--Report 1

select td.yearnumber,nd.nationality, sum(Total\_number\_\_transactions) as CountPassengers

from transaction\_fact tf,time\_dim td, nationality\_dim nd, TRAVEL\_CLASS\_DIM tc

where tf.time\_id = td.time\_id

and tf.natid = nd.natid

and tf.TRAVEL\_CLASS\_ID = tc.travel\_class\_id

and nd.nationality = 'Australian'

and td.yearnumber = '2008'

and tc.TRAVEL\_CLASS\_TYPE = 'First Class'

Group By td.yearnumber,nd.nationality;

--Report 1 Alternate Query

select /\*+ ORDERED USE\_HASH (tc nd) \*/ td.yearnumber,nd.nationality, sum(Total\_number\_\_transactions) as CountPassengers

from transaction\_fact tf,time\_dim td, nationality\_dim nd, TRAVEL\_CLASS\_DIM tc

where tf.time\_id = td.time\_id

and tf.natid = nd.natid

and tf.TRAVEL\_CLASS\_ID = tc.travel\_class\_id

and nd.nationality = 'Australian'

and td.yearnumber = '2008'

and tc.TRAVEL\_CLASS\_TYPE = 'First Class'

Group By td.yearnumber,nd.nationality;

--Report 2

select ad.name as AirlineName, sum(total\_total\_paid)-sum(tf.TOTAL\_FARE) as TotalProfit

from airline\_dim ad, transaction\_fact tf, time\_dim td

where ad.AIRLINEID = tf.AIRLINEID

and tf.TIME\_ID = td.TIME\_ID

and td.yearnumber = '2007'

Group By ad.name;

--Report 2 Alternate Query

select /\*+ USE\_MERGE (ad td) \*/ ad.name as AirlineName, sum(total\_total\_paid)-sum(tf.TOTAL\_FARE) as TotalProfit

from airline\_dim ad, transaction\_fact tf, time\_dim td

where ad.AIRLINEID = tf.AIRLINEID

and tf.TIME\_ID = td.TIME\_ID

and td.yearnumber = '2007'

Group By ad.name;

--Report 3

select

decode(grouping(sa.city),1,'Any City',sa.city) as "Departure City",

decode(grouping(sa.country),1,'Any Country',sa.country) as "Departure Country",

decode(grouping(da.city),1,'Any City',da.city) as "Arrival City",

decode(grouping(da.country),1,'Any Country',da.country) as "Arrival Country",

sum(rf.TOTAL\_ROUTES) as "Number of Routes",

round(sum(rf.TOTAL\_ROUTE\_DISTANCE)/sum(rf.TOTAL\_ROUTES),2) as "Average Distance"

from route\_fact rf, source\_airport\_dim sa, dest\_airport\_dim da

where rf.sourceairportid = sa.airportid

and rf.destairportid = da.airportid

group by cube (sa.CITY, sa.COUNTRY, da.CITY, da.COUNTRY)

order by sa.COUNTRY,sa.CITY;

--Report 3 Alternate Query

select /\*+ USE\_NL (sa rf) \*/

decode(grouping(sa.city),1,'Any City',sa.city) as "Departure City",

decode(grouping(sa.country),1,'Any Country',sa.country) as "Departure Country",

decode(grouping(da.city),1,'Any City',da.city) as "Arrival City",

decode(grouping(da.country),1,'Any Country',da.country) as "Arrival Country",

sum(rf.TOTAL\_ROUTES) as "Number of Routes",

round(sum(rf.TOTAL\_ROUTE\_DISTANCE)/sum(rf.TOTAL\_ROUTES),2) as "Average Distance"

from route\_fact rf, source\_airport\_dim sa, dest\_airport\_dim da

where rf.sourceairportid = sa.airportid

and rf.destairportid = da.airportid

group by cube (sa.CITY, sa.COUNTRY, da.CITY, da.COUNTRY)

order by sa.COUNTRY,sa.CITY;

