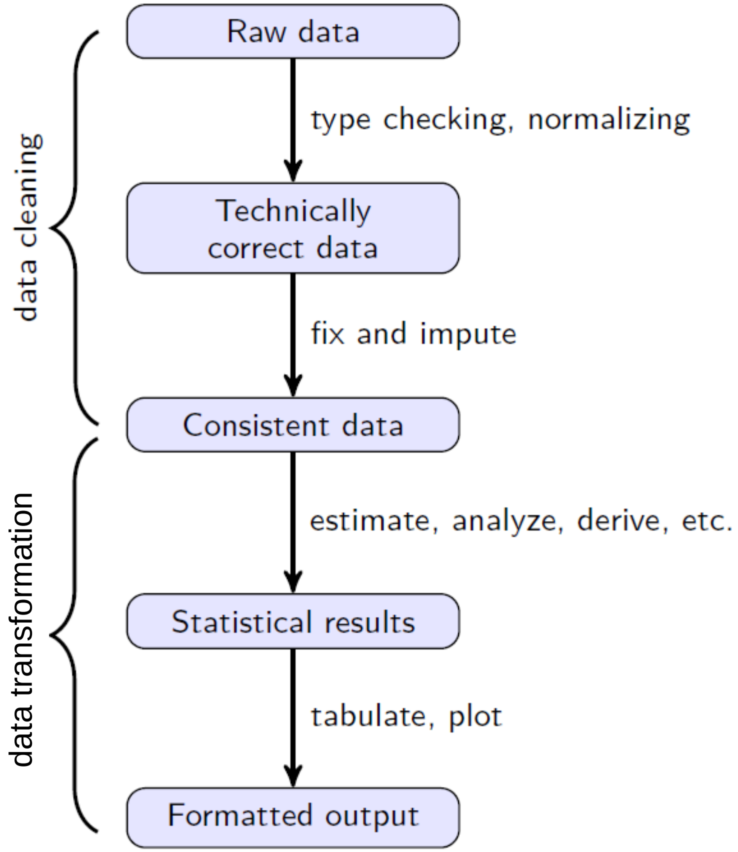
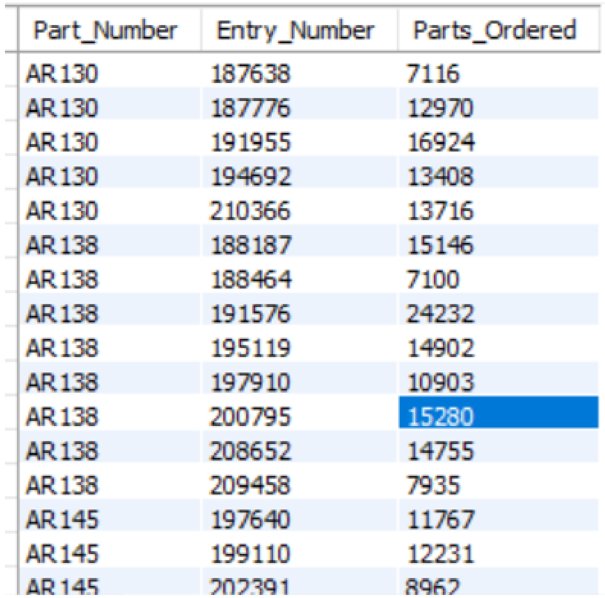
DW Fact Table Creation

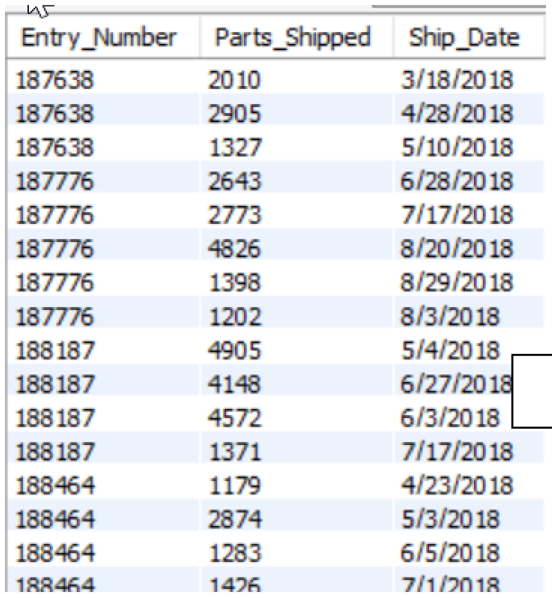
- Design of fact table
  - Granularity of fact table
  - Long vs. Wide
- Joining tables to create fact table – common issues
  - Many to many relationships
  - Outer joins
- Data Cleaning
  - Fix/delete bad records
  - Deduplication
  - Normalize units / Standardize fields for linking tables
  - Handle missing values, class imbalance, outliers
- Data Transformation/Aggregation
  - Summary fields
  - Calculated/Derived fields
  - Categories/Indicators



