and r.airlineid = al.airlineid

and r.routeid = fl.routeid

and fl.flightid = tr.flightid

and tr.passid = p.passid;

-----------------------------------------------------------

--Altering Transaction Fact

alter table temp\_Trans\_fact

add(class\_id varchar2(20));

alter table temp\_Trans\_fact

add(flighttype\_id varchar2(20));

--------------------------------------------------

update temp\_Trans\_fact

set class\_id='C1'

where totalpaid>= 2\*fare;

update temp\_Trans\_fact

set class\_id = 'C2'

where totalpaid>= 1.5\*fare and totalpaid<2\*fare;

update temp\_Trans\_fact

set class\_id = 'C3'

where totalpaid < 1.5\*fare;

-------------------------------------------------------

update temp\_Trans\_fact

set flighttype\_id = 'F1'

where sourceCountry = destCountry;

update temp\_Trans\_fact

set flighttype\_id = 'F2'

where sourceCountry != destCountry;

-------------------------------------------------------------

--Creating Transaction Fact

----------------------------------------------

--Creating Transaction Fact Table

drop table Transaction\_fact;

create table Transaction\_fact

as select

flightdateid,

class\_id,

flighttype\_id,

sourceairportid,

destairportid,

airlineid,

count(passid) as Total\_Number\_Of\_transactions,

count(passid) as number\_of\_passengers,

sum(totalpaid) as total\_paid,

sum(age) as total\_passenger\_age,

sum(totalpaid - fare) as Agent\_profit

from temp\_Trans\_fact

group by sourceairportid, destairportid,flightdateid, class\_id, flighttype\_id, airlineid;

-- Memebership Sales Join Fact

-----------------------------------------------------------------------------------------------

---Passenger\_DIM

create table passenger\_dim

(Pass\_type\_ID varchar2(10),

Pass\_type\_desc varchar2(20));

insert into passenger\_dim values('P1','Children');

insert into passenger\_dim values('P2','Teenager');

insert into passenger\_dim values('P3','Adult');

insert into passenger\_dim values('P4','Elder');

select \* from passenger\_dim;

---Membership type DIM

create table Membership\_type\_DIM as

select distinct

MembershiptypeID,

MembershipName,

membershipfee

from Membershiptype;

-----------------------------------------------

---Membership Discount DIM

create table Membership\_Discount\_DIM as

select

distinct m.membershiptypeid,

p.startdate,

p.enddate,

p.discount

from membershipjoinrecords m, promotion p

where m.promotion = p.promotionid;

---------------------------------

--membership join dim

create table Membership\_Join\_Date\_DIM

as select distinct to\_char(joindate,'YYYYMM') as joindateid,

to\_char(joindate,'YYYY') as year,

to\_char(joindate,'MM') as month

from membershipjoinrecords;

----------------------------------------------

--Createing Memebership Sales fact

drop table MembershipSales\_Fact\_TMP;

create table MembershipSales\_Fact\_TMP as

select to\_char(m.joindate, 'YYYYMM') as joindateid,

p.age,

m.membershipTypeID,

t.membershipfee,

m.passid

from membershipjoinrecords m, passengers p, membershiptype t

where m.passid = p.passid

and m.membershiptypeid = t.membershiptypeid;

alter table MembershipSales\_Fact\_TMP

add(passenger\_type\_id varchar2(50));

update MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P1'

where age < 11;

update MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P2'

where age >= 11 and age <= 17;

update MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P3'

where age >= 18 and age <= 60;

update MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P4'

where age > 60 ;

---------------------------------------------

create table Membership\_Sales\_Fact

as select joindateid,

passenger\_type\_id,

membershiptypeid,

sum(membershipfee) as total\_membership\_sale,

count(passid) as total\_number\_of\_members

from MembershipSales\_Fact\_TMP

group by joindateid, passenger\_type\_id, membershiptypeid;

--------------------------------------------------------------------

-----LEVEL\_0 Star Schema, Without Aggregation-----

--------------------------------------------------------------------

---Create L0\_Source\_Airport\_Dim table--

create table L0\_source\_airport\_dim

as select \* from source\_airport\_dim ;

---Create L0\_Destination\_Airport\_Dim table--

create table L0\_Destination\_airport\_dim

as select \* from Destination\_airport\_dim ;

---Create L0\_Source\_City\_DIM table--

create table L0\_Source\_City\_DIM

as select \* from Source\_City\_DIM ;

---Create L0\_Destination\_City\_DIM table--

create table L0\_Destination\_City\_DIM

as select \* from Destination\_City\_DIM ;

---Create L0\_Source\_Country\_DIM table--

create table L0\_Source\_Country\_DIM

as select \* from Source\_Country\_DIM ;

---Create L0\_\_Destination\_DIM table--

create table L0\_Destination\_Country\_DIM

as select \* from Destination\_Country\_DIM ;

--Create L0\_L0\_Airline\_Service\_Bridge table

create table L0\_Airline\_Service\_Bridge

as select \* from Airline\_Service\_Bridge;

--Create L0\_Service\_DIM table

create table L0\_Service\_DIM

as select \* from Service\_DIM;

--Create L0\_Airline\_DIM table

Create Table L0\_Airline\_DIM As

Select \* from Airline\_DIM;

--Level 0 Routr fact with no aggregation

create table L0\_Route\_fact

as select sourceairportid,

destairportid,

airlineid,

distance as route\_distance,

servicecost as service\_cost

from Routes;

----L0\_flight type dim

create table L0\_Flight\_type\_dim

as select \* from Flight\_type\_dim;

--create L0\_Flight\_class\_DIM table

CREATE TABLE L0\_Flight\_Class\_DIM

as select \* from Flight\_Class\_DIM;

--Create table FlightDate\_DIM

create table L0\_Flight\_Date\_DIM

as select \* from Flight\_Date\_DIM;

-----Create L0\_Temp\_Trans\_fact table

create table L0\_Temp\_Trans\_fact

as select r.sourceairportid,

r.destairportid,

to\_char(fl.flightdate,'DDMMYY') as flightdateid,

tr.totalpaid,

fl.fare,

a1.country as sourceCountry,

ap.country as destCountry,

al.airlineid,

p.passid,

p.age

from routes r, flights fl, transactions tr, airports a1, airports ap, airlines al, passengers p

where r.sourceairportid = a1.airportid

and r.destairportid = ap.airportid

and r.airlineid = al.airlineid

and r.routeid = fl.routeid

and fl.flightid = tr.flightid

and tr.passid = p.passid;

alter table L0\_Temp\_Trans\_fact

add(class\_id varchar2(20));

alter table L0\_Temp\_Trans\_fact

add(flighttype\_id varchar2(20));

-----

update L0\_Temp\_Trans\_fact

set class\_id='C1'

where totalpaid>= 2\*fare;

update L0\_Temp\_Trans\_fact

set class\_id = 'C2'

where totalpaid>= 1.5\*fare and totalpaid<2\*fare;

update L0\_Temp\_Trans\_fact

set class\_id = 'C3'

where totalpaid < 1.5\*fare;

-----

update L0\_Temp\_Trans\_fact

set flighttype\_id = 'F1'

where sourceCountry = destCountry;

update L0\_Temp\_Trans\_fact

set flighttype\_id = 'F2'

where sourceCountry != destCountry;

--Creating Transaction Fact Table

create table L0\_Transaction\_fact

as select

passid,

flightdateid,

class\_id,

flighttype\_id,

sourceairportid,

destairportid,

airlineid,

totalpaid,

age,

(totalpaid - fare) as Agent\_profit

from Temp\_Trans\_fact;

drop table L0\_Transaction\_fact ;

select \* from L0\_Transaction\_fact ;

---Passenger\_DIM

create table L0\_Passenger\_dim

as select \* from Passenger\_dim;

---Membership type DIM

create table L0\_Membership\_type\_DIM

as select \* from Membership\_type\_DIM ;

-- membership join date dim

create table L0\_Membership\_Join\_Date\_DIM

as select \* from Membership\_Join\_Date\_DIM ;

-------memb discount

---Membership Discount DIM

create table L0\_Membership\_Discount\_DIM as

select

m.membershiptypeid,

p.startdate,

p.enddate,

p.discount

from membershipjoinrecords m, promotion p

where m.promotion = p.promotionid;

------------------------------------------------------------

-----create Passenger Dec DIM (for lowering the level of aggregation)

create table Passenger\_Desc\_DIM

as select distinct passid from passengers;

--Createing Memebership Sales TMP fact

create table L0\_MembershipSales\_Fact\_TMP as

select to\_char(m.joindate,'YYYYMM') as joindateid,

p.age,

m.membershipTypeID,

r.promotionid,

t.membershipfee,

m.passid,

r.discount

from membershipjoinrecords m, promotion r, passengers p, membershiptype t

where m.passid = p.passid

and m.promotion = r.promotionid

and m.membershiptypeid = t.membershiptypeid;

alter table L0\_MembershipSales\_Fact\_TMP

add(passenger\_type\_id varchar2(50));

update L0\_MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P1'

where age < 11;

update L0\_MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P2'

where age >= 11 and age <= 17;

update L0\_MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P3'

where age >= 18 and age <= 60;

update L0\_MembershipSales\_Fact\_TMP

set passenger\_type\_id = 'P4'

where age > 60 ;

