


Quality Control Form

SELECT

Analytics

2024

Specification of Data

<u>Analyst</u>	<div>SELECT</div> <div>Analytics</div>
Team Members	Katherina Akintelure, Kemi Alake, Tai Chen, Harry Murphy
<u>Client</u>	<div><div>OFFFUTURE</div><div>THE FUTURE OF OFFICE SUPPLIES</div></div>
Date of file received	24/4/2024
Name of file received	"1 - Offfuture 2011-2014"
File Format received	CSV
Size of file (KB) received	12090
Encoding of the file	UTF-8
Recorded Number of Column	24
Recorded Number of Rows	51290
Name of Database	Pagilla
Name of Schema	student
Name of Destination Table	team_select



Page 1 - Specification of Data

Page 2 - Contents

Page 3 - Summary of Checks

Page 4 - Compulsory Checks

Page 6 - Additional Checks

Page 7 - Notes Section

Page 8 - Suggestions for Data Collection & Formatting

Page 9 - Appendix

Page 10 - Expansion of large checks

Page 15 - Description of Checks

Summary of Compulsory Checks

Checks	PASS/FAIL	Notes
Row Count	PASS	Source data equal to imported.
Column Count	PASS	Source data equal to imported.
Sum of Row Sums	PASS	Source data equal to imported.
Sum of Column Sums	PASS	Source data equal to imported.
Nulls Check	PASS	Source data equal to imported.
String Length Check	PASS	Not all source data equal to imported. *see notes (3,8)
Date Boundaries	PASS	Source data equal to imported.
SUM, MIN, and MAX - Numeric	PASS	Source data equal to imported.
Unique Rows Check	PASS	Source data equal to imported.
Eyeball Check: Random Entry Check	PASS	Source data equal to imported.
Eyeball Check: Date Formatting	PASS	Rearranged to YY-MM-DD on import.


All compulsory checks passed. Continuity between the source data and the data loaded on the Pagilla database was tested and met all checks.

*See compulsory checks & appendix for further context**
*See description of checks for explanation**

SQL ID	Description of Check	Result in Source	Result in Destination	PASS/FAIL	Notes
SQL CODE 2	Number of Nulls	41296	41296	PASS	Checking for total Null values
SQL CODE 3	Column Count	24	24	PASS	Checking for total number of columns
SQL CODE 4	Row Count	51290	51290	PASS	Checking for total number of rows
SQL CODE 5	Sum of Column sums	1331006116.61916	1331006116.61916	PASS	Checking for the sum of column sums
SQL CODE 6	Sum of Row sums	1331006116.61916	1331006116.61916	PASS	Checking for the sum of row sums
SQL CODE 7 7.1 - 7.48 (Except 7.3, 7.4, 7.12) <i>*see appendix</i>	String lengths - MIN/MAX	SAME	SAME	PASS	Checking for the MIN/MAX of the string length for each column
SQL CODE 7 (7.3, 7.4, 7.12) <i>*see appendix</i>	String lengths - MIN/MAX	7.3: 5/5 7.4: 5/5 7.12: 0/5	7.3: 10/10 7.4: 10/10 7.12: 4/5	FAIL	Checking for the MIN/MAX of the string length for dates and postal code <i>*See notes (3,8)</i>

SQL ID	Description of Check	Result in Source	Result in Destination	PASS/FAIL	Notes
SQL CODE 8.1	Min/Max Date - order_date	01/01/2011 31/12/2014	2011-01-01 2014-12-31	PASS	Checking oldest/most recent order date
SQL CODE 8.2	Min/Max Date - ship_date	03/01/2011 07/01/2015	2011-01-03 2015-01-07	PASS	Checking oldest/most recent ship date
SQL CODE 9 9.1-9.6 *see appendix	SUM/MIN/MAX of numeric tables	SAME	SAME	PASS	Checking for SUM/MIN/MAX for row_id, sales, quantity, discount and profit
SQL CODE 10	# of unique row IDs	51,290	51,290	PASS	Checking for the number of unique ids
SQL CODE 11	Column entry check Market: 11.1 Region: 11.2 Category: 11.3	SAME	SAME	PASS	Checking that the entries for market, region, category are the same

Additional Checks

SQL ID	Description of Check	Result in Source	Result in Destination	PASS/FAIL	Notes
SQL CODE 12	Eyeball entry check -(23732, 7122, 11283, 1235, 50199)	(23732, 7122, 11283, 1235, 50199)	(23732, 7122, 11283, 1235, 50199)	PASS	Checking that all entries are the same from 5 random ids
SQL CODE 13	Check for inconsistency with 'Amman'	JUST "Amman"	JUST "Amman"	PASS	Checking that "Amman" is not in the state column <i>*See notes (6)</i>
SQL CODE 14	Secondary duplicate check	None Found	None Found	PASS	No suspicious entries detected, same results between source and destination data <i>*See notes (9)</i>
	Data type check	N/A	As expected	PASS	Data types were as expected. Results from this check can be referenced in future to ensure data type continuity. <i>*see data types summary</i>



Column names were changed from source to destination. For example, “Product Name” in the source data was changed to “product_name” in the destination. This was done to limit capital letters and spaces. This was a decision made to increase ease of use in SQL and has no effect on the content of the data or any insights to be drawn from it.



