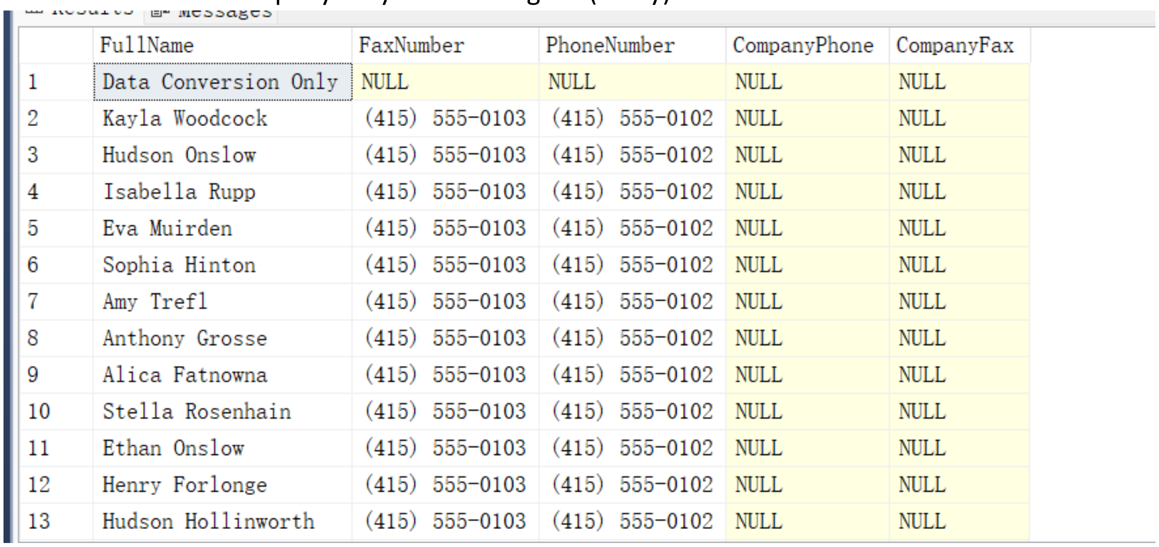
SQL Assignments

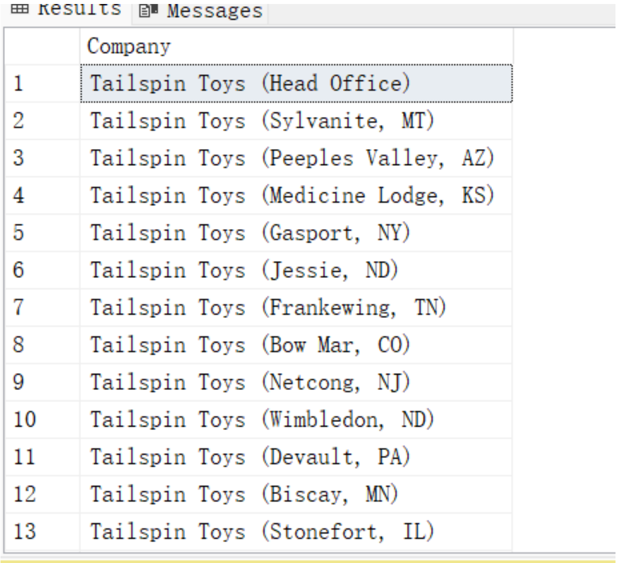
SQL related assignments will be on the Wide World Importers Database unless otherwise mentioned.

1. List of Persons' full name, all their fax and phone numbers, as well as the phone number and fax of the company they are working for (if any).



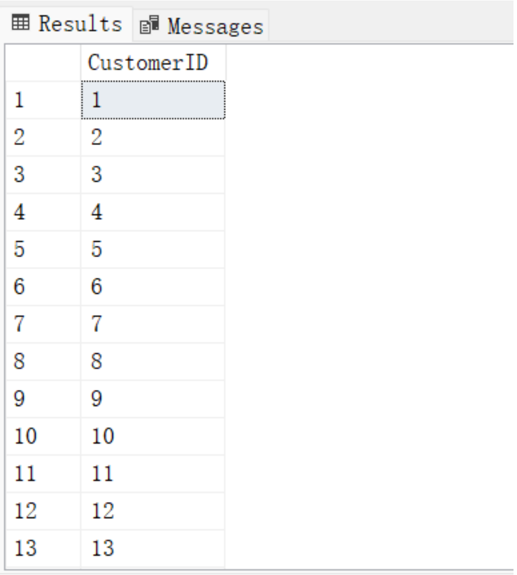
	FullName	FaxNumber	PhoneNumber	CompanyPhone	CompanyFax
1	Data Conversion Only	NULL	NULL	NULL	NULL
2	Kayla Woodcock	(415) 555-0103	(415) 555-0102	NULL	NULL
3	Hudson Onslow	(415) 555-0103	(415) 555-0102	NULL	NULL
4	Isabella Rupp	(415) 555-0103	(415) 555-0102	NULL	NULL
5	Eva Muirden	(415) 555-0103	(415) 555-0102	NULL	NULL
6	Sophia Hinton	(415) 555-0103	(415) 555-0102	NULL	NULL
7	Amy Trefl	(415) 555-0103	(415) 555-0102	NULL	NULL
8	Anthony Grosse	(415) 555-0103	(415) 555-0102	NULL	NULL
9	Alica Fatnowna	(415) 555-0103	(415) 555-0102	NULL	NULL
10	Stella Rosenhain	(415) 555-0103	(415) 555-0102	NULL	NULL
11	Ethan Onslow	(415) 555-0103	(415) 555-0102	NULL	NULL
12	Henry Forlonge	(415) 555-0103	(415) 555-0102	NULL	NULL
13	Hudson Hollinworth	(415) 555-0103	(415) 555-0102	NULL	NULL

2. If the customer's primary contact person has the same phone number as the customer's phone number, list the customer companies.



