

Homework 1

Create database from unclean datasets.

Write queries to print certain data from the database.

Writing functions for GCD and LCM in PL/SQL.

Using Cursors to print data.



Aashay thakkar

31447706

CS 632

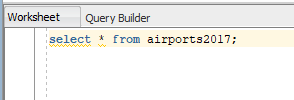
**Airports2017**

Q 1. a) The two rows that have a problem are:

Row number 46 - China Xi'an Xianyang International Airport

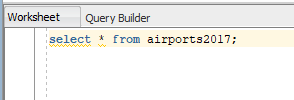
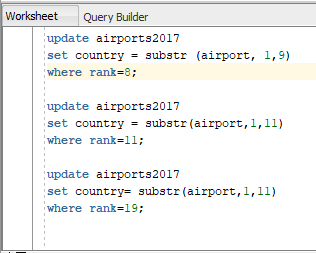
Row number 50 - Russia Sheremetyevo International Airport

* The rows have a problem because of the ‘?’ symbol in the rank change (G) row. This is caused because these airports are not present in the 2016 list and hence cannot compute rank change.
* Since the rank in the previous year should be greater than 50, according to the percent change and the total passengers, the minimum possible rank in the previous year should be 51 and 52.The rank change in the 46th row should be 5. The rank change in the 50th row should be 2.





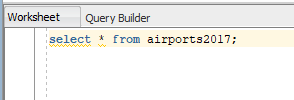
b)

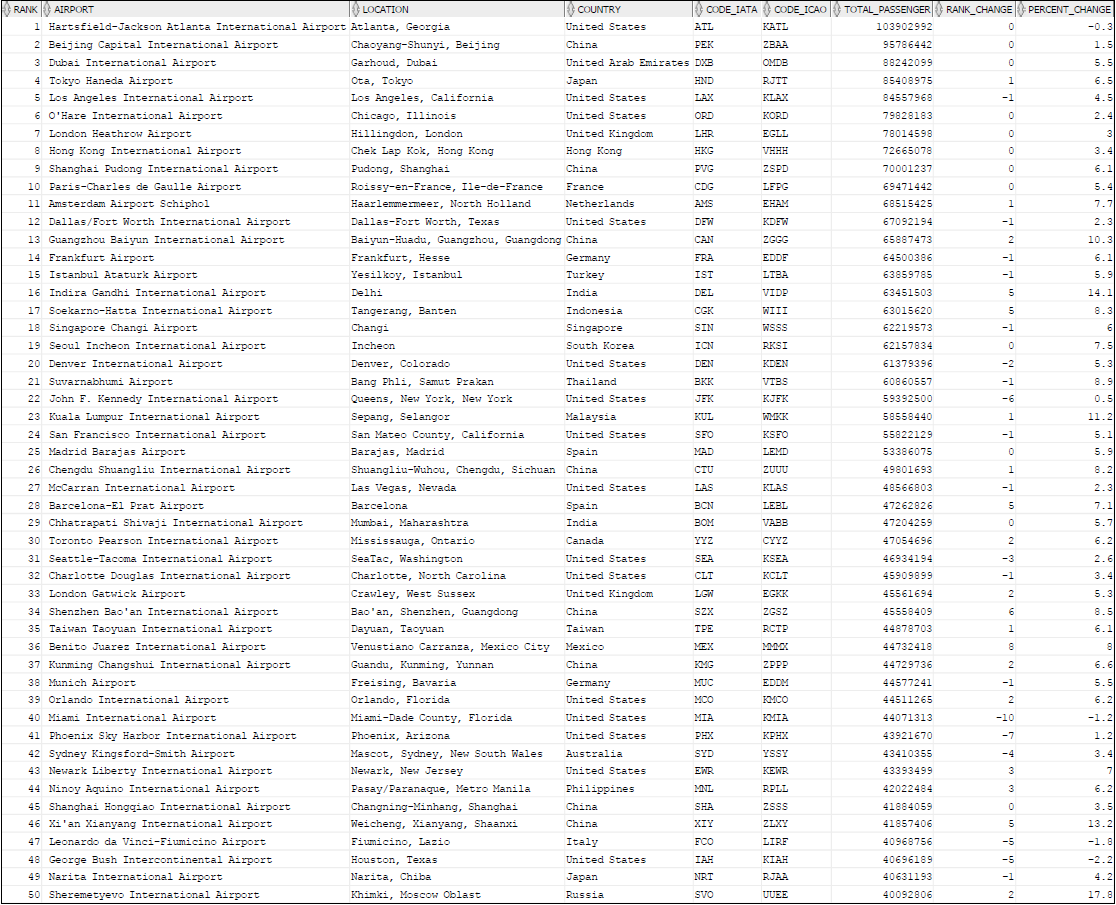


c) After running the update statement:

update AIRPORTS2017

set airport = substr(Airport, length(country)+1);