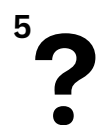
Dates are formatted DD/MM/YYYY in the source data and YYYY-MM-DD in the imported table. When eyeball checks were conducted, it could be concluded that all dates were equivalent, although they were in different date formats.



Dates are formatted DD/MM/YYYY in the source data and YYYY-MM-DD in the imported table. However, testing for the most recent and oldest dates in both the “Order Date” and “Ship Date” give the same results. The only resulting impact is length of date as a string. No impact on the utility of the data but take into consideration when using for analysis.



In the destination table, the decision was made to change ‘postal_code’ into a VARCHAR data type. This was because, although they do not appear in the data, postal codes can contain letters as well as numbers. This is especially true internationally, for example in the United Kingdom. In the source, however, the only postcodes present only contained numbers.



When investigating NULL values, it was found that there were 41296 NULL values. All NULL values found were from postal codes, in both source and destination. This seems to be due to a data collection issue, in which many customers did not provide a postal code. The only postal codes were provided from customers in the United States.



Amman appears as “’Amman” in the column, ‘state’ and as “Amman” in the column ‘city’. Checks were conducted to ensure that this was consistent throughout the table. The inclusion of an apostrophe at the start of this string appears intentional. However, this may cause issues within SQL.



The MAXIMUM and MINIMUM values of every column were calculated using string length. All calculations were done using the ‘LEN()’ function in PostgreSQL and applied to data represented as VARCHAR. In the case where the data in a column was not represented as VARCHAR, this data was cast as a VARCHAR and then calculations were made. Manipulated data types included dates and any numeric data types.



The string length count for “Postal Code” differs between the source data and SQL imported data. This is only due to Excel reading blank cells as the minimum length (with length = 0) and SQL ignoring nulls when taking minimum length. All post codes are the same in both tables.



Data was checked for duplicates outside of row_ID. For example an order for the exact same price, on the same day, by the same person. No duplicates of this kind were identified.

'Amman

There is a lack of consistency between the naming scheme for the state “ 'Amman ” and the city “ Amman ”. This lack of consistency may hinder analyses. Reformation to “ Amman ” for this state is recommended.

**See notes (6) for details.*

Date Format

Intentionally reformatting dates to be 'SQL friendly' would enable faster loading and less data manipulation, reducing risk of data being misrepresented or accidentally altered (*WHAT WOULD BE AN SQL FRIENDLY FORMAT?*).

**See notes (3) for details.*

Postal codes

At present postal codes can be represented by both numeric and text datatypes. This could lead to confusion surrounding the appropriate datatype choice during data loading into the database. This could be improved by collecting postal codes from outside the United States or by specifying that this data category should be represented as text.

Numerical rounding

Consistency of precision for values representing currency could be improved due to many results stating prices with more than 2 decimal places. This could be due to the exact exchanged value of a currency but could lead to inaccurate results. For example it is not possible to have 0.444 of a pound.

Price currency

The currency used for 'Sales', 'Profit', or 'Shipping Cost' is not mentioned. This would be a nice addition to help further understand the meaning of the data provided.

Appendix

SQL ID	Description of Check	Result in Source	Result in Destination
SQL CODE 7.1 and 7.25	minimum length & maximum length in column row_id	min: 1 max: 5	min: 1 max: 5
SQL CODE 7.2 and 7.26	minimum length & maximum length in column order_id	min: 9 max: 15	min: 9 max: 15
SQL CODE 7.3 and 7.27	minimum length & maximum length in column order_date	min: 5 max: 5	min: 10 max: 10
SQL CODE 7.4 and 7.28	minimum & maximum lengths in column ship_date	min: 5 max: 5	min: 10 max: 10
SQL CODE 7.5 and 7.29	minimum & maximum lengths in column ship_mode	min: 8 max: 14	min: 8 max: 14
SQL CODE 7.6 and 7.30	minimum & maximum lengths in column customer_id	min: 5 max: 8	min: 5 max: 8
SQL CODE 7.7 and 7.31	minimum & maximum lengths in column customer_name	min: 7 max: 22	min: 7 max: 22
SQL CODE 7.8 and 7.32	minimum & maximum lengths in column segment	min: 8 max: 11	min: 8 max: 11