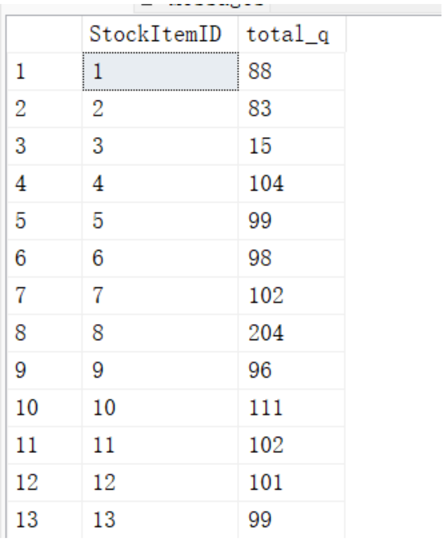
	Company
1	Tailspin Toys (Head Office)
2	Tailspin Toys (Sylvanite, MT)
3	Tailspin Toys (Peeples Valley, AZ)
4	Tailspin Toys (Medicine Lodge, KS)
5	Tailspin Toys (Gasport, NY)
6	Tailspin Toys (Jessie, ND)
7	Tailspin Toys (Frankewing, TN)
8	Tailspin Toys (Bow Mar, CO)
9	Tailspin Toys (Netcong, NJ)
10	Tailspin Toys (Wimbledon, ND)
11	Tailspin Toys (Devault, PA)
12	Tailspin Toys (Biscay, MN)
13	Tailspin Toys (Stonefort, IL)

3. List of customers to whom we made a sale prior to 2016 but no sale since 2016-01-01.



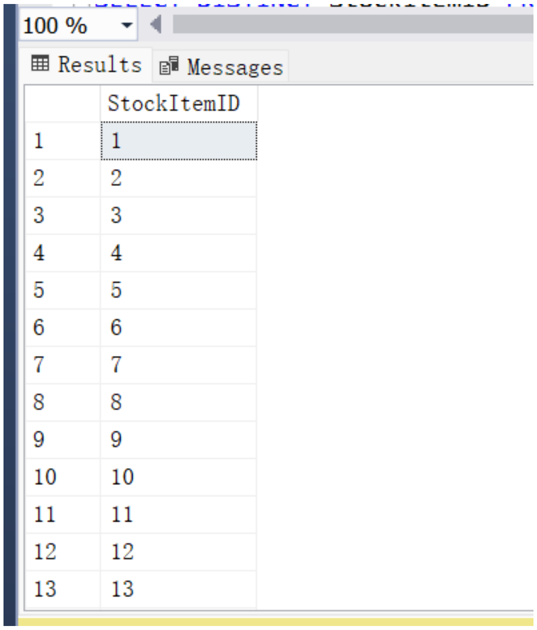
	CustomerID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13

4. List of Stock Items and total quantity for each stock item in Purchase Orders in Year 2013.



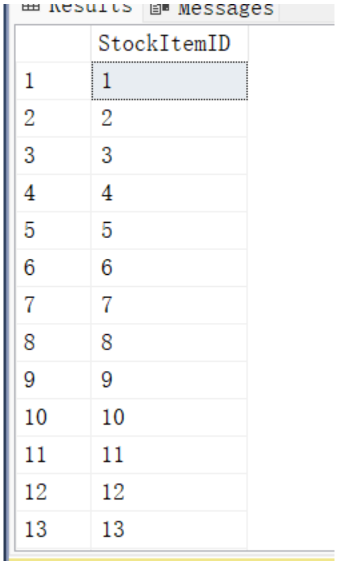
	StockItemID	total_q
1	1	88
2	2	83
3	3	15
4	4	104
5	5	99
6	6	98
7	7	102
8	8	204
9	9	96
10	10	111
11	11	102
12	12	101
13	13	99

5. List of stock items that have at least 10 characters in description.



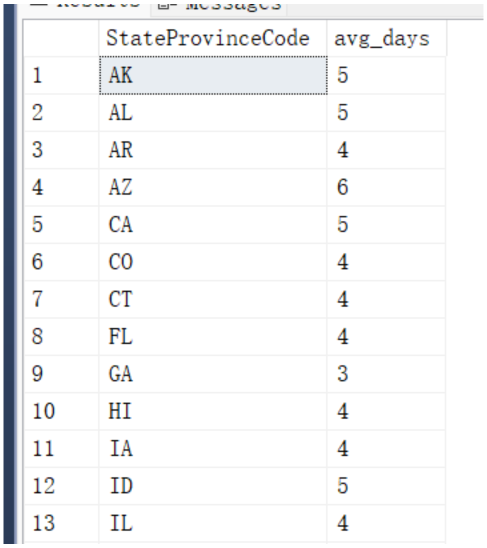
	StockItemID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13

