Details of Oracle Account used:

Username: S27148572

Password: student

Task C.1.

1. **Copy the operational database from dw\_oft into your account. Create the necessary primary keys and foreign keys for each table and draw E/R diagram using SQL Developer**

create table Airport

as

select \* from dw\_oft.Airports1;

create table Airline

as

select \* from dw\_oft.Airlines1;

create table Route

as

select \* from dw\_oft.Routes1;

create table Aircraft

as

select \* from dw\_oft.Aircrafts1;

create table Provides

as

select \* from dw\_oft.Provides1;

create table Flight

as

select \* from dw\_oft.Flights1;

create table Passenger

as

select \* from dw\_oft.Passengers1;

create table Transaction

as

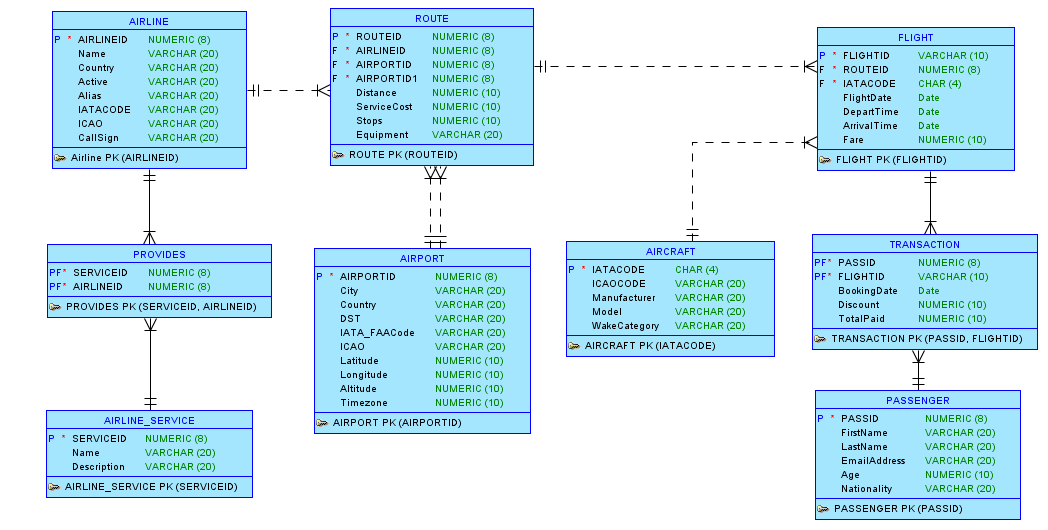
select \* from dw\_oft.Transactions1;

create table Airline\_Service

as

select \* from DW\_OFT.AIRLINE\_SERVICES1;

E/R Diagram with required primary keys and foreign keys



1. If you have done the data cleaning, explain what kind of data cleaning process that you have done (you need to show the SQL to explore the operational database, and SQL of the data cleaning, as well as the screenshot of data before and after data cleaning)

**Doing data cleaning for Route table**

--Total Records

Select count(\*) from ROUTE;

--58445

-- Checking whether source airport id is null

Select COUNT(\*) from ROUTE where SOURCEAIRPORTID is null;

--0

--Checking total number of source airport

Select COUNT(\*) from ROUTE where SOURCEAIRPORTID is not null;

--58445

-- Checking whether destination airport id is null

Select COUNT(\*) from ROUTE where DESTAIRPORTID is null;

--0

--Checking total number of destination airport

Select COUNT(\*) from ROUTE where DESTAIRPORTID is not null;

--58445

--Checking whether source and destination airport id are same

Select COUNT(\*) from ROUTE where DESTAIRPORTID = SOURCEAIRPORTID;

--0

--Conclusion

--source airport id and destination airport id is not null in routes

--source airport id is not same as destination airport id in any row

--Checking for 0 distance travelled

Select COUNT(\*) from ROUTE where distance = 0;

--0

--Checking for 0 service cost

Select COUNT(\*) from ROUTE where SERVICECOST = 0;

--0

--Conclusion no distance and service cost is zero so value is present

-- Checking negative values for distance

select \* from ROUTE where DISTANCE <= 0;

-- Found one value with negative distance. We are deleting the record having routeid = -1

-- Checking negative values for servicecost

select \* from ROUTE where SERVICECOST <= 0;

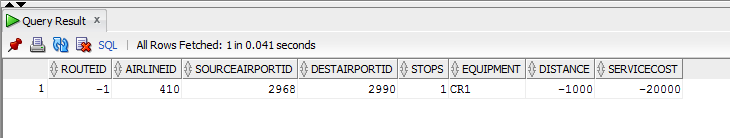
-- Found one value with negative servicecost. We are deleting the record having routeid = -1

-- Checking negative values for routeid

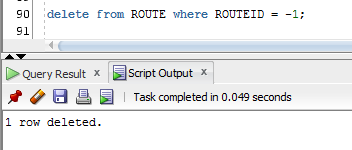
select \* from ROUTE where ROUTEID <= 0;

-- Found one value with negative routeID. We are deleting the record having routeid = -1

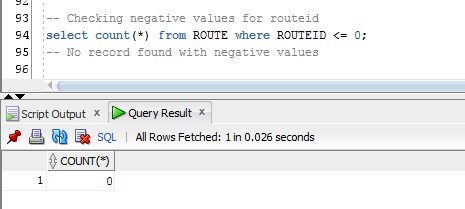
Screenshot showing data before data cleaning



Deleting the record with routeid = -1 as distance, route ID, and service cost cannot be negative using the query shown below:



Screenshot after data cleaning to show that no record with negative value exists



-- Checking whether any source airport id does not match airport id

Select count(\*) from ROUTE where SOURCEAIRPORTID not IN (Select a.AIRPORTID from airport a);

--0

-- Checking whether any destination airport id does not match airport id

Select count(\*) from ROUTE where DESTAIRPORTID not IN (Select a.AIRPORTID from airport a);

--0

--Checking whether airline id in route is matching airline id in airline table

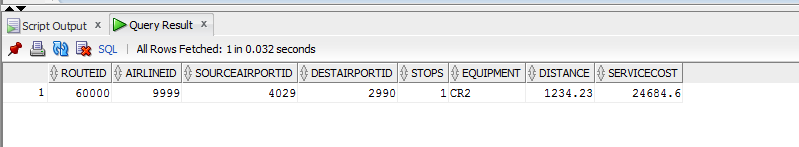
Select count(\*) from ROUTE where AIRLINEID not IN (Select a.AIRLINEID from AIRLINE a);

--1

-- fetching that 1 record

Select \* from ROUTE where AIRLINEID not IN (Select a.AIRLINEID from AIRLINE a);

Screenshot showing that record with invalid airlineid



--Validating the record with airlineid = 9999

Select count(\*) from AIRLINE where AIRLINEID = 9999;

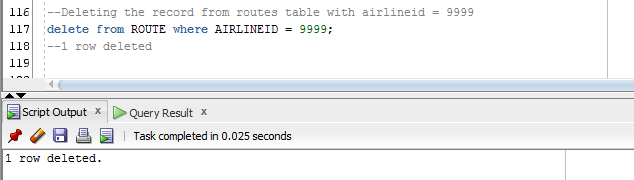
--0

--The row with airline id = 9999 does not exist so we delete this record as it is invalid

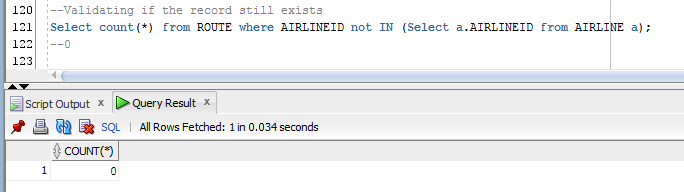
--Deleting the record from routes table with airlineid = 9999

delete from ROUTE where AIRLINEID = 9999;

--1 row deleted



Screenshot showing that the invalid record does not exist now



--checking route id is unique

Select distinct count(\*) from ROUTE;

--58443

Select count(\*) from ROUTE;

--58443

--Checking whether there is any route is without airline

Select COUNT(\*) from ROUTE where AIRLINEID is not null;

--58443

**Doing data cleaning for airport table**

Select COUNT(\*) from AIRPORT where AIRPORTID is null;

--0

Select COUNT(\*) from AIRPORT where AIRPORTID is not null;

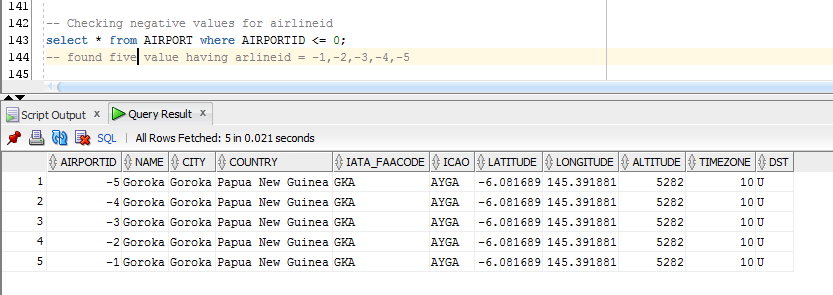
--7738

-- Checking negative values for airlineid

select \* from AIRPORT where AIRPORTID <= 0;

-- found five value having arlineid = -1,-2,-3,-4,-5

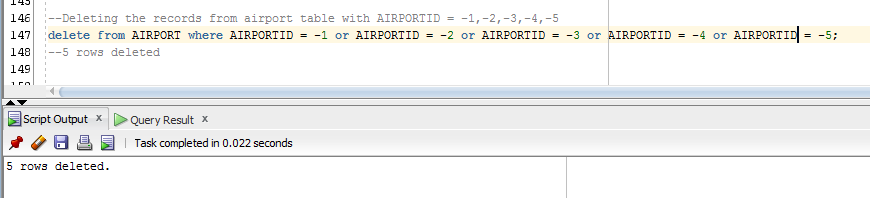
Screenshot showing that records with invalid AirportId:



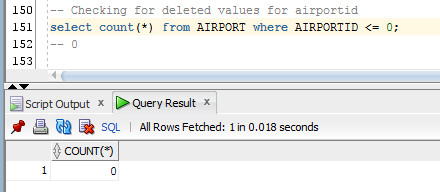
--Deleting the records from airport table with AIRPORTID = -1,-2,-3,-4,-5

delete from AIRPORT where AIRPORTID = -1 or AIRPORTID = -2 or AIRPORTID = -3 or AIRPORTID = -4 or AIRPORTID = -5;

--5 rows deleted



Screenshot showing that the invalid record does not exist now



**Doing data cleaning for Airline service table**

--Checking airline\_service

select \* from AIRLINE\_SERVICE;

-- 11

-- checking whether there is any null values

Select COUNT(\*) from AIRLINE\_SERVICE where SERVICEID is null;

-- 0

-- checking for any Repetitions is any name values

select distinct count(name) from AIRLINE\_SERVICE;

-- 11

-- Checking negative values for serviceid

select count(\*) from AIRLINE\_SERVICE where SERVICEID <= 0;

-- 0

**Doing data cleaning for airline table**

-- Checking Airline

-- Checking for number of records in airline

Select COUNT(\*) from AIRLINE;

--5987 records

-- Checking for null values in AirlineID

Select COUNT(\*) from AIRLINE where AIRLINEID is null;

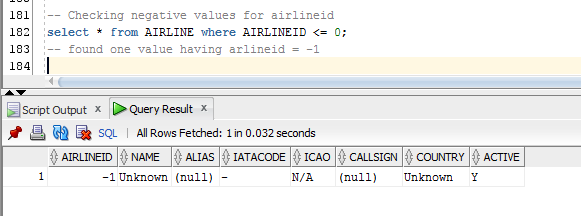
-- 0

-- Checking negative values for airlineid

select \* from AIRLINE where AIRLINEID <= 0;

-- found one value having arlineid = -1

Screenshot showing record with airlineID = -1

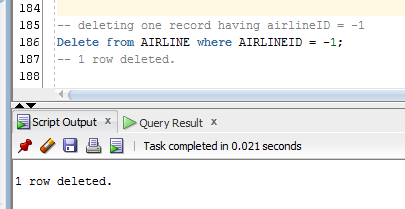


-- deleting one record having airlineID = -1

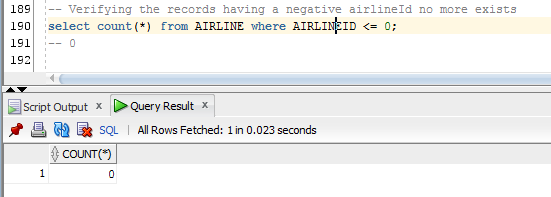
Delete from AIRLINE where AIRLINEID = -1;

-- 1 row deleted.

Screenshot showing the deletion of invalid record



Screenshot showing that the invalid records don’t exist anymore.



**Doing data cleaning for provides table**

--checking provides

-- Checking for the number of records in Provides

select count(\*) from PROVIDES;

-- 20981

--Checking for any unmatched airline and service IDs from airline and airline\_service

Select count(\*) from PROVIDES where AIRLINEID not IN (Select a.AIRLINEID

from AIRLINE a);

-- 0

-- Checking for any serviceID that does not exist in Airline\_Service table

Select count(\*) from PROVIDES where SERVICEID not IN (Select s.SERVICEID

from AIRLINE\_SERVICE s);

-- 0

-- Checking negative values for serviceid

select count(\*) from PROVIDES where SERVICEID <= 0;

-- found 0 value having ServiceID

-- Checking negative values for airlineID

select count(\*) from PROVIDES where AIRLINEID <= 0;

-- found 0 value having AirlineID

**Doing data cleaning for flights table**

-- Checking Flights

select count(\*) from FLIGHT;

--50002

--checking for null values

select count(\*) from FLIGHT where ROUTEID is null;

--0

select count(\*) from FLIGHT where AIRCRAFTID is null;

--0

select count(\*) from FLIGHT where FLIGHTID is null;

--0

-- Checking for unmatched values for routeid

Select count(\*) from FLIGHT where ROUTEID not IN (Select r.ROUTEID from ROUTE r);

--0

-- Checking for unmatched values for aircraftID

Select count(\*) from FLIGHT where AIRCRAFTID not IN (Select a.IATACODE from AIRCRAFT a);

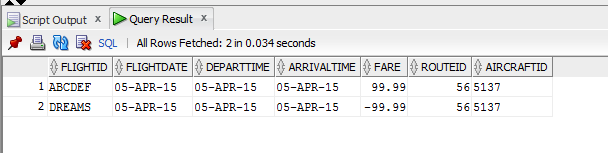
--2

--fetching 2 invalid rows

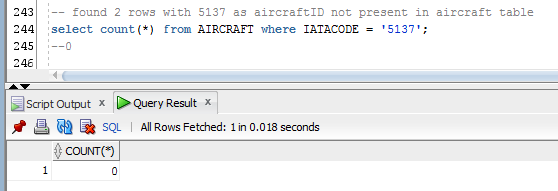
Select \* from FLIGHT where AIRCRAFTID not IN (Select a.IATACODE from AIRCRAFT a);

--We are deleting these two rows as they have invalid aircraftIDs

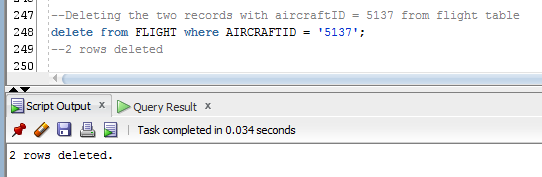
Screenshot showing records with invalid aircraftID values



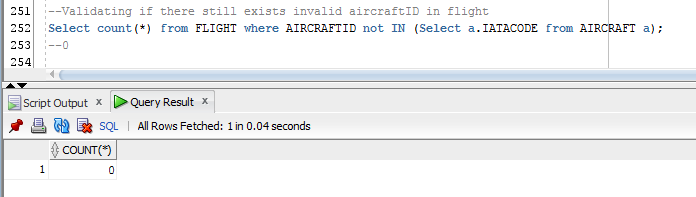
Validating if the above records with aircraftid = 5137 exists in aircraft table:



Deleting these invalid records from flight table



Screenshot showing that there are no more invalid records in the flight table



--checking for negative fares

Select count(\*) from FLIGHT where fare <= 0;

--0

-- Checking invalid values for FlightID

Select count(\*) from FLIGHT where FLIGHTID LIKE '-%';

-- 0

--Checking for negative values for routeid

Select count(\*) from FLIGHT where ROUTEID <= 0;

--0

--Checking for invalid values for aircraftid

Select count(\*) from FLIGHT where AIRCRAFTID LIKE '-%';

-- 0

**Doing data cleaning for aircraft table**

--Checking aircraft

select count(\*) from AIRCRAFT;

-- 357

--checking for negative values

select count(\*) from AIRCRAFT where IATACODE like '-%';

-- 0

--checking for invalid wakecategory

select \* from AIRCRAFT where WAKECATEGORY not in ('M','H','L');

-- 4 rows

-- Confused if we should delete these

-- Some rows are with model as null.. will it matter?

select \* from AIRCRAFT where MODEL LIKE '-%';

--We found 5 rows with - in front.. should we remove the rows or update them by removing –

select count(\*) from AIRCRAFT where MANUFACTURER LIKE '-%';

--found zero rows

**Doing data cleaning for passenger table**

--Checking passenger

select count(\*) from PASSENGER;

-- 10004 records found but assignment specification tells that we have records of 10000 passengers

-- Checking for negative values of PASSID

select \* from PASSENGER where PASSID <= 0;

-- found 0 rows

-- Checking for negative values in Age

select count(\*) from PASSENGER where AGE < 0;

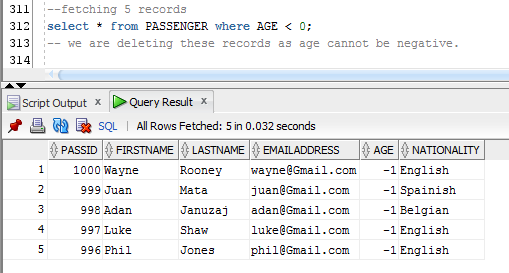
-- 5 rows

--fetching 5 records

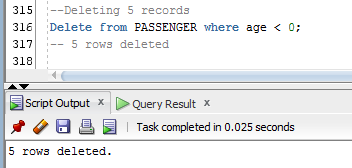
select \* from PASSENGER where AGE < 0;

-- we are deleting these records as age cannot be negative.

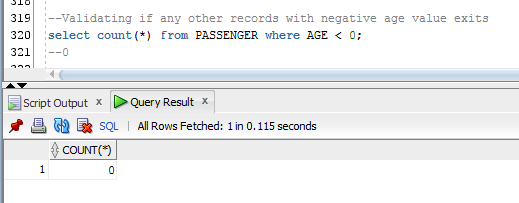
Screenshot showing records with negative age values



Deleting these records as age cannot be negative



Screenshot after the records with negative age value are deleted



-- Checking for 0 values in Age

select count(\*) from PASSENGER where AGE = 0;

-- found 149 rows but we are keeping these values as infant can also travel

**Doing data cleaning for transaction table**

-- Checking for values

select count(\*) from TRANSACTION;

-- 25005 records found.