Data Warehouse Example 4 – Ordered vs Shipped

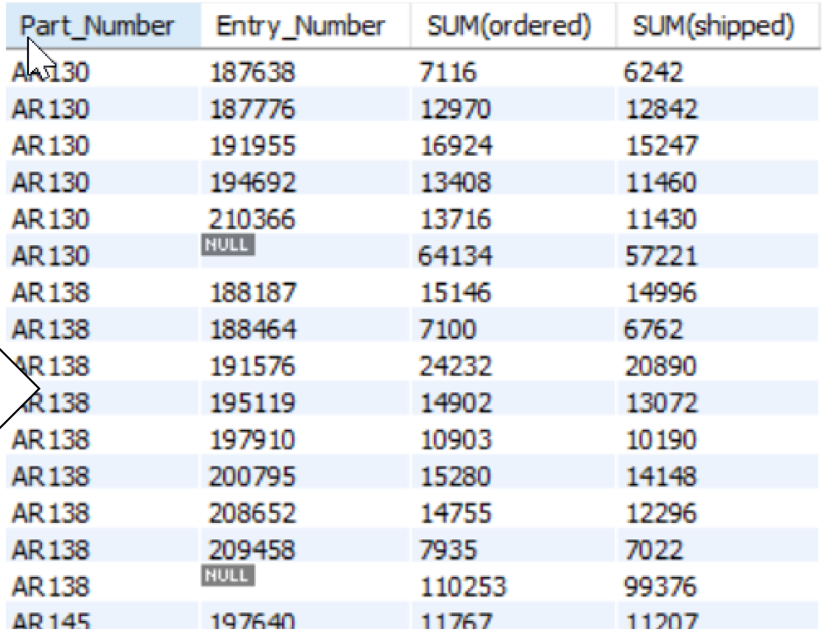


Part_Number	Entry_Number	Parts_Ordered
AR130	187638	7116
AR130	187776	12970
AR130	191955	16924
AR130	194692	13408
AR130	210366	13716
AR138	188187	15146
AR138	188464	7100
AR138	191576	24232
AR138	195119	14902
AR138	197910	10903
AR138	200795	15280
AR138	208652	14755
AR138	209458	7935
AR145	197640	11767
AR145	199110	12231
AR145	202391	8962



Entry_Number	Parts_Shipped	Ship_Date
187638	2010	3/18/2018
187638	2905	4/28/2018
187638	1327	5/10/2018
187776	2643	6/28/2018
187776	2773	7/17/2018
187776	4826	8/20/2018
187776	1398	8/29/2018
187776	1202	8/3/2018
188187	4905	5/4/2018
188187	4148	6/27/2018
188187	4572	6/3/2018
188187	1371	7/17/2018
188464	1179	4/23/2018
188464	2874	5/3/2018
188464	1283	6/5/2018
188464	1426	7/1/2018

ETL

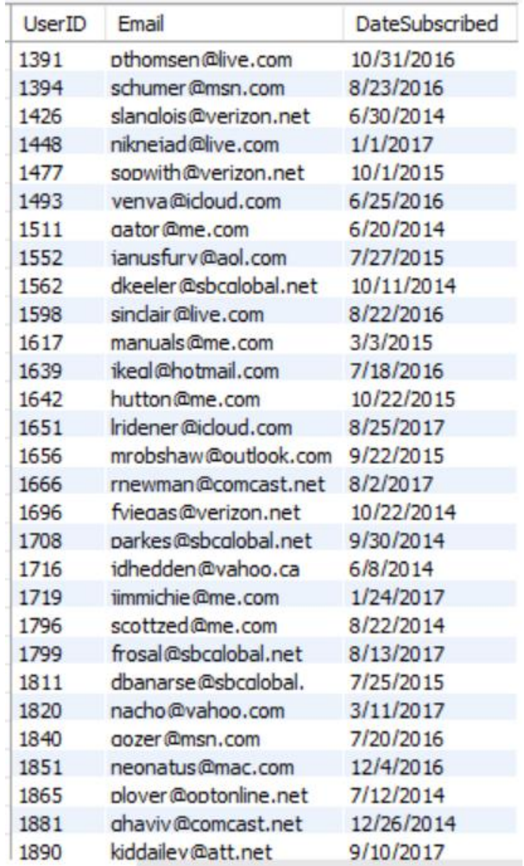


Part_Number	Entry_Number	SUM(ordered)	SUM(shipped)
AR130	187638	7116	6242
AR130	187776	12970	12842
AR130	191955	16924	15247
AR130	194692	13408	11460
AR130	210366	13716	11430
AR130	NULL	64134	57221
AR138	188187	15146	14996
AR138	188464	7100	6762
AR138	191576	24232	20890
AR138	195119	14902	13072
AR138	197910	10903	10190
AR138	200795	15280	14148
AR138	208652	14755	12296
AR138	209458	7935	7022
AR138	NULL	110253	99376
AR145	197640	11767	11207