6. List of stock items that are not sold to the state of Alabama and Georgia in 2014.



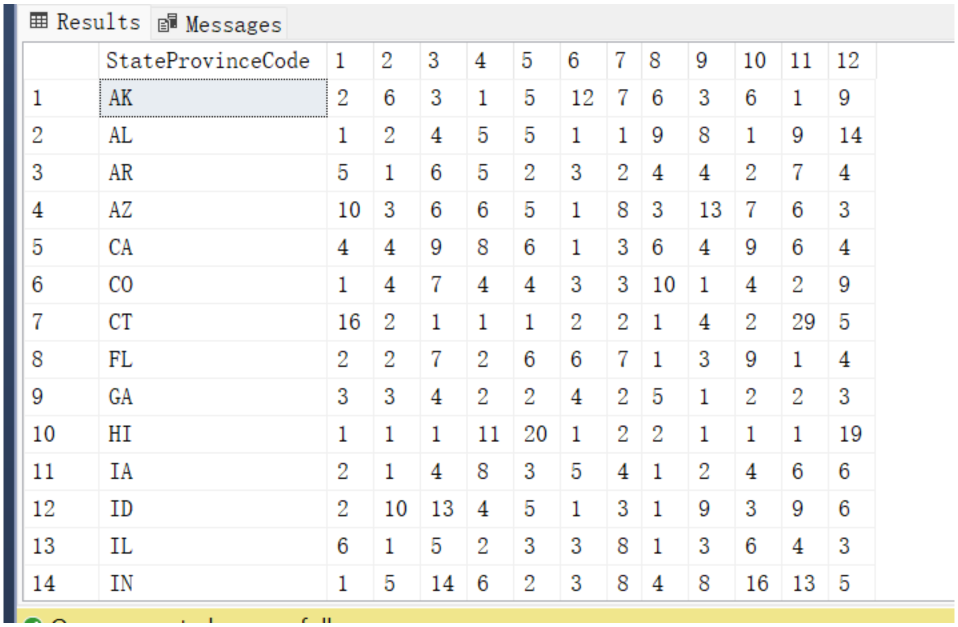
	StockItemID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13

7. List of States and Avg dates for processing (confirmed delivery date – order date).



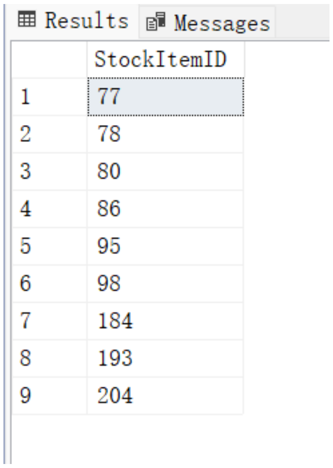
	StateProvinceCode	avg_days
1	AK	5
2	AL	5
3	AR	4
4	AZ	6
5	CA	5
6	CO	4
7	CT	4
8	FL	4
9	GA	3
10	HI	4
11	IA	4
12	ID	5
13	IL	4

8. List of States and Avg dates for processing (confirmed delivery date – order date) by month.



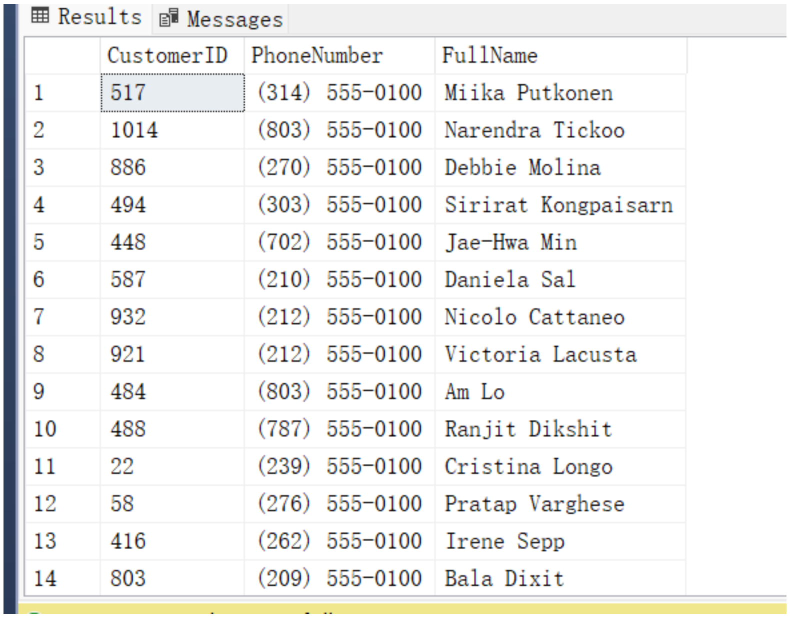
	StateProvinceCode	1	2	3	4	5	6	7	8	9	10	11	12
1	AK	2	6	3	1	5	12	7	6	3	6	1	9
2	AL	1	2	4	5	5	1	1	9	8	1	9	14
3	AR	5	1	6	5	2	3	2	4	4	2	7	4
4	AZ	10	3	6	6	5	1	8	3	13	7	6	3
5	CA	4	4	9	8	6	1	3	6	4	9	6	4
6	CO	1	4	7	4	4	3	3	10	1	4	2	9
7	CT	16	2	1	1	1	2	2	1	4	2	29	5
8	FL	2	2	7	2	6	6	7	1	3	9	1	4
9	GA	3	3	4	2	2	4	2	5	1	2	2	3
10	HI	1	1	1	11	20	1	2	2	1	1	1	19
11	IA	2	1	4	8	3	5	4	1	2	4	6	6
12	ID	2	10	13	4	5	1	3	1	9	3	9	6
13	IL	6	1	5	2	3	3	8	1	3	6	4	3
14	IN	1	5	14	6	2	3	8	4	8	16	13	5