--Report 4

select

decode(grouping(sa.city),1,'Any City',sa.city) as "Departure City",

decode(grouping(sa.country),1,'Any Country',sa.country) as "Departure Country",

decode(grouping(da.city),1,'Any City',da.city) as "Arrival City",

decode(grouping(da.country),1,'Any Country',da.country) as "Arrival Country",

sum(rf.TOTAL\_ROUTES) as "Number of Routes",

round(sum(rf.TOTAL\_ROUTE\_DISTANCE)/sum(rf.TOTAL\_ROUTES),2) as "Average Distance"

from route\_fact rf, source\_airport\_dim sa, dest\_airport\_dim da

where rf.sourceairportid = sa.airportid

and rf.destairportid = da.airportid

group by cube (sa.CITY, sa.COUNTRY, da.CITY, da.COUNTRY)

order by sa.COUNTRY,sa.CITY;

--Report(4) Alternate Query

select /\*+ USE\_NL (sa rf) \*/

decode(grouping(sa.city),1,'Any City',sa.city) as "Departure City",

decode(grouping(sa.country),1,'Any Country',sa.country) as "Departure Country",

decode(grouping(da.city),1,'Any City',da.city) as "Arrival City",

decode(grouping(da.country),1,'Any Country',da.country) as "Arrival Country",

sum(rf.TOTAL\_ROUTES) as "Number of Routes",

round(sum(rf.TOTAL\_ROUTE\_DISTANCE)/sum(rf.TOTAL\_ROUTES),2) as "Average Distance"

from route\_fact rf, source\_airport\_dim sa, dest\_airport\_dim da

where rf.sourceairportid = sa.airportid

and rf.destairportid = da.airportid

group by cube (sa.CITY, sa.COUNTRY, da.CITY, da.COUNTRY)

order by sa.COUNTRY,sa.CITY;

---------------------Report(5) Original Query----------------------------------

select decode(grouping(td.YEARNUMBER),1,'Any Year',td.YEARNUMBER) as "Flight Year",

decode(grouping(ad.NAME),1,'Any Airline',ad.NAME) as "Airline Name",

decode(grouping(ft.FLIGHT\_TYPE),1,'All Flight Type',ft.FLIGHT\_TYPE) as "Flight Type",

decode(grouping(sa.COUNTRY),1,'Any Country',sa.COUNTRY) as "Source Country",

decode(grouping(da.COUNTRY),1,'Any Country',da.COUNTRY) as "Destination Country",

sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Number of Transactions",

round(sum(tf.TOTAL\_TOTAL\_PAID) - sum(tf.TOTAL\_FARE)/sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS),2) as "Average Agent Profit(USD)"

from transaction\_fact tf, airline\_dim ad, flight\_type\_dim ft,

time\_dim td, source\_airport\_dim sa, dest\_airport\_dim da

where tf.AIRLINEID = ad.AIRLINEID

and tf.FLIGHT\_TYPE\_ID = ft.FLIGHT\_TYPE\_ID

and tf.time\_ID = td.time\_ID

and tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.DEST\_AIRPORT\_ID = da.AIRPORTID

and (sa.COUNTRY = da.COUNTRY

or sa.COUNTRY != da.COUNTRY)

group by td.YEARNUMBER,ad.NAME, rollup (ft.FLIGHT\_TYPE,sa.COUNTRY,da.COUNTRY);

--report 5 (Alternate Query)

select /\*+ USE\_MERGE (sa tf) \*/ decode(grouping(td.YEARNUMBER),1,'Any Year',td.YEARNUMBER) as "Flight Year",

decode(grouping(ad.NAME),1,'Any Airline',ad.NAME) as "Airline Name",

decode(grouping(ft.FLIGHT\_TYPE),1,'All Flight Type',ft.FLIGHT\_TYPE) as "Flight Type",

decode(grouping(sa.COUNTRY),1,'Any Country',sa.COUNTRY) as "Source Country",

decode(grouping(da.COUNTRY),1,'Any Country',da.COUNTRY) as "Destination Country",

sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Number of Transactions",

round(sum(tf.TOTAL\_TOTAL\_PAID) - sum(tf.TOTAL\_FARE)/sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS),2) as "Average Agent Profit(USD)"

from transaction\_fact tf, airline\_dim ad, flight\_type\_dim ft,

time\_dim td, source\_airport\_dim sa, dest\_airport\_dim da

where tf.AIRLINEID = ad.AIRLINEID

and tf.FLIGHT\_TYPE\_ID = ft.FLIGHT\_TYPE\_ID

and tf.time\_ID = td.time\_ID

and tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.DEST\_AIRPORT\_ID = da.AIRPORTID

and (sa.COUNTRY = da.COUNTRY

or sa.COUNTRY != da.COUNTRY)

group by td.YEARNUMBER,ad.NAME, rollup (ft.FLIGHT\_TYPE,sa.COUNTRY,da.COUNTRY);