See create_orders.sql and 5_3_OrdersDW.sql on Blackboard

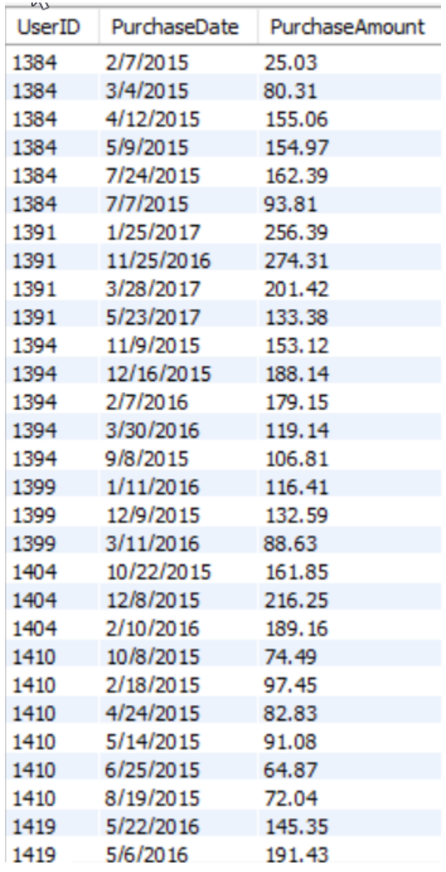
Data Warehouse Example 5 – Book Sales

Newsletter Subscription List



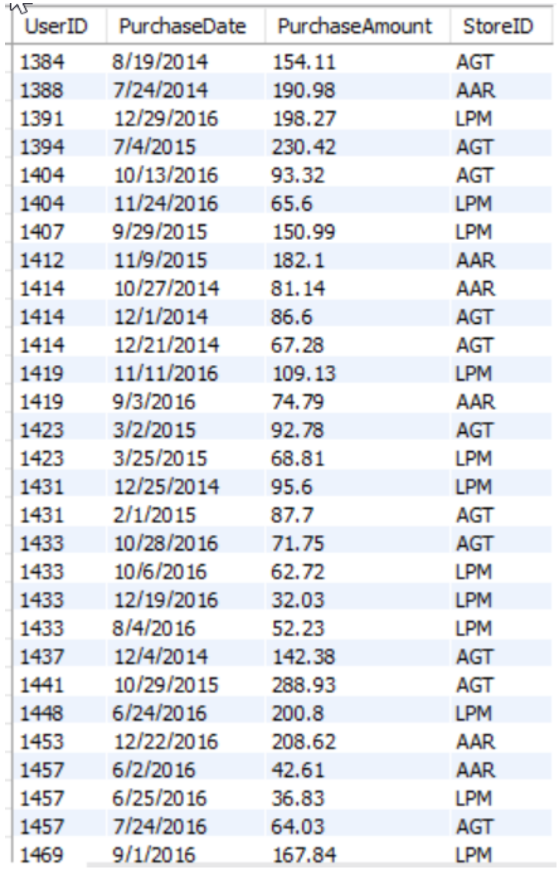
UserID	Email	DateSubscribed
1391	pthomsen@live.com	10/31/2016
1394	schumer@msn.com	8/23/2016
1426	slanglois@verizon.net	6/30/2014
1448	nikneiad@live.com	1/1/2017
1477	sopwith@verizon.net	10/1/2015
1493	venva@icloud.com	6/25/2016
1511	gator@me.com	6/20/2014
1552	ianusfury@aol.com	7/27/2015
1562	dkeeler@sbcglobal.net	10/11/2014
1598	sinclair@live.com	8/22/2016
1617	manuals@me.com	3/3/2015
1639	ikeal@hotmail.com	7/18/2016
1642	hutton@me.com	10/22/2015
1651	lridener@icloud.com	8/25/2017
1656	mrobshaw@outlook.com	9/22/2015
1666	rnewman@comcast.net	8/2/2017
1696	fviegas@verizon.net	10/22/2014
1708	parkes@sbcglobal.net	9/30/2014
1716	idhedden@yahoo.ca	6/8/2014
1719	iimmichie@me.com	1/24/2017
1796	scottzed@me.com	8/22/2014
1799	frosal@sbcglobal.net	8/13/2017
1811	dbanarse@sbcglobal.	7/25/2015
1820	nacho@yahoo.com	3/11/2017
1840	gozer@msn.com	7/20/2016
1851	neonatus@mac.com	12/4/2016
1865	plover@optonline.net	7/12/2014
1881	ghaviv@comcast.net	12/26/2014
1890	kiddailey@att.net	9/10/2017

Online Purchases by UserID

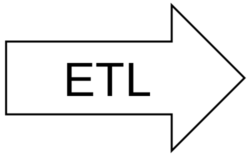


UserID	PurchaseDate	PurchaseAmount
1384	2/7/2015	25.03
1384	3/4/2015	80.31
1384	4/12/2015	155.06
1384	5/9/2015	154.97
1384	7/24/2015	162.39
1384	7/7/2015	93.81
1391	1/25/2017	256.39
1391	11/25/2016	274.31
1391	3/28/2017	201.42
1391	5/23/2017	133.38
1394	11/9/2015	153.12
1394	12/16/2015	188.14
1394	2/7/2016	179.15
1394	3/30/2016	119.14
1394	9/8/2015	106.81
1399	1/11/2016	116.41
1399	12/9/2015	132.59
1399	3/11/2016	88.63
1404	10/22/2015	161.85
1404	12/8/2015	216.25
1404	2/10/2016	189.16
1410	10/8/2015	74.49
1410	2/18/2015	97.45
1410	4/24/2015	82.83
1410	5/14/2015	91.08
1410	6/25/2015	64.87
1410	8/19/2015	72.04
1419	5/22/2016	145.35
1419	5/6/2016	191.43