9. List of StockItems that the company purchased more than sold in the year of 2015.



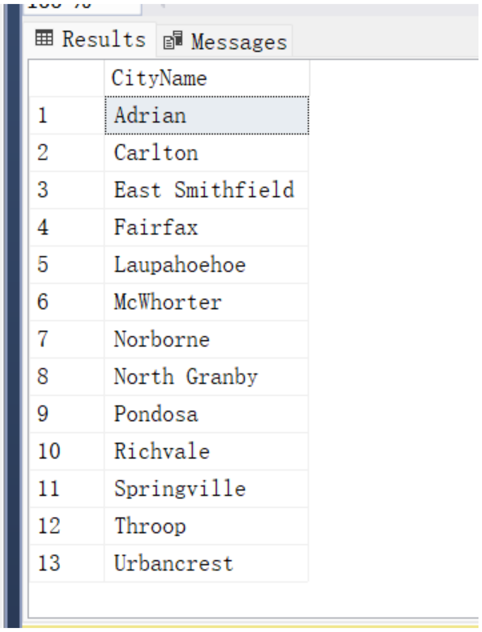
	StockItemID
1	77
2	78
3	80
4	86
5	95
6	98
7	184
8	193
9	204

10. List of Customers and their phone number, together with the primary contact person's name, to whom we did not sell more than 10 mugs (search by name) in the year 2016.



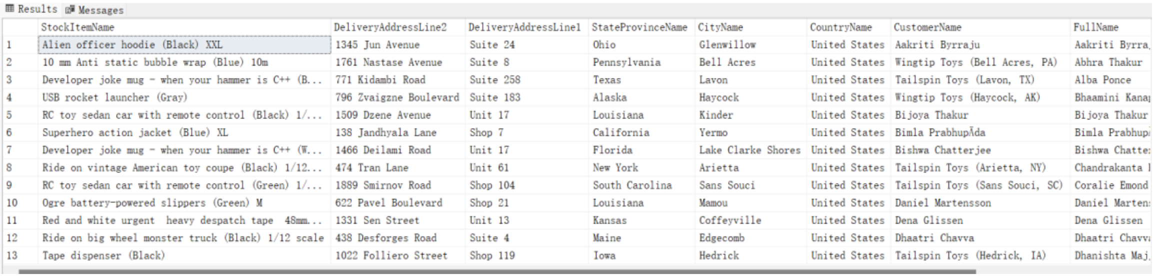
	CustomerID	PhoneNumber	FullName
1	517	(314) 555-0100	Miika Putkonen
2	1014	(803) 555-0100	Narendra Tickoo
3	886	(270) 555-0100	Debbie Molina
4	494	(303) 555-0100	Sirirat Kongpaisarn
5	448	(702) 555-0100	Jae-Hwa Min
6	587	(210) 555-0100	Daniela Sal
7	932	(212) 555-0100	Nicolo Cattaneo
8	921	(212) 555-0100	Victoria Lacusta
9	484	(803) 555-0100	Am Lo
10	488	(787) 555-0100	Ranjit Dikshit
11	22	(239) 555-0100	Cristina Longo
12	58	(276) 555-0100	Pratap Varghese
13	416	(262) 555-0100	Irene Sepp
14	803	(209) 555-0100	Bala Dixit

11. List all the cities that were updated after 2015-01-01.



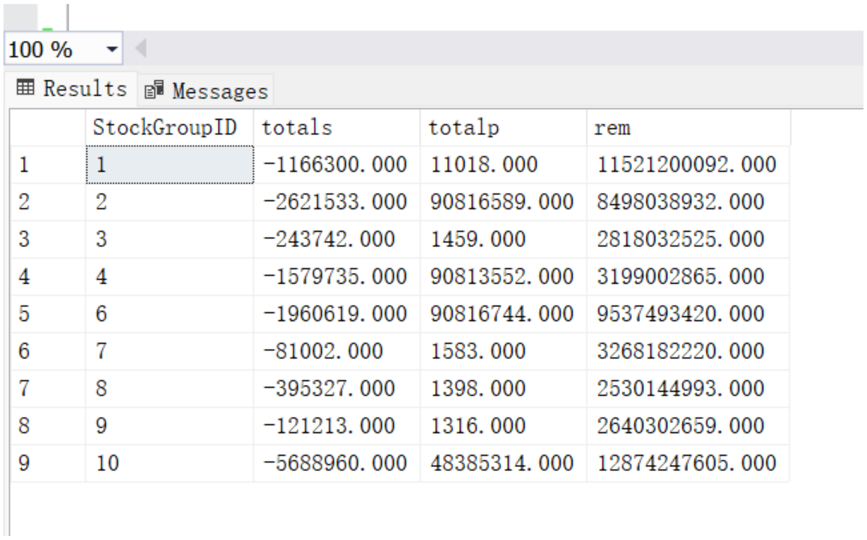
	CityName
1	Adrian
2	Carlton
3	East Smithfield
4	Fairfax
5	Laupahoehoe
6	McWhorter
7	Norborne
8	North Granby
9	Pondosa
10	Richvale
11	Springville
12	Throop
13	Urbancrest

