Chapter 1: Reducing Rows and Columns in Your Result Sets

Table	Column	Database	Schema
Order	City Key	WideWorldImportersDW	Fact
Order	Customer Key	WideWorldImportersDW	Fact
Order	Description	WideWorldImportersDW	Fact
Order	Lineage Key	WideWorldImportersDW	Fact
Order	Order Date Key	WideWorldImportersDW	Fact
Order	Order Key	WideWorldImportersDW	Fact
Order	Package	WideWorldImportersDW	Fact
Order	Picked Date Key	WideWorldImportersDW	Fact
Order	Picker Key	WideWorldImportersDW	Fact
Order	Quantity	WideWorldImportersDW	Fact
Order	Salesperson Key	WideWorldImportersDW	Fact
Order	Stock Item Key	WideWorldImportersDW	Fact
Order	Tax Amount	WideWorldImportersDW	Fact
Order	Tax Rate	WideWorldImportersDW	Fact
Order	Total Excluding Tax	WideWorldImportersDW	Fact
Order	Total Including Tax	WideWorldImportersDW	Fact
Order	Unit Price	WideWorldImportersDW	Fact
Order	WWI Backorder ID	WideWorldImportersDW	Fact
Order	WWI Order ID	WideWorldImportersDW	Fact

Order Key	City Key	Customer	Stock Item Key	Order Date Key	Picked Date Key	Salesperson Key	Picker Key	WWI Order ID	WWI Backorder ID	Description	Package	Quantity
702	50969	0	163	2013-01-05	2013-01-05	11	22	291	NULL	IT joke mug - hardware: ...	Each	3
894	53636	0	182	2013-01-07	2013-01-07	12	17	357	NULL	Developer joke mug - in ...	Each	3
924	41165	0	197	2013-01-07	2013-01-07	23	17	365	NULL	DBA joke mug - it depen ...	Each	3
1087	72156	0	176	2013-01-07	2013-01-07	6	17	414	NULL	Developer joke mug - th ...	Each	3
1292	47741	0	182	2013-01-09	2013-01-09	7	22	474	NULL	Developer joke mug - in ...	Each	3
1941	51415	0	198	2013-01-11	2013-01-11	23	9	662	NULL	DBA joke mug - it depen ...	Each	3
1981	68127	0	166	2013-01-12	2013-01-12	27	22	676	NULL	IT joke mug - that behav ...	Each	3
2148	50119	0	164	2013-01-14	2013-01-14	26	12	727	NULL	IT joke mug - hardware: ...	Each	3
2216	70334	0	167	2013-01-14	2013-01-14	27	12	748	NULL	IT joke mug - keyboard r ...	Each	3
2225	47741	0	178	2013-01-14	2013-01-14	15	12	750	NULL	Developer joke mug - un ...	Each	3
2238	60505	0	174	2013-01-14	2013-01-14	12	12	753	NULL	Developer joke mug - a f ...	Each	3
2271	43128	0	173	2013-01-14	2013-01-14	21	12	762	NULL	Developer joke mug - a f ...	Each	3
2498	49849	0	163	2013-01-15	2013-01-15	12	17	831	NULL	IT joke mug - hardware: ...	Each	3
2727	72539	0	204	2013-01-17	2013-01-17	11	17	905	NULL	DBA joke mug - mind if I ...	Each	3
2869	64965	0	199	2013-01-17	2013-01-17	15	17	948	NULL	DBA joke mug - you mig ...	Each	3
3133	52423	0	195	2013-01-18	2013-01-18	28	20	1025	NULL	DBA joke mug - I will get ...	Each	3
3229	48762	0	178	2013-01-19	2013-01-19	11	22	1056	NULL	Developer joke mug - un ...	Each	3
3324	59392	0	196	2013-01-21	2013-01-21	12	29	1082	NULL	DBA joke mug - I will get ...	Each	3
3463	60508	0	189	2013-01-22	2013-01-22	19	19	1127	NULL	Developer joke mug - O ...	Each	3
3506	45901	0	176	2013-01-22	2013-01-22	35	19	1140	NULL	Developer joke mug - th ...	Each	3
3514	54129	0	176	2013-01-22	2013-01-22	28	19	1143	NULL	Developer joke mug - th ...	Each	3
3627	41981	0	172	2013-01-23	2013-01-23	19	28	1178	NULL	Developer joke mug - th ...	Each	3
3673	49849	0	169	2013-01-23	2013-01-23	19	28	1191	NULL	Developer joke mug - ol ...	Each	3
3730	72610	0	179	2013-01-23	2013-01-23	11	28	1208	NULL	Developer joke mug - (hi ...	Each	3

Order Date	Order	Stock Item	Customer	WWI Order	WWI Backorder
2013-01-01	26	176	330	12	54
2013-01-03	407	195	61	177	222
2013-01-07	809	173	125	334	421
2013-01-10	1578	165	258	558	592
2013-01-15	2287	192	197	768	858
2013-02-12	6908	168	162	2156	2163
2013-03-12	11187	195	25	3483	3523
2013-03-13	11417	201	207	3555	3585
2013-03-14	11599	168	379	3610	3675
2013-03-21	13017	173	206	4056	4094
2013-03-22	13187	199	170	4114	4160
2013-03-26	13692	170	11	4272	4305
2013-03-28	14125	198	34	4403	4428
2013-03-29	14278	171	313	4454	4471
2013-04-02	14738	166	7	4606	4645
2013-04-03	14945	196	303	4667	4703
2013-04-05	15446	183	380	4829	4885
2013-04-09	15920	186	177	4982	5007

Order Year	Order Month	Order	Stock Item	Customer	WWI Order	WWI Backorder
2013	1	26	176	330	12	54
2013	1	407	195	61	177	222
2013	1	809	173	125	334	421
2013	1	1578	165	258	558	592
2013	1	2287	192	197	768	858
2013	2	6908	168	162	2156	2163
2013	3	11187	195	25	3483	3523
2013	3	11417	201	207	3555	3585
2013	3	11599	168	379	3610	3675
2013	3	13017	173	206	4056	4094
2013	3	13187	199	170	4114	4160
2013	3	13692	170	11	4272	4305
2013	3	14125	198	34	4403	4428
2013	3	14278	171	313	4454	4471
2013	4	14738	166	7	4606	4645
2013	4	14945	196	303	4667	4703
2013	4	15446	183	380	4829	4885
2013	4	15920	186	177	4982	5007

Order Year	Order Month	Order	Stock Item	Customer	WWI Order	WWI Backorder
2013	1	26	176	330	12	54
2013	1	407	195	61	177	222
2013	1	809	173	125	334	421
2013	1	1578	165	258	558	592
2013	1	2287	192	197	768	858
2013	2	6908	168	162	2156	2163
2013	3	11187	195	25	3483	3523
2013	3	11417	201	207	3555	3585
2013	3	11599	168	379	3610	3675
2013	3	13017	173	206	4056	4094
2013	3	13187	199	170	4114	4160
2013	3	13692	170	11	4272	4305
2013	3	14125	198	34	4403	4428
2013	3	14278	171	313	4454	4471
2013	4	14738	166	7	4606	4645
2013	4	14945	196	303	4667	4703
2013	4	15446	183	380	4829	4885
2013	4	15920	186	177	4982	5007
2013	4	16517	181	7	5171	5223
2013	4	17541	192	46	5504	5551
2013	4	19192	187	87	6030	6052
2013	4	19321	188	372	6068	6106

Order Year	Order Month	number of backorders	number of impacted customers	number of orders
2015	4	556	556	556
2013	2	209	209	209
2013	7	495	495	495
2014	5	575	575	575
2014	10	602	602	602
2015	6	603	603	603
2016	1	571	571	571
2013	9	466	466	466
2014	7	544	544	544
2013	11	442	442	442
2014	11	498	498	498
2013	5	510	510	510
2014	3	398	398	398
2014	4	486	486	486
2015	5	561	561	561
2016	3	512	512	512
2015	1	523	523	523
2013	1	481	481	481
2015	8	531	531	531
2013	3	378	378	378
2014	1	453	453	453

Order Year	Order Month	Impacted Customers	Backorder Items	Number of backorders	orders	Customers
2013	1	105	189	481	5281	367
2013	2	44	139	209	3726	340
2013	3	81	175	378	5389	375
2013	4	105	190	487	5314	373
2013	5	108	201	510	5699	373
2013	6	75	190	390	5338	364
2013	7	107	199	495	5896	375
2013	8	88	187	393	4819	358
2013	9	94	192	466	5099	364
2013	10	85	187	433	5171	361
2013	11	82	188	442	4957	350
2013	12	83	186	411	4966	356
2014	1	99	191	453	5682	375
2014	2	80	191	419	4830	352
2014	3	78	176	398	5118	362
2014	4	99	197	486	5513	371
2014	5	112	203	575	6027	377
2014	6	114	199	585	5888	374
2014	7	111	200	544	6310	371
2014	8	94	195	454	5130	360
2014	9	88	185	453	5117	357

Order Year	Order Month	Customers	orders	total backorders	impacted customers
2013	1	367	5281	481	105
2013	2	340	3726	209	44
2013	3	375	5389	378	81
2013	4	373	5314	487	105
2013	5	373	5699	510	108
2013	6	364	5338	390	75
2013	7	375	5896	495	107
2013	8	358	4819	393	88
2013	9	364	5099	466	94
2013	10	361	5171	433	85
2013	11	350	4957	442	82
2013	12	356	4966	411	83
2014	1	375	5682	453	99
2014	2	352	4830	419	80
2014	3	362	5118	398	78
2014	4	371	5513	486	99
2014	5	377	6027	575	112
2014	6	374	5888	585	114
2014	7	371	6310	544	111
2014	8	360	5130	454	94
2014	9	357	5117	453	88

Chapter 2: Efficiently Aggregating Data in Your Results

Year	Month	Average # of Items Sold
2013	1	36
2013	2	38
2013	3	38
2013	4	40
2013	5	41
2013	6	40
2013	7	39
2013	8	40
2013	9	37
2013	10	38
2013	11	39
2013	12	39
2014	1	38
2014	2	38
2014	3	38

Year	Month	Total # of Items Sold
2013	1	193271
2013	2	142120
2013	3	207486
2013	4	212995
2013	5	230725
2013	6	213468
2013	7	232599
2013	8	192199
2013	9	190567
2013	10	198476
2013	11	194290
2013	12	193461
2014	1	216337
2014	2	182103
2014	3	196451

Year	Month	# of Items Sold
2013	1	5246
2013	2	3707
2013	3	5330
2013	4	5254
2013	5	5617
2013	6	5287
2013	7	5834
2013	8	4767
2013	9	5021
2013	10	5097
2013	11	4899
2013	12	4909
2014	1	5610
2014	2	4768
2014	3	5070

Invoice Date Key	# of Items Sold
2013-01-01	96
2013-01-02	288
2013-01-03	260
2013-01-04	250
2013-01-05	216
2013-01-07	288
2013-01-08	360
2013-01-09	260
2013-01-10	252
2013-01-11	324
2013-01-12	225
2013-01-14	260
2013-01-15	288
2013-01-16	260
2013-01-17	216

Invoice Date Key	Delivery Date Key	# of items sold	profit	total bill with taxes
2013-02-08	2013-02-09	7434	79058.6	178077.76
2013-02-25	2013-02-26	4925	41569.15	93578.39
2013-03-02	2013-03-03	3736	43854.85	88493.94
2013-03-14	2013-03-15	14002	119923.05	250612.44
2013-03-20	2013-03-21	9334	86824.65	185902.85
2013-03-26	2013-03-27	5714	47061.4	114092.32
2013-05-11	2013-05-12	4596	28995.75	71460.65
2013-05-21	2013-05-22	7531	96755.95	217180.22
2013-05-27	2013-05-28	10093	97419.9	214008.99
2013-05-28	2013-05-29	12974	110398.7	242219.34
2013-06-14	2013-06-15	9012	95209.8	217710.55
2013-06-26	2013-06-27	6914	76876.65	186906.23
2013-07-18	2013-07-19	6936	62780.8	153203.6
2013-08-21	2013-08-22	11592	75627.8	185726.16
2013-08-26	2013-08-27	9811	79456.7	183368.28
2013-08-27	2013-08-28	6490	59667.85	141011.2
2013-09-26	2013-09-27	6661	72937	161039.4
2013-10-18	2013-10-19	4715	39938.65	89821.57
2013-11-04	2013-11-05	8177	62764.95	146858.6
2013-11-14	2013-11-15	6670	60927	136476.5

Date	Average profits by date	Average bill amount by date
2013-02-08	10.63	23.95
2013-02-05	8.44	19

Year	# of items sold	profit	total bill with taxes
2013	2401657	22768352.25	52563272.64
2016	1241304	11174765.55	25971029.11
2014	2567401	24828462.45	57418916.89
2015	2740266	26957600.65	62090220.81

Year	Deliver Date	# of items sold	profit	Total bill will taxes
2013	2013-01-04	8256	70075.75	155966.93
2013	2013-02-09	7434	79058.6	178077.76
2013	2013-02-26	4925	41569.15	93578.39
2013	2013-04-17	11898	118855.4	269476.07
2013	2013-04-18	11710	92089.45	210607.35
2013	2013-04-24	9884	96883.3	222505.81
2013	2013-05-05	6253	58012.65	124110.89
2013	2013-06-06	9526	112814.35	250679.35
2013	2013-07-12	12607	149956.9	358280.5
2013	2013-07-27	6487	72803.85	175664.82
2013	2013-09-18	6900	68950.05	160869.91
2013	2013-10-03	6912	69076.05	150202.66
2013	2013-10-22	9164	79122.95	193849.8
2013	2013-03-15	14002	119923.05	250612.44
2013	2013-07-21	5998	42889.25	103548.25
2013	2013-03-30	5115	45994.8	115153.42
2013	2013-05-22	7531	96755.95	217180.22
2013	2013-10-29	9297	82435.65	195568.9
2013	2013-11-08	5139	50314	113162.37
2013	2013-11-23	4283	42874	95749.82
2013	2013-12-29	1509	14555.7	34713.84

AVG()	Year	Earliest Delivery Date
COUNT()	2013	2013-01-02
MAX()	2014	2014-01-02
MIN()	2015	2015-01-02
SUM()	2016	2016-01-02

Invoice Year	Invoice Month	# of Customers with Orders	# of Customers Received Orders
2013	7	5834	5637
2013	8	4767	4877
2013	9	5021	4934
2013	10	5097	5092
2013	11	4899	5022
2013	12	4909	4675
2014	1	5610	5595
2014	2	4768	4911
2014	3	5070	5006
2014	4	5443	5427
2014	5	5930	6053
2014	6	5798	5772
2014	7	6227	6229
2014	8	5059	5202
2014	9	5059	4914

Invoice Year	Invoice Month	# of Customers with Orders	# of Customers Received Orders
2013	1	5246	5013
2013	2	3707	3805
2013	3	5330	5465
2013	4	5254	4992
2013	5	5617	5648
2013	6	5287	5518
2013	7	5834	5637
2013	8	4767	4877
2013	9	5021	4934
2013	10	5097	5092
2013	11	4899	5022
2013	12	4909	4675
2014	1	5610	5595
2014	2	4768	4911
2014	3	5070	5006
2014	4	5443	5427
2014	5	5930	6053
2014	6	5798	5772
2014	7	6227	6229
2014	8	5059	5202
2014	9	5059	4914

Chapter 3: Formatting Your Results for Easier Consumption

	US English	British English	German	Chinese Simplified (PRC)
1	1/1/2013	01/01/2013	01.01.2013	2013/1/1
2	1/2/2013	02/01/2013	02.01.2013	2013/1/2
3	1/3/2013	03/01/2013	03.01.2013	2013/1/3

	US English	British English	German	Chinese Simplified (PRC)
1	Tuesday, January 1, 2013	01 January 2013	Dienstag, 1. Januar 2013	2013年1月1日
2	Wednesday, January 2, 2013	02 January 2013	Mittwoch, 2. Januar 2013	2013年1月2日
3	Thursday, January 3, 2013	03 January 2013	Donnerstag, 3. Januar 2013	2013年1月3日