-----------------

create table L0\_Membership\_Sales\_Fact

as select

passid,

joindateid,

passenger\_type\_id,

membershiptypeid,

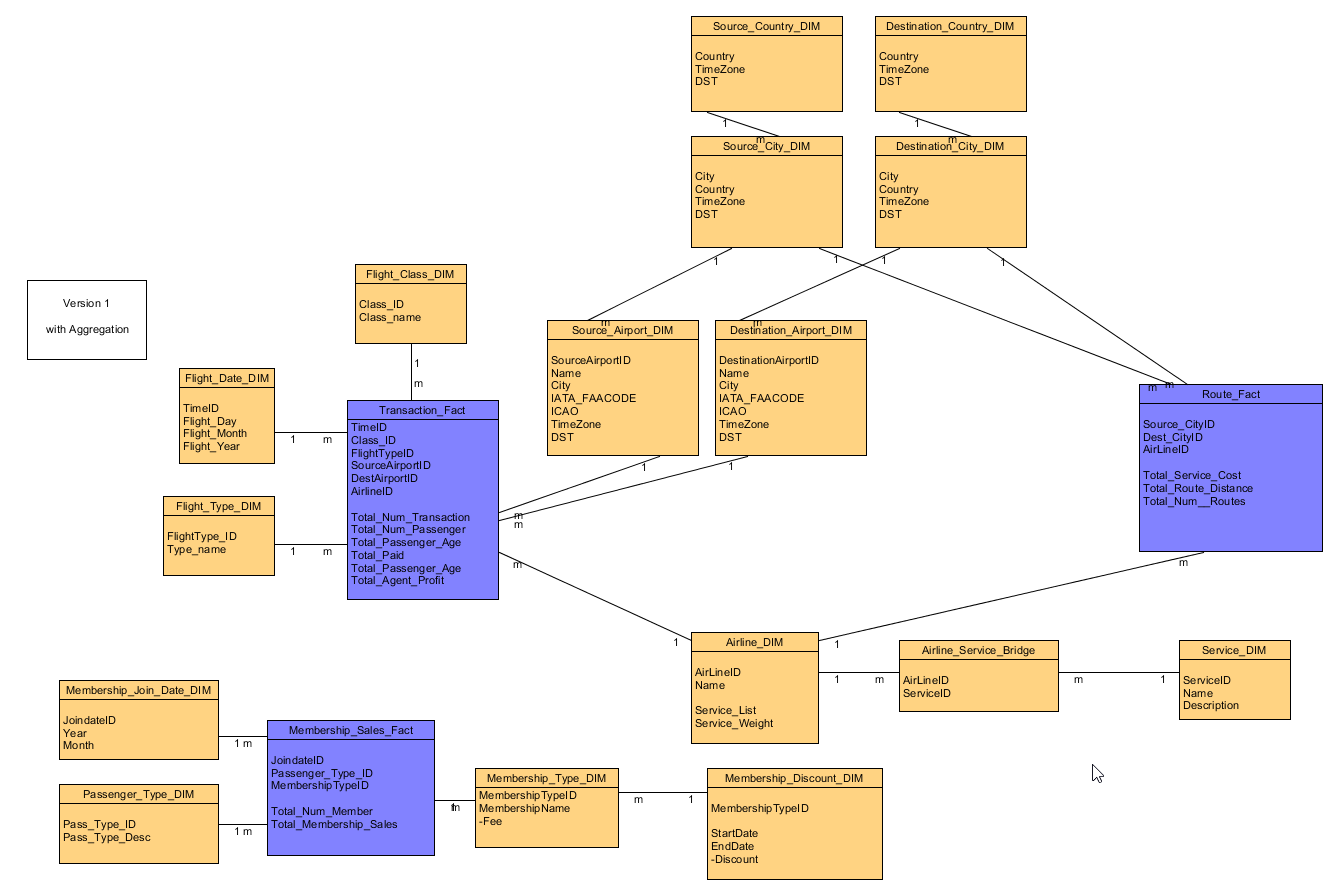
promotionid,

(membershipfee \* (1- discount)) as membership\_sale

from L0\_MembershipSales\_Fact\_TMP;

C. 2

a) Star Schema for version\_1



b) Star Schema for version\_2

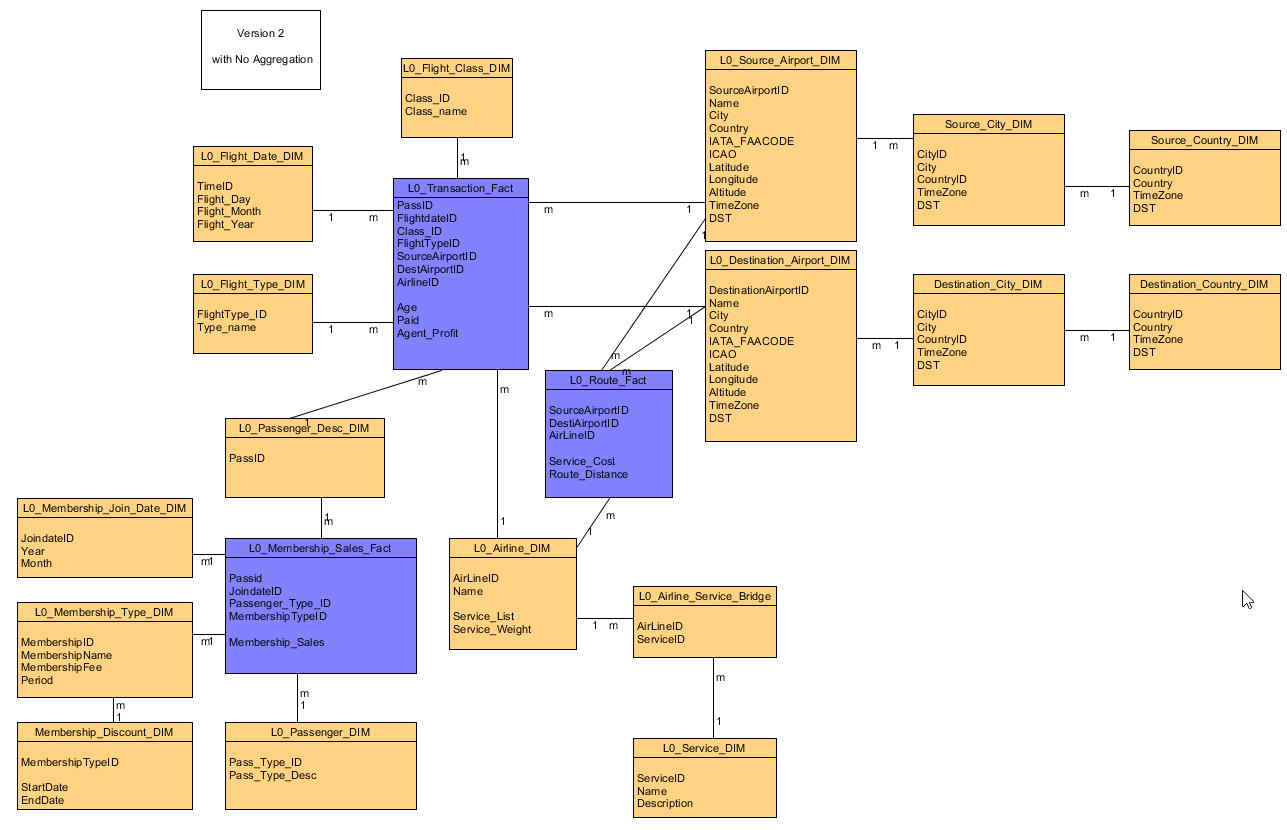
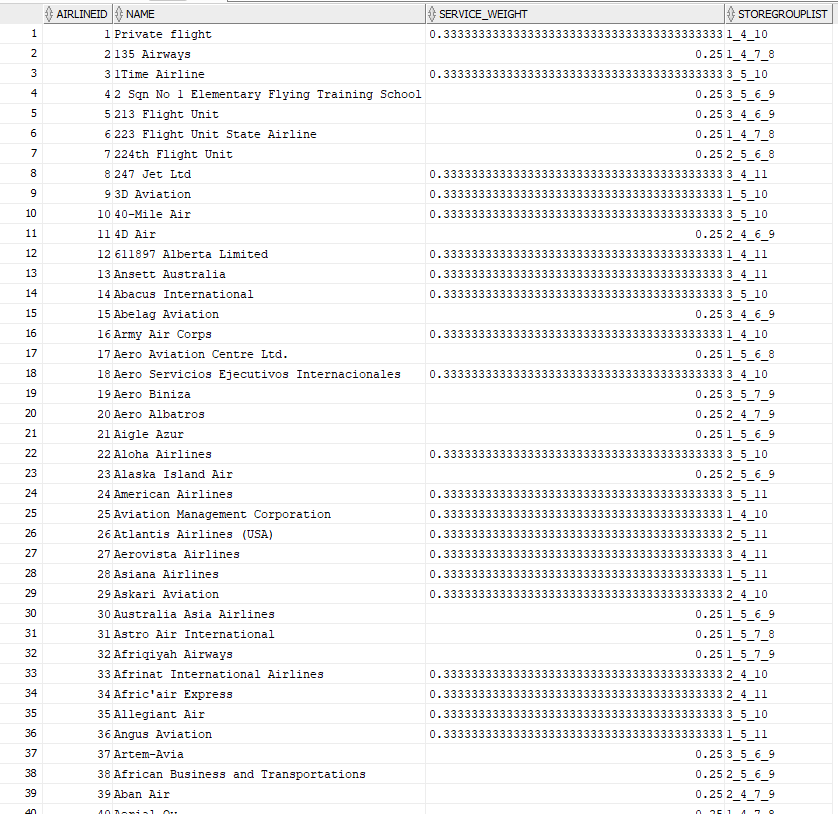
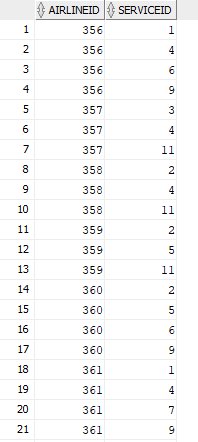