------------------------Report 6 (Original Query)----------------------------

select

decode(grouping(td.DAYNAME),1,'Any Day',td.DAYNAME) as "Flight Day",

decode(grouping(ft.FLIGHT\_TYPE),1,'All Flight Type',ft.FLIGHT\_TYPE) as "Flight Type",

decode(grouping(tc.TRAVEL\_CLASS\_TYPE),1,'Any Class',tc.TRAVEL\_CLASS\_TYPE) as "Flight Class",

decode(grouping(sa.COUNTRY),1,'Any Country',sa.COUNTRY) as "Source Country",

decode(grouping(da.COUNTRY),1,'Any Country',da.COUNTRY) as "Destination Country",

sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Number of Transactions",

round(sum(tf.TOTAL\_TOTAL\_PAID)/ sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS),2) as "Average Paid Ticket(USD)"

from

transaction\_fact tf, time\_dim td, flight\_type\_dim ft,

travel\_class\_dim tc, source\_airport\_dim sa, dest\_airport\_dim da

where

tf.time\_id = td.time\_id

and tf.FLIGHT\_TYPE\_ID = ft.FLIGHT\_TYPE\_ID

and tf.TRAVEL\_CLASS\_ID = tc.TRAVEL\_CLASS\_ID

and tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.DEST\_AIRPORT\_ID = da.AIRPORTID

group by td.DAYNAME, cube

(ft.FLIGHT\_TYPE,tc.TRAVEL\_CLASS\_TYPE,sa.COUNTRY,da.COUNTRY);

---Report 6 (Alternate Query)

select /\*+ ORDERED \*/

decode(grouping(td.DAYNAME),1,'Any Day',td.DAYNAME) as "Flight Day",

decode(grouping(ft.FLIGHT\_TYPE),1,'All Flight Type',ft.FLIGHT\_TYPE) as "Flight Type",

decode(grouping(tc.TRAVEL\_CLASS\_TYPE),1,'Any Class',tc.TRAVEL\_CLASS\_TYPE) as "Flight Class",

decode(grouping(sa.COUNTRY),1,'Any Country',sa.COUNTRY) as "Source Country",

decode(grouping(da.COUNTRY),1,'Any Country',da.COUNTRY) as "Destination Country",

sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Number of Transactions",

round(sum(tf.TOTAL\_TOTAL\_PAID)/ sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS),2) as "Average Paid Ticket(USD)"

from

flight\_type\_dim ft,

travel\_class\_dim tc,transaction\_fact tf,time\_dim td,source\_airport\_dim sa, dest\_airport\_dim da

where

tf.time\_id = td.time\_id

and tf.FLIGHT\_TYPE\_ID = ft.FLIGHT\_TYPE\_ID

and tf.TRAVEL\_CLASS\_ID = tc.TRAVEL\_CLASS\_ID

and tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.DEST\_AIRPORT\_ID = da.AIRPORTID

group by td.DAYNAME, cube

(ft.FLIGHT\_TYPE,tc.TRAVEL\_CLASS\_TYPE,sa.COUNTRY,da.COUNTRY);

--Report 7 (Original)

select

decode(grouping(sa.city),1,'Any City',sa.city) as "Departure City",

decode(grouping(sa.country),1,'Any Country',sa.country) as "Departure Country",

decode(grouping(da.city),1,'Any City',da.city) as "Arrival City",

decode(grouping(da.country),1,'Any Country',da.country) as "Arrival Country",

sum(rf.TOTAL\_SERVICE\_COST) as "Total Service Cost"

from route\_fact rf, source\_airport\_dim sa, dest\_airport\_dim da

where rf.sourceairportid = sa.airportid

and rf.destairportid = da.airportid

group by cube (sa.CITY, sa.COUNTRY, da.CITY, da.COUNTRY)

order by sa.COUNTRY,sa.CITY;