	Quantity	Unit Price	Tax Rate
1	10	230,00 €	15,00 %
2	9	13,00 €	15,00 %
3	9	32,00 €	15,00 %

	Date
1	01-01-2013
2	01-02-2013
3	01-03-2013

	Date
1	2013/01/01
2	2013/01/02
3	2013/01/03

	Year
1	2013
2	2013
3	2013

	Date
1	01/01/2013 00:00 AM
2	01/02/2013 00:00 AM
3	01/03/2013 00:00 AM

	Tax Rate
1	15.00%
2	15.00%
3	15.00%

	Tax
1	15.0%

	Date
1	20130101
2	20130102
3	20130103

	Date
1	130101
2	130102
3	130103

	Date
1	01 Jan 2013 00:00:00:000
2	02 Jan 2013 00:00:00:000
3	03 Jan 2013 00:00:00:000

	Number
1	102.55

	NumericNumber	IntegerNumber
1	103	102

	Number
1	$102.55268

	number
1	102.55

	Number
1	102.60000

	Number
1	102.6

	Number
1	102

	Number
1	-102

	Order Key	Description
1	1	Ride on toy sedan car (Black) 1/12 scale
2	2	Developer joke mug - old C developers never die (...
3	3	USB food flash drive - chocolate bar

	Order Key	Product Description
1	1	Ride on toy sedan car (Black) 1/12 scale
2	2	Developer joke mug - old C developers never die (...
3	3	USB food flash drive - chocolate bar

Chapter 4: Manipulating Your Data Results Using Conditional SQL

	Employee Key	WWI Employee ID	Employee	Is Salesperson
1	212	20	Jack Potter	1
2	208	16	Archer Lamble	1
3	207	15	Taj Shand	1
4	206	14	Lily Code	1
5	205	13	Hudson Hollinworth	1
6	200	8	Anthony Grosse	1
7	199	7	Amy Trefl	1
8	198	6	Sophia Hinton	1
9	195	3	Hudson Onslow	1
10	194	2	Kayla Woodcock	1
11	196	4	Isabella Rupp	0
12	197	5	Eva Muirden	0
13	201	9	Alica Fatnowna	0
14	202	10	Stella Rosenhain	0
15	203	11	Ethan Onslow	0
16	204	12	Henry Forlonge	0
17	209	17	Piper Koch	0
18	210	18	Katie Darwin	0
19	211	19	Jai Shand	0

	Employee Key	Employee	Preferred Name
1	204	Henry Forlonge	Henry
2	205	Hudson Hollinworth	Hudson
3	195	Hudson Onslow	Hudson
4	210	Katie Darwin	Katie
5	194	Kayla Woodcock	Kayla
6	201	Alica Fatnowna	Alica
7	199	Amy Trefl	Amy
8	200	Anthony Grosse	Anthony
9	208	Archer Lamble	Archer
10	209	Piper Koch	Piper
11	203	Ethan Onslow	Ethan
12	197	Eva Muirden	Eva
13	196	Isabella Rupp	Isabella
14	212	Jack Potter	Jack
15	211	Jai Shand	Jai
16	206	Lily Code	Lily
17	198	Sophia Hinton	Sophia
18	202	Stella Rosenhain	Stella
19	207	Taj Shand	Taj

	City	State Province	Sales Territory
1	Carrollton	New York	Mideast
2	Carrollton	Virginia	Southeast
3	Carrollton	Illinois	Great Lakes
4	Carrollton	Missouri	Plains
5	Carrollton	Ohio	Great Lakes
6	Carrollton	Kentucky	Southeast
7	Carrollton	Georgia	Southeast
8	Carrollton	Alabama	Southeast
9	Carrollton	Mississippi	Southeast
10	Carrollton	Texas	Southwest

	City	State Province	Sales Territory
1	Carrollton	New York	Eastcoast
2	Carrollton	Virginia	Southeast
3	Carrollton	Illinois	Midwest
4	Carrollton	Missouri	Midwest
5	Carrollton	Ohio	Great Lakes
6	Carrollton	Kentucky	Southeast
7	Carrollton	Georgia	Southeast
8	Carrollton	Alabama	Southeast
9	Carrollton	Mississippi	Southeast
10	Carrollton	Texas	Southwest

	Customer Key	Customer Total Spending
1	378	154555.45
2	374	169931.60
3	111	181002.15
4	197	183277.70
5	227	183321.40
6	340	185707.00
7	66	188680.45
8	292	190017.30
9	305	195208.05
10	258	197314.45
11	225	198555.45
12	369	199441.80
13	358	201364.45
14	251	203346.30
15	126	203960.40
16	91	204303.60
17	348	204815.50
18	303	205312.55
19	24	206388.85
20	175	207512.50

	Customer Key	Customer Total Spending	Spending Group
1	378	154555.45	High
2	374	169931.60	High
3	111	181002.15	High
4	197	183277.70	High
5	227	183321.40	High
6	340	185707.00	High
7	66	188680.45	High
8	292	190017.30	High
9	305	195208.05	High
10	258	197314.45	High
11	225	198555.45	High
12	369	199441.80	High
13	358	201364.45	High
14	251	203346.30	High
15	126	203960.40	High
16	91	204303.60	High
17	348	204815.50	High
18	303	205312.55	High
19	24	206388.85	High
20	175	207512.50	High

	Output
1	8

	Stock Holding Key	Bin Location	BinLocationDetailed	Target Stock Level
1	1	L-1	LowerLevel1	100
2	2	L-1	LowerLevel1	100
3	3	L-2	LowerLevel2	120

	Order Key	WWI Order ID	Total Excluding Tax	Sales Size
1	1	1	2300.00	Extra Large
2	2	2	117.00	Medium
3	3	2	288.00	Medium

	Output
1	A

	Output
1	2

```
Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varchar value 'A' to data type int.
```

	TransactionKey	WWIInvoiceID	WWICustomerTransactionID	WWI ID	WWI ID with NULL
1	40	40	127	000004040127	000004040127
2	41	41	130	000004141130	000004141130
3	42	NULL	231	000004200231	NULL
4	43	NULL	232	000004300232	NULL

	Output
1	Hello

	Output
1	World

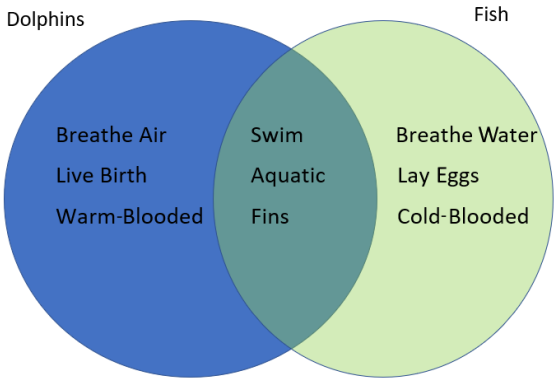
	Output
1	Hello

	Transaction Key	WWI Invoice ID	WWIInvoiceID
1	40	40	40
2	41	41	41
3	42	NULL	0
4	43	NULL	0

Chapter 5: Using Common Table Expressions

```
with ISODates as (
SELECT distinct [Date]
      ,[ISO Week Number] as [ISOWeekNumber]
  FROM [Dimension].[Date])

SELECT Date, [ISOWeekNumber]
FROM ISODates
```

	Date	ISOWeekNumber
1	2013-01-01	1
2	2013-01-02	1
3	2013-01-03	1
4	2013-01-04	1
5	2013-01-05	1
6	2013-01-06	1
7	2013-01-07	2



	Date	ISOWeekNumber	DailyTotalDry	DailyTotalChiller	TotalDailyItems
1	2016-01-01	53	5891	0	5891
2	2016-01-02	53	7260	96	7356
3	2016-01-03	53	0	0	0
4	2016-01-04	1	13140	1032	14172
5	2016-01-05	1	9942	1392	11334
6	2016-01-06	1	14130	1752	15882
7	2016-01-07	1	14388	1968	16356
8	2016-01-08	1	10934	912	11846
9	2016-01-09	1	5384	1188	6572
10	2016-01-10	1	0	0	0
11	2016-01-11	2	5733	528	6261
12	2016-01-12	2	12271	1464	13735
13	2016-01-13	2	10565	516	11081

```
    GROUP BY [Invoice Date Key])

    --end of CTEs

    --Query selecting from CTEs:
    SELECT
      [Date]
    , [ISOWeekNumber]
    , ISNULL(DailyTotalDry,0) as DailyTotalDry
    , ISNULL(DailyTotalChiller,0) as DailyTotalChiller
    , ISNULL(DailyTotalDry,0)  +  ISNULL(DailyTotalChiller,0) as TotalDailyItems
    FROM ISODates d
    LEFT OUTER JOIN DryTotalQuery dtq on d.[Date] = dtq.DryInvoiceDateKey
    LEFT OUTER JOIN ChillerTotalQuery ctq on d.[Date] = ctq.ChillerInvoiceDateKey
    WHERE YEAR(d.Date) = 2016
    ORDER BY [Date]
--try to reuse CTE: will fail
    SELECT * FROM ISODates
```

```
(366 rows affected)
Msg 208, Level 16, State 1, Line 36
Invalid object name 'ISODates'.

Completion time: 2022-11-27T12:20:56.0445313-07:00
```

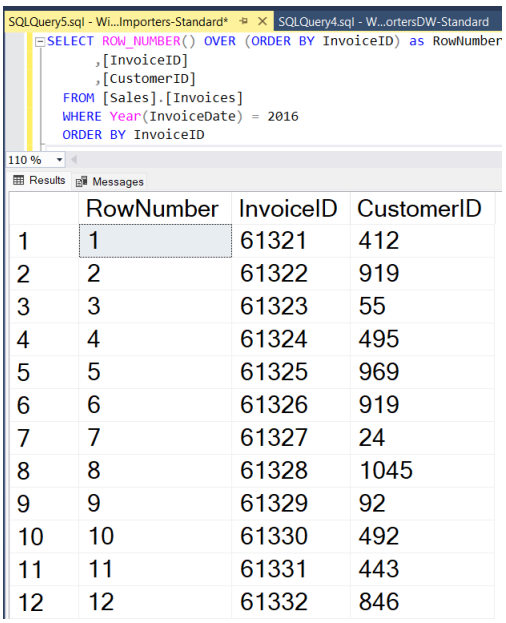
Query completed with errors.

```
WITH DirectReports AS
(   --the anchor member is the top level where there is no ManagerID
    SELECT ManagerID, EmpID, JobTitle, cast('' as nvarchar(50)) as ManagerTitle,
    0 AS EmployeeLevel --create a level field
    FROM dbo.Employee
    WHERE ManagerID IS NULL
    UNION ALL
    SELECT e.ManagerID, e.EmpID, e.JobTitle, cast(d.JobTitle as nvarchar(50)) as ManagerTitle,
    EmployeeLevel + 1 --increase the level by 1 for each level of recursiveness
    FROM dbo.Employee AS e
        INNER JOIN DirectReports AS d --referencing the CTE name here is the RECURSIVE technique
        ON e.ManagerID = d.EmpID
)
SELECT ManagerID, EmpID, JobTitle, ManagerTitle, EmployeeLevel
FROM DirectReports
```

	ManagerID	EmpID	JobTitle	ManagerTitle	EmployeeLevel
1	NULL	1	Chief Executive Officer		0
2	1	16	Human Resources Manager	Chief Executive Officer	1
3	1	273	Vice President of Sales	Chief Executive Officer	1
4	16	23	HR Specialist	Human Resources Manager	2
5	273	274	West US Sales Manager	Vice President of Sales	2
6	273	285	East US Sales Manager	Vice President of Sales	2
7	274	275	Sales Representative	West US Sales Manager	3
8	274	276	Sales Representative	West US Sales Manager	3
9	285	286	Sales Representative	East US Sales Manager	3

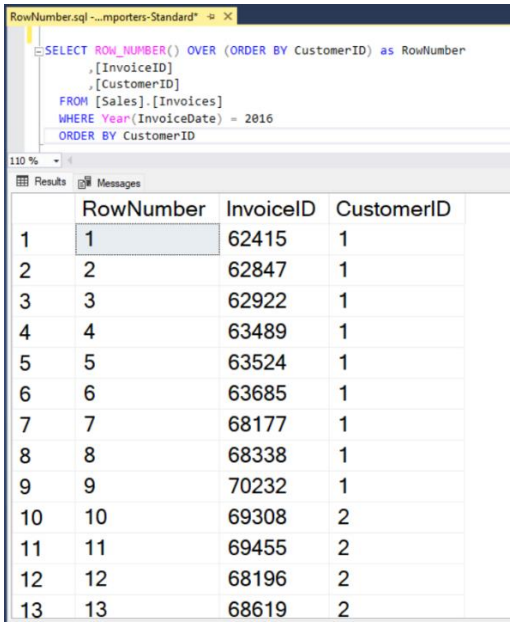
Query executed successfully.

Chapter 6: Analyze Your Data Using Window Functions



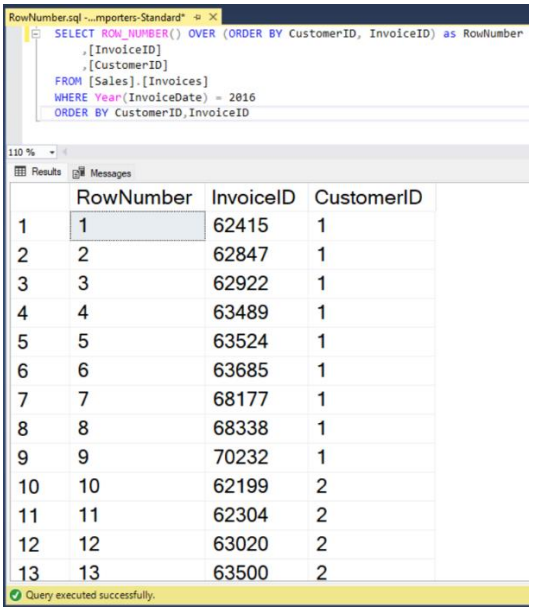
```
SELECT ROW_NUMBER() OVER (ORDER BY InvoiceID) as RowNumber
      ,[InvoiceID]
      ,[CustomerID]
  FROM [Sales].[Invoices]
  WHERE Year(InvoiceDate) = 2016
  ORDER BY InvoiceID
```

	RowNumber	InvoiceID	CustomerID
1	1	61321	412
2	2	61322	919
3	3	61323	55
4	4	61324	495
5	5	61325	969
6	6	61326	919
7	7	61327	24
8	8	61328	1045
9	9	61329	92
10	10	61330	492
11	11	61331	443
12	12	61332	846



```
SELECT ROW_NUMBER() OVER (ORDER BY CustomerID) as RowNumber
      ,[InvoiceID]
      ,[CustomerID]
  FROM [Sales].[Invoices]
  WHERE Year(InvoiceDate) = 2016
  ORDER BY CustomerID
```

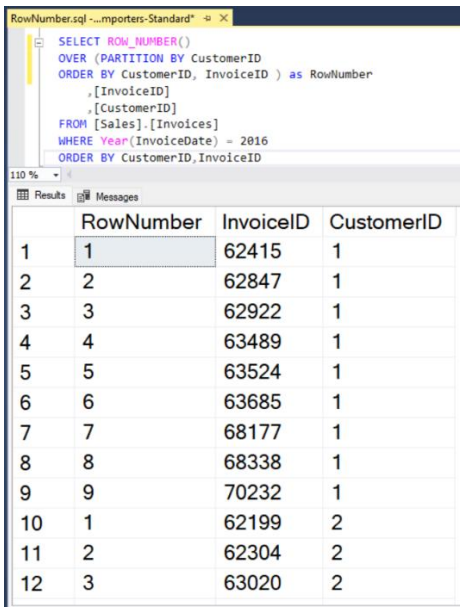
	RowNumber	InvoiceID	CustomerID
1	1	62415	1
2	2	62847	1
3	3	62922	1
4	4	63489	1
5	5	63524	1
6	6	63685	1
7	7	68177	1
8	8	68338	1
9	9	70232	1
10	10	69308	2
11	11	69455	2
12	12	68196	2
13	13	68619	2



```
SELECT ROW_NUMBER() OVER (ORDER BY CustomerID, InvoiceID) as RowNumber
      ,[InvoiceID]
      ,[CustomerID]
  FROM [Sales].[Invoices]
  WHERE Year(InvoiceDate) = 2016
  ORDER BY CustomerID,InvoiceID
```

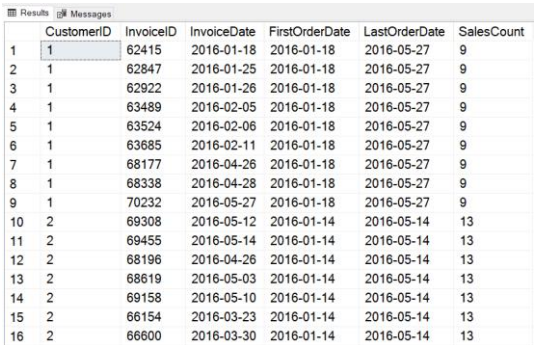
	RowNumber	InvoiceID	CustomerID
1	1	62415	1
2	2	62847	1
3	3	62922	1
4	4	63489	1
5	5	63524	1
6	6	63685	1
7	7	68177	1
8	8	68338	1
9	9	70232	1
10	10	62199	2
11	11	62304	2
12	12	63020	2
13	13	63500	2

Query executed successfully.