Table Screenshots

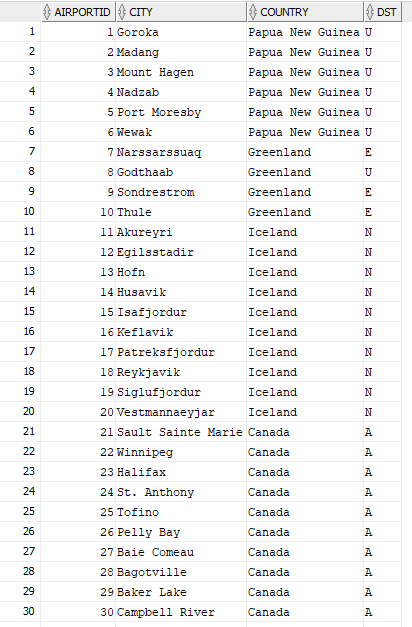
Airline\_DIM



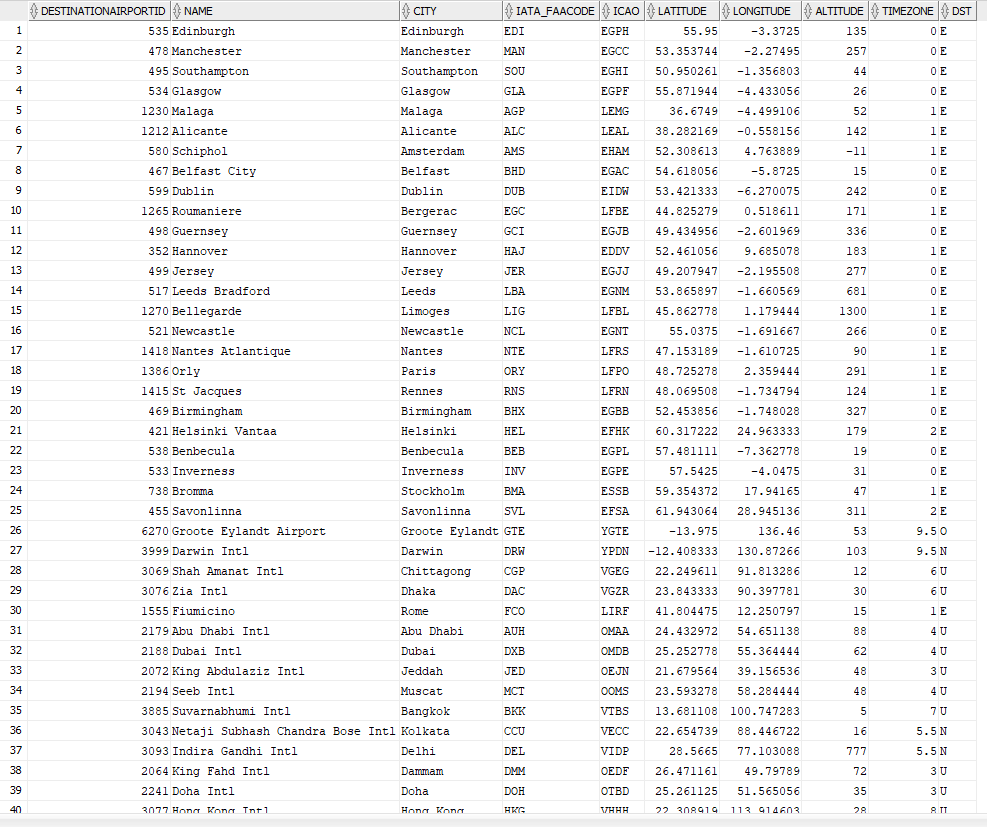
Airline\_Service\_Bridge



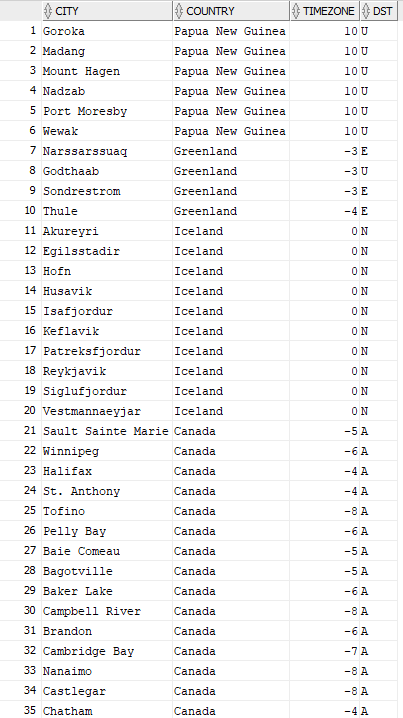
Airport\_DIM



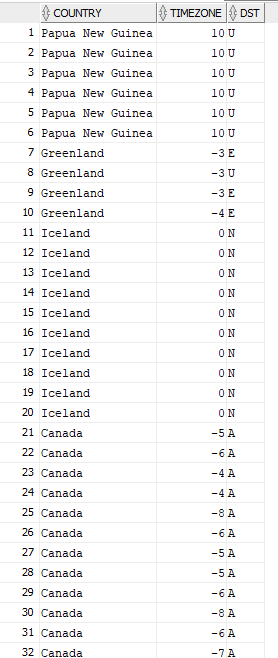
Destination Airport DIM / Source Airport DIM