12. List all the Order Detail (Stock Item name, delivery address, delivery state, city, country, customer name, customer contact person name, customer phone, quantity) for the date of 2014-07-01. Info should be relevant to that date.



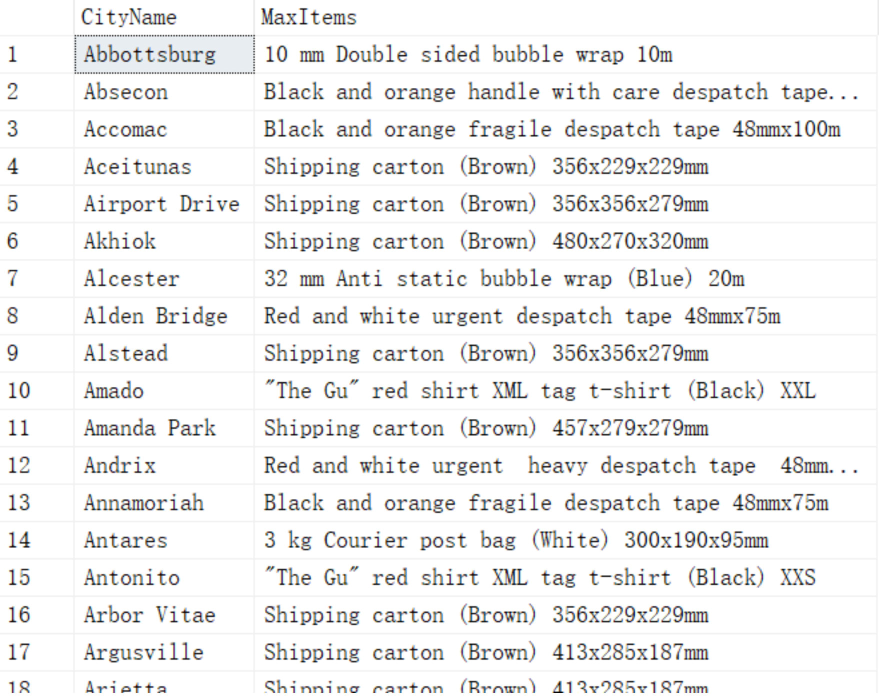
	StockItemName	DeliveryAddressLine2	DeliveryAddressLine1	StateProvinceName	CityName	CountryName	CustomerName	FullName
1	Alien officer hoodie (Black) XXL	1345 Jun Avenue	Suite 24	Ohio	Glenwillow	United States	Aakriti Byrraju	Aakriti Byrra
2	10 mm Anti static bubble wrap (Blue) 10m	1761 Nastase Avenue	Suite 8	Pennsylvania	Bell Acres	United States	Wingtip Toys (Bell Acres, PA)	Abhra Thakur
3	Developer joke mug - when your hammer is C++ (B...	771 Kidambi Road	Suite 258	Texas	Lavon	United States	Tailspin Toys (Lavon, TX)	Alba Ponce
4	USB rocket launcher (Gray)	796 Zvaigzne Boulevard	Suite 183	Alaska	Haycock	United States	Wingtip Toys (Haycock, AK)	Bhaamini Kana
5	RC toy sedan car with remote control (Black) 1/...	1509 Dzene Avenue	Unit 17	Louisiana	Kinder	United States	Bijoya Thakur	Bijoya Thakur
6	Superhero action jacket (Blue) XL	138 Jandhyala Lane	Shop 7	California	Yermo	United States	Bimla PrabhupÃda	Bimla Prabhup
7	Developer joke mug - when your hammer is C++ (W...	1466 Deilami Road	Unit 17	Florida	Lake Clarke Shores	United States	Bishwa Chatterjee	Bishwa Chatte
8	Ride on vintage American toy coupe (Black) 1/12...	474 Tran Lane	Unit 61	New York	Arietta	United States	Tailspin Toys (Arietta, NY)	Chandrakanta
9	RC toy sedan car with remote control (Green) 1/...	1889 Smirnov Road	Shop 104	South Carolina	Sans Souci	United States	Tailspin Toys (Sans Souci, SC)	Coralie Emond
10	Ogre battery-powered slippers (Green) M	622 Pavel Boulevard	Shop 21	Louisiana	Mamou	United States	Daniel Martensson	Daniel Marten
11	Red and white urgent heavy despatch tape 48mm...	1331 Sen Street	Unit 13	Kansas	Coffeyville	United States	Dena Glissen	Dena Glissen
12	Ride on big wheel monster truck (Black) 1/12 scale	438 Desforges Road	Suite 4	Maine	Edgecomb	United States	Dhaatri Chavva	Dhaatri Chavv
13	Tape dispenser (Black)	1022 Folliero Street	Shop 119	Iowa	Hedrick	United States	Tailspin Toys (Hedrick, IA)	Dhanishta Maj