-- Checking for unmatched PassId values

Select count(\*) from TRANSACTION where PASSID not IN (Select p.PASSID

from PASSENGER p);

-- found 0 rows.

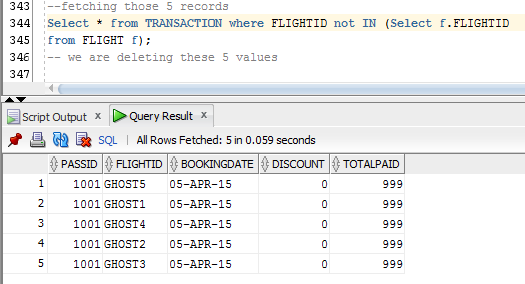
-- Checking for unmatched FlightID values

Select count(\*) from TRANSACTION where FLIGHTID not IN (Select f.FLIGHTID

from FLIGHT f);

-- 5 values found

Screenshot showing invalid records in transaction table

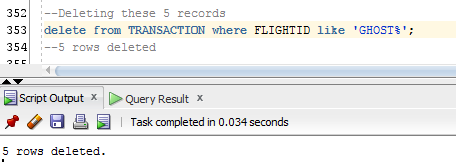


--Checking for these values in flight table

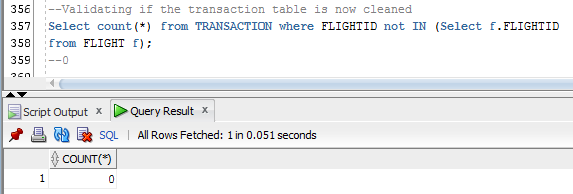
select \* from FLIGHT where FLIGHTID like 'GHOST%';

--0

Screenshot showing the deletion of the invalid records



Screenshot showing that the transaction table does not have any records with invalid flightID values



-- Checking for negative values in Totalpaid

select count(\*) from TRANSACTION where TOTALPAID <= 0;

-- found 0 rows

-- Checking for negative values in Discount ????

select count(\*) from TRANSACTION where DISCOUNT < 0;

-- found 0 rows

-- Checking for booking date between 2006-09

select count(\*) from TRANSACTION where to\_char(BOOKINGDATE,'YYYY') < '2006' OR to\_char(BOOKINGDATE,'YYYY') > '2009';

-- 0 rows

Queries to select the data from the tables:

select \* from AIRLINE;

Select \* from AIRCRAFT;

Select \* from ROUTE;

Select \* from Airport;

select \* from PASSENGER;

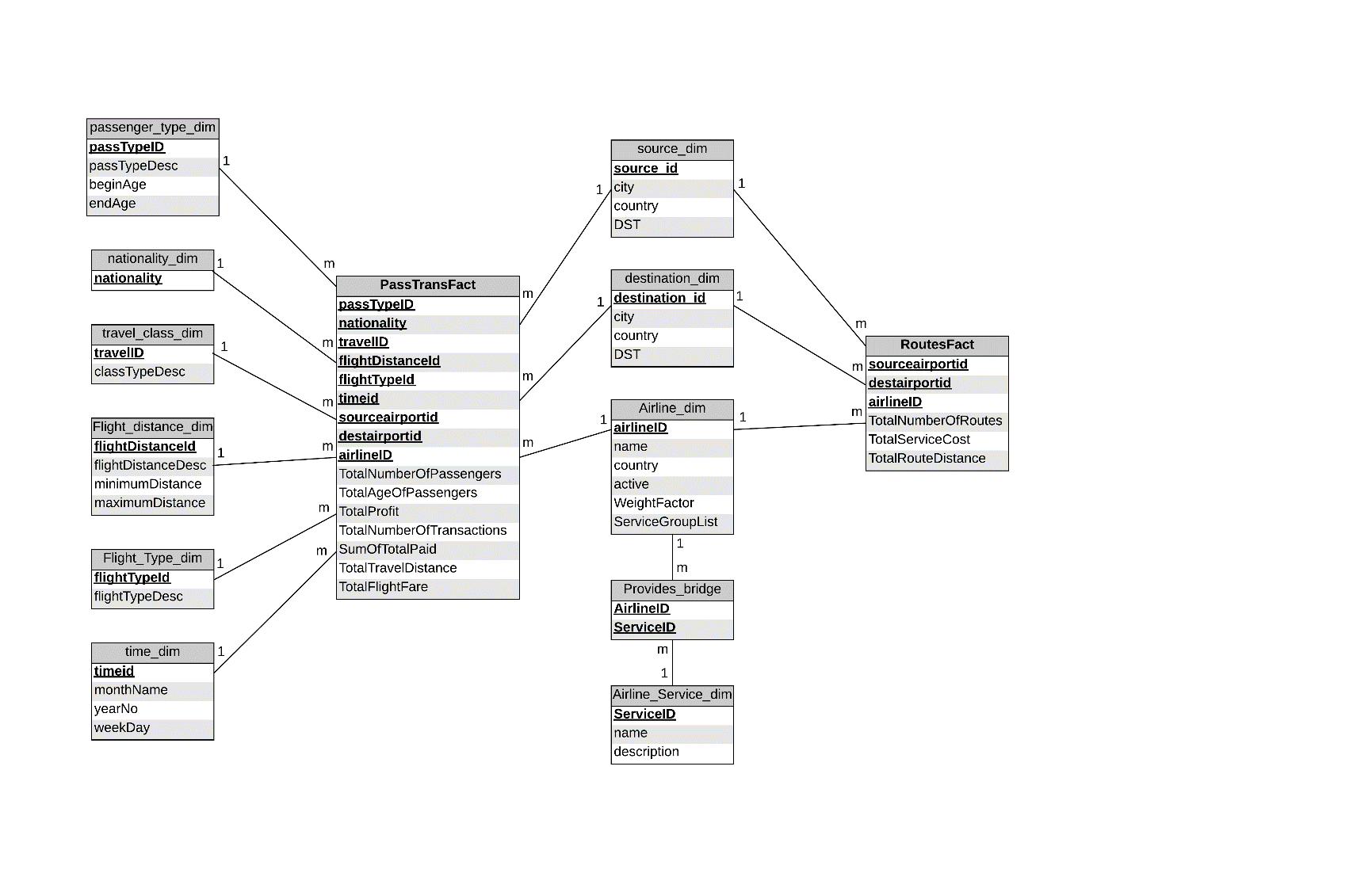
select \* from TRANSACTION;

select \* from PROVIDES;

select \* from AIRLINE\_SERVICE;

select \* from FLIGHT;

1. A star/snowflake schema (diagram) for your data warehouse



1. A short explanation of your star schema

The three main components of the above shown multi-fact snowflake schema are:

**Facts**

Facts are numeric measurements and are aggregate values. We reviewed the requirements of the management and found that some of the dimensions are not common to both the fact measures. Hence, we decided to create two fact tables as follows:

* PassTransFact: It contains measures related to passengers and flight transactions. It includes the following measures:
* TotalNumberOfPassengers
* TotalAgeOfPassengers
* TotalProfit (Total paid – flight fare)
* TotalNumberOfTransactions
* SumOfTotalPaid
* TotalTravelDistance
* TotalFlightFare
* RoutesFact: It contains measures related to routes which are as follows:
* TotalNumberOfRoutes
* TotalServiceCost
* TotalRouteDistance

**Dimensions**

Dimensions are qualifying characteristics that provide additional perspectives to a given fact. Considering management’s requirements, we have identified ten dimensions, out of which three dimensions are common to both fact tables. One of these common dimensions is connected to another dimension table through a bridge table because

* There is no direct relationship between routes and airline services
* The airline service information is not available in the service cost of a particular route

Dimensions related to PassTransFact only are:

* Passenger\_type\_dim
* Nationality\_dim
* travel\_Class\_dim
* flight\_distance\_dim
* flight\_type\_dim
* time\_dim

Dimensions related to both PassTransFact and RoutesFact are:

* source\_dim
* destination\_dim
* airline\_dim
* airline\_service\_dim

There is no such dimension which is unique to RoutesFact.

Airline\_dim dimension is connected to airline\_service\_dim dimension through a bridge table called provides\_bridge and is linked through airlineid and serviceid with airline\_dim and airline\_service\_dim respectively.

**Attributes**

These are the additional description of the dimension table and are optional. For example, the attributes for airline\_dim are:

* AirlineID
* Name
* Country
* Active

We have also added WeightFactor and ServiceGroupList in airline\_dim to depict the share of each service provided by that airline in the total service cost and the services which are available for that airline respectively.