--Report 7 (Alternate)

select \* from

(select /\*+ordered no\_merge\*/

decode(grouping(sa.city),1,'Any City',sa.city) as "Departure City",

decode(grouping(sa.country),1,'Any Country',sa.country) as "Departure Country",

decode(grouping(da.city),1,'Any City',da.city) as "Arrival City",

decode(grouping(da.country),1,'Any Country',da.country) as "Arrival Country",

sum(rf.TOTAL\_SERVICE\_COST) as "Total Service Cost"

from source\_airport\_dim sa, dest\_airport\_dim da, route\_fact rf

where rf.sourceairportid = sa.airportid

and rf.destairportid = da.airportid

group by cube (sa.CITY, sa.COUNTRY, da.CITY, da.COUNTRY)) q1

order by q1."Departure Country",q1."Departure City";

--Report 8 (Original)

select decode(grouping(td.YEARNUMBER),1,'Any Year',td.YEARNUMBER) as "Flight Year",

decode(grouping(ad.NAME),1,'Any Airline',ad.NAME) as "Airline Name",

decode(grouping(tc.TRAVEL\_CLASS\_TYPE),1,'All Class Type',tc.TRAVEL\_CLASS\_TYPE) as "Travel Class Type",

decode(grouping(fd.FLIGHT\_DESCRIPTION),1,'All Flight Description',fd.FLIGHT\_DESCRIPTION) as "Flight Description",

decode(grouping(sa.COUNTRY),1,'Any Country',sa.COUNTRY) as "Source Country",

decode(grouping(da.COUNTRY),1,'Any Country',da.COUNTRY) as "Destination Country",

sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Number of Transactions",

sum(tf.TOTAL\_TOTAL\_PAID) - sum(tf.TOTAL\_FARE) as "Total Agent Profit(USD)"

from transaction\_fact tf, airline\_dim ad, travel\_class\_dim tc, FLIGHT\_DISTANCE\_DIM fd,

time\_dim td, source\_airport\_dim sa, dest\_airport\_dim da

where tf.AIRLINEID = ad.AIRLINEID

and tf.TRAVEL\_CLASS\_ID = tc.TRAVEL\_CLASS\_ID

and tf.FLIGHT\_DISTANCE\_ID = fd.FLIGHT\_DITANCE\_ID

and tf.time\_ID = td.time\_ID

and tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.DEST\_AIRPORT\_ID = da.AIRPORTID

and (sa.COUNTRY = da.COUNTRY

or sa.COUNTRY != da.COUNTRY)

group by td.YEARNUMBER,ad.NAME, rollup (tc.TRAVEL\_CLASS\_TYPE,fd.FLIGHT\_DESCRIPTION,sa.COUNTRY,da.COUNTRY);

--Report 8 (Alternate)

select /\*+ordered\*/

decode(grouping(td.YEARNUMBER),1,'Any Year',td.YEARNUMBER) as "Flight Year",

decode(grouping(ad.NAME),1,'Any Airline',ad.NAME) as "Airline Name",

decode(grouping(tc.TRAVEL\_CLASS\_TYPE),1,'All Class Type',tc.TRAVEL\_CLASS\_TYPE) as "Travel Class Type",

decode(grouping(fd.FLIGHT\_DESCRIPTION),1,'All Flight Description',fd.FLIGHT\_DESCRIPTION) as "Flight Description",

decode(grouping(sa.COUNTRY),1,'Any Country',sa.COUNTRY) as "Source Country",

decode(grouping(da.COUNTRY),1,'Any Country',da.COUNTRY) as "Destination Country",

sum(tf.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Number of Transactions",

sum(tf.TOTAL\_TOTAL\_PAID) - sum(tf.TOTAL\_FARE) as "Total Agent Profit(USD)"