13. List of stock item groups and total quantity purchased, total quantity sold, and the remaining stock quantity (quantity purchased – quantity sold)



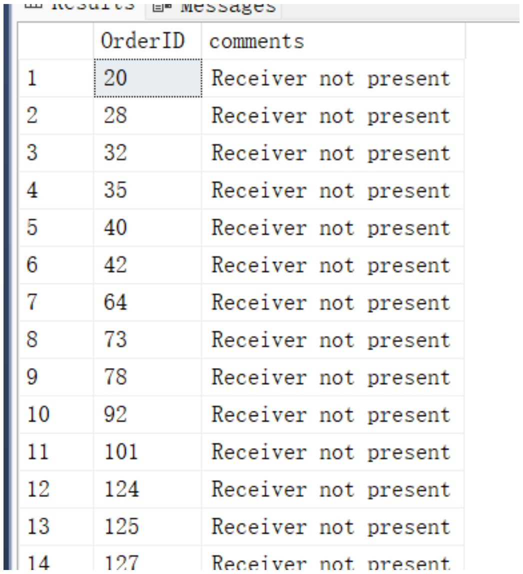
	StockGroupID	totals	totalp	rem
1	1	-1166300.000	11018.000	11521200092.000
2	2	-2621533.000	90816589.000	8498038932.000
3	3	-243742.000	1459.000	2818032525.000
4	4	-1579735.000	90813552.000	3199002865.000
5	6	-1960619.000	90816744.000	9537493420.000
6	7	-81002.000	1583.000	3268182220.000
7	8	-395327.000	1398.000	2530144993.000
8	9	-121213.000	1316.000	2640302659.000
9	10	-5688960.000	48385314.000	12874247605.000

14. List of Cities in the US and the stock item that the city got the most deliveries in 2016. If the city did not purchase any stock items in 2016, print “No Sales”.



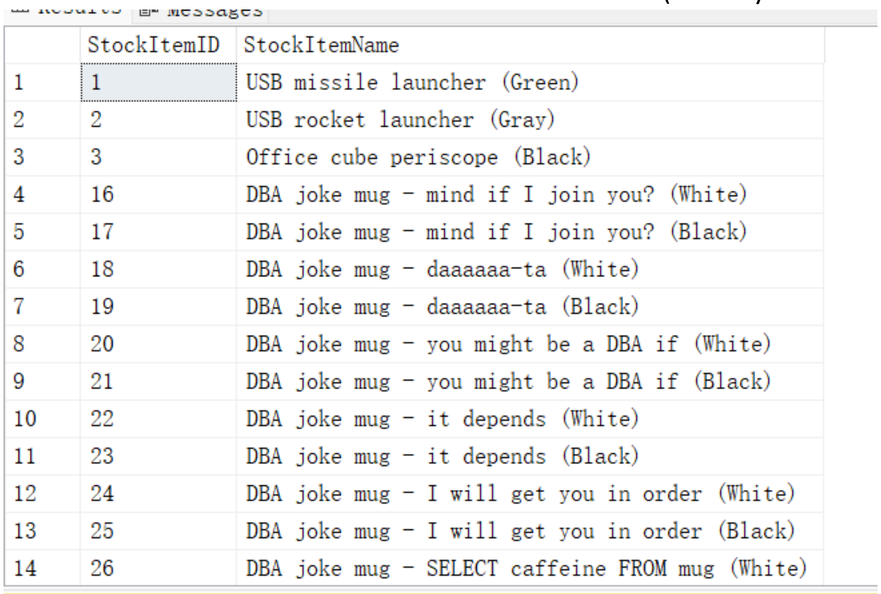
	CityName	MaxItems
1	Abbottsburg	10 mm Double sided bubble wrap 10m
2	Absecon	Black and orange handle with care despatch tape...
3	Accomac	Black and orange fragile despatch tape 48mmx100m
4	Aceitunas	Shipping carton (Brown) 356x229x229mm
5	Airport Drive	Shipping carton (Brown) 356x356x279mm
6	Akhiok	Shipping carton (Brown) 480x270x320mm
7	Alcester	32 mm Anti static bubble wrap (Blue) 20m
8	Alden Bridge	Red and white urgent despatch tape 48mmx75m
9	Alstead	Shipping carton (Brown) 356x356x279mm
10	Amado	"The Gu" red shirt XML tag t-shirt (Black) XXL
11	Amanda Park	Shipping carton (Brown) 457x279x279mm
12	Andrix	Red and white urgent heavy despatch tape 48mm...
13	Annamoriah	Black and orange fragile despatch tape 48mmx75m
14	Antares	3 kg Courier post bag (White) 300x190x95mm
15	Antonito	"The Gu" red shirt XML tag t-shirt (Black) XXS
16	Arbor Vitae	Shipping carton (Brown) 356x229x229mm
17	Argusville	Shipping carton (Brown) 413x285x187mm
18	Arietta	Shipping carton (Brown) 413x285x187mm