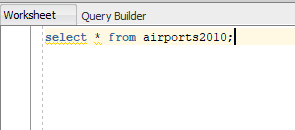
**Airports2010**

Q1. a) The two rows that have a problem are:

Row number 47 – Brazil São Paulo-Guarulhos International Airport

Row number 49 - China Shenzhen Bao'an International Airport

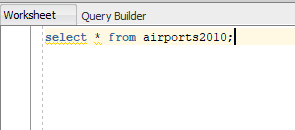
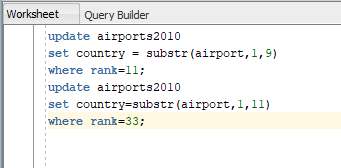
* The rows have a problem because of the ‘?’ symbol in the rank change (G) row. This is caused because these airports are not present in the 2009 list and hence cannot compute rank change.
* Since the rank in the previous year should be greater than 50, according to the total passengers, the minimum possible rank in the previous year should be 51 and 52. The rank change in the 47th row should be 4. The rank change in the 49th row should be 3.





b)

Since the countries in the airport column and the country column are different for rows 11 and 33, we will use the update command to change the country column for the same.

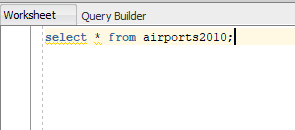




c) After the update statement:

update AIRPORTS2010

set airport = substr(Airport, length(country)+1);





2) Write SQL queries for the following:

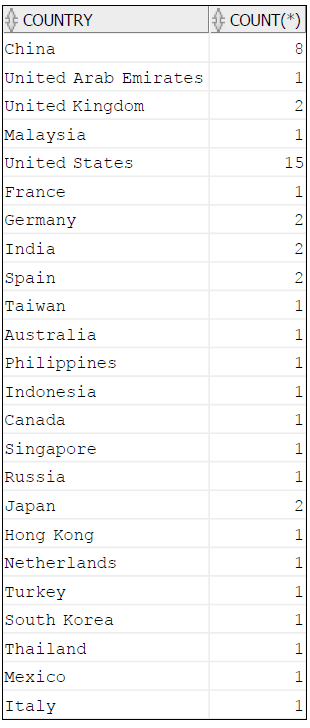
1. Show AIRPORTS2017 in alphabetical order by Country
   * **select \* from airports2017**

**order by country;**

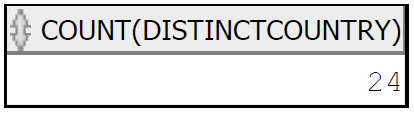


1. Using Group By, show each country once and how often it appears in the list.
   * **select country, count(\*) from airports2017**

**group by country;**



1. Show the total number of countries in the table. With a single SELECT statement.
   * **select count(distinct country) from airports2017;**



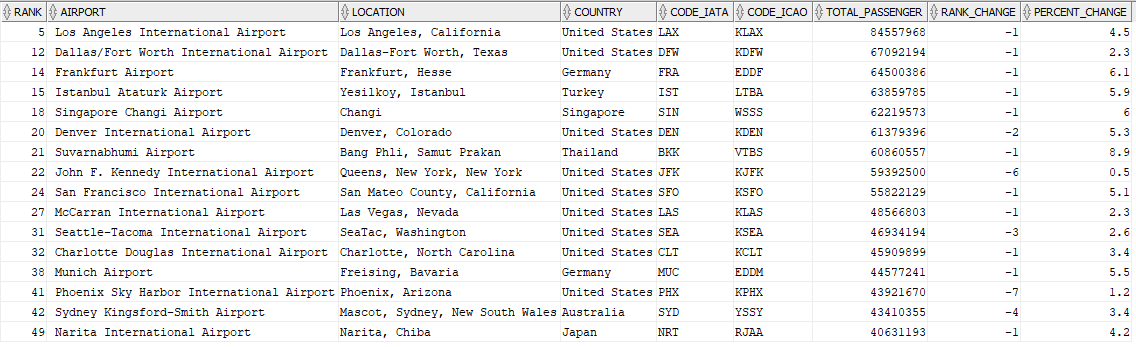
1. Show all information about all airports that had more than 7 million flights in 2017.
   * **select \* from airports2017**