1. **SQL statements (e.g. create table, insert into, etc) to create the star schema**

**--Creating Source Dimension**

Create table source\_dim as

Select distinct a.AIRPORTID as source\_id, a.city, a.country, a.DST

from AIRPORT a, ROUTE r

where a.AIRPORTID = r.SOURCEAIRPORTID

order by a.AIRPORTID;

**--Creating Destination Dimension**

Create table destination\_dim as

Select distinct a.AIRPORTID as destination\_id, a.city, a.country,a.DST

from AIRPORT a, ROUTE r

where a.AIRPORTID = r.DESTAIRPORTID

order by a.AIRPORTID;

**--Creating Passenger\_Type Dimension (children,Teenager,Young, Adult)**

Create table passenger\_type\_dim

(passTypeID number,

passTypeDesc varchar2 (20),

beginAge number,

endAge number

);

Select max(age) from PASSENGER;

--87

--Inserting values into passenger type dimension

Insert into passenger\_type\_dim values (1, 'Children', 0, 10);

Insert into passenger\_type\_dim values (2, 'Teenager', 11, 17);

Insert into passenger\_type\_dim values (3, 'Young Adult', 18, 35);

Insert into passenger\_type\_dim values (4, 'Middle Adult', 36, 60);

--Add a assumption for max age 87 so we hav taken 99

Insert into passenger\_type\_dim values (5, 'Senior Adult', 61, 99);

**--Creating Nationality Dimension**

Create table nationality\_dim as

Select distinct p.NATIONALITY

from PASSENGER p;

**--Create travel class Dimension**

Create table travel\_class\_dim

(travelID number,

classTypeDesc varchar2 (20)

);

-- Inserting values into travel class dimension

Insert into travel\_class\_dim values (1, 'Business Class');

Insert into travel\_class\_dim values (2, 'First Class');

Insert into travel\_class\_dim values (3, 'Economy Class');

**--Create flight distance dimension**

Create table Flight\_distance\_dim

(flightDistanceId number,

flightDistanceDesc varchar2 (20),

minimumDistance number,

maximumDistance number

);

--Inserting data into flight Distance dimension

Insert into Flight\_distance\_dim values (1, 'Small', 0,1199);

Insert into Flight\_distance\_dim values (2, 'Medium', 1200,4000);

Insert into Flight\_distance\_dim values (3, 'Large', 4001,10000);

Insert into Flight\_distance\_dim values (4, 'Very Large', 10001,19999);

**--Create Airline dimension**

Create Table Airline\_dim As

Select a.AIRLINEID, a.NAME, a.COUNTRY, a.ACTIVE, round(1.0/count(p.SERVICEID),2) As

WeightFactor , LISTAGG (p.SERVICEID, '\_') Within Group (Order By

p.SERVICEID) As ServiceGroupList

From Airline a, Provides p

Where a.AIRLINEID = p.AIRLINEID

Group By a.AIRLINEID, a.NAME, a.COUNTRY, a.ACTIVE;

**--Create flight\_type dimension**

Create table Flight\_Type\_dim

(flightTypeId number,

flightTypeDesc varchar2 (20)

);

--Insert data into flight type dimension

Insert into Flight\_Type\_dim values (1, 'Domestic');

Insert into Flight\_Type\_dim values (2, 'International');

**--Creating Airline service dimension**

Create table Airline\_Service\_dim

AS

Select \* from AIRLINE\_SERVICE;

**--Creating provides bridge table**

Create table Provides\_bridge

AS

Select \* from PROVIDES;

**--Create time dimension**

Create table time\_dim as

Select distinct to\_char(Flightdate,'MM') || to\_char(Flightdate, 'YYYY') || to\_char(Flightdate, 'DAY')as timeId, to\_char(Flightdate, 'YYYY') as yearNo,

to\_char(Flightdate,'MM') as monthName,

to\_char(Flightdate, 'DAY') as weekDay from flight

order by yearNo;

--Create temporary passtransfact table

Create table temp\_passtransfact as

Select r.sourceairportid, r.DESTAIRPORTID, p.nationality, ar.airlineid, count(p.passid) as TotalNumberOfPassengers,

sum(p.age) as TotalAgeOfPassengers, count(t.passid) as TotalNumberOfTransactions, sum(t.totalpaid) as SumOfTotalPaid,

sum(r.Distance) as TotalTravelDistance, sum(f.fare) as TotalFlightFare, p.AGE, f.fare, t.TOTALPAID, r.DISTANCE, a1.COUNTRY as SourceCountry,

a2.country as DestCountry, f.FLIGHTDATE

from Route r, airport a1, airport a2, flight f, transaction t, passenger p, airline ar

where r.sourceairportid = a1.AIRPORTID and

r.DESTAIRPORTID = a2.AIRPORTID and

r.routeid = f.routeid and f.flightid = t.flightid and

p.passid = t.passid and

ar.AIRLINEID = r.AIRLINEID

group by r.sourceairportid, r.DESTAIRPORTID, p.nationality, ar.airlineid, p.AGE, f.FARE, t.TOTALPAID, r.distance, a1.COUNTRY, a2.country, f.FLIGHTDATE;

--Alter for passenger type

Alter table temp\_passtransfact

add(passTypeID number);

--update for passenger type

update temp\_passtransfact

set passTypeID =1

where Age >= 0

and Age <= 10;

update temp\_passtransfact

set passTypeID =2

where Age >= 11

and Age <= 17;

update temp\_passtransfact

set passTypeID =3

where Age >= 18

and Age <= 35;

update temp\_passtransfact

set passTypeID =4

where Age >= 36

and Age <= 60;

update temp\_passtransfact

set passTypeID =5

where Age >= 61

and Age <= 99;

-- Alter travel class dimension

Alter table temp\_passtransfact

add(travelID number);

-- Update queries for travel class dimension

update temp\_passtransfact

set TRAVELID = 1

where TOTALPAID >= 1.8\*FARE;

update temp\_passtransfact

set TRAVELID = 2

where TOTALPAID >= 1.3\*FARE

and TOTALPAID < 1.8\*FARE;

update temp\_passtransfact

set TRAVELID = 3

where TOTALPAID < 1.3\*FARE;

-- Alter table for Flight Distance Dim

Alter table temp\_passtransfact

add(flightDistanceid number);

-- Update queries for Flight Distance Dim

update temp\_passtransfact

set FLIGHTDISTANCEID = 1

where DISTANCE >= 0

AND DISTANCE <= 1199;

update temp\_passtransfact

set FLIGHTDISTANCEID = 2

where DISTANCE >= 1200

AND DISTANCE <= 4000;

update temp\_passtransfact

set FLIGHTDISTANCEID = 3

where DISTANCE >= 4001

AND DISTANCE <= 10000;

update temp\_passtransfact

set FLIGHTDISTANCEID = 4

where DISTANCE >= 10001

AND DISTANCE <= 19999;

-- Alter table for Flight Type Dim

Alter table temp\_passtransfact

add(flightTypeid number);

-- Update queries for Flight Type Dim

update temp\_passtransfact

set flightTypeid = 1

where SOURCECOUNTRY = DESTCOUNTRY;

update temp\_passtransfact

set flightTypeid = 2

where SOURCECOUNTRY != DESTCOUNTRY;

Create table passtransfact as

Select f.passtypeid, f.NATIONALITY, f.TRAVELID, f.FLIGHTDISTANCEID, f.FLIGHTTYPEID,

to\_char(f.Flightdate,'MM') || to\_char(f.Flightdate, 'YYYY') || to\_char(f.Flightdate, 'DAY')as timeId,

f.SOURCEAIRPORTID, f.DESTAIRPORTID, f.AIRLINEID, f.TOTALNUMBEROFPASSENGERS, f.TOTALAGEOFPASSENGERS,

sum (f.SUMOFTOTALPAID - f.TOTALFLIGHTFARE) as TotalProfit, f.TOTALNUMBEROFTRANSACTIONS, f.SUMOFTOTALPAID,

f.TOTALTRAVELDISTANCE, f.TOTALFLIGHTFARE

from temp\_passtransfact f

group by f.passtypeid, f.NATIONALITY, f.TRAVELID, f.FLIGHTDISTANCEID, f.FLIGHTTYPEID,

to\_char(f.Flightdate,'MM') || to\_char(f.Flightdate, 'YYYY') || to\_char(f.Flightdate, 'DAY'),

f.SOURCEAIRPORTID, f.DESTAIRPORTID, f.AIRLINEID, f.TOTALNUMBEROFPASSENGERS, f.TOTALAGEOFPASSENGERS,

f.TOTALNUMBEROFTRANSACTIONS, f.SUMOFTOTALPAID,

f.TOTALTRAVELDISTANCE, f.TOTALFLIGHTFARE;

--Fact Table 2

Create table Routesfact as

Select r.sourceairportid, r.DESTAIRPORTID, ar.airlineid, count(r.ROUTEID) as TotalNumberOfRoutes,

sum(r.SERVICECOST) as TotalServiceCost, sum (r.DISTANCE) as TotalRouteDistance

from Route r, airport a1, airport a2, airline ar

where r.sourceairportid = a1.AIRPORTID and

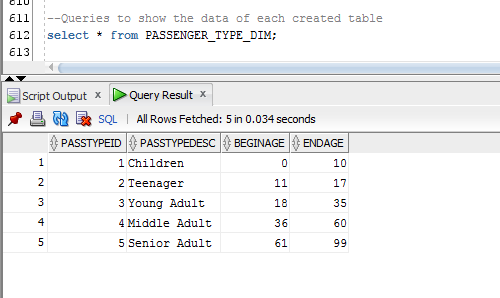
r.DESTAIRPORTID = a2.AIRPORTID and

ar.AIRLINEID = r.AIRLINEID

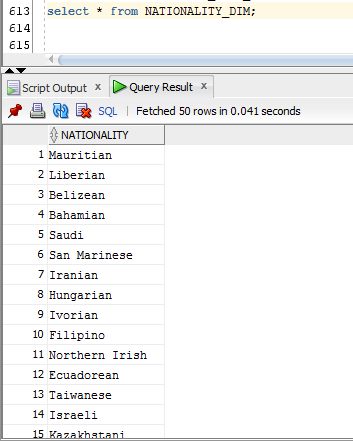
group by r.sourceairportid, r.DESTAIRPORTID, ar.airlineid;

1. Screen shots of the tables that you have created; this includes the contents of each table that you have created. If the table is very big, you can print a snapshot of the contents of the table

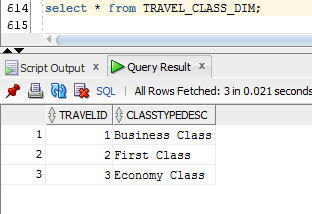
Screenshot showing passenger\_type\_dim table