15. List any orders that had more than one delivery attempt (located in invoice table).



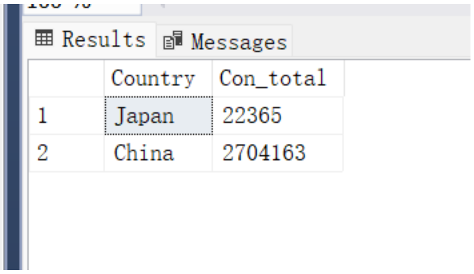
	OrderID	comments
1	20	Receiver not present
2	28	Receiver not present
3	32	Receiver not present
4	35	Receiver not present
5	40	Receiver not present
6	42	Receiver not present
7	64	Receiver not present
8	73	Receiver not present
9	78	Receiver not present
10	92	Receiver not present
11	101	Receiver not present
12	124	Receiver not present
13	125	Receiver not present
14	127	Receiver not present

16. List all stock items that are manufactured in China. (Country of Manufacture)



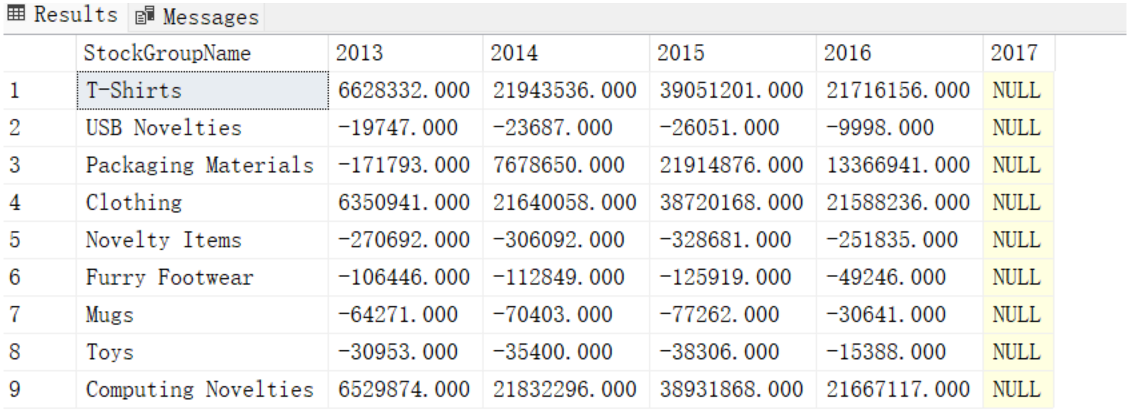
	StockItemID	StockItemName
1	1	USB missile launcher (Green)
2	2	USB rocket launcher (Gray)
3	3	Office cube periscope (Black)
4	16	DBA joke mug - mind if I join you? (White)
5	17	DBA joke mug - mind if I join you? (Black)
6	18	DBA joke mug - daaaaaa-ta (White)
7	19	DBA joke mug - daaaaaa-ta (Black)
8	20	DBA joke mug - you might be a DBA if (White)
9	21	DBA joke mug - you might be a DBA if (Black)
10	22	DBA joke mug - it depends (White)
11	23	DBA joke mug - it depends (Black)
12	24	DBA joke mug - I will get you in order (White)
13	25	DBA joke mug - I will get you in order (Black)
14	26	DBA joke mug - SELECT caffeine FROM mug (White)

17. Total quantity of stock items sold in 2015, group by country of manufacturing.



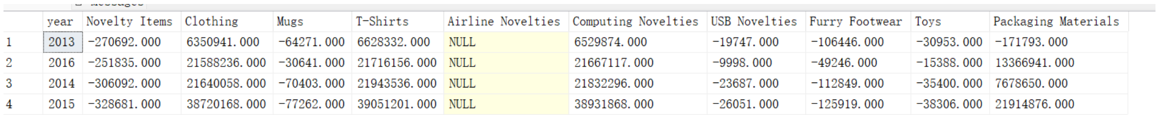
	Country	Con_total
1	Japan	22365
2	China	2704163

18. Create a view that shows the total quantity of stock items of each stock group sold (in orders) by year 2013-2017. [Stock Group Name, 2013, 2014, 2015, 2016, 2017]



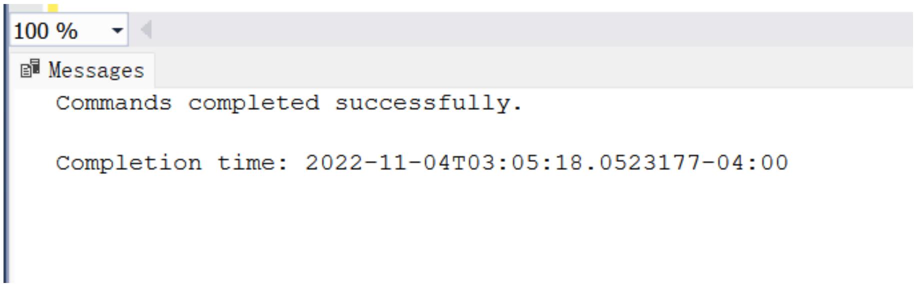
	StockGroupName	2013	2014	2015	2016	2017
1	T-Shirts	6628332.000	21943536.000	39051201.000	21716156.000	NULL
2	USB Novelties	-19747.000	-23687.000	-26051.000	-9998.000	NULL
3	Packaging Materials	-171793.000	7678650.000	21914876.000	13366941.000	NULL
4	Clothing	6350941.000	21640058.000	38720168.000	21588236.000	NULL
5	Novelty Items	-270692.000	-306092.000	-328681.000	-251835.000	NULL
6	Furry Footwear	-106446.000	-112849.000	-125919.000	-49246.000	NULL
7	Mugs	-64271.000	-70403.000	-77262.000	-30641.000	NULL
8	Toys	-30953.000	-35400.000	-38306.000	-15388.000	NULL
9	Computing Novelties	6529874.000	21832296.000	38931868.000	21667117.000	NULL