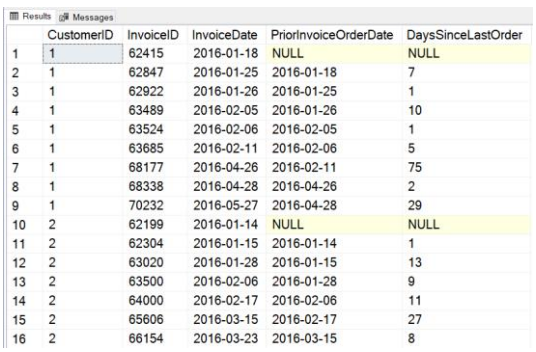


```
SELECT ROW_NUMBER()
OVER (PARTITION BY CustomerID
ORDER BY CustomerID, InvoiceID ) as RowNumber
      ,[InvoiceID]
      ,[CustomerID]
  FROM [Sales].[Invoices]
  WHERE Year(InvoiceDate) = 2016
  ORDER BY CustomerID,InvoiceID
```

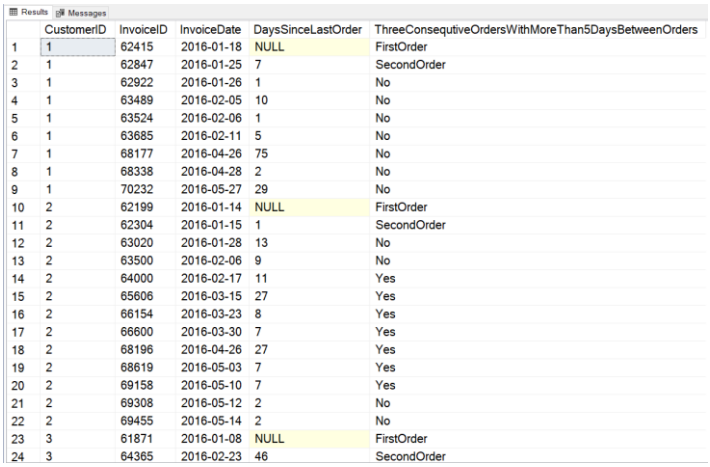
	RowNumber	InvoiceID	CustomerID
1	1	62415	1
2	2	62847	1
3	3	62922	1
4	4	63489	1
5	5	63524	1
6	6	63685	1
7	7	68177	1
8	8	68338	1
9	9	70232	1
10	1	62199	2
11	2	62304	2
12	3	63020	2



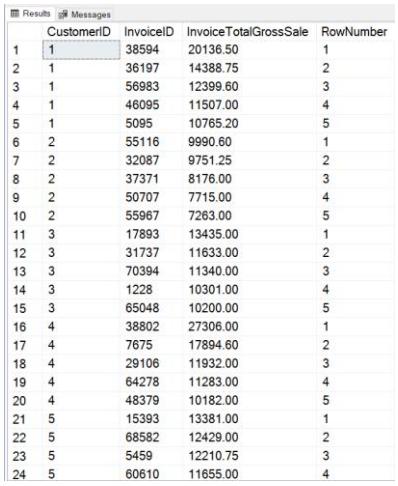
	CustomerID	InvoiceID	InvoiceDate	FirstOrderDate	LastOrderDate	SalesCount
1	1	62415	2016-01-18	2016-01-18	2016-05-27	9
2	1	62847	2016-01-25	2016-01-18	2016-05-27	9
3	1	62922	2016-01-26	2016-01-18	2016-05-27	9
4	1	63489	2016-02-05	2016-01-18	2016-05-27	9
5	1	63524	2016-02-06	2016-01-18	2016-05-27	9
6	1	63685	2016-02-11	2016-01-18	2016-05-27	9
7	1	68177	2016-04-26	2016-01-18	2016-05-27	9
8	1	68338	2016-04-28	2016-01-18	2016-05-27	9
9	1	70232	2016-05-27	2016-01-18	2016-05-27	9
10	2	69308	2016-05-12	2016-01-14	2016-05-14	13
11	2	69455	2016-05-14	2016-01-14	2016-05-14	13
12	2	68196	2016-04-26	2016-01-14	2016-05-14	13
13	2	68619	2016-05-03	2016-01-14	2016-05-14	13
14	2	69158	2016-05-10	2016-01-14	2016-05-14	13
15	2	66154	2016-03-23	2016-01-14	2016-05-14	13
16	2	66600	2016-03-30	2016-01-14	2016-05-14	13



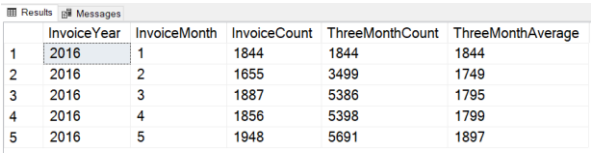
	CustomerID	InvoiceID	InvoiceDate	PriorInvoiceOrderDate	DaysSinceLastOrder
1	1	62415	2016-01-18	NULL	NULL
2	1	62847	2016-01-25	2016-01-18	7
3	1	62922	2016-01-26	2016-01-25	1
4	1	63489	2016-02-05	2016-01-26	10
5	1	63524	2016-02-06	2016-02-05	1
6	1	63685	2016-02-11	2016-02-06	5
7	1	68177	2016-04-26	2016-02-11	75
8	1	68338	2016-04-28	2016-04-26	2
9	1	70232	2016-05-27	2016-04-28	29
10	2	62199	2016-01-14	NULL	NULL
11	2	62304	2016-01-15	2016-01-14	1
12	2	63020	2016-01-28	2016-01-15	13
13	2	63500	2016-02-06	2016-01-28	9
14	2	64000	2016-02-17	2016-02-06	11
15	2	65606	2016-03-15	2016-02-17	27
16	2	66154	2016-03-23	2016-03-15	8



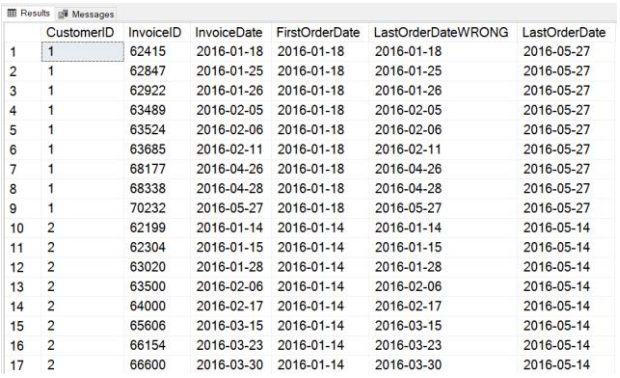
	CustomerID	InvoiceID	InvoiceDate	DaysSinceLastOrder	ThreeConsequtiveOrdersWithMoreThan5DaysBetweenOrders
1	1	62415	2016-01-18	NULL	FirstOrder
2	1	62847	2016-01-25	7	SecondOrder
3	1	62922	2016-01-26	1	No
4	1	63489	2016-02-05	10	No
5	1	63524	2016-02-06	1	No
6	1	63685	2016-02-11	5	No
7	1	68177	2016-04-26	75	No
8	1	68338	2016-04-28	2	No
9	1	70232	2016-05-27	29	No
10	2	62199	2016-01-14	NULL	FirstOrder
11	2	62304	2016-01-15	1	SecondOrder
12	2	63020	2016-01-28	13	No
13	2	63500	2016-02-06	9	No
14	2	64000	2016-02-17	11	Yes
15	2	65606	2016-03-15	27	Yes
16	2	66154	2016-03-23	8	Yes
17	2	66600	2016-03-30	7	Yes
18	2	68196	2016-04-26	27	Yes
19	2	68619	2016-05-03	7	Yes
20	2	69158	2016-05-10	7	Yes
21	2	69308	2016-05-12	2	No
22	2	69455	2016-05-14	2	No
23	3	61871	2016-01-08	NULL	FirstOrder
24	3	64365	2016-02-23	46	SecondOrder



	CustomerID	InvoiceID	InvoiceTotalGrossSale	RowNumber
1	1	38594	20136.50	1
2	1	36197	14388.75	2
3	1	56983	12399.60	3
4	1	46095	11507.00	4
5	1	5095	10765.20	5
6	2	55116	9990.60	1
7	2	32087	9751.25	2
8	2	37371	8176.00	3
9	2	50707	7715.00	4
10	2	55967	7263.00	5
11	3	17893	13435.00	1
12	3	31737	11633.00	2
13	3	70394	11340.00	3
14	3	1228	10301.00	4
15	3	65048	10200.00	5
16	4	38802	27306.00	1
17	4	7675	17894.60	2
18	4	29106	11932.00	3
19	4	64278	11283.00	4
20	4	48379	10182.00	5
21	5	15393	13381.00	1
22	5	68582	12429.00	2
23	5	5459	12210.75	3
24	5	60610	11655.00	4



	InvoiceYear	InvoiceMonth	InvoiceCount	ThreeMonthCount	ThreeMonthAverage
1	2016	1	1844	1844	1844
2	2016	2	1655	3499	1749
3	2016	3	1887	5386	1795
4	2016	4	1856	5398	1799
5	2016	5	1948	5691	1897



	CustomerID	InvoiceID	InvoiceDate	FirstOrderDate	LastOrderDateWRONG	LastOrderDate
1	1	62415	2016-01-18	2016-01-18	2016-01-18	2016-05-27
2	1	62847	2016-01-25	2016-01-18	2016-01-25	2016-05-27
3	1	62922	2016-01-26	2016-01-18	2016-01-26	2016-05-27
4	1	63489	2016-02-05	2016-01-18	2016-02-05	2016-05-27
5	1	63524	2016-02-06	2016-01-18	2016-02-06	2016-05-27
6	1	63685	2016-02-11	2016-01-18	2016-02-11	2016-05-27
7	1	68177	2016-04-26	2016-01-18	2016-04-26	2016-05-27
8	1	68338	2016-04-28	2016-01-18	2016-04-28	2016-05-27
9	1	70232	2016-05-27	2016-01-18	2016-05-27	2016-05-27
10	2	62199	2016-01-14	2016-01-14	2016-01-14	2016-05-14
11	2	62304	2016-01-15	2016-01-14	2016-01-15	2016-05-14
12	2	63020	2016-01-28	2016-01-14	2016-01-28	2016-05-14
13	2	63500	2016-02-06	2016-01-14	2016-02-06	2016-05-14
14	2	64000	2016-02-17	2016-01-14	2016-02-17	2016-05-14
15	2	65606	2016-03-15	2016-01-14	2016-03-15	2016-05-14
16	2	66154	2016-03-23	2016-01-14	2016-03-23	2016-05-14
17	2	66600	2016-03-30	2016-01-14	2016-03-30	2016-05-14

Results Messages

	CustomerID	InvoiceID	InvoiceDate	FirstOrderDate	LastOrderDateWRONG	LastOrderDate
1	1	62415	2016-01-18	2016-01-18	2016-01-18	2016-05-27
2	1	62847	2016-01-25	2016-01-18	2016-01-25	2016-05-27
3	1	62922	2016-01-26	2016-01-18	2016-01-26	2016-05-27
4	1	63489	2016-02-05	2016-01-18	2016-02-05	2016-05-27
5	1	63524	2016-02-06	2016-01-18	2016-02-06	2016-05-27
6	1	63685	2016-02-11	2016-01-18	2016-02-11	2016-05-27
7	1	68177	2016-04-26	2016-01-18	2016-04-26	2016-05-27
8	1	68338	2016-04-28	2016-01-18	2016-04-28	2016-05-27
9	1	70232	2016-05-27	2016-01-18	2016-05-27	2016-05-27
10	2	62199	2016-01-14	2016-01-14	2016-01-14	2016-05-14
11	2	62304	2016-01-15	2016-01-14	2016-01-15	2016-05-14
12	2	63020	2016-01-28	2016-01-14	2016-01-28	2016-05-14
13	2	63500	2016-02-06	2016-01-14	2016-02-06	2016-05-14
14	2	64000	2016-02-17	2016-01-14	2016-02-17	2016-05-14
15	2	65606	2016-03-15	2016-01-14	2016-03-15	2016-05-14
16	2	66154	2016-03-23	2016-01-14	2016-03-23	2016-05-14
17	2	66600	2016-03-30	2016-01-14	2016-03-30	2016-05-14
18	2	68196	2016-04-26	2016-01-14	2016-04-26	2016-05-14
19	2	68619	2016-05-03	2016-01-14	2016-05-03	2016-05-14
20	2	69158	2016-05-10	2016-01-14	2016-05-10	2016-05-14
21	2	69308	2016-05-12	2016-01-14	2016-05-12	2016-05-14
22	2	69455	2016-05-14	2016-01-14	2016-05-14	2016-05-14
23	3	61871	2016-01-08	2016-01-08	2016-01-08	2016-05-30
24	3	64365	2016-02-23	2016-01-08	2016-02-23	2016-05-30
25	3	65000	2016-03-02	2016-01-08	2016-03-02	2016-05-30
26	3	65048	2016-03-03	2016-01-08	2016-03-03	2016-05-30

Results Messages

	OrderYear	OrderMonth	TotalOrders	PreviousYearsOrder	YOY_Change
1	2014	1	4202578.80	3824842.85	9.88%
2	2014	2	3572744.40	2821282.20	26.64%
3	2014	3	3955257.55	3966078.10	-0.27%
4	2014	4	4212856.25	4155710.05	1.38%
5	2014	5	4753224.10	4562830.35	4.17%
6	2014	6	4427573.80	4150098.60	6.69%
7	2014	7	4919791.85	4502741.85	9.26%
8	2014	8	4197257.40	3601220.60	16.55%
9	2014	9	3973877.85	3916003.25	1.48%
10	2014	10	4606478.45	3879872.45	18.73%
11	2014	11	4157270.55	3819809.10	8.83%
12	2014	12	4513092.40	3728103.40	21.06%
13	2015	1	4556065.25	4202578.80	8.41%
14	2015	2	4307819.25	3572744.40	20.57%
15	2015	3	4644642.35	3955257.55	17.43%
16	2015	4	5222594.85	4212856.25	23.97%
17	2015	5	4636628.45	4753224.10	-2.45%

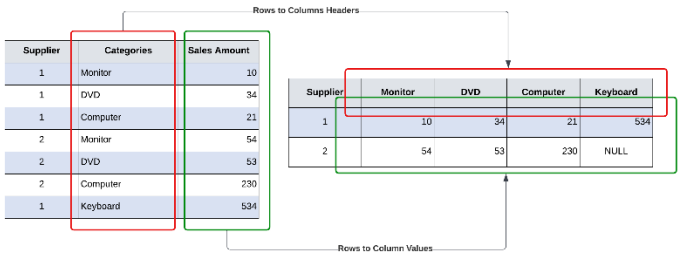
Chapter 7: Reshaping Your Data with Advanced Techniques