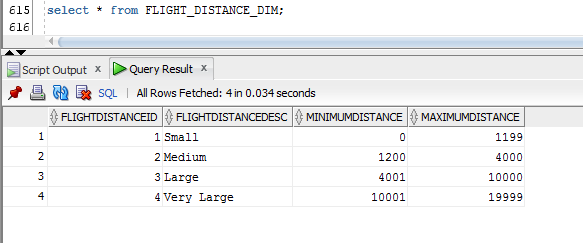
Screenshot showing nationality\_dim table



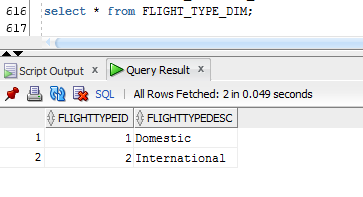
Screenshot showing travel\_class\_dim table



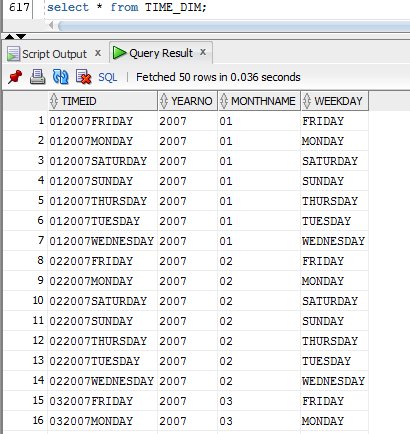
Screenshot showing flight\_distance\_dim table



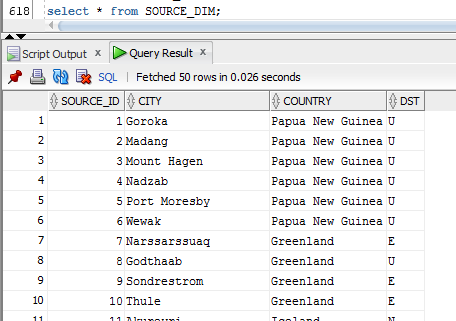
Screenshot showing flight\_type\_dim table



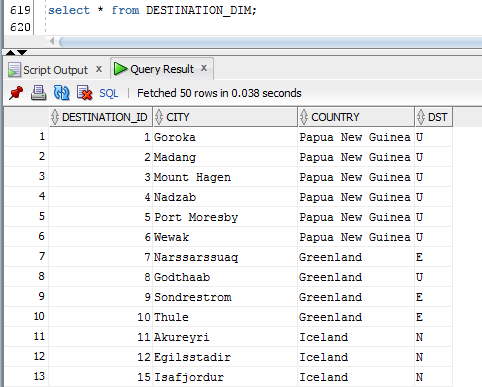
Screenshot showing time\_dim table



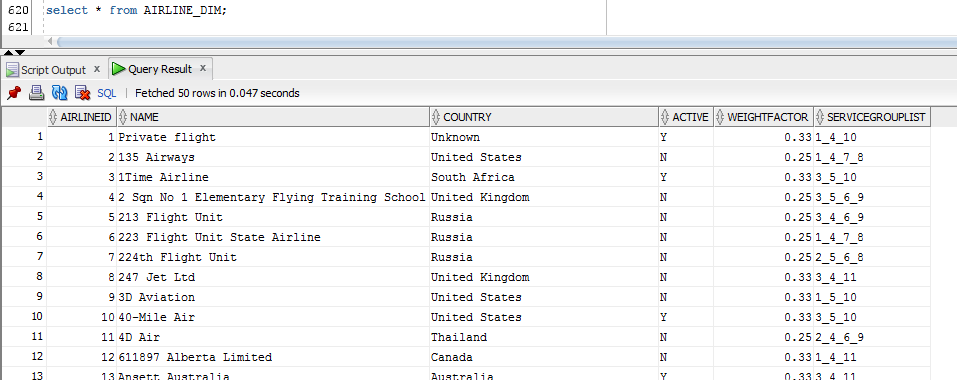
Screenshot showing source\_dim table



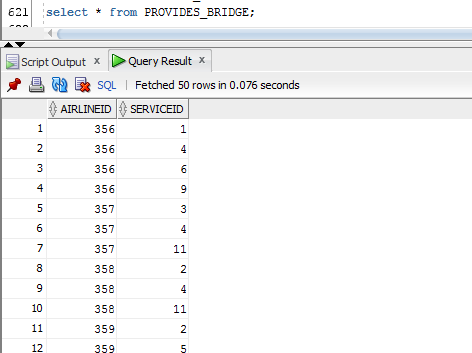
Screenshot showing destination\_dim table



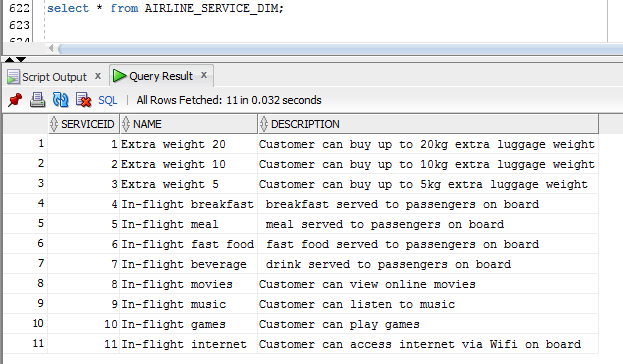
Screenshot showing airline\_dim table



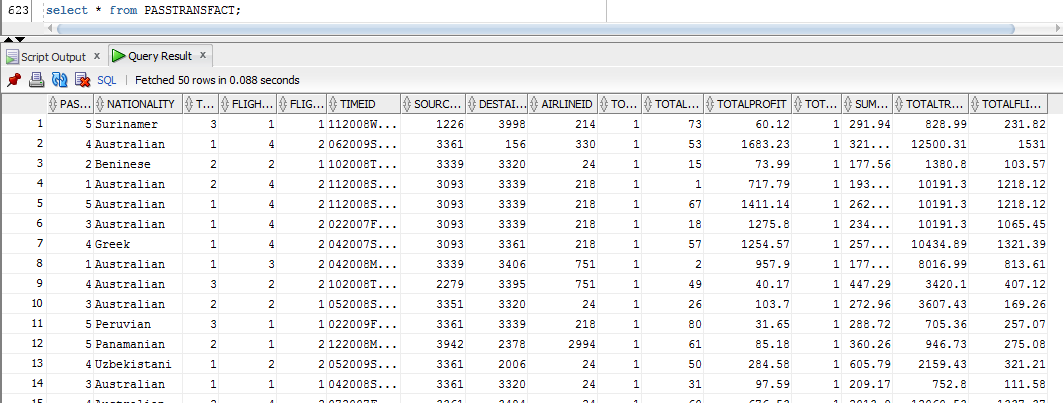
Screenshot showing provides\_bridge table



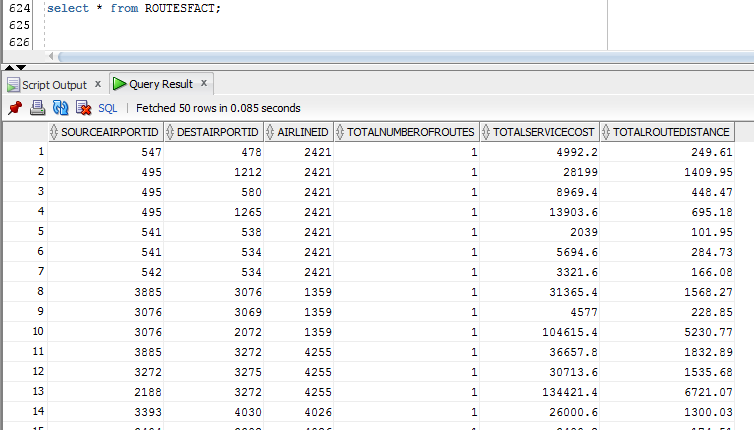
Screenshot showing airline\_service\_dim table



Screenshot showing PassTransFact table



Screenshot showing RoutesFact table



# Task C.2

## Simple reports

### Report 1: Show All Report

1. Query Question

What is the total number of passengers who departed from cities of Australia each year?

1. Explanation

The top management would be interested in the total number of passengers travelling in a particular year from a particular country to take decisions related to the frequency of the flights from the source airport.

1. SQL command

Select t.YEARNO as Year, s.city as Source\_City, sum(f.TOTALNUMBEROFPASSENGERS) as Total\_Number\_Of\_Passengers

from source\_dim s, passtransfact f, time\_dim t

where s.SOURCE\_ID = f.SOURCEAIRPORTID and

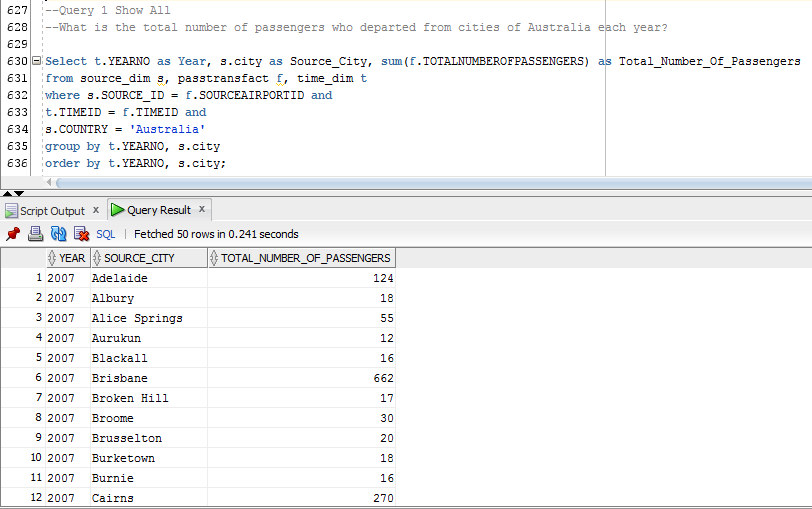
t.TIMEID = f.TIMEID and

s.COUNTRY = 'Australia'

group by t.YEARNO, s.city

order by t.YEARNO, s.city;

1. Screenshot of Query result



### Report 2: Top k Report

1. Query Question

Which are the top 3 airlines with maximum profit for Australia in the year 2007 and show whether they are active or not?