***see notes (7)**

SQL ID	Description of Check	Result in Source	Result in Destination
SQL CODE 7.9 and 7.33	minimum & maximum lengths in column city	min: 2 max: 35	min: 2 max: 35
SQL CODE 7.10 and 7.34	minimum & maximum lengths in column state	min: 3 max: 36	min: 3 max: 36
SQL CODE 7.11 and 7.35	minimum & maximum lengths in column country	min: 4 max: 32	min: 4 max: 32
SQL CODE 7.12 and 7.36	minimum & maximum lengths in column postal_code	min: 0 max: 5	min: 4 max: 5
SQL CODE 7.13 and 7.37	minimum & maximum values in column market	min: 2 max: 6	min: 2 max: 6
SQL CODE 7.14 and 7.38	minimum & maximum values in column region	min: 4 max: 14	min: 4 max: 14
SQL CODE 7.15 and 7.39	minimum & maximum values in column product_id	min: 15 max: 16	min: 15 max: 16
SQL CODE 7.16 and 7.40	minimum & maximum values in column category	min: 9 max: 15	min: 9 max: 15

***see notes (7)**

SQL ID	Description of Check	Result in Source	Result in Destination
SQL CODE 7.17 and 7.41	minimum & maximum values in column sub_category	min: 3 max: 11	min: 3 max: 11
SQL CODE 7.18 and 7.42	minimum & maximum values in column product_name	min: 5 max: 127	min: 5 max: 127
SQL CODE 7.19 and 7.43	minimum & maximum values in sales	min: 1 max: 10	min: 1 max: 10
SQL CODE 7.20 and 7.44	minimum & maximum values in column quantity	min: 2 max: 1	min: 1 max: 2
SQL CODE 7.21 and 7.45	minimum & maximum values in column discount	min: 1 max: 5	min: 1 max: 5
SQL CODE 7.22 and 7.46	minimum & maximum values in column profit	min: 1 max: 10	min: 1 max: 10
SQL CODE 7.23 and 7.47	minimum & maximum values in column shipping_cost	min: 1 max: 6	min: 1 max: 6
SQL CODE 7.24 and 7.48	minimum & maximum values in column order_priority	min: 3 max: 8	min: 3 max: 8

***see notes (7)**

Sum, Min & Max calculation of each numerical column in the table

SQL ID	Description of Check	Result in Source	Result in Destination
SQL CODE 9.1	SUM/MIN/MAX of row_id	sum: 1315357695 min: 1 max: 51290	sum: 1315357695 min: 1 max: 51290
SQL CODE 9.2	SUM/MIN/MAX of sales	sum: 12642501.90988 min: 0.444 max: 22638.48	sum: 12642501.90988 min: 0.444 max: 22638.48
SQL CODE 9.3	SUM/MIN/MAX of quantity	sum: 178312 min: 1 max: 14	sum: 178312 min: 1 max: 14
SQL CODE 9.4	SUM/MIN/MAX of discount	sum: 7329.728 min: 0 max: 0.85	sum: 7329.728 min: 0 max: 0.85
SQL CODE 9.5	SUM/MIN/MAX of profit	sum: 1467457.29128 min: -6599.978 max: 8399.976	sum: 1467457.29128 min: -6599.978 max: 8399.976
SQL CODE 9.6	SUM/MIN/MAX of shipping_cost	sum: 1352820.69 min: 0 max: 933.57	sum: 1352820.69 min: 0 max: 933.57

Unique entries of each category/column

SQL ID	Description of Check	Result in Source	Result in Destination
SQL CODE 11	Column entry check - Market	Africa APAC Canada EMEA EU LATAM US	Africa APAC Canada EMEA EU LATAM US
SQL CODE 11	Column entry check - Region	Africa Canada Caribbean Central Central Asia East EMEA North North Asia Oceania South Southeast Asia West	Africa Canada Caribbean Central Central Asia East EMEA North North Asia Oceania South Southeast Asia West
SQL CODE 11	Column entry check - Category	Furniture Office Supplies Technology	Furniture Office Supplies Technology

Row count

Check there is the same amount of rows between *source data* and *destination data*.

Column count

Check there is the same amount of columns between source data and *destination data*.

Sum of row sums

Calculate the sum of each row's numeric data, then calculate the sum of all these values together. Check this matches between the *source data* and *destination data*.

Sum of column sums

Calculate the sum of each column's numeric data, then sum of all these values together. Check this matches between the two data sets and that for each dataset this matches the value of the 'sum of row sums'.

Nulls check

Check there is the same count of NULL values in the *source data* and the *destination data*.

String length check

Check the maximum and minimum string length by column match between the *source data* and the *destination data*.

Date boundaries

Check the date boundaries are as expected based on the client's description of the data: between 2011 and 2015.

Check that the date boundaries match between the *source data* and the *destination data*.

SUM, MIN and MAX - Numeric

For columns containing numerical data types *only*, check the total, minimum, and maximum values for each column. Check that these values match between the *source data* and the *destination data*.

Unique Rows Check

Check the count of unique rows (observations that do not match any others in the dataset) matches between the *source data* and the *destination data*.

Eyeball Check: Random Entry Check

Use the SQL ROUND() function to select 5 rows from the *destination data* at random. Compare these to the equivalent rows (matched by row_ID) in the *source data*.

Eyeball Check: Date Formatting

Compare the date formats between the *source data* and the *destination data*. Ensure the dates have been populated as expected in the *destination data*. Look for potential mis-interpretations, e.g. MONTH being interpreted as DAY.