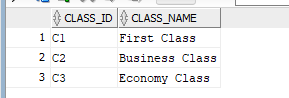
Destination CITY DIM / Source CITY DIM



Destination Country Dim / Source Country DIM



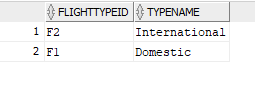
Flight Class DIM



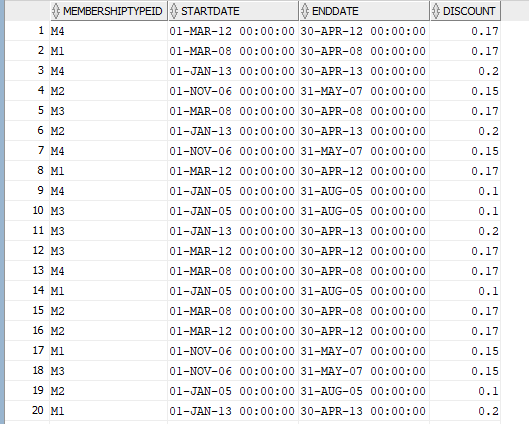
Flight Date DIM



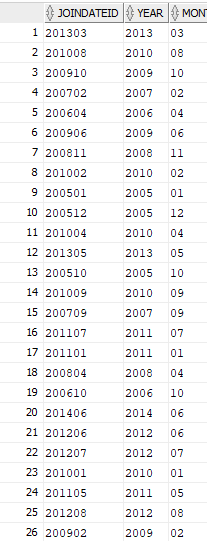
Flight Type DIM



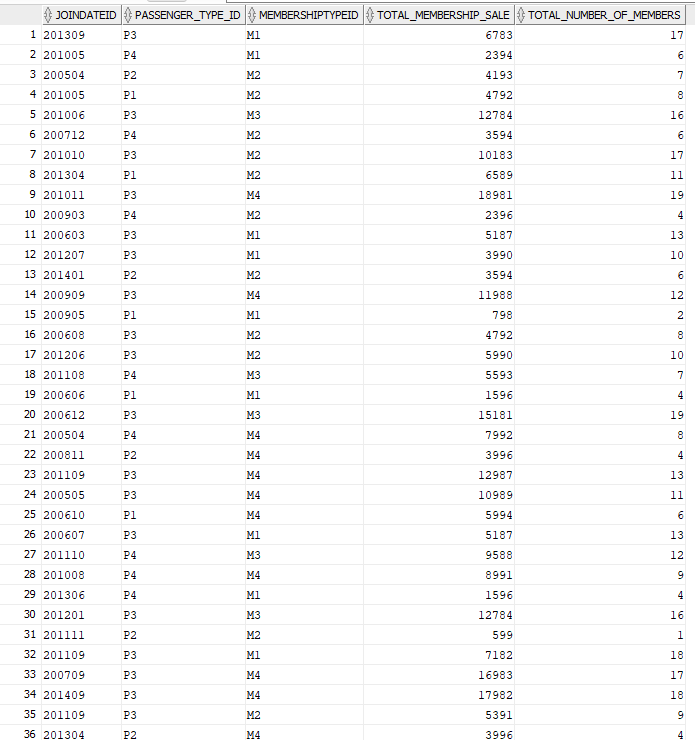
Membership Discount DIM



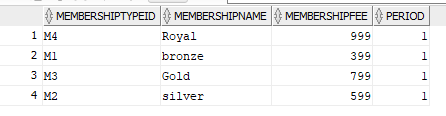
Membership Join Date DIM



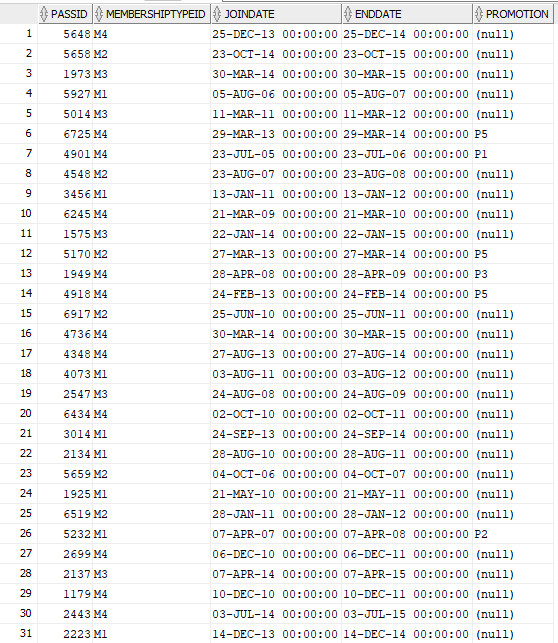
Membership Sales FACT



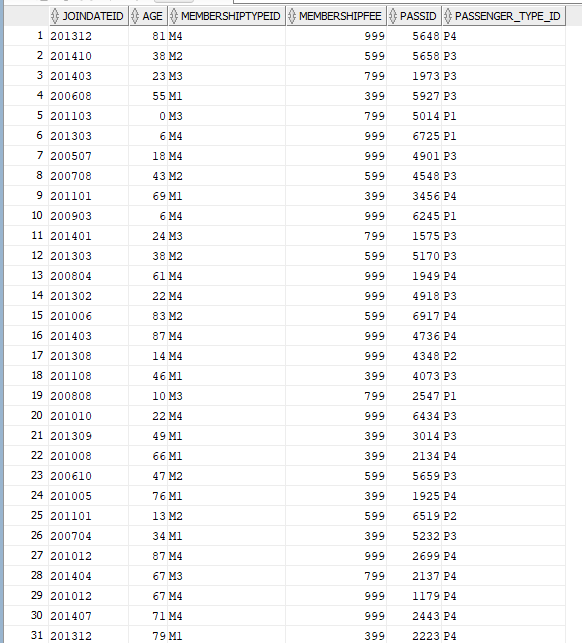
Membership Type DIM



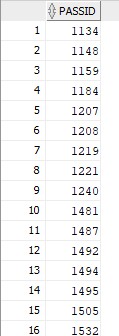