In-store Purchases by UserID

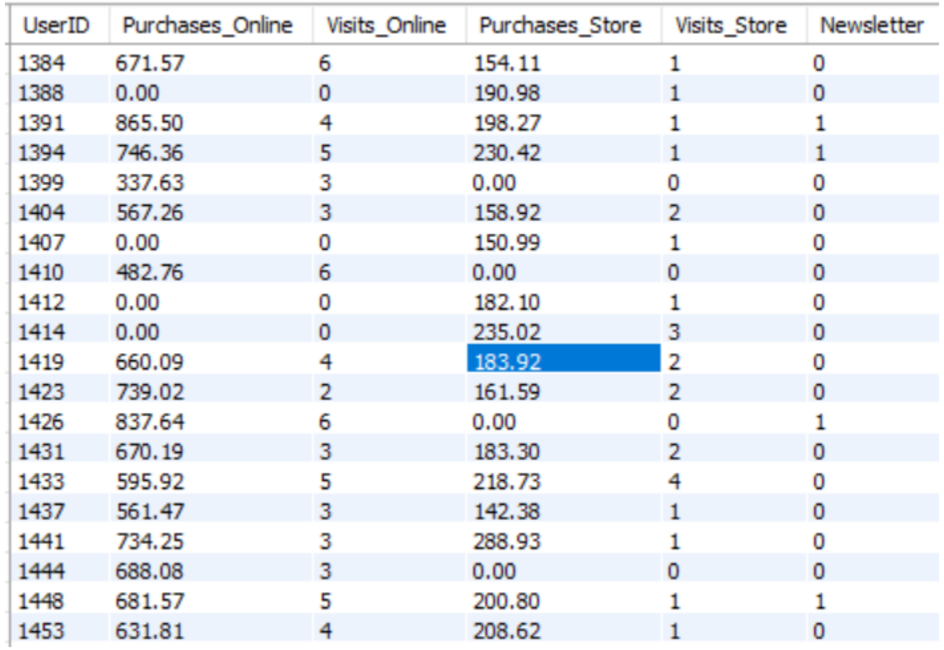


UserID	PurchaseDate	PurchaseAmount	StoreID
1384	8/19/2014	154.11	AGT
1388	7/24/2014	190.98	AAR
1391	12/29/2016	198.27	LPM
1394	7/4/2015	230.42	AGT
1404	10/13/2016	93.32	AGT
1404	11/24/2016	65.6	LPM
1407	9/29/2015	150.99	LPM
1412	11/9/2015	182.1	AAR
1414	10/27/2014	81.14	AAR
1414	12/1/2014	86.6	AGT
1414	12/21/2014	67.28	AGT
1419	11/11/2016	109.13	LPM
1419	9/3/2016	74.79	AAR
1423	3/2/2015	92.78	AGT
1423	3/25/2015	68.81	LPM
1431	12/25/2014	95.6	LPM
1431	2/1/2015	87.7	AGT
1433	10/28/2016	71.75	AGT
1433	10/6/2016	62.72	LPM
1433	12/19/2016	32.03	LPM
1433	8/4/2016	52.23	LPM
1437	12/4/2014	142.38	AGT
1441	10/29/2015	288.93	AGT
1448	6/24/2016	200.8	LPM
1453	12/22/2016	208.62	AAR
1457	6/2/2016	42.61	AAR
1457	6/25/2016	36.83	LPM
1457	7/24/2016	64.03	AGT
1469	9/1/2016	167.84	LPM



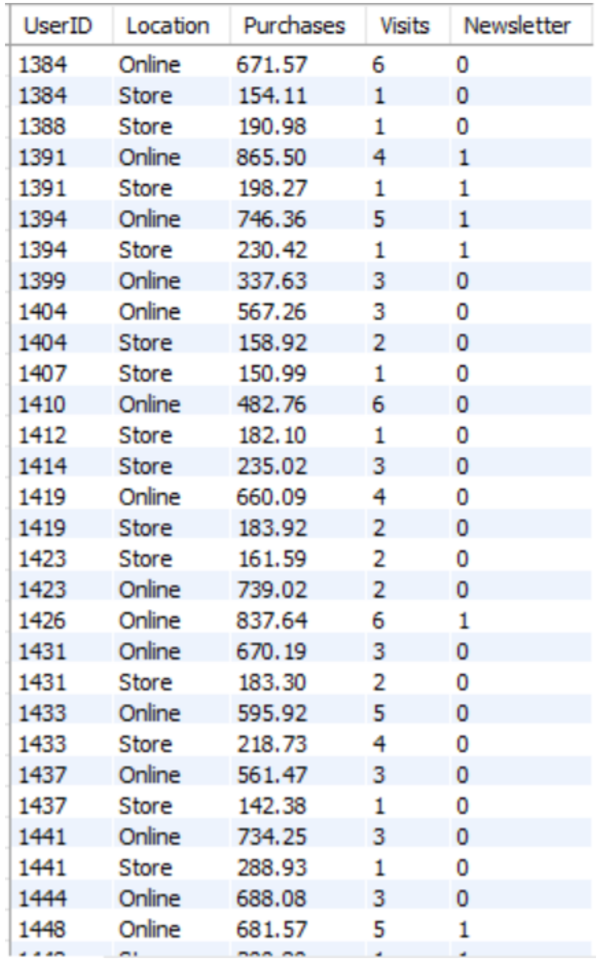
See create_booksales.sql and 5_4_BookSales_long.sql, 5_5_BookSales_wide on Blackboard