from airline\_dim ad, travel\_class\_dim tc, FLIGHT\_DISTANCE\_DIM fd,

time\_dim td, source\_airport\_dim sa, dest\_airport\_dim da,transaction\_fact tf

where tf.AIRLINEID = ad.AIRLINEID

and tf.TRAVEL\_CLASS\_ID = tc.TRAVEL\_CLASS\_ID

and tf.FLIGHT\_DISTANCE\_ID = fd.FLIGHT\_DITANCE\_ID

and tf.time\_ID = td.time\_ID

and tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.DEST\_AIRPORT\_ID = da.AIRPORTID

and (sa.COUNTRY = da.COUNTRY

or sa.COUNTRY != da.COUNTRY)

group by td.YEARNUMBER,ad.NAME, rollup (tc.TRAVEL\_CLASS\_TYPE,fd.FLIGHT\_DESCRIPTION,sa.COUNTRY,da.COUNTRY);

--Report 9 (Original)

select td.MONTH, sa.CITY, sa.country, fd.FLIGHT\_DESCRIPTION,

(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) as "Total profit",

to\_char(sum(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) over

(order by substr(td.time\_id,5,2) rows unbounded preceding),

'9,999,999,999.99') as "Cumulative Monthly Profit"

from

transaction\_fact tf, flight\_distance\_dim fd, time\_dim td,

source\_airport\_dim sa

where tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.FLIGHT\_DISTANCE\_ID = fd.FLIGHT\_DITANCE\_ID

and tf.time\_ID = td.time\_ID

and td.YEARNUMBER = '2007'

and fd.FLIGHT\_DESCRIPTION = 'Small'

and sa.city = 'Sydney'

group by

td.MONTH, sa.CITY, sa.country,fd.FLIGHT\_DESCRIPTION,substr(td.time\_id,5,2);

--Report 9 (Alternate)

select q1.MONTH,q1.CITY,q1.country,q1.FLIGHT\_DESCRIPTION,q1."Total profit",

to\_char(sum(q1."Total profit")over (order by q1."monthID" rows unbounded preceding ),'9,999,999,999.99') as "Cumulative Profit"

from

(

select /\*+no\_merge\*/

substr(td.time\_id,5,2) as "monthID", td.MONTH, sa.CITY, sa.country, fd.FLIGHT\_DESCRIPTION,

(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) as "Total profit"

from

transaction\_fact tf, flight\_distance\_dim fd, time\_dim td,

source\_airport\_dim sa

where tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.FLIGHT\_DISTANCE\_ID = fd.FLIGHT\_DITANCE\_ID

and tf.time\_ID = td.time\_ID

and td.YEARNUMBER = '2007'

and fd.FLIGHT\_DESCRIPTION = 'Small'

and sa.city = 'Sydney'

group by

td.MONTH, sa.CITY, sa.country,fd.FLIGHT\_DESCRIPTION,substr(td.time\_id,5,2)

) q1;

--Report 10 (Original)

select sub.MONTH as "Month",

sum(sub.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Total Transactions",

to\_char(avg(sum(sub.TOTAL\_NUMBER\_\_TRANSACTIONS)) over

(order by sub.RepPeriod rows 2 preceding),

'9,999,999,999.99') as moving\_avg\_transactions

from (Select substr(td.time\_id,5,2) || td.YEARNUMBER as RepPeriod, td.MONTH, tf.TOTAL\_NUMBER\_\_TRANSACTIONS from transaction\_fact tf, time\_dim td, nationality\_dim nd

where

tf.time\_id = td.time\_id

and tf.natid = nd.natid

and nd.NATIONALITY = 'Australian'

and td.yearnumber = '2009') sub

group by

sub.MONTH,sub.RepPeriod ;

--Report 10 (Alternate)

select sub.MONTH as "Month",

sum(sub.TOTAL\_NUMBER\_\_TRANSACTIONS) as "Total Transactions",

to\_char(avg(sum(sub.TOTAL\_NUMBER\_\_TRANSACTIONS)) over

(order by sub.RepPeriod rows 2 preceding),

'9,999,999,999.99') as moving\_avg\_transactions

from (Select /\*+No\_merge ordered \*/ substr(td.time\_id,5,2) || td.YEARNUMBER as RepPeriod, td.MONTH, tf.TOTAL\_NUMBER\_\_TRANSACTIONS from transaction\_fact tf, time\_dim td, nationality\_dim nd

where

tf.time\_id = td.time\_id

and tf.natid = nd.natid

and nd.NATIONALITY = 'Australian'

and td.yearnumber = '2009') sub

group by

sub.MONTH,sub.RepPeriod ;