Membership Join Records



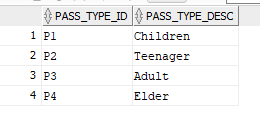
Membership Sales Fact TEMP



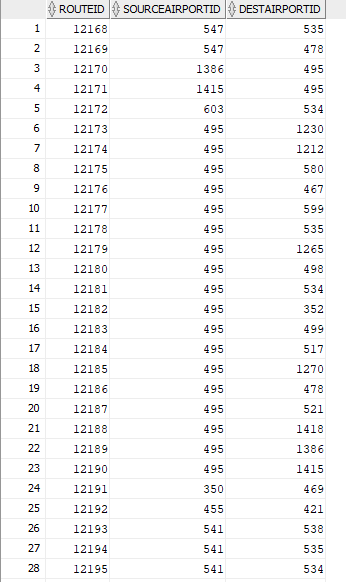
Passenger Desc DIM



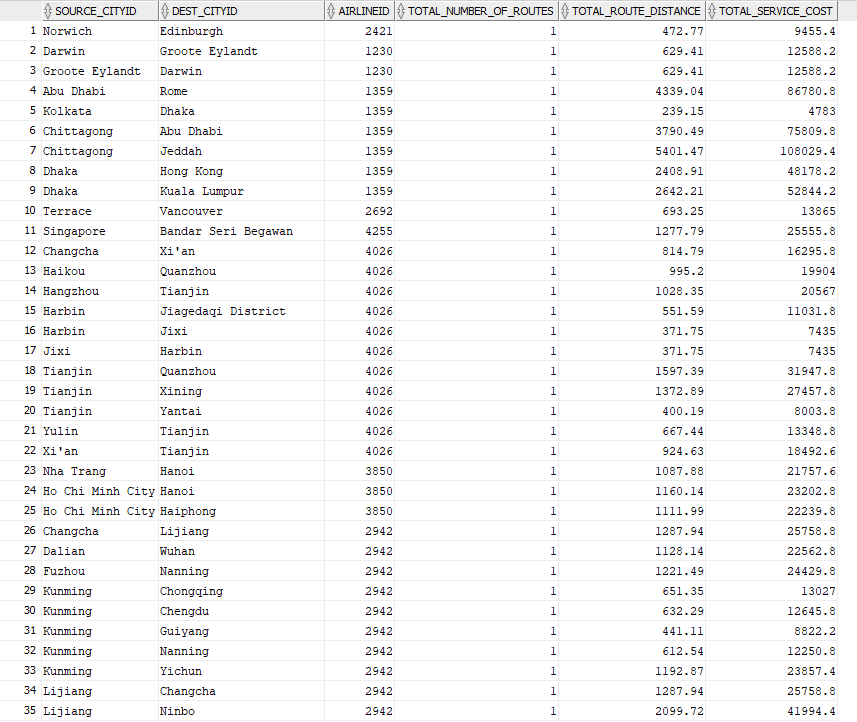
Passenger DIM



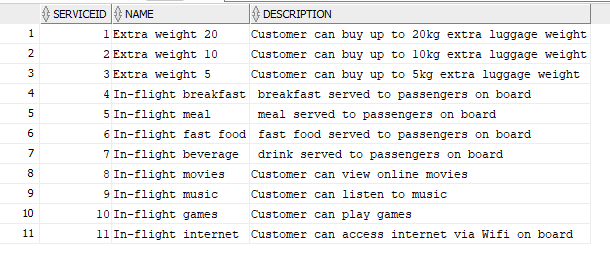
Route DIM



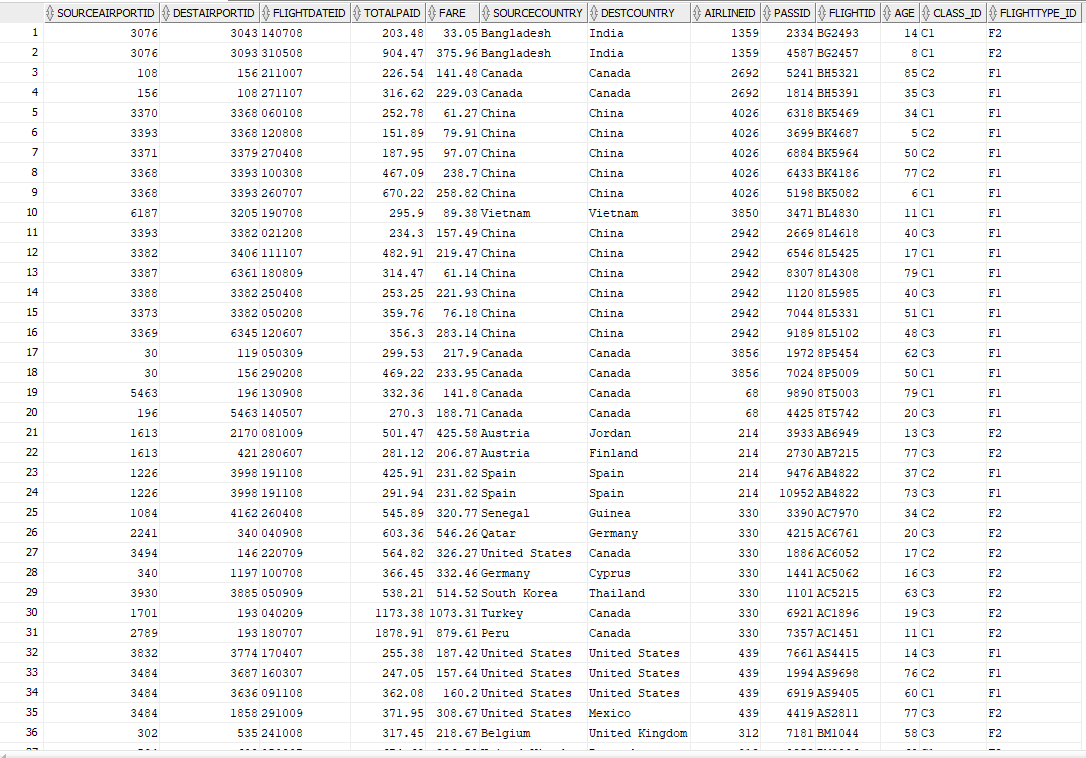
RouteFACT



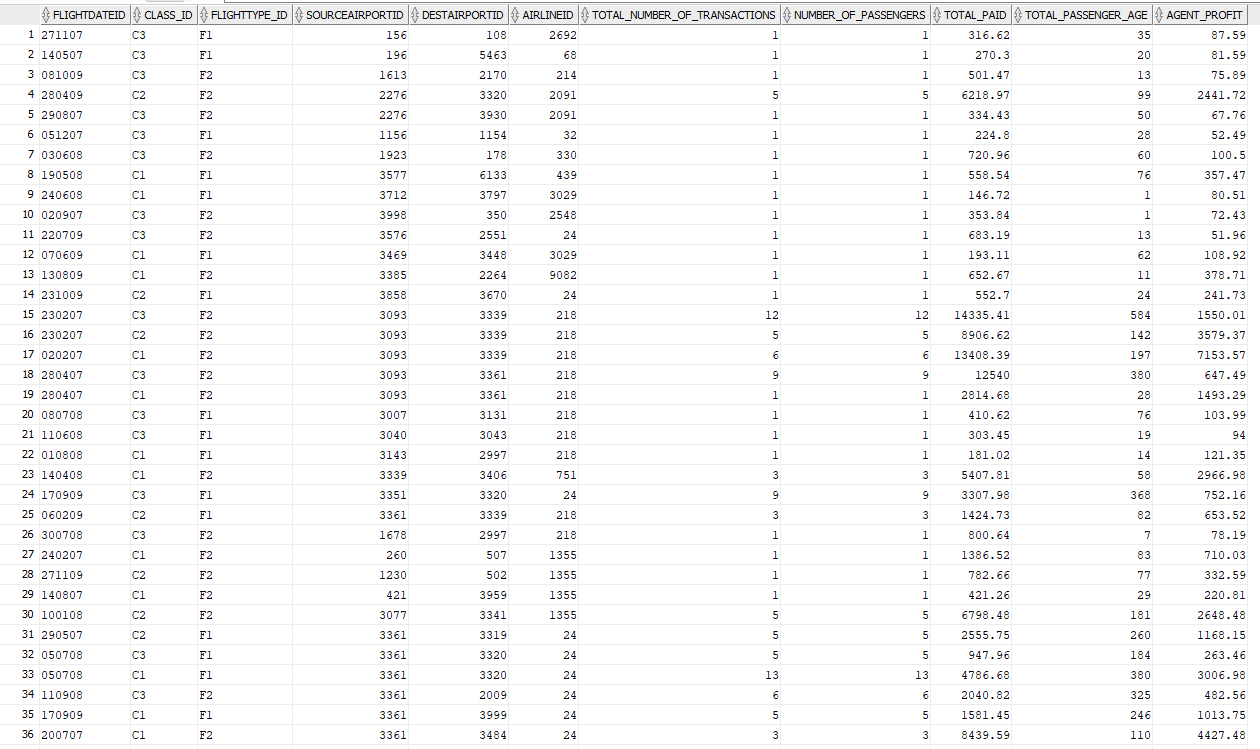
Service DIM



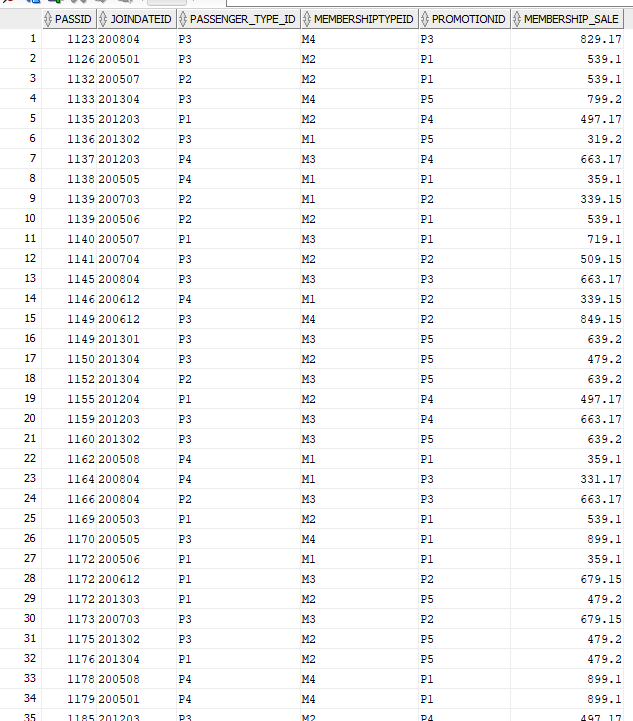
TEMP Trans Fact



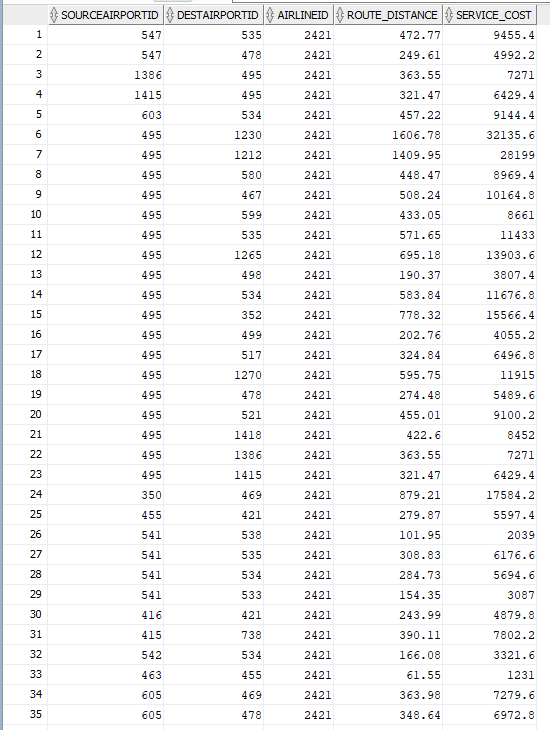
Transaction Fact



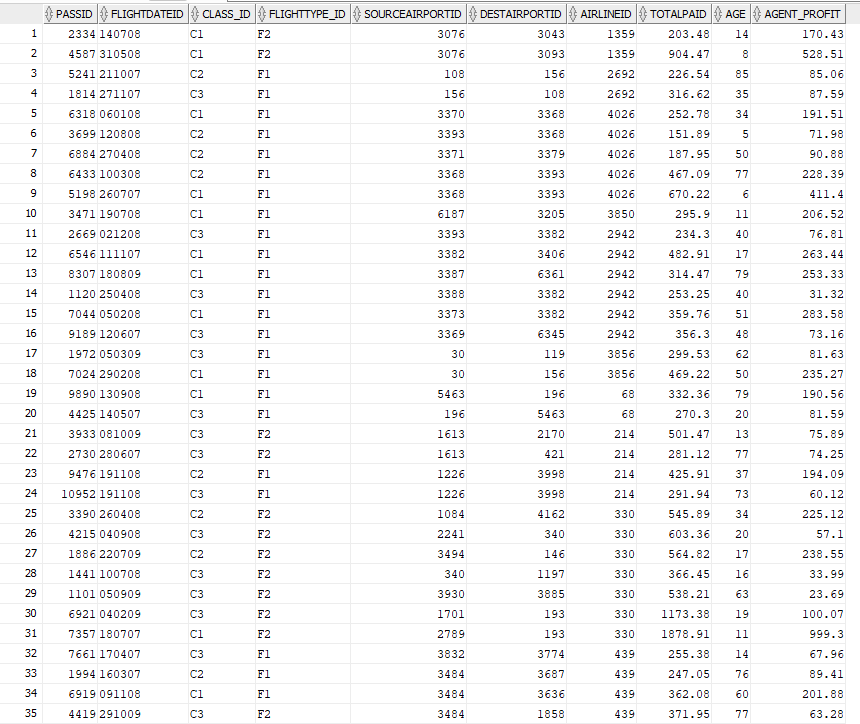
L0\_Membership Sales Fact