Data Warehouse Example 5 – Book Sales



UserID	Purchases_Online	Visits_Online	Purchases_Store	Visits_Store	Newsletter
1384	671.57	6	154.11	1	0
1388	0.00	0	190.98	1	0
1391	865.50	4	198.27	1	1
1394	746.36	5	230.42	1	1
1399	337.63	3	0.00	0	0
1404	567.26	3	158.92	2	0
1407	0.00	0	150.99	1	0
1410	482.76	6	0.00	0	0
1412	0.00	0	182.10	1	0
1414	0.00	0	235.02	3	0
1419	660.09	4	183.92	2	0
1423	739.02	2	161.59	2	0
1426	837.64	6	0.00	0	1
1431	670.19	3	183.30	2	0
1433	595.92	5	218.73	4	0
1437	561.47	3	142.38	1	0
1441	734.25	3	288.93	1	0
1444	688.08	3	0.00	0	0
1448	681.57	5	200.80	1	1
1453	631.81	4	208.62	1	0

Wide DW Design



UserID	Location	Purchases	Visits	Newsletter
1384	Online	671.57	6	0
1384	Store	154.11	1	0
1388	Store	190.98	1	0
1391	Online	865.50	4	1
1391	Store	198.27	1	1
1394	Online	746.36	5	1
1394	Store	230.42	1	1
1399	Online	337.63	3	0
1404	Online	567.26	3	0
1404	Store	158.92	2	0
1407	Store	150.99	1	0
1410	Online	482.76	6	0
1412	Store	182.10	1	0
1414	Store	235.02	3	0
1419	Online	660.09	4	0
1419	Store	183.92	2	0
1423	Store	161.59	2	0
1423	Online	739.02	2	0
1426	Online	837.64	6	1
1431	Online	670.19	3	0
1431	Store	183.30	2	0
1433	Online	595.92	5	0
1433	Store	218.73	4	0
1437	Online	561.47	3	0
1437	Store	142.38	1	0
1441	Online	734.25	3	0
1441	Store	288.93	1	0
1444	Online	688.08	3	0
1448	Online	681.57	5	1

Long DW Design

DW Fact Table Creation

- Design of fact table
  - Long vs. Wide
  - Granularity of fact table
- Columns in fact table
  - Summary fields
  - Calculated/Derived fields
  - Categories/Indicators
- Joining tables to create fact table – common issues
  - Many to many relationships
  - Outer joins

Long vs wide data

Department	Manager	Cost Center	Month	Cost
A	Casey	115Q	May	1365
A	Casey	115Q	Aug	1338
A	Casey	115Q	Sep	1305
A	Casey	115Q	Dec	497
A	Casey	116V	May	1455
A	Casey	116V	Jun	1485
A	Casey	116V	Aug	1482
A	Casey	116V	Nov	499
A	Casey	12N	Feb	469
A	Casey	12N	Mar	924
A	Casey	12N	Jun	1473
A	Casey	12N	Sep	1278
A	Casey	130T	May	1221
A	Casey	130T	Jul	1257
A	Casey	130T	Sep	1371
A	Casey	146W	Jan	455
A	Casey	146W	Jun	1395
A	Casey	146W	Jul	1482
A	Casey	146W	Aug	1305
A	Casey	146W	Oct	856