**where total\_passenger > 7000000;**



1. Show all information about all the airports that increased their passengers, yet decreased their ranking.
   * **select \* from airports2017**

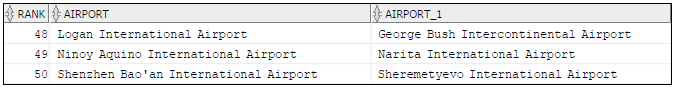
**where percent\_change > 0 and rank\_change < 0;**



1. Now we compare 2010 and 2017. Show Rank number, Airport in 2010 and Airport in 2017 whenever the two airports for that Rank are DIFFERENT. If the airports are the same, don’t show that row. This requires a JOIN. Use simple JOIN syntax, without the keyword “JOIN.”
   * **select airports2010.rank, airports2010.airport, airports2017.airport from airports2010, airports2017**

**where airports2010.airport != airports2017.airport**

**and airports2010.rank = airports2017.rank;**



1. Compute for each rank the percentage increase (decrease) from 2010 to 2017.

Hint:

[(Total\_Passengers(2017)–Total\_Passengers(2010))/Total\_Passengers(2010)]\*100

WITHIN SQL Round the result to two digits after the decimal point.

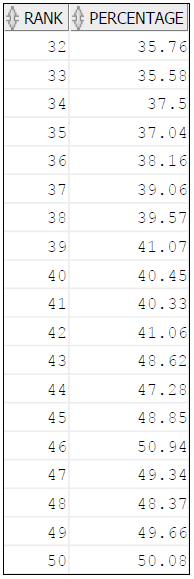
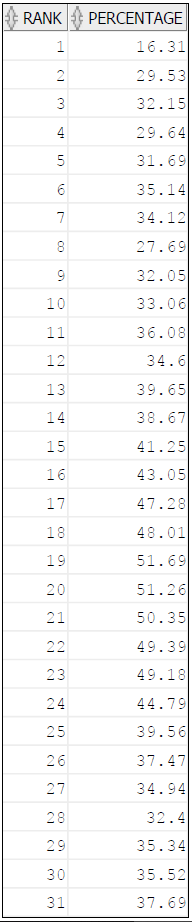
Show the result in two columns: Rank and Percentage.

* + **select airports2017.rank, round(((airports2017.total\_passenger - airports2010.total\_passenger)/(airports2010.total\_passenger))\*100,2)**

**as percentage**

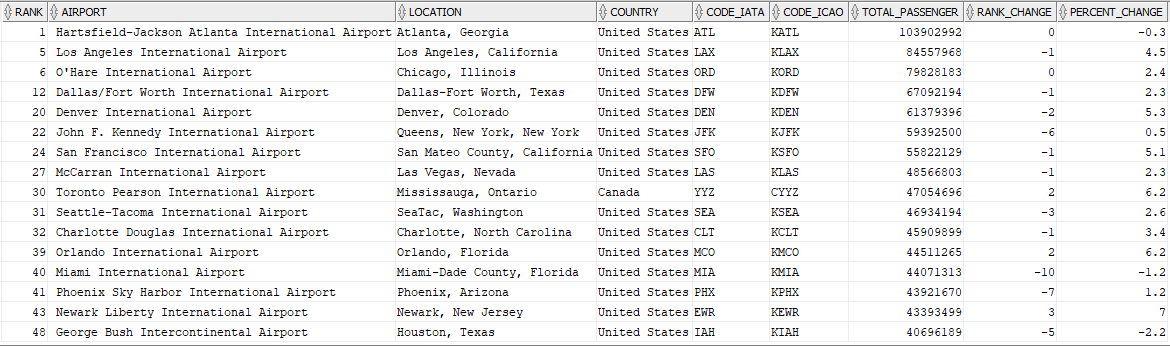
**from airports2010, airports2017**

**where airports2017.rank = airports2010.rank;**



1. List all the airports where the IATA code is the same as the last 3 letters of the ICAO code. List complete information for those airports.
   * **select \* from airports2017**

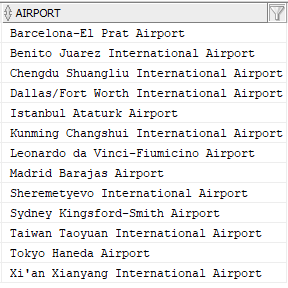
**where code\_iata = substr (code\_icao,-3) ;**



1. Show all airports that appear in any position in 2017 but did not appear in 2010 at all. Hint: Use set operations in Oracle. In your select statements. Show only the Airport name.
   * **select airports2017.airport from airports2017**

**minus**

**select airports2010.airport from airports2010;**



1. Observe the following. In many of the rows (by far not all) in the column Location, after the first comma comes the name of a city or state.