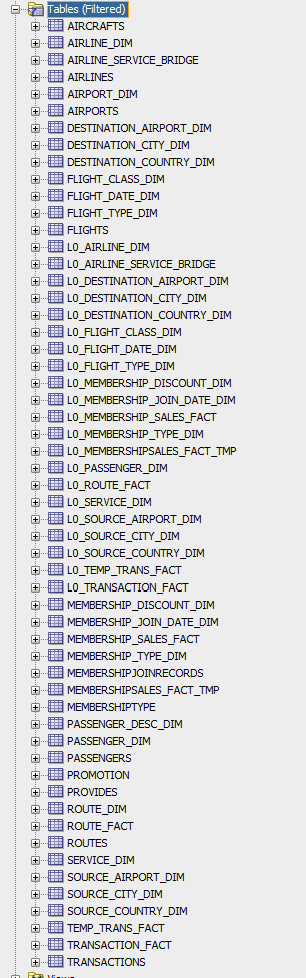
L0\_Route\_Fact



L0\_Transaction Fact

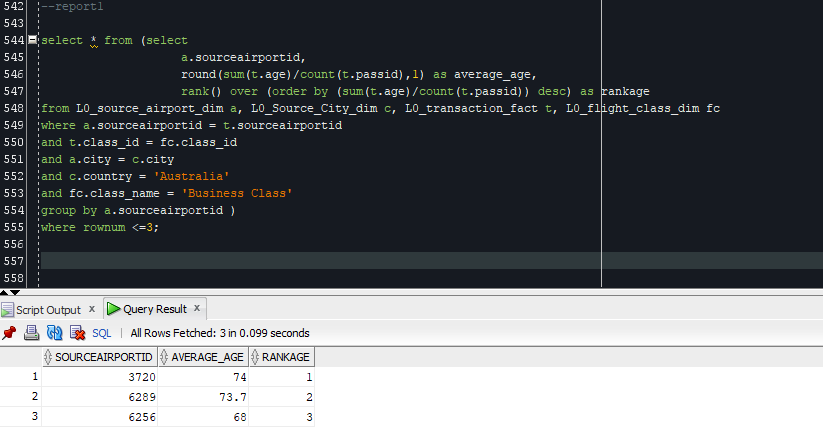


List of All Tables

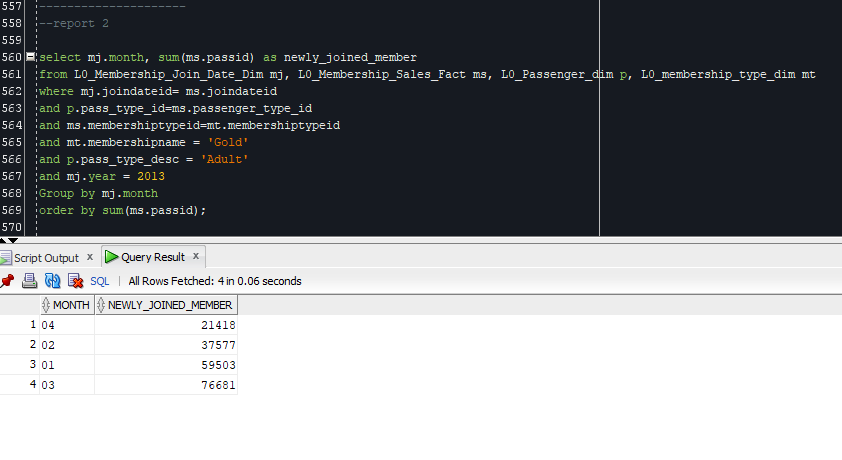


**C.3**

Report 1

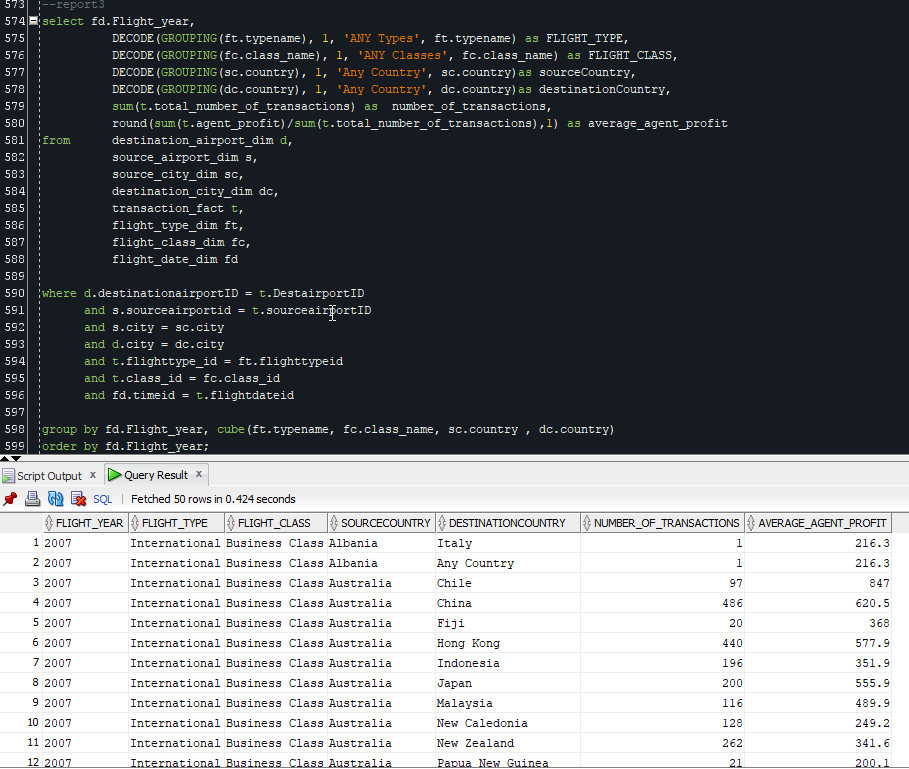


Report 2

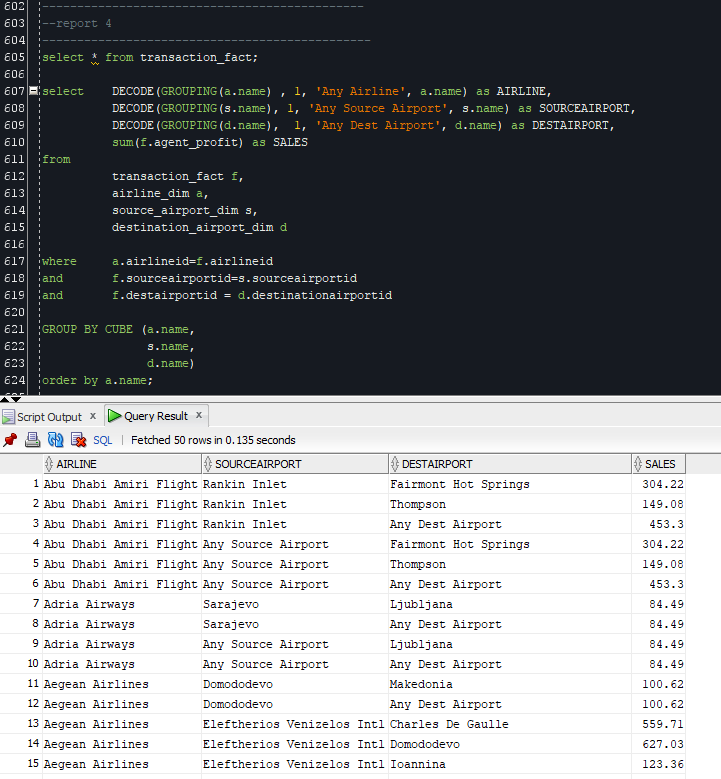


**C.4**

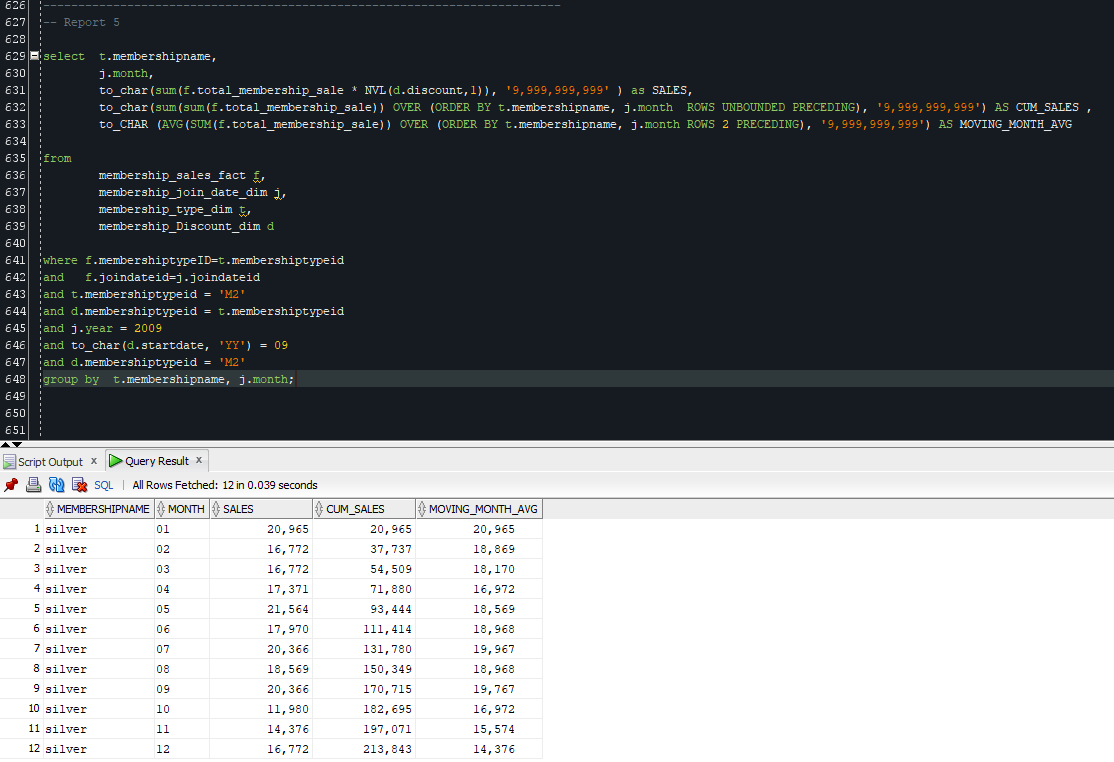
Report 3



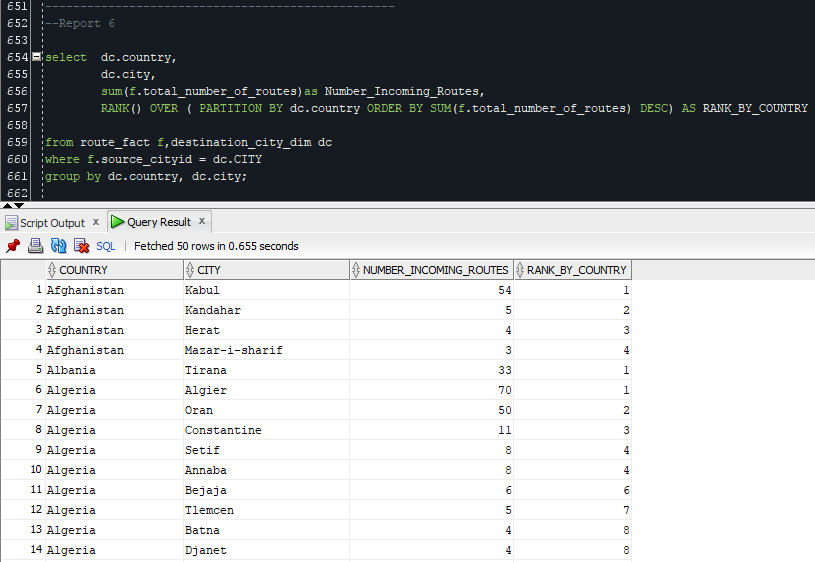
Report 4



Report 5



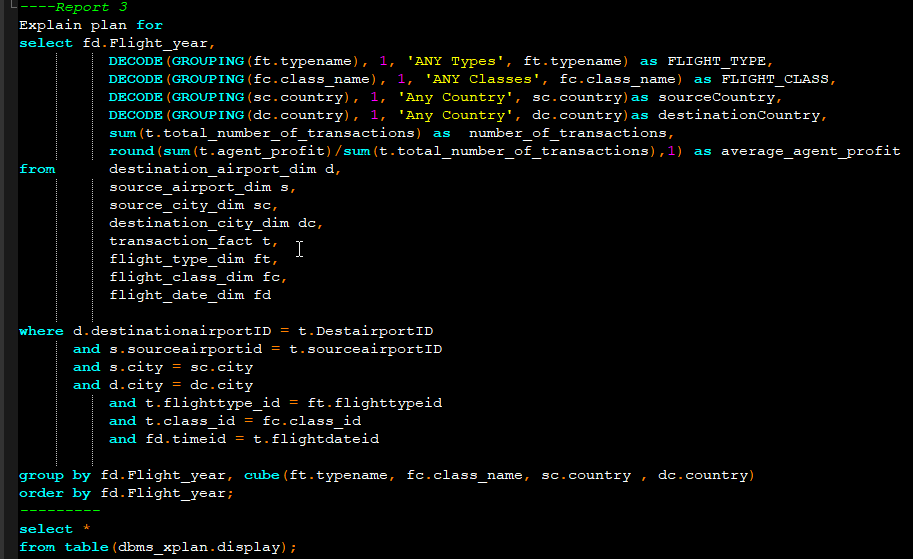
Report 6

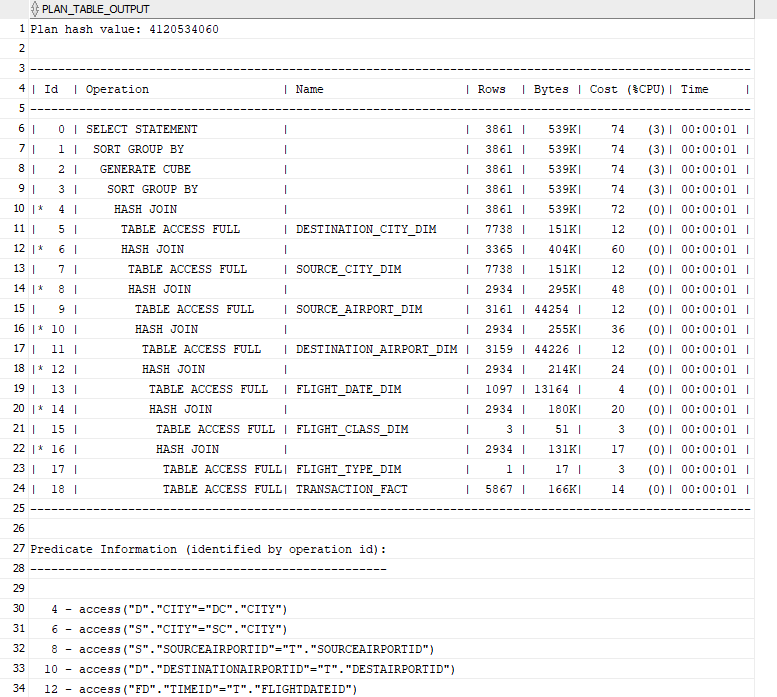


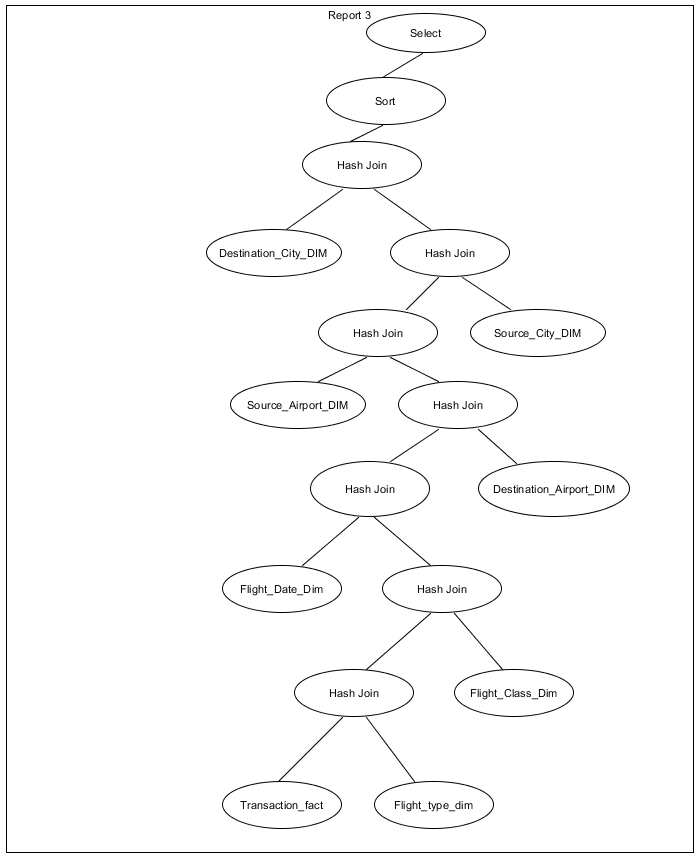
**C.5 Explain plan**

For Original Query:

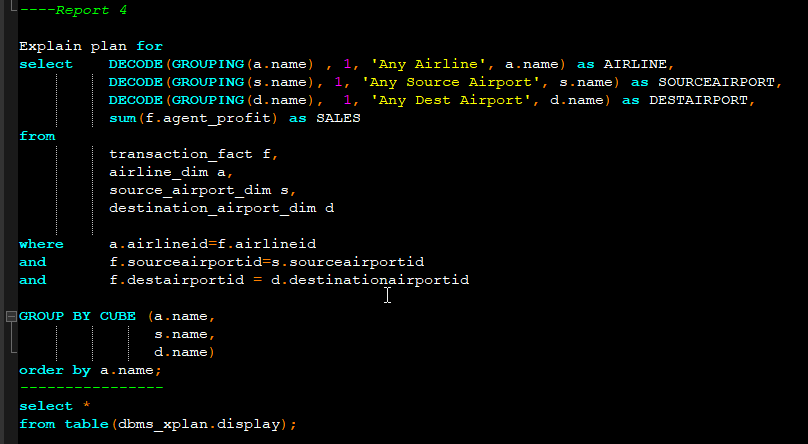
Report 3

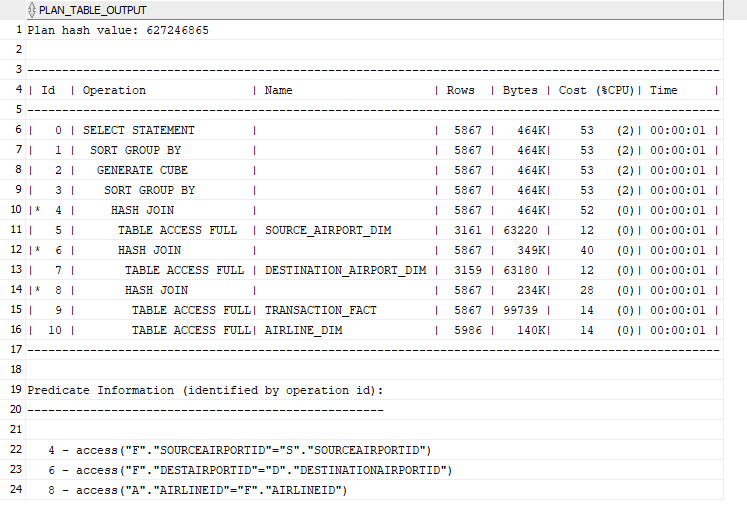


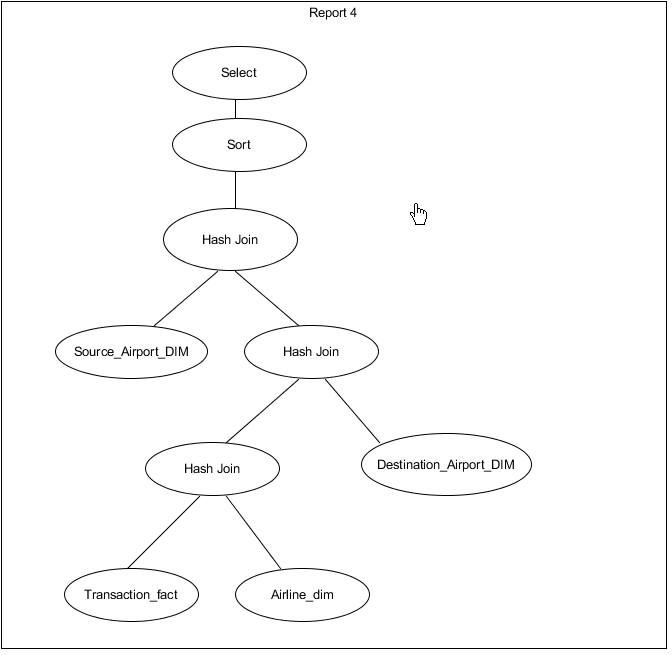




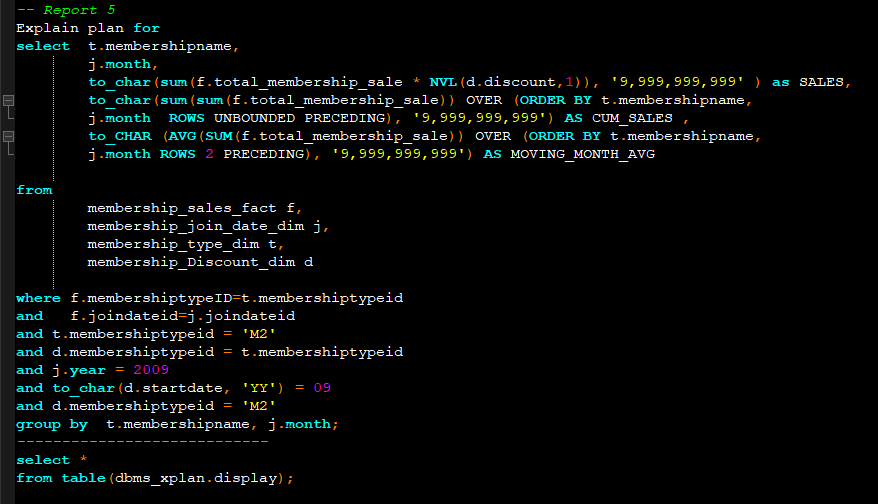
Report 4

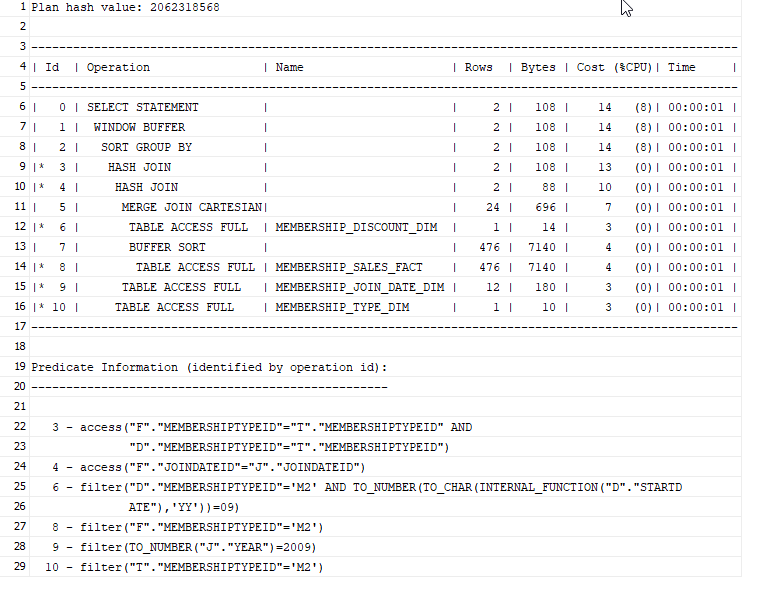


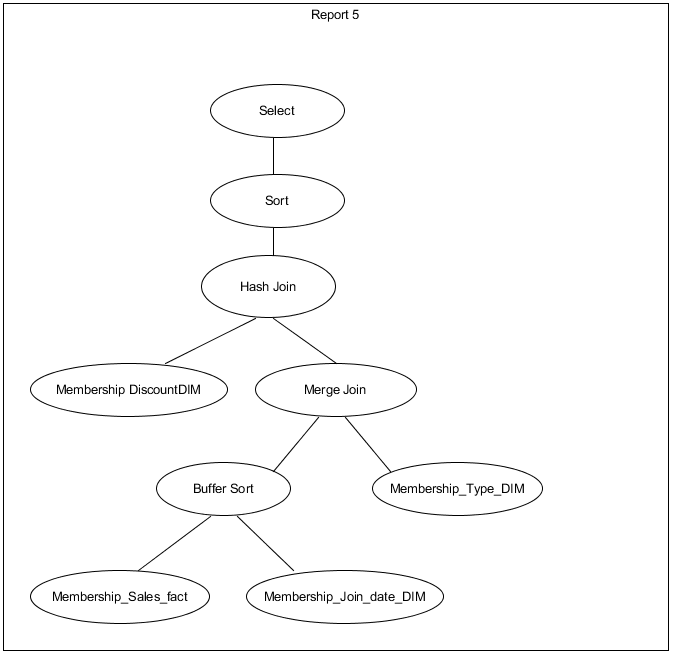




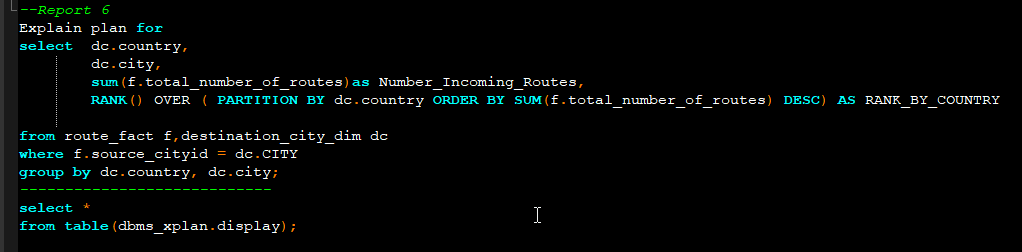
Report 5

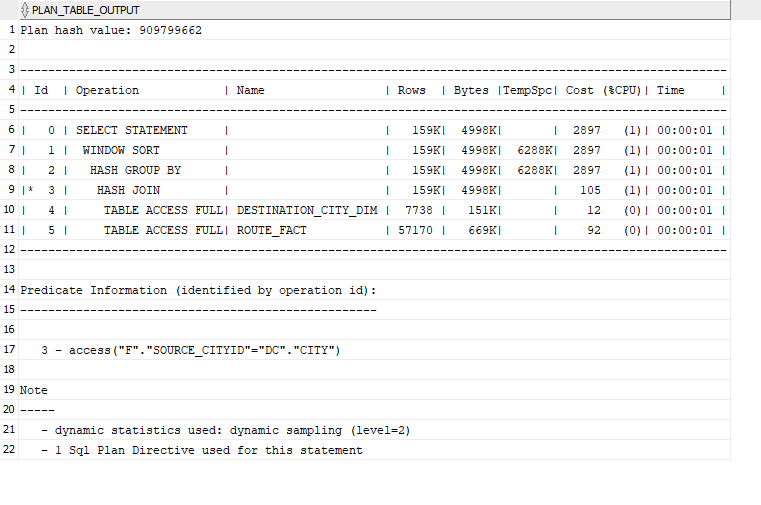


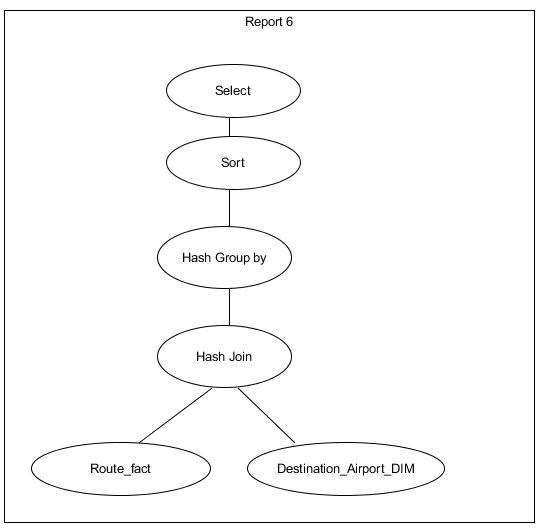




Report 6

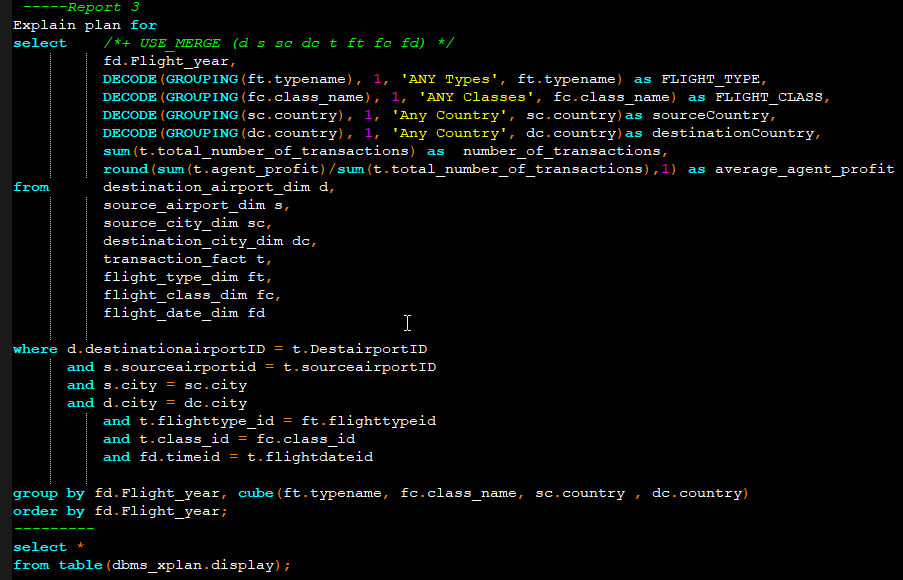




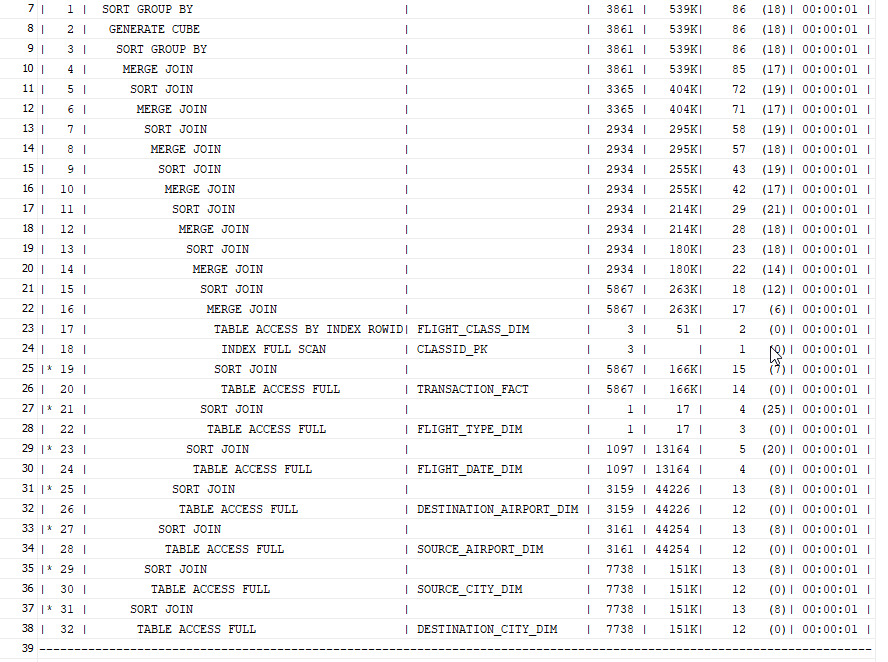


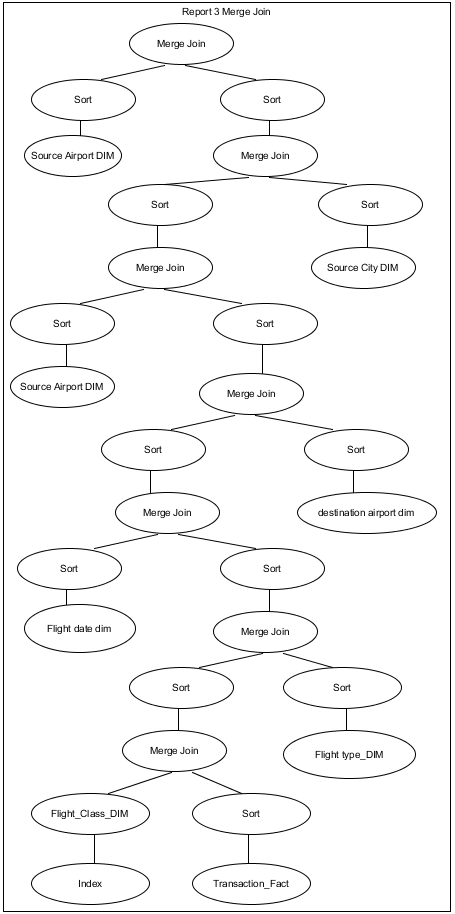
For New Query:

Report 3

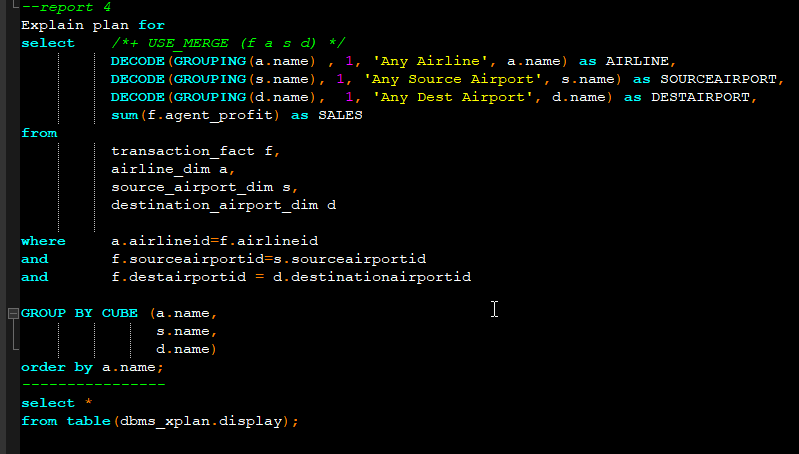


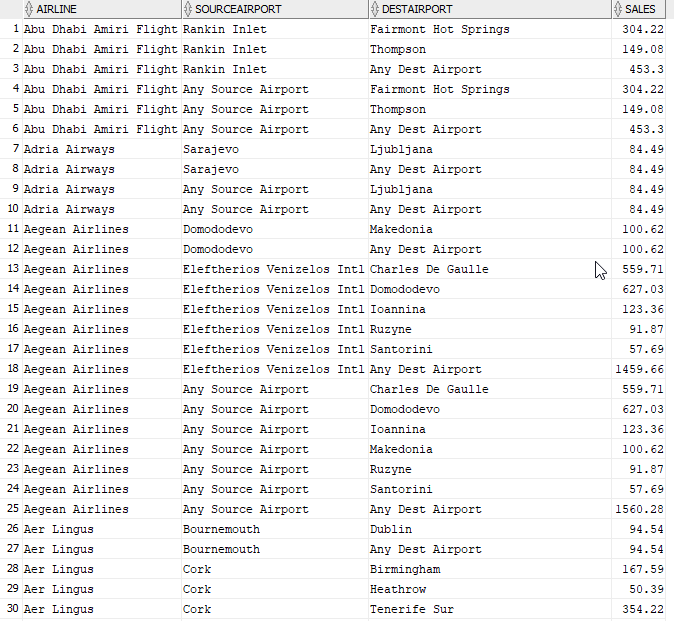


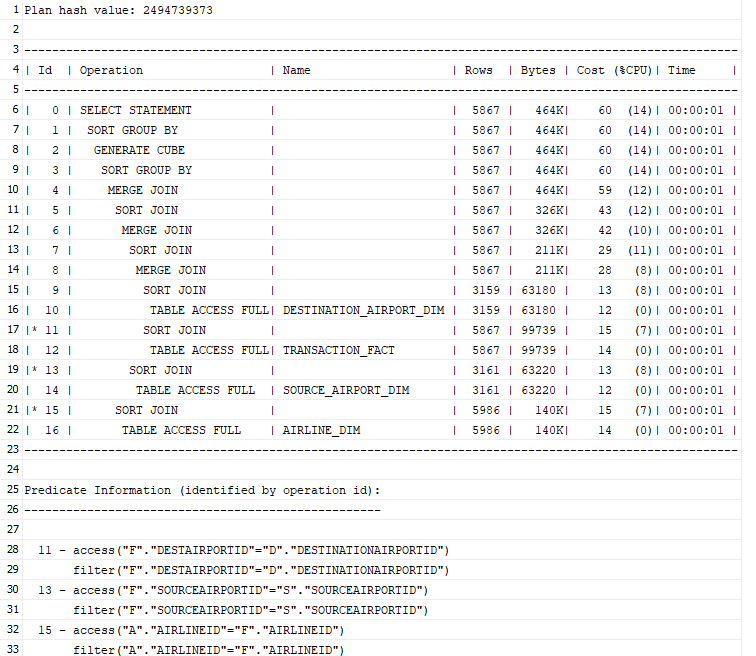


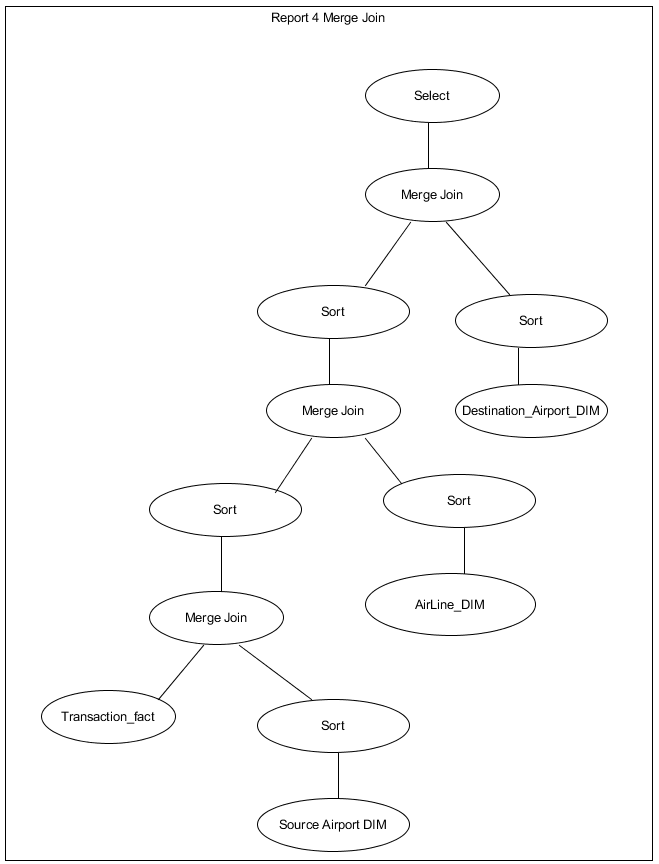


Report 4

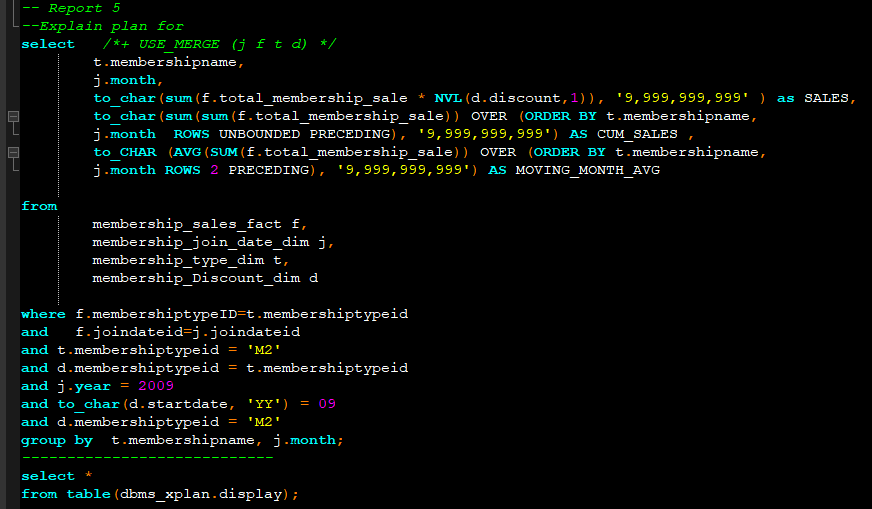


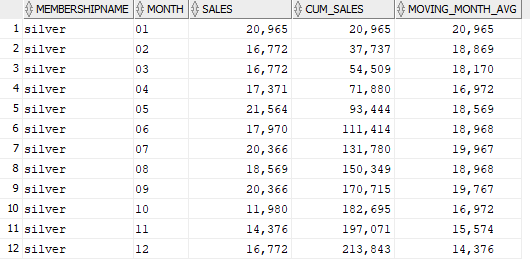


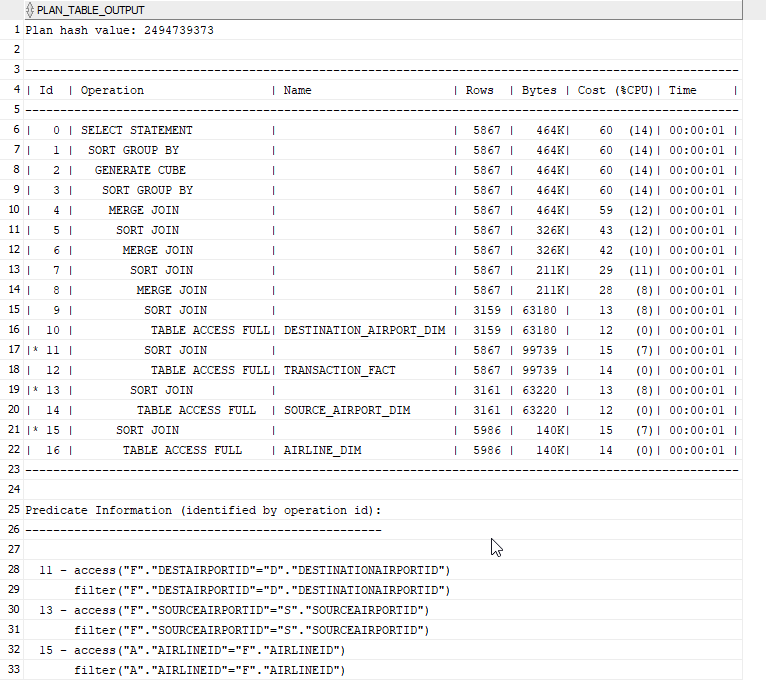


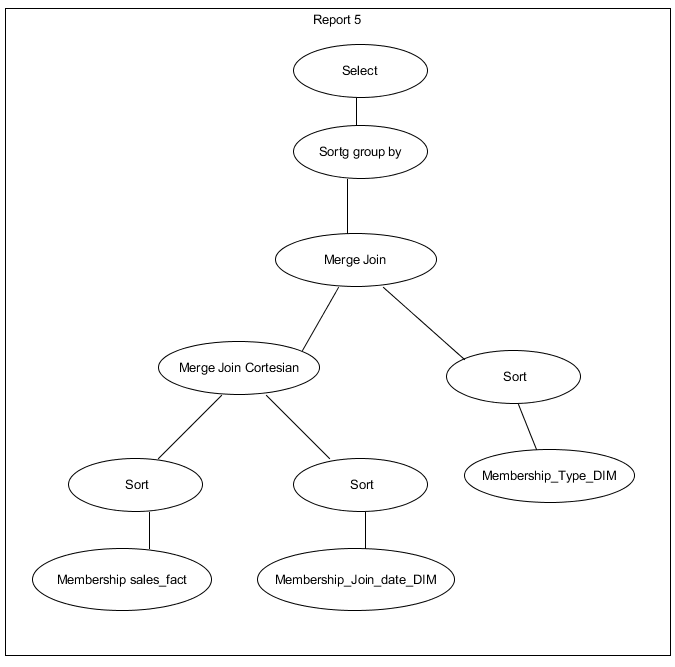


Report 5









Report 6