--Report 11 (Original)

select substr(td.time\_id,5,2) || substr(td.time\_id,3,2) as RepPeriod, td.month, td.YEARNUMBER,

(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) as "Total profit",

to\_char(sum(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) over

( partition by td.YEARNUMBER order by td.YEARNUMBER desc rows unbounded preceding),

'9,999,999,999.99') as "Cumulative Monthly Profit"

from

transaction\_fact tf, flight\_distance\_dim fd,

source\_airport\_dim sa, TIME\_DIM td

where tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.FLIGHT\_DISTANCE\_ID = fd.FLIGHT\_DITANCE\_ID

and td.time\_id = tf.time\_id

group by

substr(td.time\_id,5,2) || substr(td.time\_id,3,2),td.month,td.YEARNUMBER

order by td.YEARNUMBER, substr(td.time\_id,5,2) || substr(td.time\_id,3,2);

--Report 11 (Alternate)

select /\*+ Ordered \*/ substr(td.time\_id,5,2) || substr(td.time\_id,3,2) as RepPeriod, td.month, td.YEARNUMBER,

(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) as "Total profit",

to\_char(sum(sum(tf.total\_total\_paid) - sum(tf.total\_fare)) over

( partition by td.YEARNUMBER order by td.YEARNUMBER desc rows unbounded preceding),

'9,999,999,999.99') as "Cumulative Monthly Profit"

from

transaction\_fact tf, flight\_distance\_dim fd,

source\_airport\_dim sa, TIME\_DIM td

where tf.SOURCE\_AIRPORT\_ID = sa.AIRPORTID

and tf.FLIGHT\_DISTANCE\_ID = fd.FLIGHT\_DITANCE\_ID

and td.time\_id = tf.time\_id

group by

substr(td.time\_id,5,2) || substr(td.time\_id,3,2),td.month,td.YEARNUMBER

order by td.YEARNUMBER, substr(td.time\_id,5,2) || substr(td.time\_id,3,2);

--Report 12 (Original Execution plan)

select sub.aln as airlines\_name, sub.sn as source\_name, sub.month as month\_name,sub.YEARNUMBER,

(sum(sub.total\_total\_paid)-sum(sub.TOTAL\_FARE)) as "Total Profit",

to\_char(avg(sum(sub.total\_total\_paid)-sum(sub.TOTAL\_FARE)) over

(partition by sub.aln, sub.sn order by sub.aln, sub.sn, sub.YEARNUMBER rows 2 preceding),

'9,999,999,999.99') as moving\_avg\_profit

from (Select td.YEARNUMBER || td.month as RepDate, ad.NAME as aln, sa.NAME sn, td.month, tf.total\_total\_paid, tf.TOTAL\_FARE, ad.airlineid, td.yearnumber

from transaction\_fact tf, time\_dim td, airline\_dim ad, SOURCE\_AIRPORT\_DIM sa

where

tf.time\_ID = td.time\_ID

and tf.airlineid = ad.airlineid

and sa.airportid = tf.SOURCE\_AIRPORT\_ID) sub

group by

sub.Month,sub.YEARNUMBER,sub.airlineid,sub.aln, sub.sn ;

--Report 12 (Alternate)

select sub.aln as airlines\_name, sub.sn as source\_name, sub.month as month\_name,sub.YEARNUMBER,

sub.total\_profit as "Total Profit",

to\_char(avg(sub.total\_profit) over

(partition by sub.aln, sub.sn order by sub.aln, sub.sn, sub.YEARNUMBER rows 2 preceding),