Long data

ID

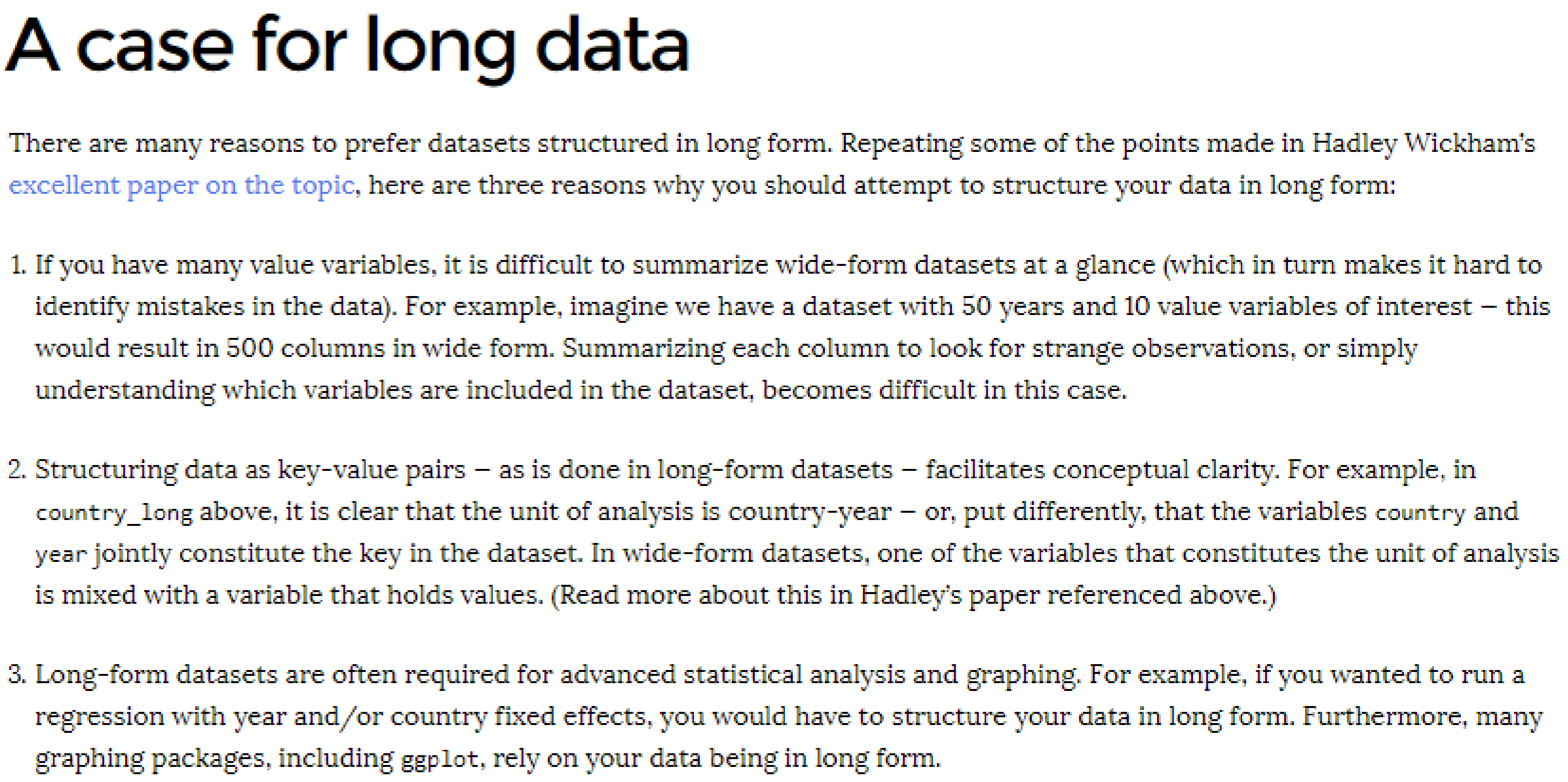
Variables

Values

Department	Manager	Cost Center	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
C	Eng	99Y			916	1210		921	1191	1350				633
A	Shah	107C					1197	1068					415	411
B	Casey	76C			920			1374	1212	1419	1209			
A	Shah	50Y	647			944	1005	1680	1278	1638				
A	Shah	116V			720		984	1971	1194	1296		1324	662	
B	Shah	68U				874	1743		1566	2100	1863	1152		
B	Casey	50Y		418		862		1383	1446					
A	Shah	99Y						1992		1500		1138		
A	Eng	12N			1624	1030	2067	1974	2376					
B	Eng	99Y					2373	1551		1551	2148	1328		
A	Shah	66O				726		1410	1359			1372		578
D	Eng	66O					1815				2163		823	
D	Casey	68U			612	888	2013	1170	1635	2088		722	445	
D	Eng	99Y			1140				2190			1344		
A	Casey	7E				1184	1581	1638	1953	1602				
C	Eng	107C			1084			2043	2025	2469	1533			
A	Eng	107C			1638		1734	1944	1920	2451			840	546
C	Shah	12N	461		1330			1185	1272		1428	602		

Wide data

Long vs wide data



A case for long data

There are many reasons to prefer datasets structured in long form. Repeating some of the points made in Hadley Wickham's excellent paper on the topic, here are three reasons why you should attempt to structure your data in long form:

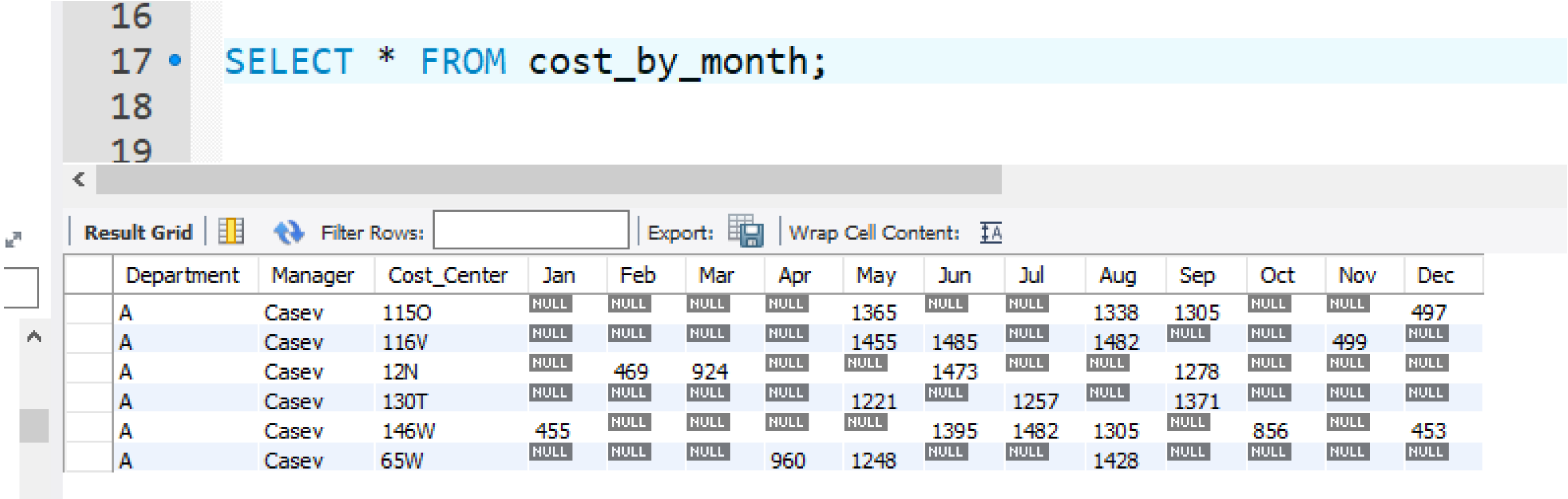
1. If you have many value variables, it is difficult to summarize wide-form datasets at a glance (which in turn makes it hard to identify mistakes in the data). For example, imagine we have a dataset with 50 years and 10 value variables of interest – this would result in 500 columns in wide form. Summarizing each column to look for strange observations, or simply understanding which variables are included in the dataset, becomes difficult in this case.