Results Messages

	TotalPurchaseOrders	Suppliers	PurchaseOrder
1	Total Purchase Orders	A Datum Corporation	5
2	Total Purchase Orders	Contoso, Ltd.	1
3	Total Purchase Orders	Fabrikam, Inc.	1055
4	Total Purchase Orders	Graphic Design Institute	13
5	Total Purchase Orders	Litware, Inc.	985
6	Total Purchase Orders	Northwind Electric Cars	10
7	Total Purchase Orders	The Phone Company	5

Results Messages

	EmpNode	Text_EmpNode	EmpLevel	EmpID	EmpName	EmpTitle
1	0x	/	0	0	John	Manager



Results Messages

	SupplierID	CTPurchaseOrder
1	1	5
2	2	1
3	4	1055
4	5	13
5	7	985
6	10	10
7	12	5

Results Messages

	TotalPurchaseOrders	SalesPer1	SalesPer2	SalesPer3	SalesPer4	SalesPer5	SalesPer6	SalesPer7
1	Total Purchase Orders	5	1	1055	13	985	10	5

Results Messages

	TotalPurchaseOrders	A Datum Corporation	Contoso, Ltd.	Fabrikam, Inc.	Graphic Design Institute	Litware, Inc.	Northwind Electric Cars	The Phone Company
1	Total Purchase Orders	5	1	1055	13	985	10	5

Messages

```
[A Datum Corporation], [Contoso, Ltd.], [Fabrikam, Inc.], [Graphic Design Institute], [Litware, Inc.], [Northwind Electric Cars], [The Phone Company]

Completion time: 2023-02-20T13:35:04.4733395-06
```

Results Messages

	TotalPurchaseOrders	A Datum Corporation	Contoso, Ltd.	Fabrikam, Inc.	Graphic Design Institute	Litware, Inc.	Northwind Electric Cars	The Phone Company
1	Total Purchase Orders	5	1	1055	13	985	10	5

Results Messages

	TotalPurchaseOrders	A Datum Corporation	Contoso, Ltd.	Fabrikam, Inc.	Graphic Design Institute	Litware, Inc.	Northwind Electric Cars	The Phone Company
1	Total Purchase Orders	5	1	1055	13	985	10	5

Results Messages

	EmpNode	Text_EmpNode	EmpLevel	EmpID	EmpName	EmpTitle
1	0x	/	0	0	John	Manager
2	0x58	/1/	1	17	Jim	Assistant Manager
3	0x68	/2/	1	24	Kim	Assistant Manager

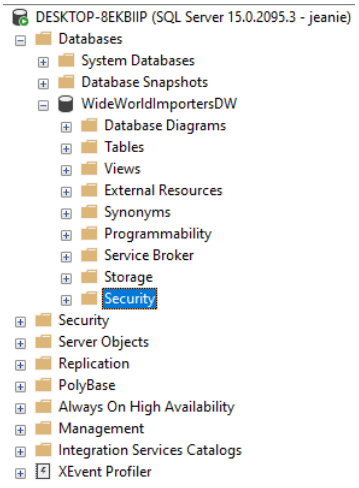
Results Messages

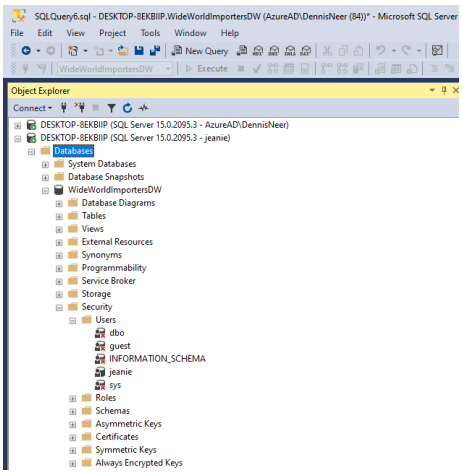
	EmpNode	Text_EmpNode	EmpLevel	EmpID	EmpName	EmpTitle
1	0x	/	0	0	John	Manager
2	0x58	/1/	1	17	Jim	Assistant Manager
3	0x5AC0	/1/1/	2	32	Jack	Team Member
4	0x5B40	/1/2/	2	25	Frank	Team Member
5	0x68	/2/	1	24	Kim	Assistant Manager

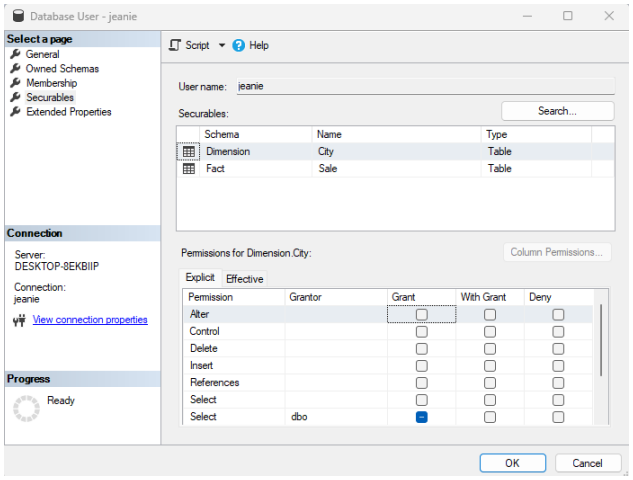
Chapter 8: Impact of SQL Security on Query Results

City	Region	Quanity	Profit
Abbottsburg	Americas	17359	173946.95
Absecon	Americas	12415	129358.35
Accomac	Americas	16472	157768.4
Aceitunas	Americas	12693	119283
Airport Drive	Americas	16445	162500
Akhiok	Americas	30999	259554.3
Alcester	Americas	12802	127040.25
Alden Bridge	Americas	14645	152137.85
Alstead	Americas	12073	106146.95
Amado	Americas	14722	136717.8
Amanda Park	Americas	12221	117443.85
Andrix	Americas	14664	130710
Annamoriah	Americas	15326	139498.5
Antares	Americas	15363	147561.8
Antonito	Americas	12873	113055.75
Arbor Vitae	Americas	14334	135056.25

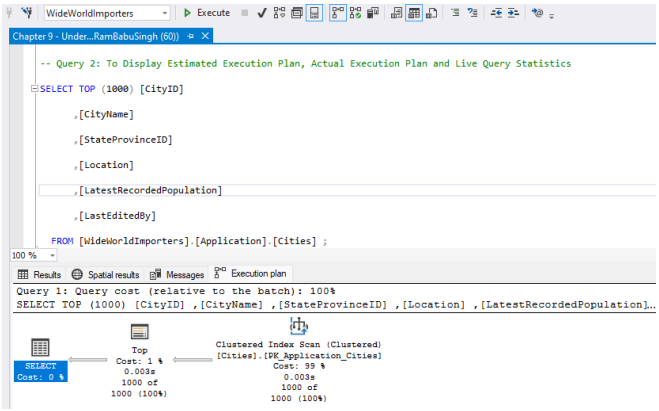
City	Region	Quanity	Profit
Absecon	Americas	12415	129358.35
Accomac	Americas	16472	157768.4
Airport Drive	Americas	16445	162500
Akhiok	Americas	30999	259554.3
Alcester	Americas	12802	127040.25
Alden Bridge	Americas	14645	152137.85
Amado	Americas	14722	136717.8
Amanda Park	Americas	12221	117443.85
Andrix	Americas	14664	130710
Annamoriah	Americas	15326	139498.5
Antares	Americas	15363	147561.8
Antonito	Americas	12873	113055.75
Arbor Vitae	Americas	14334	135056.25