1. Explanation

The top management would be interested in determing which of the active airlines is generating maximum profit in a particular year and particular country.

1. SQL command

SELECT \*

FROM

(SELECT a.Name, a.ACTIVE, SUM(f.TOTALPROFIT) AS Profit,

RANK() OVER (ORDER BY SUM(f.TOTALPROFIT) DESC ) AS

Airline\_RANK

FROM passtransfact f, AIRLINE\_DIM a, TIME\_DIM t

WHERE f.TIMEID = t.TIMEID

AND f.AIRLINEID = a.AIRLINEID

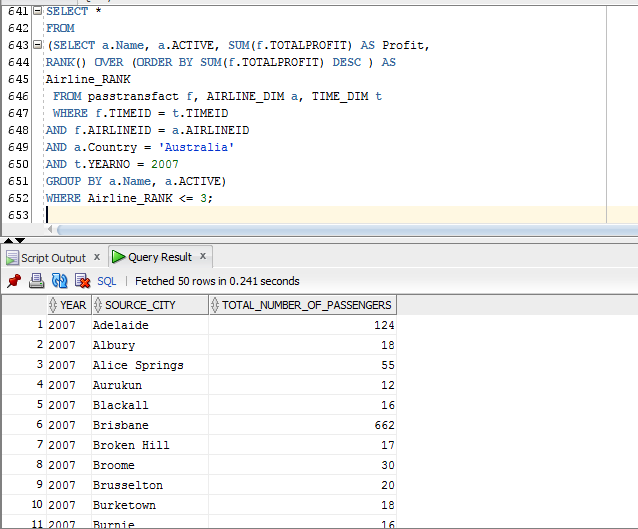
AND a.Country = 'Australia'

AND t.YEARNO = 2007

GROUP BY a.Name, a.ACTIVE)

WHERE Airline\_RANK <= 3;

1. Screenshot of Query result



### Report 3: Top n% Report

1. Query Question

Calculate the top 5% of the total service cost of qantas airlines from different source cities?

1. Explanation

The top management would be interested in knowing the top 5% airlines based on their spending on providing services.

1. SQL command

SELECT \*

FROM (

SELECT

s.CITY as source\_city, ar.NAME,

sum(f.TotalServiceCost) AS Total\_Service\_Cost,

percent\_rank() over

(order by sum(f.TotalServiceCost) ) as "Percent Rank"

FROM Routesfact f, source\_dim s, AIRLINE\_DIM ar

WHERE f.SOURCEAIRPORTID = s.source\_id

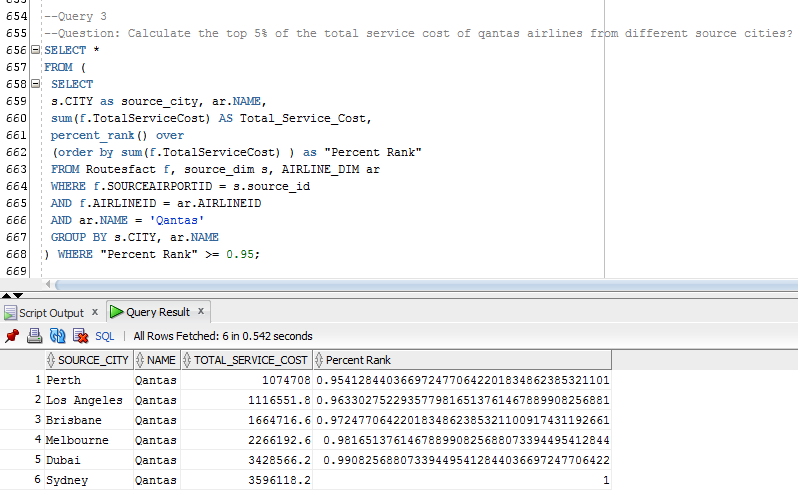
AND f.AIRLINEID = ar.AIRLINEID

AND ar.NAME = 'Qantas'

GROUP BY s.CITY, ar.NAME

) WHERE "Percent Rank" >= 0.95;

1. Screenshot of Query result



## More Advanced Reports

### Report 4: City-to-City Routes’ Report

1. Query question

What is the average route distance between source cities and destination cities?

1. SQL command

select decode(grouping(s.CITY),1,'Any City',s.CITY) as "Departure City",

decode(grouping(s.COUNTRY),1,'Any Country',s.COUNTRY) as "Departure Country",

decode(grouping(d.CITY),1,'Any City',d.CITY) as "Arrival City",

decode(grouping(d.COUNTRY),1,'Any Country',d.COUNTRY) as "Arrival Country",

sum(r.TOTALNUMBEROFROUTES) as "Number of routes",

sum(r.TOTALROUTEDISTANCE)/sum(r.TOTALNUMBEROFROUTES) as "Average Distance"

from destination\_dim d, source\_dim s, routesfact r

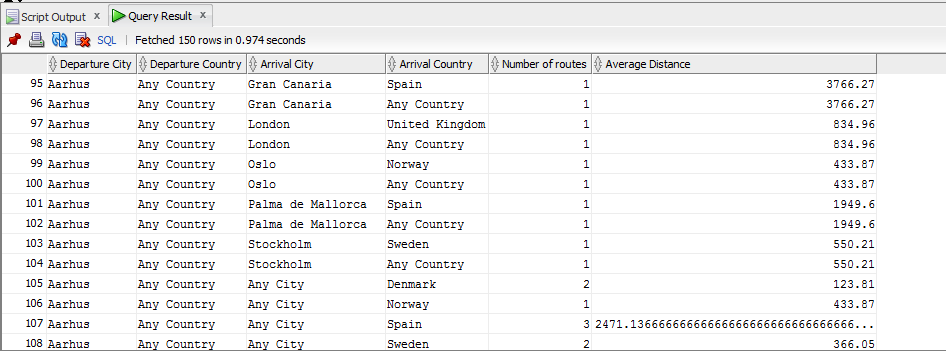
where d.DESTINATION\_ID= r.DESTAIRPORTID

and s.SOURCE\_ID = r.SOURCEAIRPORTID

group by cube(s.CITY,s.COUNTRY,d.CITY, d.COUNTRY)

order by s.CITY,s.COUNTRY,d.CITY,d.COUNTRY;

1. Screenshot of query result



### Report 5: Airline’s Report

1. Query question

What is the average agent profit every year for each airline from source countries to destination countries with respect to each flight type?

1. SQL command

SELECT t.YEARNO, ar.NAME,

decode(grouping(ft.FLIGHTTYPEDESC),1,'All Flight Type',ft.FLIGHTTYPEDESC) AS FlightType,

decode(grouping(s.COUNTRY),1,'Any Country',s.COUNTRY) AS SourceCountry,

decode(grouping(d.COUNTRY),1,'Any Country',d.COUNTRY) AS DestinationCountry,

sum(f.TOTALNUMBEROFTRANSACTIONS) AS "Number of Transactions",

sum(f.TOTALPROFIT) AS "Average Agent Profit"

FROM passtransfact f, time\_dim t, airline\_dim ar, Flight\_Type\_dim ft, source\_dim s, destination\_dim d

WHERE f.TIMEID = t.TIMEID

and f.AIRLINEID = ar.AIRLINEID

and f.FLIGHTTYPEID = ft.FLIGHTTYPEID

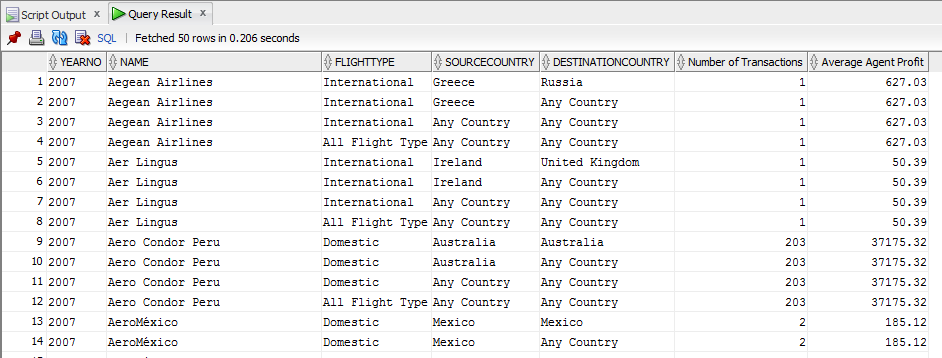
and f.SOURCEAIRPORTID = s.SOURCE\_ID

and f.DESTAIRPORTID = d.DESTINATION\_ID

GROUP by t.YEARNO, ar.NAME, rollup(ft.FLIGHTTYPEDESC, s.COUNTRY, d.COUNTRY)

order by t.YEARNO, ar.NAME, ft.FLIGHTTYPEDESC, s.COUNTRY, d.COUNTRY;

1. Screenshot of query result



### Report 6: Flight’s Report

1. Query question

What is the average paid ticket per weekday for each flight type and class from source country to destination country?