2. Structuring data as key-value pairs – as is done in long-form datasets – facilitates conceptual clarity. For example, in `country_long` above, it is clear that the unit of analysis is country-year – or, put differently, that the variables `country` and `year` jointly constitute the key in the dataset. In wide-form datasets, one of the variables that constitutes the unit of analysis is mixed with a variable that holds values. (Read more about this in Hadley's paper referenced above.)

3. Long-form datasets are often required for advanced statistical analysis and graphing. For example, if you wanted to run a regression with year and/or country fixed effects, you would have to structure your data in long form. Furthermore, many graphing packages, including `ggplot`, rely on your data being in long form.

https://sejdemyr.github.io/r-tutorials/basics/wide-and-long/

Data Warehouse Example 6 – Cost By Month

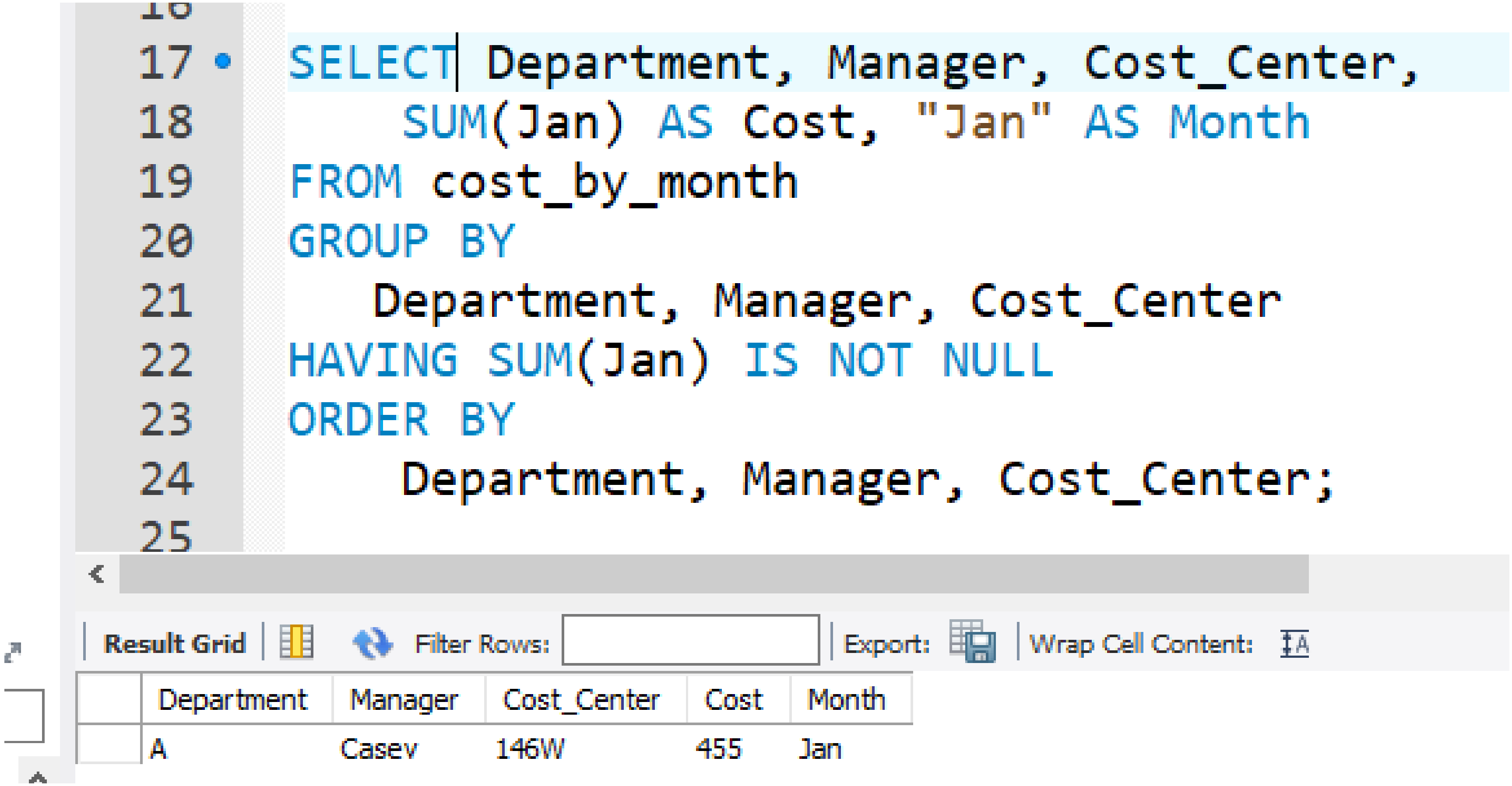


```
SELECT * FROM cost_by_month;
```

Department	Manager	Cost_Center	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A	Casev	115O	NULL	NULL	NULL	NULL	1365	NULL	NULL	1338	1305	NULL	NULL	497
A	Casev	116V	NULL	NULL	NULL	NULL	1455	1485	NULL	1482	NULL	NULL	499	NULL
A	Casev	12N	NULL	469	924	NULL	NULL	1473	NULL	NULL	1278	NULL	NULL	NULL
A	Casev	130T	NULL	NULL	NULL	NULL	1221	NULL	1257	NULL	1371	NULL	NULL	NULL
A	Casev	146W	455	NULL	NULL	NULL	NULL	1395	1482	1305	NULL	856	NULL	453
A	Casev	65W	NULL	NULL	NULL	960	1248	NULL	NULL	1428	NULL	NULL	NULL	NULL