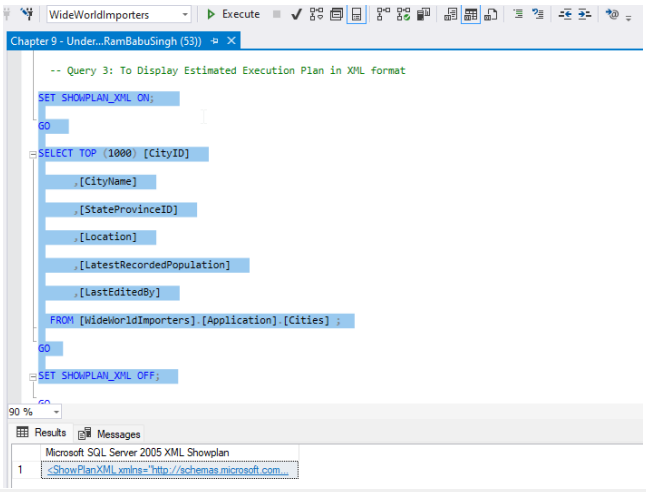


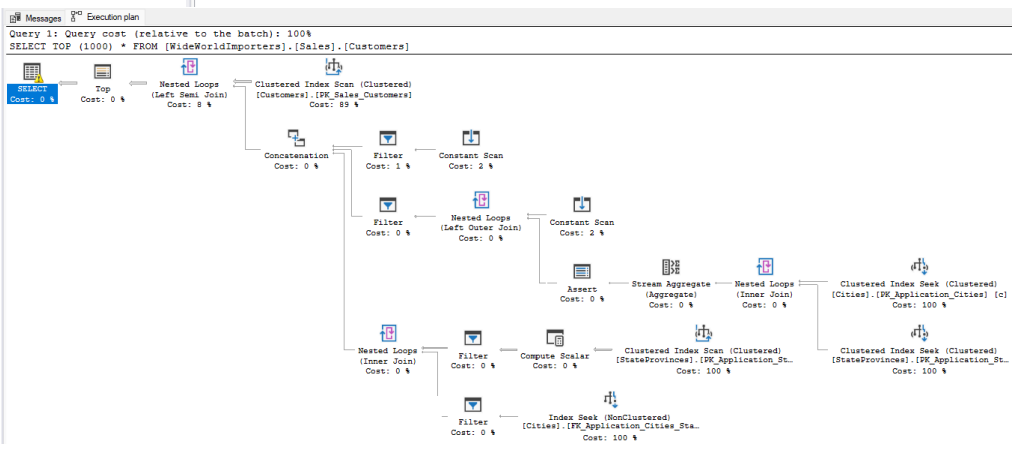


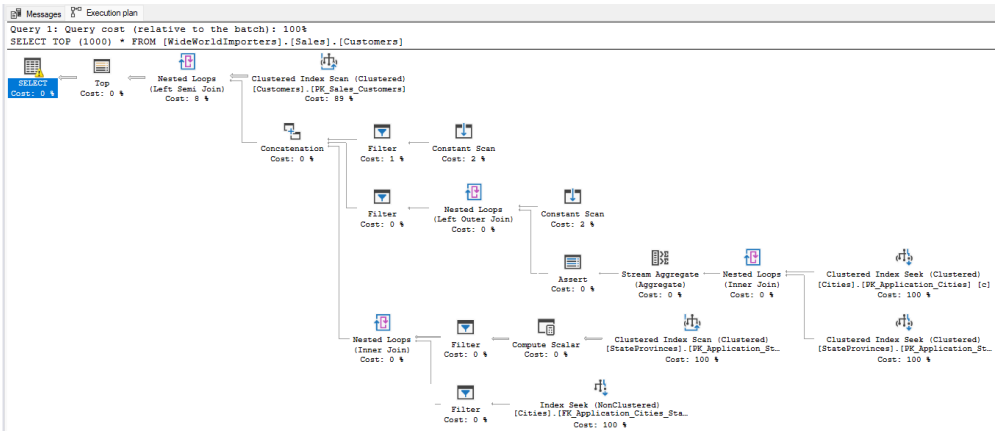


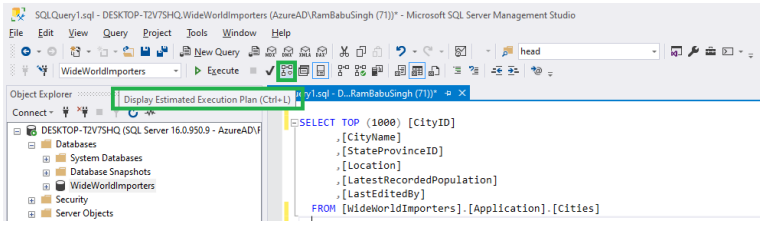
Chapter 9: Understanding Query Plans

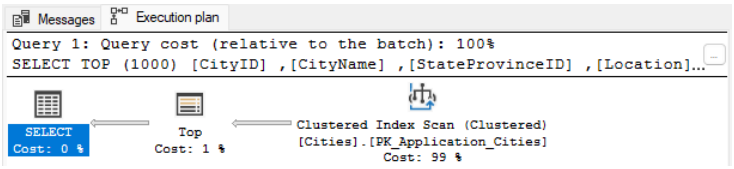


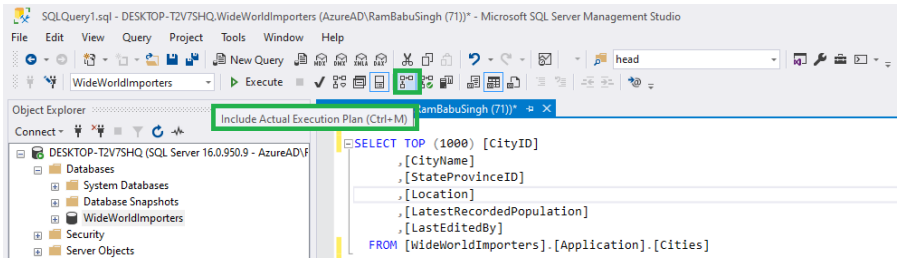


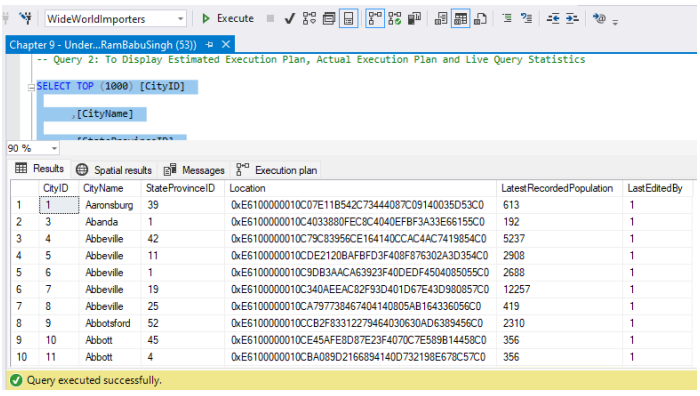




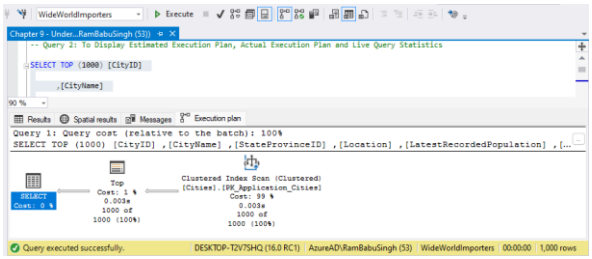


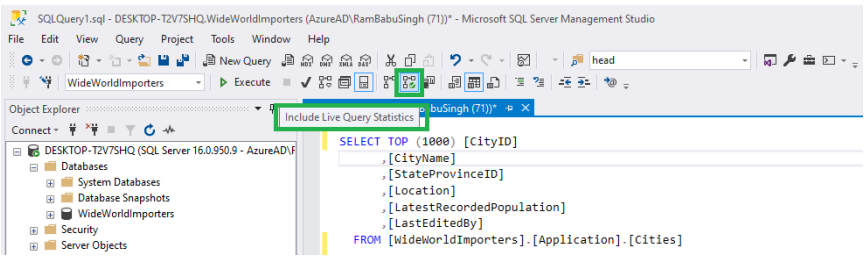


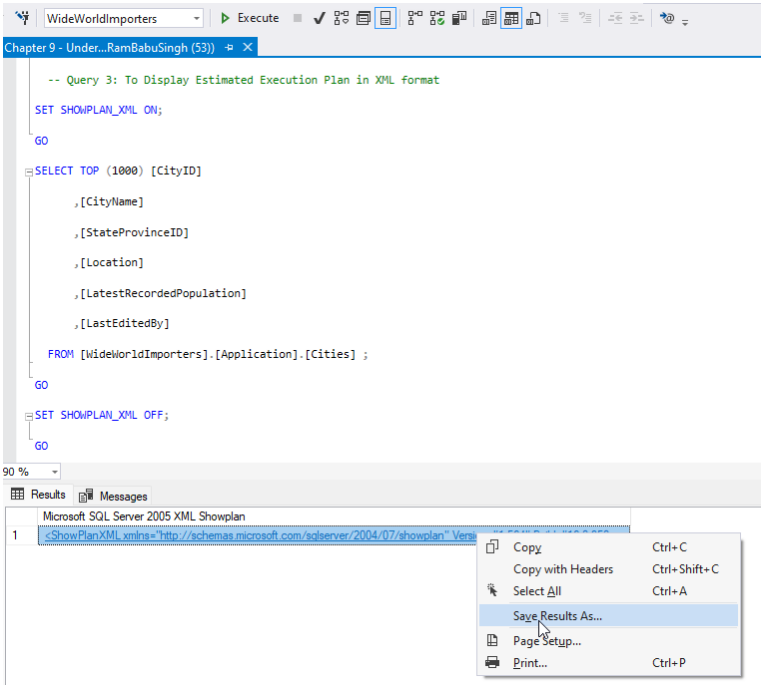


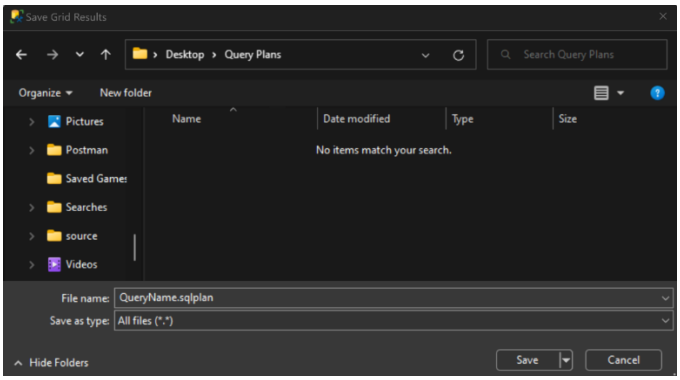


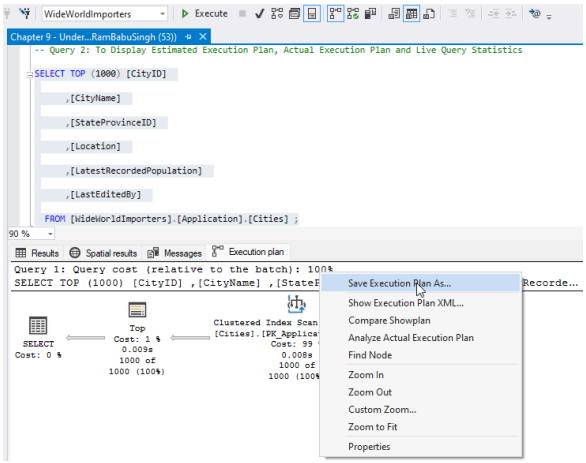
	CityID	CityName	StateProvinceID	Location	LatestRecordedPopulation	LastEditedBy
1	1	Aaronsburg	39	0xE6100000010C07E11B542C73444087C09140035D53C0	613	1
2	3	Abanda	1	0xE6100000010C4033880FEC8C4040EFBF3A33E66155C0	192	1
3	4	Abbeville	42	0xE6100000010C79C83956CE164140CCAC4AC7419854C0	5237	1
4	5	Abbeville	11	0xE6100000010CDE2120BAFBFD3F408F876302A3D354C0	2908	1
5	6	Abbeville	1	0xE6100000010C9DB3AACA63923F40DEDF4504085055C0	2688	1
6	7	Abbeville	19	0xE6100000010C340AEEAC82F93D401D67E43D980857C0	12257	1
7	8	Abbeville	25	0xE6100000010CA797738467404140805AB164336056C0	419	1
8	9	Abbotsford	52	0xE6100000010CCB2F8331227946403063OAD6389456C0	2310	1
9	10	Abbott	45	0xE6100000010CE45AFE8D87E23F4070C7E589B14458C0	356	1
10	11	Abbott	4	0xE6100000010CBA089D2166894140D732198E678C57C0	356	1

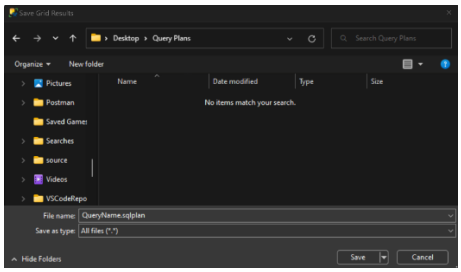

WideWorldImporters
Execute
Chapter 9 - Under...RamBabuSingh (53))
-- Query 2: To Display Estimated Execution Plan, Actual Execution Plan and Live Query Statistics
SELECT TOP (1000) [CityID]
,[CityName]
Results
Spatial results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[LatestRecordedPopulation] ,[...
SELECT
Top
Clustered Index Scan (Clustered)
[Cities].[PK_Application_Cities]
Query executed successfully.
DESKTOP-T2V7SHQ (16.0 RC1)
AzureAD\RamBabuSingh (53)
WideWorldImporters
00:00:00
1,000 rows

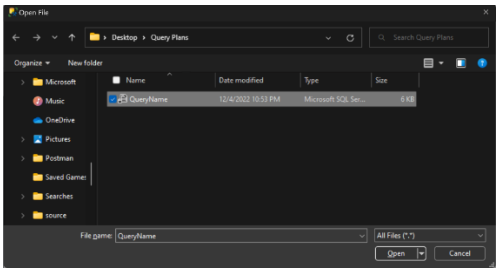

SQLQuery1.sql - DESKTOP-T2V7SHQ.WideWorldImporters (AzureAD\RamBabuSingh (71))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
WideWorldImporters
Execute
Object Explorer
Connect
Include Live Query Statistics
DESKTOP-T2V7SHQ (SQL Server 16.0.950.9 - AzureAD\
Databases
System Databases
Database Snapshots
WideWorldImporters
Security
Server Objects
SELECT TOP (1000) [CityID]
,[CityName]
,[StateProvinceID]
,[Location]
,[LatestRecordedPopulation]
,[LastEditedBy]
FROM [WideWorldImporters].[Application].[Cities]

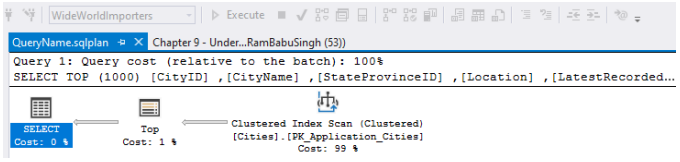

WideWorldImporters
Execute
Chapter 9 - Under...RamBabuSingh (53))
-- Query 3: To Display Estimated Execution Plan in XML format
SET SHOWPLAN_XML ON;
GO
SELECT TOP (1000) [CityID]
,[CityName]
,[StateProvinceID]
,[Location]
,[LatestRecordedPopulation]
,[LastEditedBy]
FROM [WideWorldImporters].[Application].[Cities] ;
GO
SET SHOWPLAN_XML OFF;
GO
90 %
Results
Messages
Microsoft SQL Server 2005 XML Showplan
1
<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan"
Copy Ctrl+C
Copy with Headers Ctrl+Shift+C
Select All Ctrl+A
Save Results As...
Page Setup...
Print... Ctrl+P

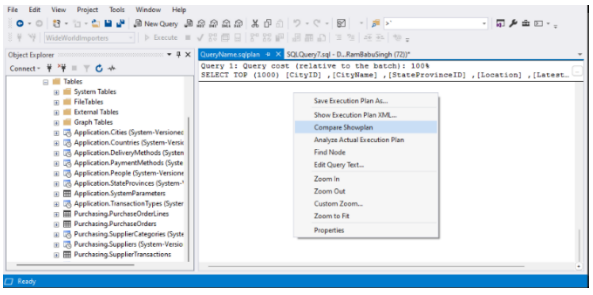

Save Grid Results
Desktop › Query Plans
Search Query Plans
Organize
New folder
Pictures
Postman
Saved Games
Searches
source
Videos
Name
Date modified
Type
Size
No items match your search.
File name: QueryName.sqlplan
Save as type: All files (*.*)
Hide Folders
Save
Cancel

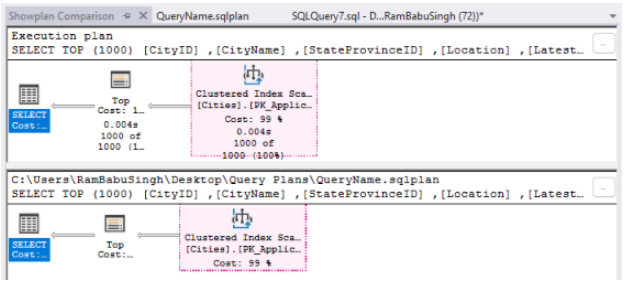

WideWorldImporters
Execute
Chapter 9 - Under...RamBabuSingh (53))
-- Query 2: To Display Estimated Execution Plan, Actual Execution Plan and Live Query Statistics
SELECT TOP (1000) [CityID]
,[CityName]
,[StateProvinceID]
,[Location]
,[LatestRecordedPopulation]
,[LastEditedBy]
FROM [WideWorldImporters].[Application].[Cities] ;
Results
Spatial results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
Save Execution Plan As...
Show Execution Plan XML...
Compare Showplan
Analyze Actual Execution Plan
Find Node
Zoom In
Zoom Out
Custom Zoom...
Zoom to Fit
Properties
SELECT
Cost: 0 %
Top
Cost: 1 %
0.009s
1000 of
1000 (100%)
Clustered Index Scan
[Cities].[PK_Applica
Cost: 99
0.008s
1000 of
1000 (100%

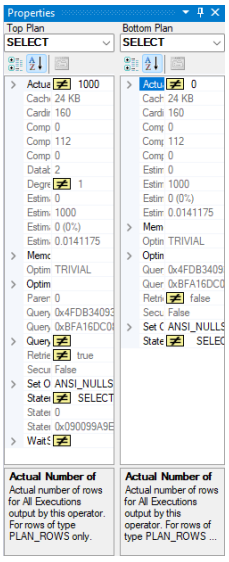

Save Grid Results
Desktop › Query Plans
Search Query Plans
Organize
New folder
Name
Date modified
Type
Size
No items match your search.
Pictures
Postman
Saved Games
Searches
source
Videos
VSCodeRepo
File name: QueryName.sqlplan
Save as type: All files (*.*)
Hide Folders
Save
Cancel

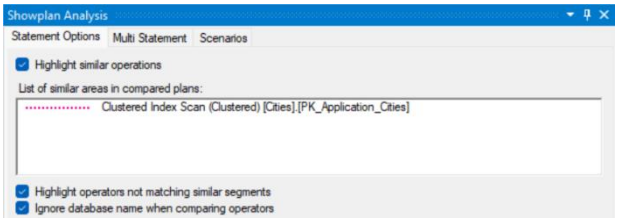

Open File
Desktop › Query Plans
Search Query Plans
Organize
New folder
Name
Date modified
Type
Size
Microsoft
Music
OneDrive
Pictures
Postman
Saved Games
Searches
source
QueryName
12/4/2022 10:53 PM
Microsoft SQL Ser...
6 KB
File name: QueryName
All Files (*.*)
Open
Cancel

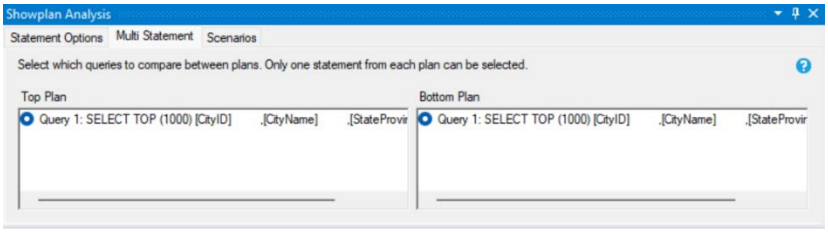

WideWorldImporters
Execute
QueryName.sqlplan
Chapter 9 - Under...RamBabuSingh (53))
Query 1: Query cost (relative to the batch): 100%
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[LatestRecorded...
SELECT
Cost: 0 %
Top
Cost: 1 %
Clustered Index Scan (Clustered)
[Cities].[PK_Application_Cities]
Cost: 99 %

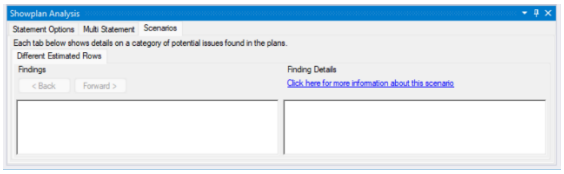

File Edit View Project Tools Window Help
New Query
Object Explorer
Connect
Tables
System Tables
FileTables
External Tables
Graph Tables
Application.Cities (System-Versioned)
Application.Countries (System-Versioned)
Application.DeliveryMethods (System-Versioned)
Application.PaymentMethods (System-Versioned)
Application.People (System-Versioned)
Application.StateProvinces (System-Versioned)
Application.SystemParameters
Application.TransactionTypes (System-Versioned)
Purchasing.PurchaseOrderLines
Purchasing.PurchaseOrders
Purchasing.SupplierCategories (System-Versioned)
Purchasing.Suppliers (System-Versioned)
Purchasing.SupplierTransactions
QueryName.sqlplan
SQLQuery7.sql - D...RamBabuSingh (72))*
Query 1: Query cost (relative to the batch): 100%
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[Latest...
Save Execution Plan As...
Show Execution Plan XML...
Compare Showplan
Analyze Actual Execution Plan
Find Node
Edit Query Text...
Zoom In
Zoom Out
Custom Zoom...
Zoom to Fit
Properties
Ready

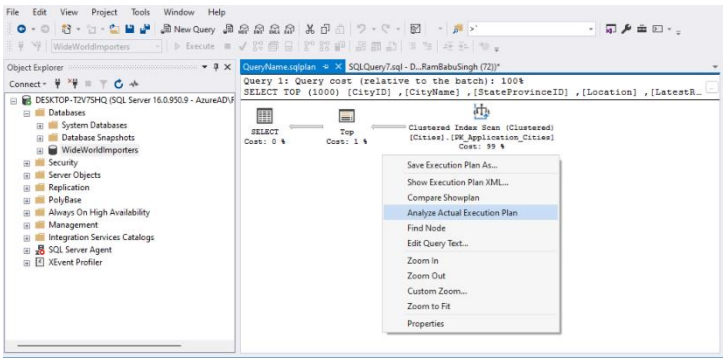

Showplan Comparison
QueryName.sqlplan
SQLQuery7.sql - D...RamBabuSingh (72))*
Execution plan
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[Latest...
SELECT
Cost:...
Top
Cost: 1...
0.004s
1000 of
1000 (1...
Clustered Index Sca...
[Cities].[PK_Applic...
Cost: 99 %
0.004s
1000 of
1000 (100%)
C:\Users\RamBabuSingh\Desktop\Query Plans\QueryName.sqlplan
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[Latest...
SELECT
Cost:...
Top
Cost:...
Clustered Index Sca...
[Cities].[PK_Applic...
Cost: 99 %

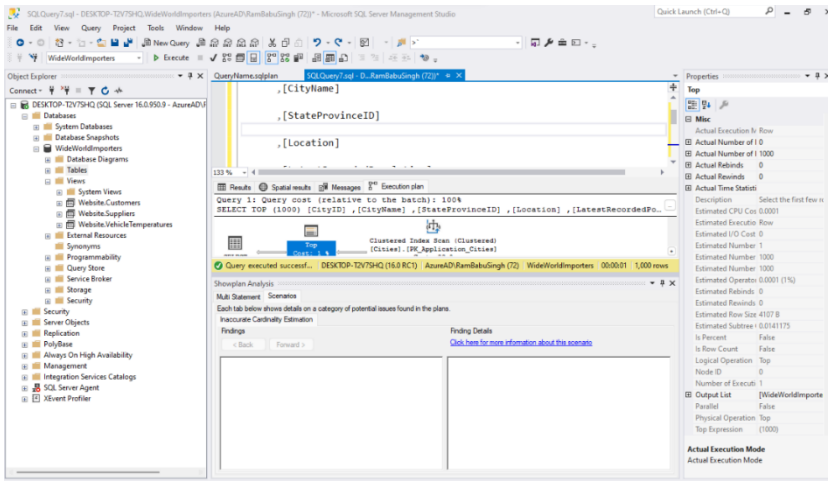

Properties
Top Plan
Bottom Plan
SELECT
SELECT
Actua 1000
Cach 24 KB
Cardi 160
Comp 0
Comp 112
Comp 0
Datat 2
Degre 1
Estim 0
Estim 1000
Estim 0 (0%)
Estim 0.0141175
Memc
Optim TRIVIAL
Optim
Parer 0
Query 0x4FDB34093
Query 0xBFA16DC0
Query
Retrie true
Secur False
Set O ANSI_NULLS
State SELECT
State 0
State 0x090099A9E
Wait
Actual Number of
Actual number of rows for All Executions output by this operator. For rows of type PLAN_ROWS only.
Actual Number of
Actual number of rows for All Executions output by this operator. For rows of type PLAN_ROWS ...