1. SQL command

SELECT t.WEEKDAY,

decode(grouping(ft.FLIGHTTYPEDESC),1,'All Flight Type',ft.FLIGHTTYPEDESC) AS FlightType,

decode(grouping(tr.CLASSTYPEDESC),1,'Any Class',tr.CLASSTYPEDESC) AS FlightClass,

decode(grouping(s.COUNTRY),1,'Any Country',s.COUNTRY) AS SourceCountry,

decode(grouping(d.COUNTRY),1,'Any Country',d.COUNTRY) AS DestinationCountry,

sum(f.TOTALNUMBEROFTRANSACTIONS) AS "Number of Transactions",

(sum(f.SUMOFTOTALPAID)/ sum(f.TOTALNUMBEROFTRANSACTIONS)) AS "Average Paid Ticket (USD)"

FROM passtransfact f, time\_dim t, TRAVEL\_CLASS\_DIM tr, Flight\_Type\_dim ft, source\_dim s, destination\_dim d

WHERE f.TIMEID = t.TIMEID

and f.TRAVELID = tr.TRAVELID

and f.FLIGHTTYPEID = ft.FLIGHTTYPEID

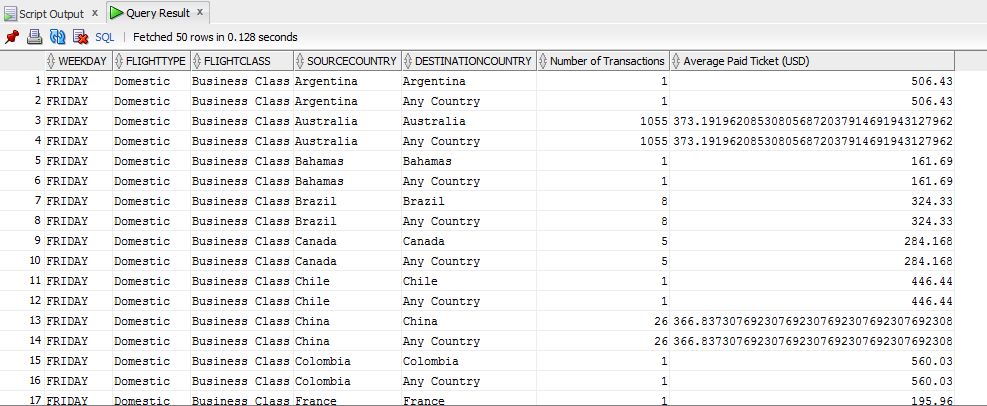
and f.SOURCEAIRPORTID = s.SOURCE\_ID

and f.DESTAIRPORTID = d.DESTINATION\_ID

GROUP by t.WEEKDAY, rollup(ft.FLIGHTTYPEDESC,tr.CLASSTYPEDESC, s.COUNTRY, d.COUNTRY)

order by t.WEEKDAY, ft.FLIGHTTYPEDESC,tr.CLASSTYPEDESC, s.COUNTRY, d.COUNTRY;

1. Screenshot of query result



## Reports with Cube and Rollup and the comparison

### Report 7: Report using Cube

1. Query question

What is the total revenue generated in the month of September for each flight type per travel class?

1. Explanation

The top management would be interested in finding the total revenue generated in a particular month, and a particular year and produced for all possible combinations of subtotals for the attributes Year, Monthname, flightTypeDesc and ClassTypeDesc

1. SQL command

SELECT t.YEARNO, t.MONTHNAME,

decode(grouping(ft.FLIGHTTYPEDESC),1,'All Flight Type',ft.FLIGHTTYPEDESC) AS FlightType,

decode(grouping(tr.CLASSTYPEDESC),1,'Any Class',tr.CLASSTYPEDESC) AS FlightClass,

sum(f.SUMOFTOTALPAID) as Total\_Revenue

FROM TIME\_DIM t, FLIGHT\_TYPE\_DIM ft, TRAVEL\_CLASS\_DIM tr, PASSTRANSFACT f

WHERE f.TIMEID = t.TIMEID

AND ft.FLIGHTTYPEID = f.FLIGHTTYPEID

AND tr.TRAVELID = f.TRAVELID

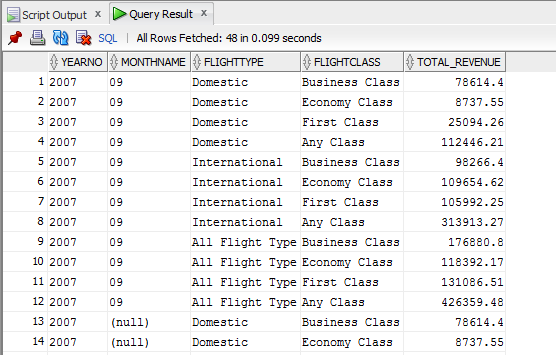
and t.YEARNO = 2007

and t.MONTHNAME = 09

GROUP BY CUBE(t.YEARNO, t.MONTHNAME, ft.FLIGHTTYPEDESC, tr.CLASSTYPEDESC)

Order by t.YEARNO, t.MONTHNAME;

1. Screenshot of query result



### Report 8: Report using Rollup

1. Query question

What is the total travel distance travelled by selected airlines ('China Eastern Airlines', 'IndiGo Airlines', 'Virgin America') for each year per flight type?

1. Explanation

The top management is interested in determining the total travel distance of selected few airlines for particular year for domestic and international flights.

1. SQL command

SELECT

decode(grouping(t.YEARNO),1,'All Years',t.YEARNO) as Year,

decode(grouping(a.NAME),1,'All AirLines',a.NAME) AS AirlineName,

decode(grouping(ft.FLIGHTTYPEDESC),1,'All Types',ft.FLIGHTTYPEDESC) AS FlightType,

sum(f.TotalTravelDistance) as Total\_Distance

FROM TIME\_DIM t, AIRLINE\_DIM a, FLIGHT\_TYPE\_DIM ft, PASSTRANSFACT f

WHERE f.TIMEID = t.TIMEID

AND ft.FLIGHTTYPEID = f.FLIGHTTYPEID

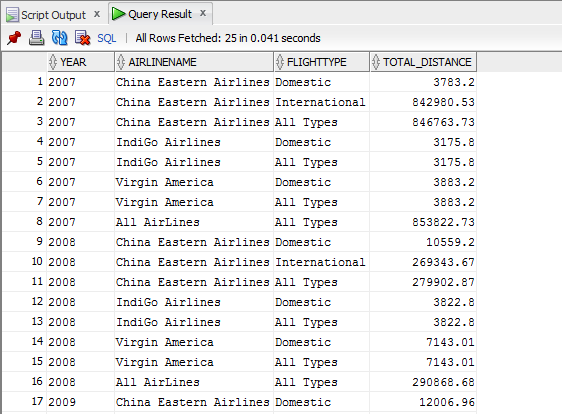
AND a.airlineid = f.airlineid

and a.name IN ('China Eastern Airlines','IndiGo Airlines','Virgin America')

GROUP BY ROLLUP(t.YEARNO, a.NAME, ft.FLIGHTTYPEDESC)

Order by t.YEARNO, a.NAME, ft.FLIGHTTYPEDESC;

1. Screenshot of query result



1. Comparison between report 7 and report 8

* Cube: Cube generates result set that shows aggregate for all possible combination of values in selected column. In the event, that we had used Cube in Report 8, additional rows showing all possible combinations and subtotals of specified attributes year, Airline name, flightTypeDesc
* RollUp: RollUp generates result set that shows aggregate hierarchy of values in the selected column. In the event, that we had used Rollup in Report 7, the rows showing subtotals and grand totals about Year, Monthname, flightTypeDesc and ClassTypeDesc attributes will not be present. Rollup is a summarized report as compared to Cube.

## Reports with Moving and Cumulative Aggregates

### Report 9

1. Query question

What are the total and cumulative monthly profits of small flight distance departing from Sydney airport in 2007?

1. Explanation

The top management would be interested in determining the total and cumulative monthly profits for flights covering small distance departing from a particular source country in particular year.

1. SQL Command

SELECT t.yearNo, t.monthname, f.flightdistancedesc, s.city as source\_city,

TO\_CHAR (SUM(p.TotalProfit), '9,999,999,999') AS TotalProfit,

TO\_CHAR (SUM(SUM(p.TotalProfit)) OVER

(ORDER BY t.yearNo, t.monthname, f.flightdistancedesc, s.city ROWS UNBOUNDED PRECEDING),

'9,999,999,999') AS CUM\_profit

FROM time\_dim t, passtransfact p, flight\_distance\_dim f, source\_dim s

WHERE t.TIMEID = p.TIMEID

AND s.SOURCE\_ID = p.SOURCEAIRPORTID

and f.FLIGHTDISTANCEID = p.FLIGHTDISTANCEID

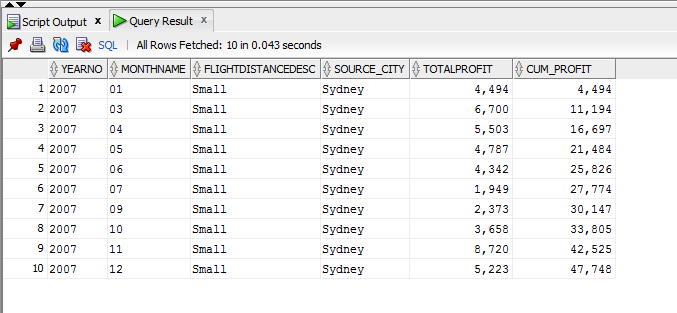
AND t.YEARNO = 2007

AND f.FLIGHTDISTANCEDESC = 'Small'

and s.CITY = 'Sydney'

GROUP BY t.yearNo, t.monthname, f.flightdistancedesc, s.city;

1. Screenshot of query result



### Report 10

1. Query question

What are the total and moving 3 monthly transactions of Australian passengers in 2009?

1. Explanation

The top management would be interested in the determining the total and moving 3 monthly transactions from passengers of a particular nationality in a particular year.