Using instr, find the position of that first comma.

Using substr, find what comes after the blank after that comma.

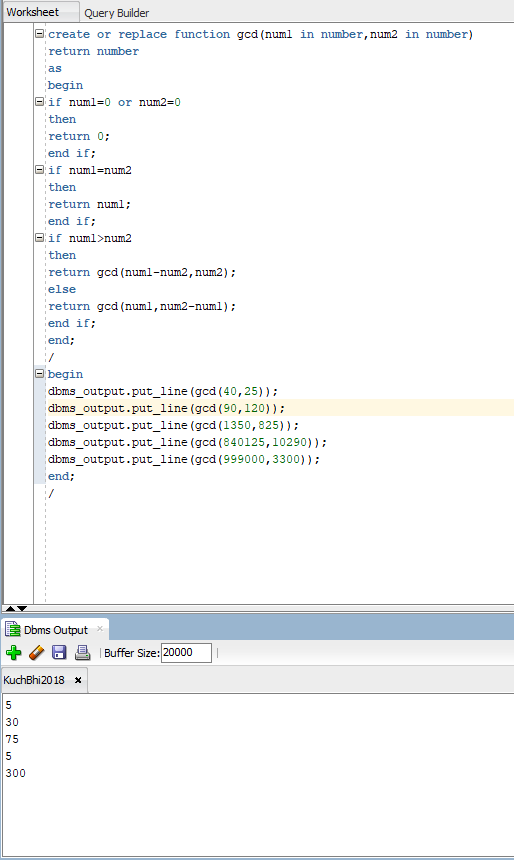
Then write ONE SINGLE select statement that shows Rank and “the text after the first comma in the Location column.”

* + **select rank, substr(location,instr(location,',',1,1)+1) as "the text after the first comma in the Location column" from airports2017;**



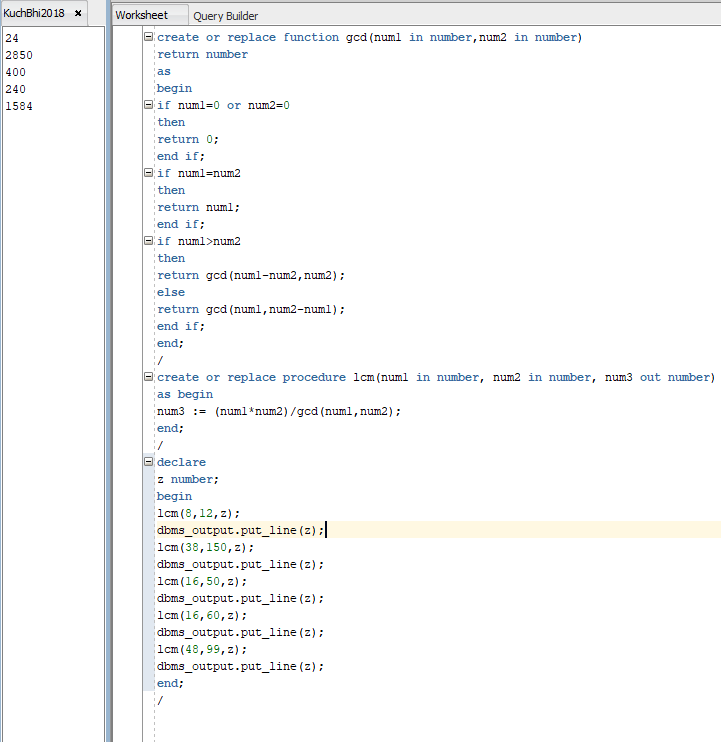
**GCD**

3) Write a main program that calls your GCD functions five times and sends the results to the screen.



**LCM**

4) Write a main program that calls LCM five times and sends the results to the screen.



**Cursor**

5) Implement a program like P18, but make the following changes:

Use your table AIRPORT2010 instead of the table X.

Use a name for the cursor variable that “makes sense.” NOT Joseph.

Do not print to the screen all the rows. Only print to the screen the rows where the total

passengers are over 50 Million.