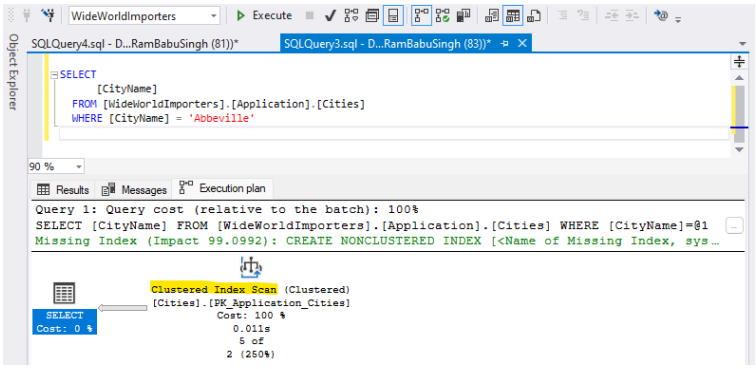

Showplan Analysis
Statement Options
Multi Statement
Scenarios
Highlight similar operations
List of similar areas in compared plans:
Clustered Index Scan (Clustered) [Cities].[PK_Application_Cities]
Highlight operators not matching similar segments
Ignore database name when comparing operators

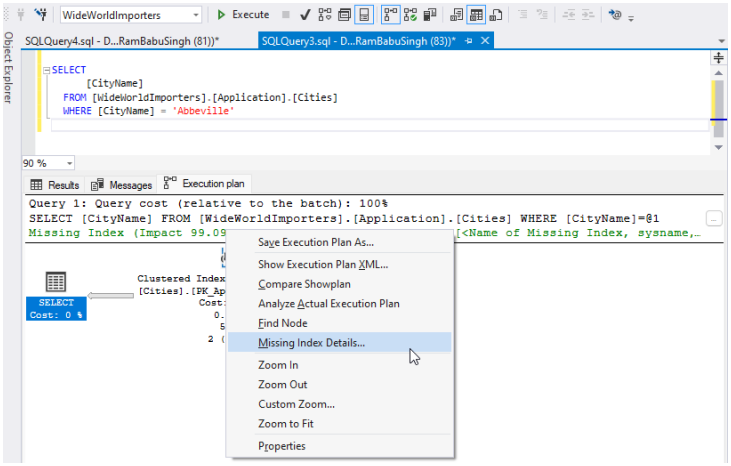

Showplan Analysis
Statement Options
Multi Statement
Scenarios
Select which queries to compare between plans. Only one statement from each plan can be selected.
Top Plan
Query 1: SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvir
Bottom Plan
Query 1: SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvir

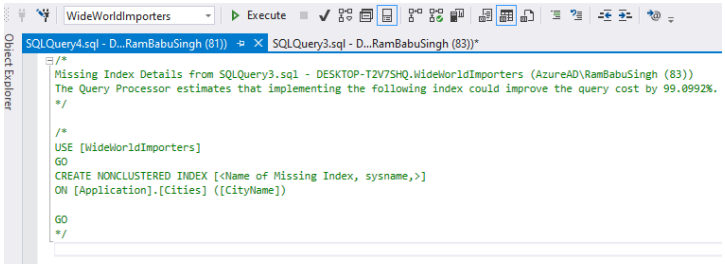

Showplan Analysis
Statement Options
Multi Statement
Scenarios
Each tab below shows details on a category of potential issues found in the plans.
Different Estimated Rows
Findings
Finding Details
Click here for more information about this scenario

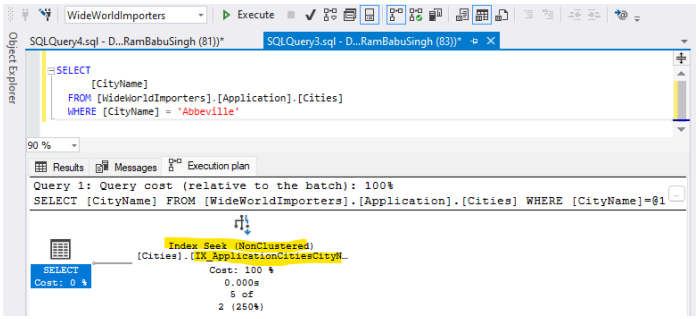

File Edit View Project Tools Window Help
Object Explorer
Connect
DESKTOP-TZV7SHQ (SQL Server 16.0.950.9 - AzureAD\
Databases
System Databases
Database Snapshots
WideWorldImporters
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
QueryName.sqlplan
SQLQuery7.sql - D...RamBabuSingh (72))*
Query 1: Query cost (relative to the batch): 100%
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[LatestR...
SELECT
Cost: 0 %
Top
Cost: 1 %
Clustered Index Scan (Clustered)
[Cities].[PK_Application_Cities]
Cost: 99 %
Save Execution Plan As...
Show Execution Plan XML...
Compare Showplan
Analyze Actual Execution Plan
Find Node
Edit Query Text...
Zoom In
Zoom Out
Custom Zoom...
Zoom to Fit
Properties


SQLQuery7.sql - DESKTOP-TZV7SHQ.WideWorldImporters (AzureAD\RamBabuSingh (72))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
Object Explorer
Connect
DESKTOP-TZV7SHQ (SQL Server 16.0.950.9 - AzureAD\
Databases
System Databases
Database Snapshots
WideWorldImporters
Database Diagrams
Tables
Views
System Views
Website.Customers
Website.Suppliers
Website.VehicleTemperatures
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
QueryName.sqlplan
SQLQuery7.sql - D...RamBabuSingh (72))*
,[CityName]
,[StateProvinceID]
,[Location]
Query 1: Query cost (relative to the batch): 100%
SELECT TOP (1000) [CityID] ,[CityName] ,[StateProvinceID] ,[Location] ,[LatestRecordedPo...
Clustered Index Scan (Clustered)
[Cities].[PK_Application_Cities]
Query executed successf...
DESKTOP-TZV7SHQ (16.0 RC1)
AzureAD\RamBabuSingh (72)
WideWorldImporters
00:00:01
1,000 rows
Showplan Analysis
Multi Statement
Scenarios
Each tab below shows details on a category of potential issues found in the plans.
Inaccurate Cardinality Estimation
Findings
Finding Details
Click here for more information about this scenario
Properties
Top
Misc
Actual Execution Mode
Actual Execution Mode

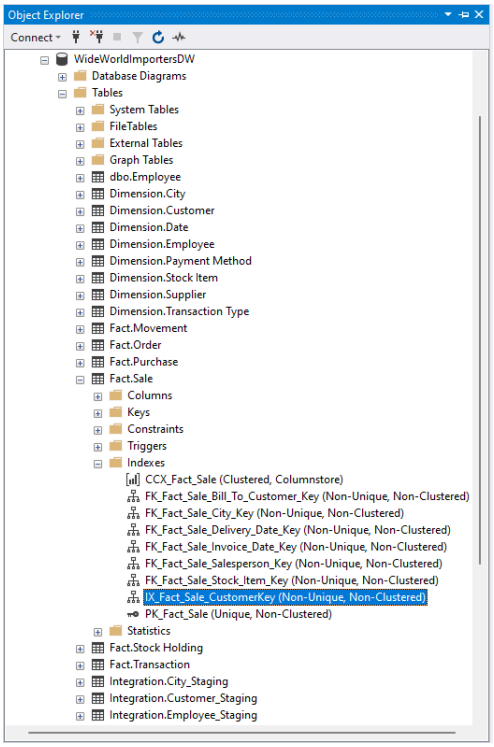

WideWorldImporters
Execute
SQLQuery4.sql - D...RamBabuSingh (81))*
SQLQuery3.sql - D...RamBabuSingh (83))*
Object Explorer
SELECT
[CityName]
FROM [WideWorldImporters].[Application].[Cities]
WHERE [CityName] = 'Abbeville'
90 %
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [CityName] FROM [WideWorldImporters].[Application].[Cities] WHERE [CityName]=@1
Missing Index (Impact 99.0992): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sys…
SELECT
Cost: 0 %
Clustered Index Scan (Clustered)
[Cities].[PK_Application_Cities]
Cost: 100 %
0.011s
5 of
2 (250%)


WideWorldImporters
Execute
SQLQuery4.sql - D...RamBabuSingh (81))*
SQLQuery3.sql - D...RamBabuSingh (83))*
Object Explorer
SELECT
[CityName]
FROM [WideWorldImporters].[Application].[Cities]
WHERE [CityName] = 'Abbeville'
90 %
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [CityName] FROM [WideWorldImporters].[Application].[Cities] WHERE [CityName]=@1
Missing Index (Impact 99.09
[<Name of Missing Index, sysname,…
SELECT
Cost: 0 %
Clustered Index
[Cities].[PK_Ap
Cost:
Save Execution Plan As...
Show Execution Plan XML...
Compare Showplan
Analyze Actual Execution Plan
Find Node
Missing Index Details...
Zoom In
Zoom Out
Custom Zoom...
Zoom to Fit
Properties


WideWorldImporters
Execute
SQLQuery4.sql - D...RamBabuSingh (81))
SQLQuery3.sql - D...RamBabuSingh (83))*
Object Explorer
/*
Missing Index Details from SQLQuery3.sql - DESKTOP-T2V7SHQ.WideWorldImporters (AzureAD\RamBabuSingh (83))
The Query Processor estimates that implementing the following index could improve the query cost by 99.0992%.
*/
/*
USE [WideWorldImporters]
GO
CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>]
ON [Application].[Cities] ([CityName])
GO
*/


WideWorldImporters
Execute
SQLQuery4.sql - D...RamBabuSingh (81))*
SQLQuery3.sql - D...RamBabuSingh (83))*
Object Explorer
SELECT
[CityName]
FROM [WideWorldImporters].[Application].[Cities]
WHERE [CityName] = 'Abbeville'
90 %
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [CityName] FROM [WideWorldImporters].[Application].[Cities] WHERE [CityName]=@1
SELECT
Cost: 0 %
Index Seek (NonClustered)
[Cities].[IX_ApplicationCitiesCityN…
Cost: 100 %
0.000s
5 of
2 (250%)

Chapter 10: Understanding the Impact of Indexes on Your Query Design




```
CREATE CLUSTERED INDEX [PK_Employee_EmployeeID] ON dbo.[Employee]
(
EmployeeID ASC
);
```

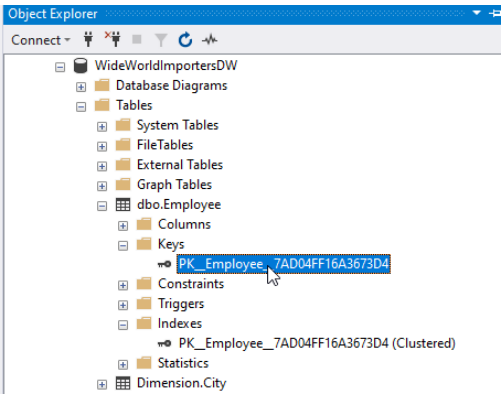
The index or statistics with name 'PK_Employee_EmployeeID' already exists on table or view 'dbo.Employee'.

```
CREATE CLUSTERED INDEX [PK_Employee_EmployeeID] ON dbo.[Employee]
(
EmployeeID ASC
);
```

100 %

Messages

```
Msg 1913, Level 16, State 1, Line 94
The operation failed because an index or statistics with name 'PK_Employee_EmployeeID' already exists on table 'dbo.Employee'.
```



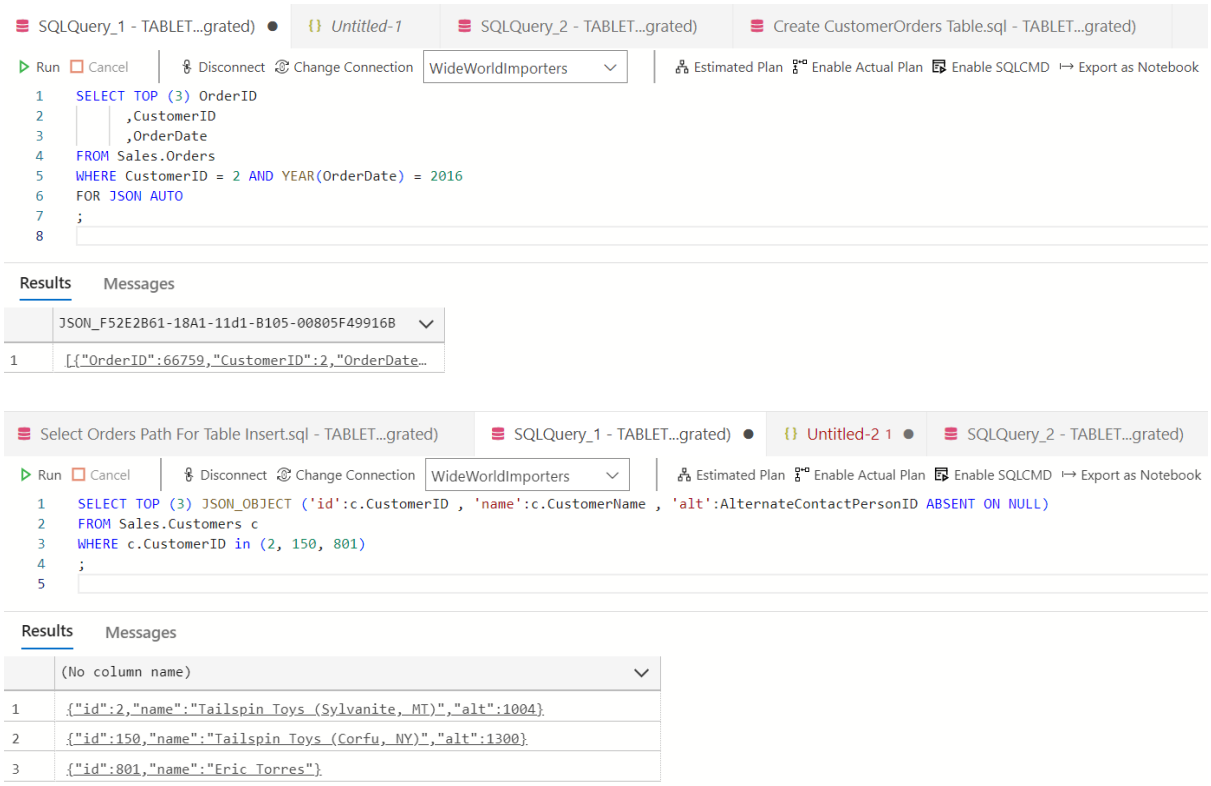
```
Query 12: query to get list of fragemented indexes with fragmentation in the WideWorldImportersDW databases
*/

SELECT OBJECT_NAME(IND.OBJECT_ID) AS [Table Name],

IND.NAME AS [Index Name], PS.INDEX_TYPE_DESC AS [Index Type],

PS.AVG_FRAGMENTATION_IN_PERCENT [Avg Fragmentation]

FROM SYS.DM_DB_INDEX_PHYSICAL_STATS(DB_ID(), NULL, NULL, NULL, NULL) PS

INNER JOIN SYS.INDEXES IND

ON IND.OBJECT_ID = PS.OBJECT_ID

AND IND.INDEX_ID = PS.INDEX_ID

WHERE PS.AVG_FRAGMENTATION_IN_PERCENT > 0
```