1. SQL Command

SELECT t.YEARNO,t.MONTHNAME, n.NATIONALITY,

TO\_CHAR (SUM(p.TotalNumberofTransactions), '9,999,999,999') AS TotalTransactions,

TO\_CHAR (AVG(SUM(p.TotalNumberofTransactions)) OVER

(ORDER BY t.YEARNO,t.MONTHNAME, n.NATIONALITY

rows 2 preceding),

'9,999,999,999') AS moving\_3\_month\_avg

FROM TIME\_DIM t, PASSTRANSFACT p, NATIONALITY\_DIM n

WHERE t.TIMEID = p.TIMEID

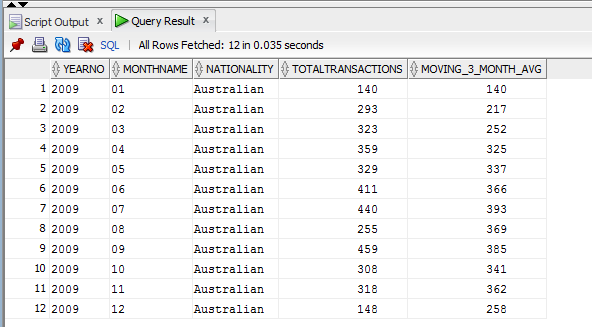
AND n.NATIONALITY = p.NATIONALITY

AND t.YEARNO=2009

and n.NATIONALITY= 'Australian'

GROUP BY t.YEARNO,t.MONTHNAME, n.NATIONALITY;

1. Screenshot of query result



### Report 11

1. Query question

What are the total and cumulative monthly profits from Middle Adult passengers travelling by Southwest Airlines every year?

1. Explanation

The top management would be interested in determining the total and cumulative monthly profits generated by the particular airlines from passengers of particular age group every year.

1. SQL Command

SELECT p.PASSTYPEDESC, a.NAME, t.YEARNO, t.MONTHNAME,

TO\_CHAR (SUM(f.TOTALPROFIT), '9,999,999,999') AS PROFIT,

TO\_CHAR (SUM(SUM(f.TOTALPROFIT)) OVER

(PARTITION BY t.YEARNO ORDER BY p.PASSTYPEDESC, a.NAME, t.YEARNO, t.MONTHNAME desc

ROWS UNBOUNDED PRECEDING),

'9,999,999,999') AS CUM\_PROFIT

FROM PASSENGER\_TYPE\_DIM p, PASSTRANSFACT f, TIME\_DIM t, AIRLINE\_DIM a

WHERE p.passtypeid = f.passtypeid

AND t.TIMEID = f.TIMEID

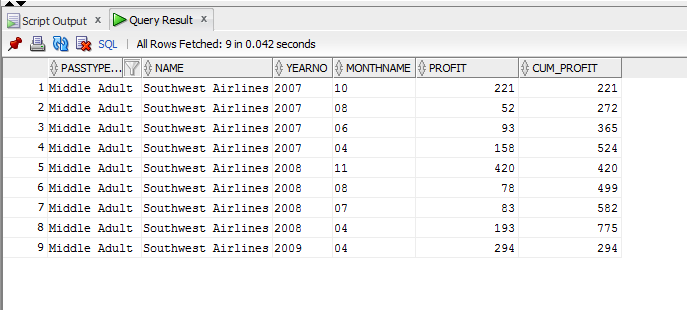
AND a.AIRLINEID = f.AIRLINEID

AND a.NAME = 'Southwest Airlines'

AND p.PASSTYPEDESC = 'Middle Adult'

GROUP BY p.PASSTYPEDESC, a.NAME, t.YEARNO, t.MONTHNAME;

1. Screenshot of query result



### Report 12

1. Query question

What are the total and moving 3 monthly number of passengers travelling in business class in flights departing from Canada?

1. Explanation

The report would interest top management as it might want to determine the total number of passengers travelling in particular flight class who are departing from particular country on moving 3 monthly basis.

1. SQL Command

SELECT tr.CLASSTYPEDESC, s.COUNTRY, t.YEARNO,

TO\_CHAR(SUM(f.TOTALNUMBEROFPASSENGERS)) AS "NumberOf Passengers",

TO\_CHAR(AVG(SUM(f.TOTALNUMBEROFPASSENGERS)) OVER

(PARTITION BY t.YEARNO ORDER BY SUM(f.TOTALNUMBEROFPASSENGERS)

ROWS 2 PRECEDING)) AS MOVING\_3\_YEAR,

TO\_CHAR (AVG(SUM(f.TOTALNUMBEROFPASSENGERS)) OVER

(partition by tr.classtypedesc ORDER BY sum(f.TOTALNUMBEROFPASSENGERS)

ROWS 2 PRECEDING)) AS MOVING\_3\_TravelClass

FROM PASSTRANSFACT f, TRAVEL\_CLASS\_DIM tr, SOURCE\_DIM s, TIME\_DIM t

WHERE t.TIMEID = f.TIMEID

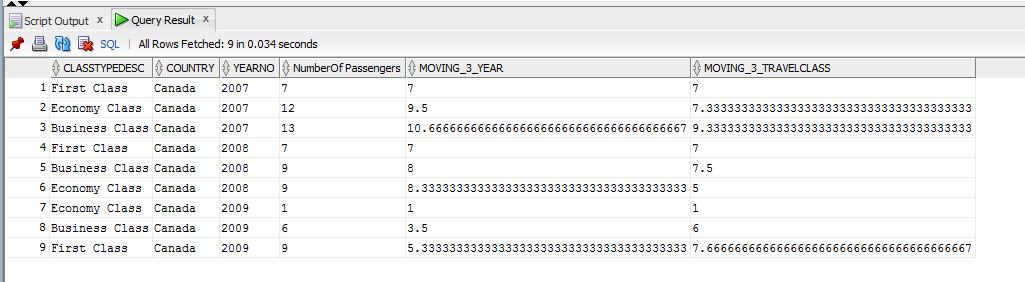
AND tr.travelid = f.travelid

and s.source\_id = f.sourceairportid

AND s.country = 'Canada'

GROUP BY tr.CLASSTYPEDESC, s.COUNTRY, t.YEARNO;

1. Screenshot of query result



## Reports with Rank and Percent Rank and the comparison

### Report 13

1. Query question

What are the city ranks by total service cost of source airports in each country?

1. Explanation

The top management would be interested in finding the rank of cities in each country based on total service cost.

1. SQL Command

SELECT s.country,s.city,

TO\_CHAR(SUM(TotalServiceCost)) AS TotalServiceCost,

RANK() OVER (PARTITION BY s.country

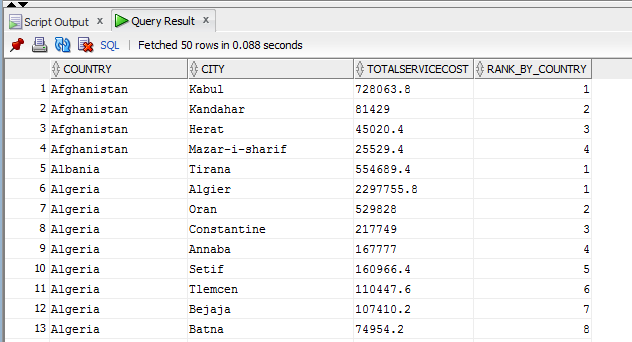
ORDER BY SUM(TotalServiceCost) DESC) AS RANK\_BY\_COUNTRY

FROM source\_dim s, routesfact r

WHERE s.source\_id=r.sourceairportid

GROUP BY s.country,s.city;

1. Screenshot of query result



### Report 14

1. Query question

What is the top 10% total revenue by nationality ('Angolan', 'Australian', 'British', 'Bangladeshi', 'Chinese', 'Batswana') and travel class?

1. Explanation

The top management would be interested in finding the total revenue from passengers of different nationalities per travel class.

1. SQL Command

select \*

from (

SELECT

n.NATIONALITY, tr.classtypedesc,

sum(f.SUMOFTOTALPAID) as "Total Revenue",

percent\_rank() over

(partition by n.nationality order by sum(f.SUMOFTOTALPAID)) as "Percent Rank by nationality",

percent\_rank() over

(partition by tr.classtypedesc order by sum(f.SUMOFTOTALPAID)) as "Percent Rank by travel class"

FROM PASSTRANSFACT f, TRAVEL\_CLASS\_DIM tr, NATIONALITY\_DIM n

WHERE tr.TRAVELID = f.TRAVELID

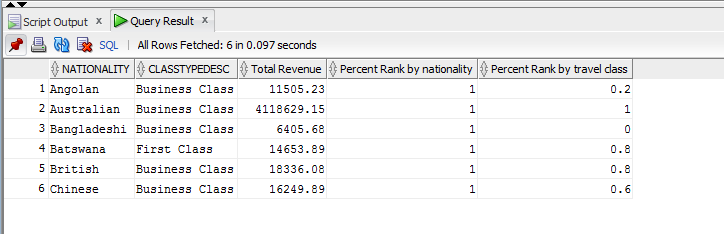
and n.NATIONALITY = f.NATIONALITY

and n.NATIONALITY IN ('Angolan','Australian','British','Bangladeshi','Chinese','Batswana')

GROUP BY n.NATIONALITY, tr.classtypedesc

) where "Percent Rank by nationality" >= 0.9;

1. Screenshot of query result



1. Comparison of rank and percent rank

* Rank: It computes the rank of a record compared to other records in the table based on the values of a set of measures. For example, in report 13, the cities in each country are ranked based on service cost.
* Percent rank: Percent rank also computes the ranking of a record but in a percentage form unlike the number form through rank. For example, in report 14, based on total revenue, we have determined the top 10% of the total revenue generated by a particular nationality of passengers, travelling a particular class.