'9,999,999,999.99') as moving\_avg\_profit

from (Select /\*+ No\_Merge use\_nl(sa tf) \*/ td.YEARNUMBER || td.month as RepDate, ad.NAME as aln, sa.NAME sn, td.month,ad.airlineid, td.yearnumber, sum(tf.total\_total\_paid) - sum(tf.TOTAL\_FARE) as total\_profit

from time\_dim td, airline\_dim ad, SOURCE\_AIRPORT\_DIM sa, transaction\_fact tf

where

tf.time\_ID = td.time\_ID

and tf.airlineid = ad.airlineid

and sa.airportid = tf.SOURCE\_AIRPORT\_ID

Group By

td.YEARNUMBER || td.month, ad.NAME, sa.NAME, td.month,ad.airlineid, td.yearnumber) sub;

--Report 13 (Original)

Select sa.country, sa.city as CityName, sa.name as SourceAirport,

To\_char(sum(rf.total\_service\_cost)) as Service$,

Rank() Over (Partition By sa.Country

ORDER BY sum(rf.total\_service\_cost) desc) as RankScore

from source\_airport\_dim sa, ROUTE\_FACT rf

where rf.SOURCEAIRPORTID = sa.airportid

group by

sa.country, sa.city, sa.name;

--Report 13 (Alternate)

Select /\*+ use\_merge (sa rf) \*/ sa.country, sa.city as CityName, sa.name as SourceAirport,

To\_char(sum(rf.total\_service\_cost)) as Service$,

Rank() Over (Partition By sa.Country

ORDER BY sum(rf.total\_service\_cost) desc) as RankScore

from source\_airport\_dim sa, ROUTE\_FACT rf

where rf.SOURCEAIRPORTID = sa.airportid

group by

sa.country, sa.city, sa.name;

--Report 14 (Original)

Select \*

from (Select

td.YEARNUMBER, nd.NATIONALITY, pt.PASSENGER\_TYPE,

To\_char(sum(tf.TOTAL\_TOTAL\_PAID)-sum(tf.total\_fare)) as Revenue$,

Percent\_Rank() Over (Partition By nd.NATIONALITY

ORDER BY sum(tf.TOTAL\_TOTAL\_PAID)-sum(tf.total\_fare) desc) as PercentRankScoreNationality,

Percent\_Rank() Over (Partition By PASSENGER\_TYPE

ORDER BY sum(tf.TOTAL\_TOTAL\_PAID)-sum(tf.total\_fare) desc) as PercentankScorePassenger

from PASSENGER\_TYPE\_DIM pt, nationality\_Dim nd, transaction\_fact tf, TIME\_DIM td

where tf.NATID = nd.NATID

And tf.PASSENGER\_TYPE\_ID = pt.PASSENGER\_TYPE\_ID

and td.TIME\_ID = tf.TIME\_ID

group by

td.YEARNUMBER, nd.NATIONALITY, pt.PASSENGER\_TYPE

order by td.yearnumber

) q1

where PercentRankScoreNationality < 0.1

Or PercentankScorePassenger < 0.1;

--Report 14 (Alternate)

Select \*

from (Select

td.YEARNUMBER, nd.NATIONALITY, pt.PASSENGER\_TYPE,

To\_char(sum(tf.TOTAL\_TOTAL\_PAID)-sum(tf.total\_fare)) as Revenue$,

Percent\_Rank() Over (Partition By nd.NATIONALITY

ORDER BY sum(tf.TOTAL\_TOTAL\_PAID)-sum(tf.total\_fare) desc) as PercentRankScoreNationality,

Percent\_Rank() Over (Partition By PASSENGER\_TYPE

ORDER BY sum(tf.TOTAL\_TOTAL\_PAID)-sum(tf.total\_fare) desc) as PercentankScorePassenger

from PASSENGER\_TYPE\_DIM pt, nationality\_Dim nd, transaction\_fact tf, TIME\_DIM td

where tf.NATID = nd.NATID

And tf.PASSENGER\_TYPE\_ID = pt.PASSENGER\_TYPE\_ID

and td.TIME\_ID = tf.TIME\_ID

group by

td.YEARNUMBER, nd.NATIONALITY, pt.PASSENGER\_TYPE

order by td.yearnumber

) q1

where PercentRankScoreNationality < 0.1

Or PercentankScorePassenger < 0.1;