91 %

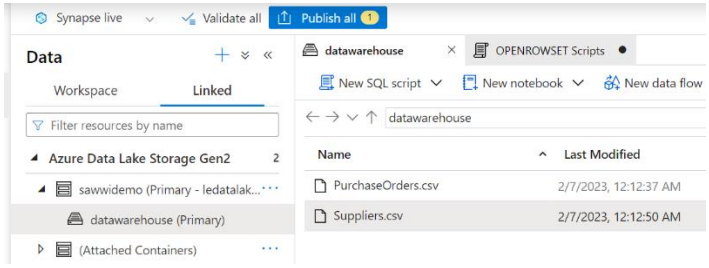
Results | Messages

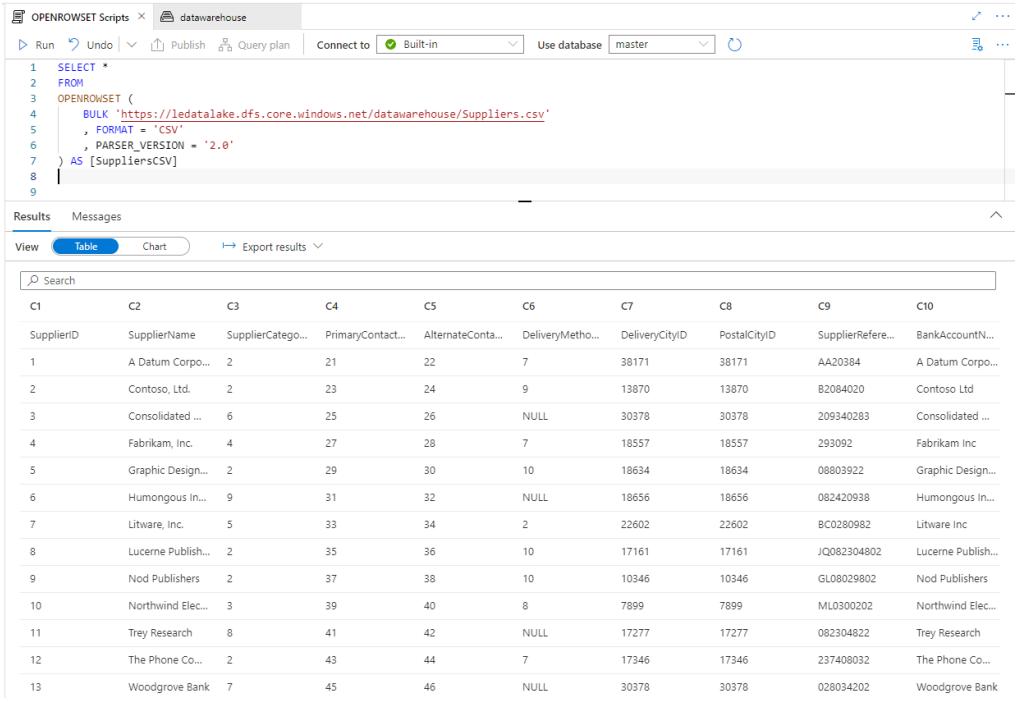
	Table Name	Index Name	Index Type	Avg Fragmentation
1	Customer	IX_Dimension_Customer_WWICustomerID	NONCLUSTERED INDEX	50
2	Stock Item	IX_Dimension_Stock_Item_WWIStockItemID	NONCLUSTERED INDEX	33.3333333333333
3	Purchase	FK_Fact_Purchase_Date_Key	NONCLUSTERED INDEX	33.3333333333333
4	Employee	PK_Dimension_Employee	CLUSTERED INDEX	33.3333333333333
5	Purchase	FK_Fact_Purchase_Supplier_Key	NONCLUSTERED INDEX	25
6	Purchase	FK_Fact_Purchase_Stock_Item_Key	NONCLUSTERED INDEX	25
7	Purchase	PK_Fact_Purchase	NONCLUSTERED INDEX	25
8	Purchase	FK_Fact_Purchase_Date_Key	NONCLUSTERED INDEX	14.2857142857143
9	Purchase	FK_Fact_Purchase_Date_Key	NONCLUSTERED INDEX	14.2857142857143
10	Purchase	FK_Fact_Purchase_Date_Key	NONCLUSTERED INDEX	14.2857142857143
11	Purchase	FK_Fact_Purchase_Stock_Item_Key	NONCLUSTERED INDEX	12.5
12	Purchase	FK_Fact_Purchase_Stock_Item_Key	NONCLUSTERED INDEX	12.5
13	Purchase	FK_Fact_Purchase_Stock_Item_Key	NONCLUSTERED INDEX	12.5
14	Purchase	FK_Fact_Purchase_Supplier_Key	NONCLUSTERED INDEX	12.5
15	Purchase	FK_Fact_Purchase_Supplier_Key	NONCLUSTERED INDEX	12.5
16	Purchase	FK_Fact_Purchase_Supplier_Key	NONCLUSTERED INDEX	12.5
17	Purchase	PK_Fact_Purchase	NONCLUSTERED INDEX	11.1111111111111
18	Purchase	PK_Fact_Purchase	NONCLUSTERED INDEX	11.1111111111111
19	Purchase	PK_Fact_Purchase	NONCLUSTERED INDEX	10
20	Customer	PK_Dimension_Customer	CLUSTERED INDEX	7.69230769230769
21	Stock Item	PK_Dimension_Stock_Item	CLUSTERED INDEX	5
22	Date	PK_Dimension_Date	CLUSTERED INDEX	3.84615384615385
23	Transaction	FK_Fact_Transaction_Date_Key	NONCLUSTERED INDEX	2.85714285714286
24	City	IX_Dimension_City_WWICityID	NONCLUSTERED INDEX	0.23094688221709
25	City	PK_Dimension_City	CLUSTERED INDEX	0.0288350634371396

Chapter 11: Handling JSON Data in SQL Server

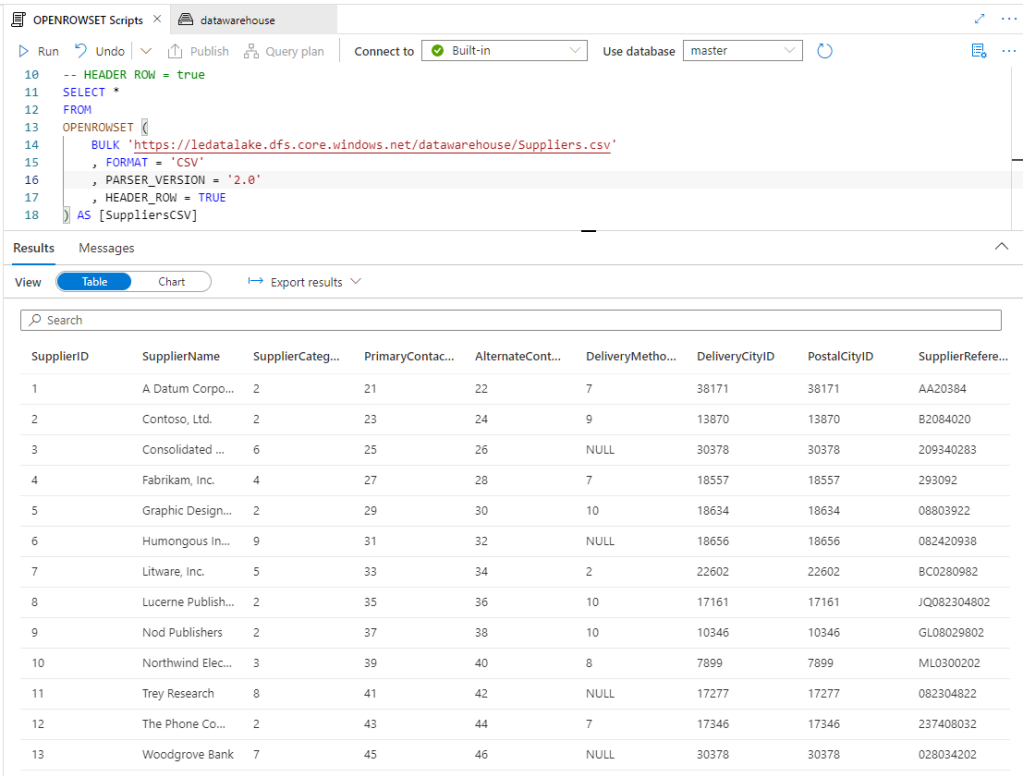


Chapter 12: Integrating File Data and Data Lake Content with SQL

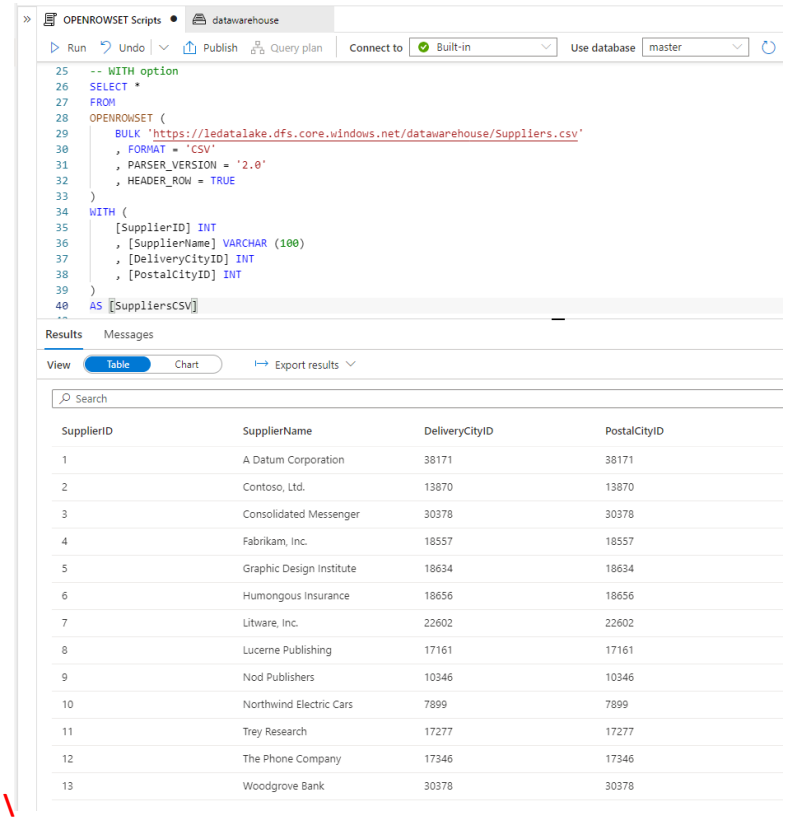




C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
SupplierID	SupplierName	SupplierCatego...	PrimaryContact...	AlternateConta...	DeliveryMetho...	DeliveryCityID	PostalCityID	SupplierRefere...	BankAccountN...
1	A Datum Corpo...	2	21	22	7	38171	38171	AA20384	A Datum Corpo...
2	Contoso, Ltd.	2	23	24	9	13870	13870	B2084020	Contoso Ltd
3	Consolidated ...	6	25	26	NULL	30378	30378	209340283	Consolidated ...
4	Fabrikam, Inc.	4	27	28	7	18557	18557	293092	Fabrikam Inc
5	Graphic Design...	2	29	30	10	18634	18634	08803922	Graphic Design...
6	Humongous In...	9	31	32	NULL	18656	18656	082420938	Humongous In...
7	Litware, Inc.	5	33	34	2	22602	22602	BC0280982	Litware Inc
8	Lucerne Publish...	2	35	36	10	17161	17161	JQ082304802	Lucerne Publish...
9	Nod Publishers	2	37	38	10	10346	10346	GL08029802	Nod Publishers
10	Northwind Elec...	3	39	40	8	7899	7899	ML0300202	Northwind Elec...
11	Trey Research	8	41	42	NULL	17277	17277	082304822	Trey Research
12	The Phone Co...	2	43	44	7	17346	17346	237408032	The Phone Co...
13	Woodgrove Bank	7	45	46	NULL	30378	30378	028034202	Woodgrove Bank



SupplierID	SupplierName	SupplierCateg...	PrimaryContac...	AlternateCont...	DeliveryMetho...	DeliveryCityID	PostalCityID	SupplierRefere...
1	A Datum Corpo...	2	21	22	7	38171	38171	AA20384
2	Contoso, Ltd.	2	23	24	9	13870	13870	B2084020
3	Consolidated ...	6	25	26	NULL	30378	30378	209340283
4	Fabrikam, Inc.	4	27	28	7	18557	18557	293092
5	Graphic Design...	2	29	30	10	18634	18634	08803922
6	Humongous In...	9	31	32	NULL	18656	18656	082420938
7	Litware, Inc.	5	33	34	2	22602	22602	BC0280982
8	Lucerne Publish...	2	35	36	10	17161	17161	JQ082304802
9	Nod Publishers	2	37	38	10	10346	10346	GL08029802
10	Northwind Elec...	3	39	40	8	7899	7899	ML0300202
11	Trey Research	8	41	42	NULL	17277	17277	082304822
12	The Phone Co...	2	43	44	7	17346	17346	237408032
13	Woodgrove Bank	7	45	46	NULL	30378	30378	028034202



SupplierID	SupplierName	DeliveryCityID	PostalCityID
1	A Datum Corporation	38171	38171
2	Contoso, Ltd.	13870	13870
3	Consolidated Messenger	30378	30378
4	Fabrikam, Inc.	18557	18557
5	Graphic Design Institute	18634	18634
6	Humongous Insurance	18656	18656
7	Litware, Inc.	22602	22602
8	Lucerne Publishing	17161	17161
9	Nod Publishers	10346	10346
10	Northwind Electric Cars	7899	7899
11	Trey Research	17277	17277
12	The Phone Company	17346	17346
13	Woodgrove Bank	30378	30378

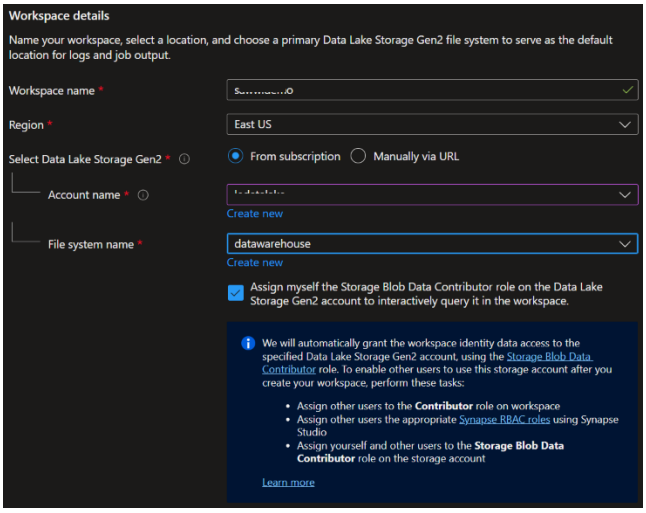
```sql
SELECT
    [SuppliersCSV].SupplierID
    , [SuppliersCSV].SupplierName
    , [PurchaseOrdersCSV].OrderDate
    , [PurchaseOrdersCSV].LastEditedBy
FROM
    OPENROWSET (
        BULK 'https://ledatalake.dfs.core.windows.net/datawarehouse/Suppliers.csv'
        , FORMAT = 'CSV'
        , PARSER_VERSION = '2.0'
        , HEADER_ROW = TRUE
    ) AS [SuppliersCSV]
    INNER JOIN
        (SELECT *
        FROM
        OPENROWSET (
            BULK 'https://ledatalake.blob.core.windows.net/datawarehouse/PurchaseOrders.csv'
            , FORMAT = 'CSV'
            , PARSER_VERSION = '2.0'
            , HEADER_ROW = TRUE
        ) AS [PurchaseOrdersCSV]) AS [PurchaseOrdersCSV]
    ON [SuppliersCSV].SupplierID = [PurchaseOrdersCSV].SupplierID
WHERE [SuppliersCSV].SupplierID = '10';
```