See create_cost_by_month.sql and 5_6_reshape_queries.sql on Blackboard

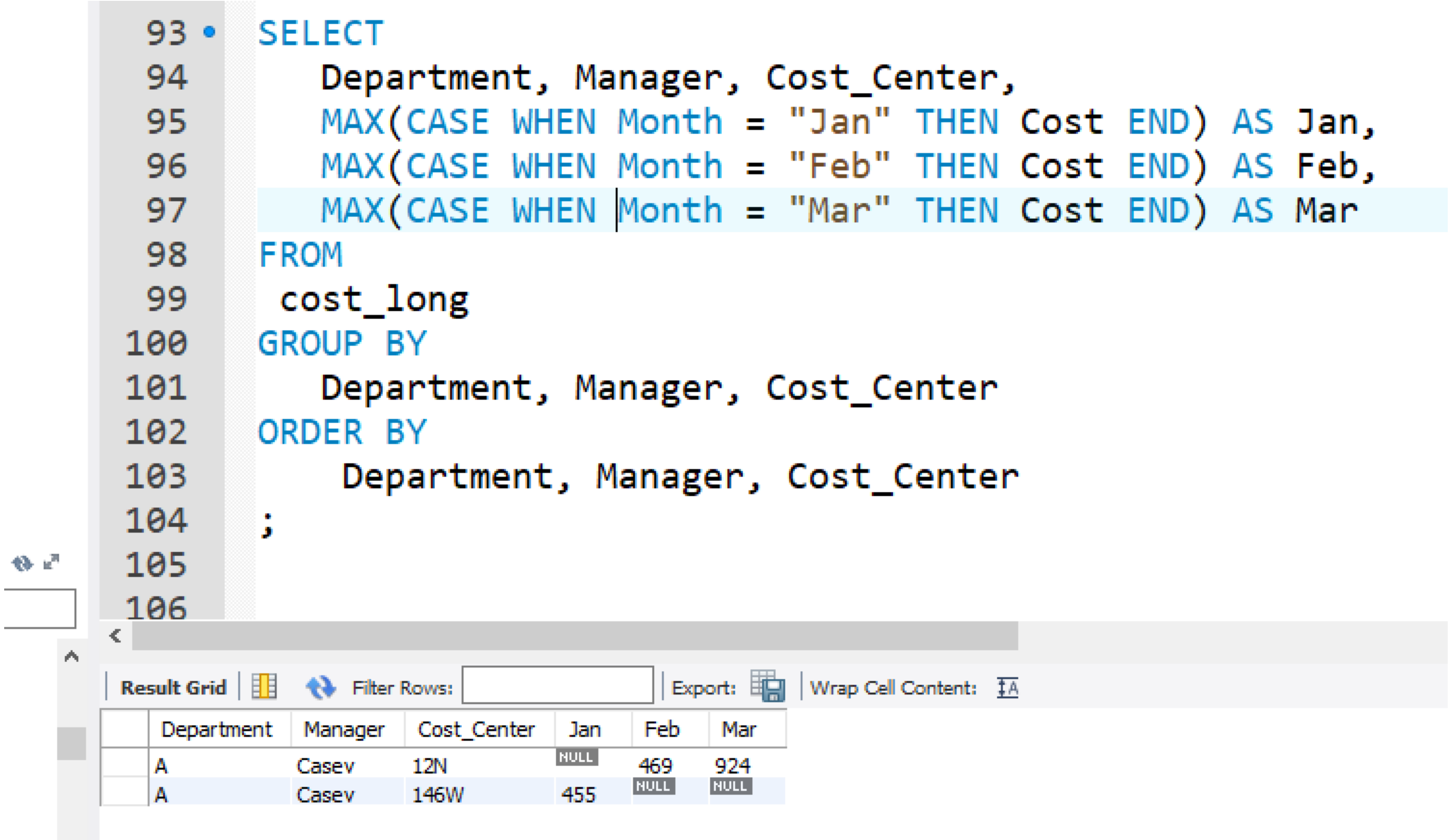
Data Warehouse Example 6 – Cost By Month



```
SELECT Department, Manager, Cost_Center,
     SUM(Jan) AS Cost, "Jan" AS Month
FROM cost_by_month
GROUP BY
   Department, Manager, Cost_Center
HAVING SUM(Jan) IS NOT NULL
ORDER BY
     Department, Manager, Cost_Center;
```

Department	Manager	Cost_Center	Cost	Month
A	Casey	146W	455	Jan

Data Warehouse Example 6 – Cost By Month



```
SELECT
   Department, Manager, Cost_Center,
   MAX(CASE WHEN Month = "Jan" THEN Cost END) AS Jan,
   MAX(CASE WHEN Month = "Feb" THEN Cost END) AS Feb,
   MAX(CASE WHEN Month = "Mar" THEN Cost END) AS Mar
FROM
 cost_long
GROUP BY
   Department, Manager, Cost_Center
ORDER BY
    Department, Manager, Cost_Center
;
```

Department	Manager	Cost_Center	Jan	Feb	Mar
A	Casey	12N	NULL	469	924
A	Casey	146W	455	NULL	NULL