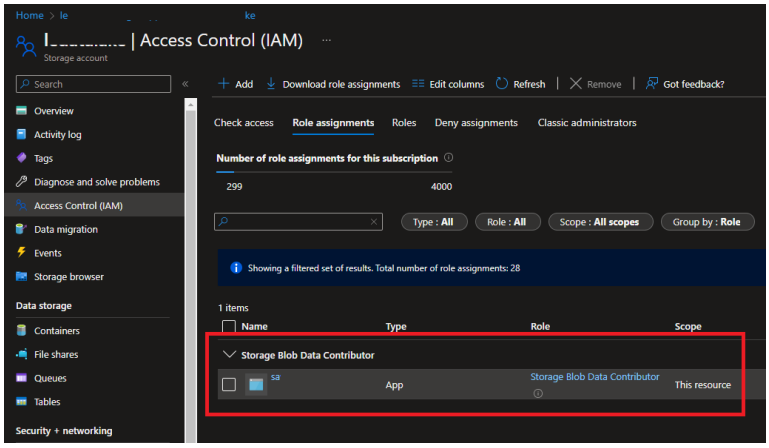
Results | Messages

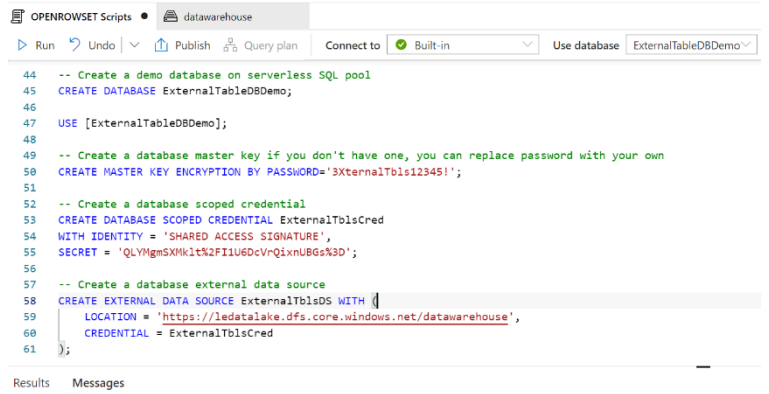
View: Table | Chart — Export results

Search

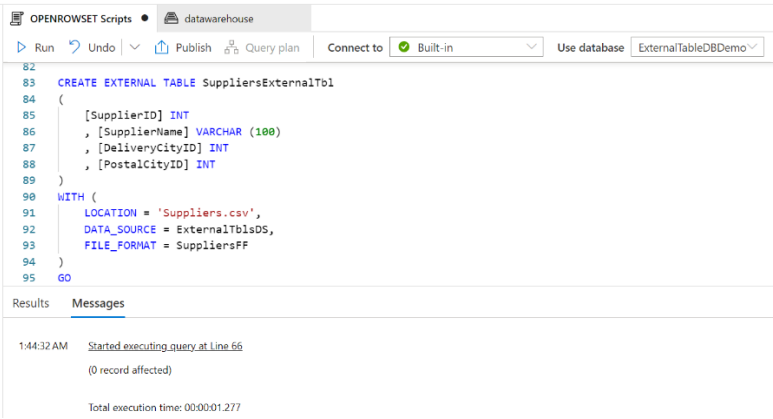
SupplierID	SupplierName	OrderDate	LastEditedBy
10	Northwind Electric Cars	2013-01-01T00:00:00.0000000	6
10	Northwind Electric Cars	2013-01-02T00:00:00.0000000	5
10	Northwind Electric Cars	2013-01-03T00:00:00.0000000	3
10	Northwind Electric Cars	2013-01-04T00:00:00.0000000	14
10	Northwind Electric Cars	2013-01-05T00:00:00.0000000	14
10	Northwind Electric Cars	2013-01-07T00:00:00.0000000	17
10	Northwind Electric Cars	2013-01-09T00:00:00.0000000	7
10	Northwind Electric Cars	2013-01-14T00:00:00.0000000	15
10	Northwind Electric Cars	2013-01-15T00:00:00.0000000	6



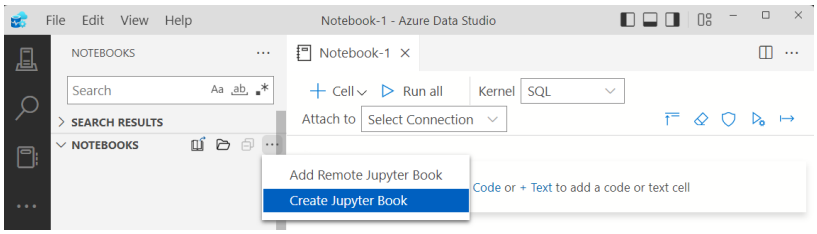

Home > le ke
| Access Control (IAM)
Storage account
Search
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Data storage
Containers
File shares
Queues
Tables
Security + networking
Add
Download role assignments
Edit columns
Refresh
Remove
Got feedback?
Check access
Role assignments
Roles
Deny assignments
Classic administrators
Number of role assignments for this subscription
299
4000
Type : All
Role : All
Scope : All scopes
Group by : Role
Showing a filtered set of results. Total number of role assignments: 28
1 items
Name
Type
Role
Scope
Storage Blob Data Contributor
sa
App
Storage Blob Data Contributor
This resource

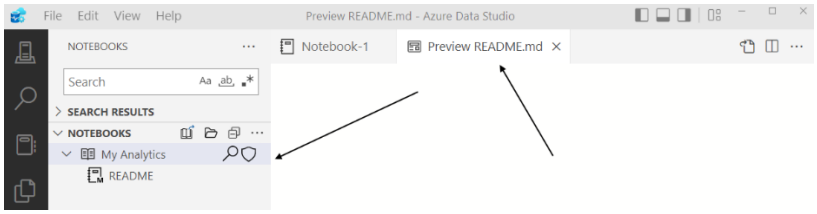

OPENROWSET Scripts
datawarehouse
Run
Undo
Publish
Query plan
Connect to
Built-in
Use database
ExternalTableDBDemo
-- Create a demo database on serverless SQL pool
CREATE DATABASE ExternalTableDBDemo;
USE [ExternalTableDBDemo];
-- Create a database master key if you don't have one, you can replace password with your own
CREATE MASTER KEY ENCRYPTION BY PASSWORD='3XternalTbls12345!';
-- Create a database scoped credential
CREATE DATABASE SCOPED CREDENTIAL ExternalTblsCred
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET = 'QLYMgmSXMklt%2FI1U6DcVrQixnUBGs%3D';
-- Create a database external data source
CREATE EXTERNAL DATA SOURCE ExternalTblsDS WITH (
LOCATION = 'https://ledatalake.dfs.core.windows.net/datawarehouse',
CREDENTIAL = ExternalTblsCred
);
Results
Messages
1:43:34 AM
Started executing query at Line 49
(0 record affected)
Total execution time: 00:00:00.482

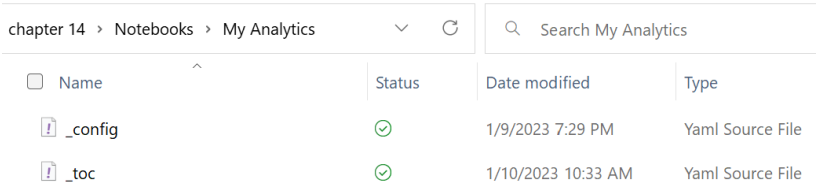

Run
Undo
Publish
Query plan
Connect to
Built-in
Use database
ExternalTableDBDemo
-- Create file format for Suppliers.csv
CREATE EXTERNAL FILE FORMAT SuppliersFF
WITH
(
FORMAT_TYPE = DELIMITEDTEXT,
FORMAT_OPTIONS (
FIELD_TERMINATOR = ','
, PARSER_VERSION = '1.0'
, FIRST_ROW = 2
, ENCODING = 'UTF8'
)
);
Results
Messages
1:44:32 AM
Started executing query at Line 66
(0 record affected)
Total execution time: 00:00:01.277

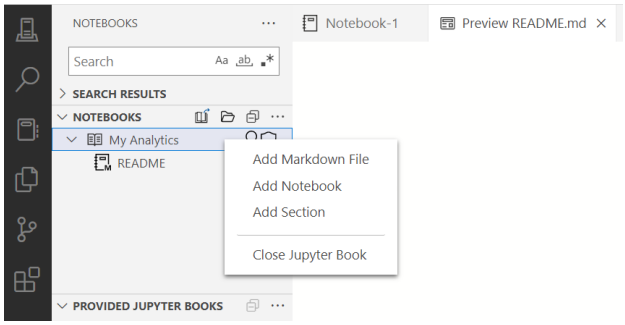

OPENROWSET Scripts
datawarehouse
Run
Undo
Publish
Query plan
Connect to
Built-in
Use database
ExternalTableDBDemo
CREATE EXTERNAL TABLE SuppliersExternalTbl
(
[SupplierID] INT
, [SupplierName] VARCHAR (100)
, [DeliveryCityID] INT
, [PostalCityID] INT
)
WITH (
LOCATION = 'Suppliers.csv',
DATA_SOURCE = ExternalTblsDS,
FILE_FORMAT = SuppliersFF
)
GO
Results
Messages
1:44:32 AM
Started executing query at Line 66
(0 record affected)
Total execution time: 00:00:01.277

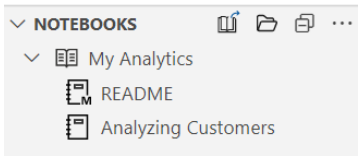
Chapter 13: Organizing and Sharing Your Queries with Jupyter Notebooks

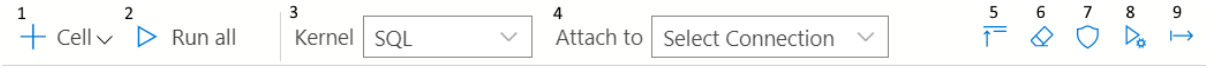


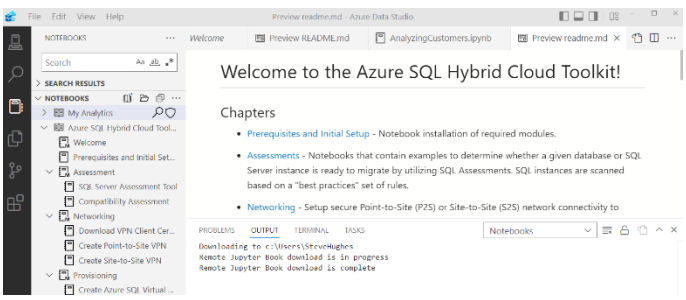


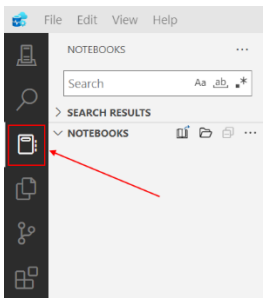


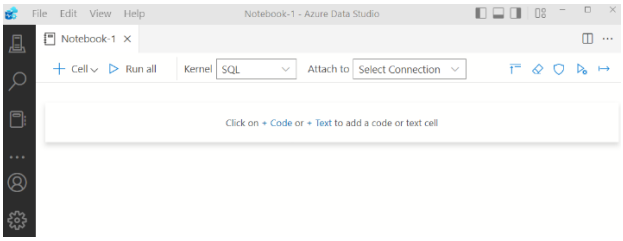


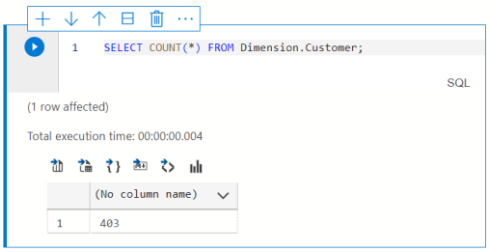




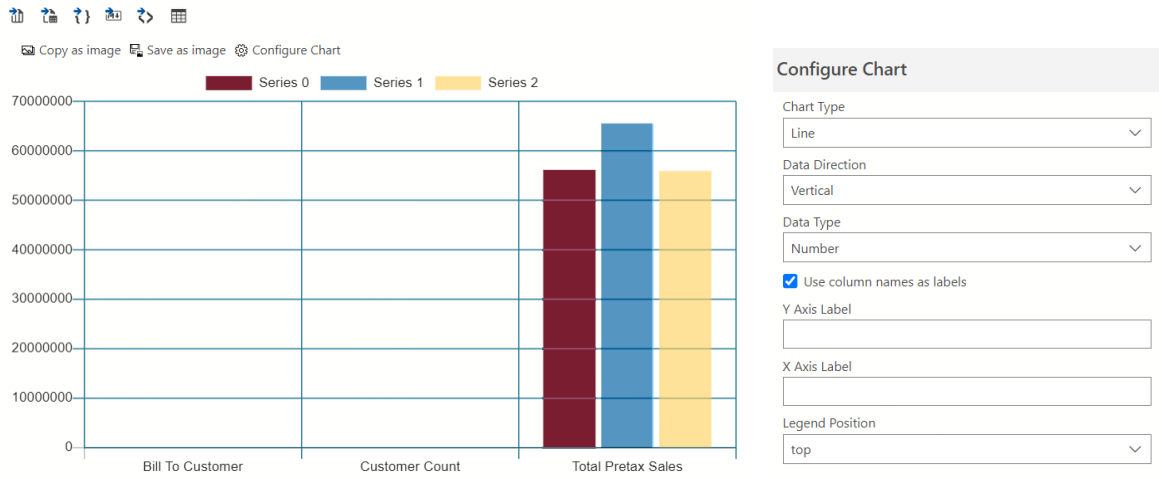


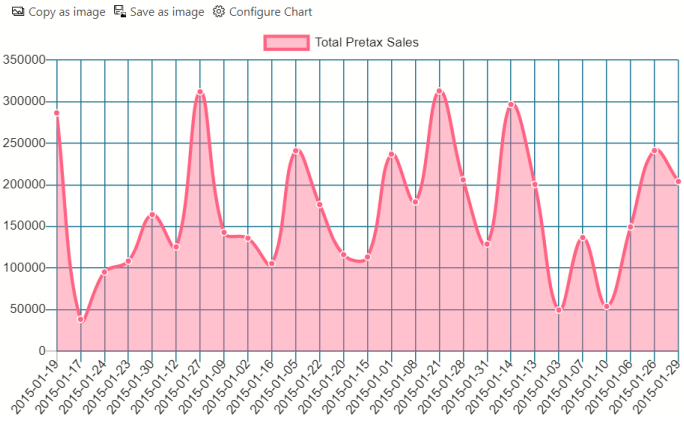


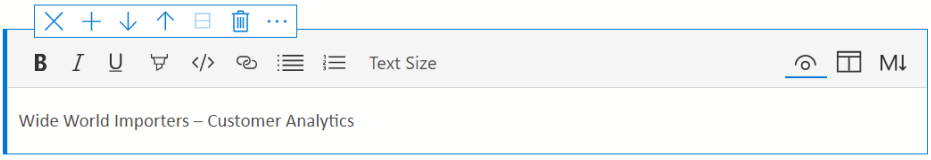


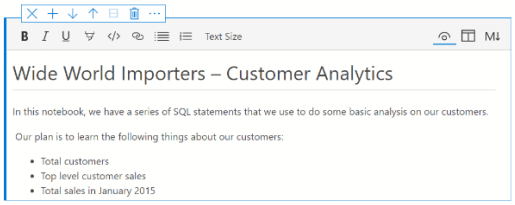


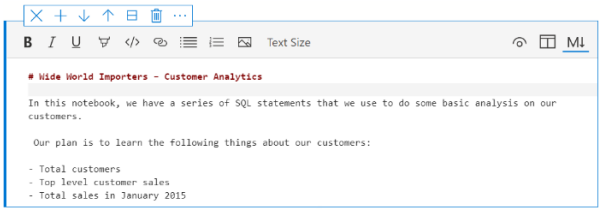
	Bill To Customer	Customer Count	Total Pretax Sales
1	Tailspin Toys (Head Office)	201	56171644.00
2	N/A	1	65545913.60
3	Wingtip Toys (Head Office)	201	55916718.80











To create this format	Use this markdown
Header 1 (H1)	# header
Header 2 (H2)	## header
Header 3 (H3)	### header
Header 4 (H4)	#### header
Bold	**insert word here** __insert word here__
Italics	*insert word here* _insert word here_
Ordered list	1. Maple 2. Oak 3. Elm
Unordered list	* maple * oak * elm