19. Create a view that shows the total quantity of stock items of each stock group sold (in orders) by year 2013-2017. [Year, Stock Group Name1, Stock Group Name2, Stock Group Name3, … , Stock Group Name10]



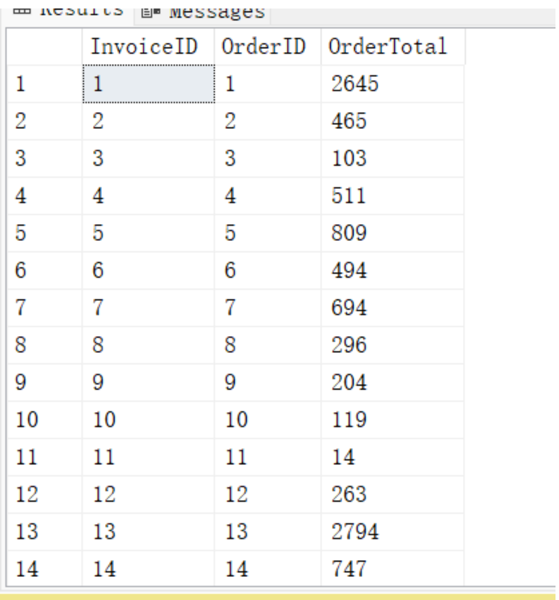
	year	Novelty Items	Clothing	Mugs	T-Shirts	Airline Novelties	Computing Novelties	USB Novelties	Furry Footwear	Toys	Packaging Materials
1	2013	-270692.000	6350941.000	-64271.000	6628332.000	NULL	6529874.000	-19747.000	-106446.000	-30953.000	-171793.000
2	2016	-251835.000	21588236.000	-30641.000	21716156.000	NULL	21667117.000	-9998.000	-49246.000	-15388.000	13366941.000
3	2014	-306092.000	21640058.000	-70403.000	21943536.000	NULL	21832296.000	-23687.000	-112849.000	-35400.000	7678650.000
4	2015	-328681.000	38720168.000	-77262.000	39051201.000	NULL	38931868.000	-26051.000	-125919.000	-38306.000	21914876.000

20. Create a function, input: order id; return: total of that order. List invoices and use that function to attach the order total to the other fields of invoices.



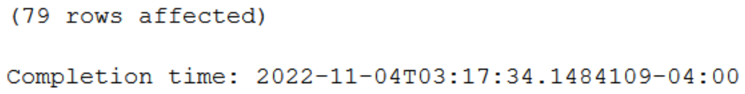
```
Commands completed successfully.

Completion time: 2022-11-04T03:05:18.0523177-04:00
```



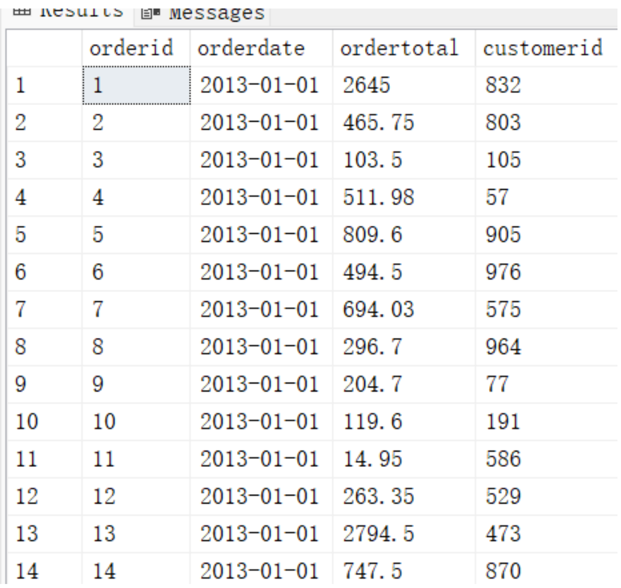
	InvoiceID	OrderID	OrderTotal
1	1	1	2645
2	2	2	465
3	3	3	103
4	4	4	511
5	5	5	809
6	6	6	494
7	7	7	694
8	8	8	296
9	9	9	204
10	10	10	119
11	11	11	14
12	12	12	263
13	13	13	2794
14	14	14	747

21. Create a new table called ods.Orders. Create a stored procedure, with proper error handling and transactions, that input is a date; when executed, it would find orders of that day, calculate order total, and save the information (order id, order date, order total, customer id) into the new table. If a given date is already existing in the new table, throw an error and roll back. Execute the stored procedure 5 times using different dates.



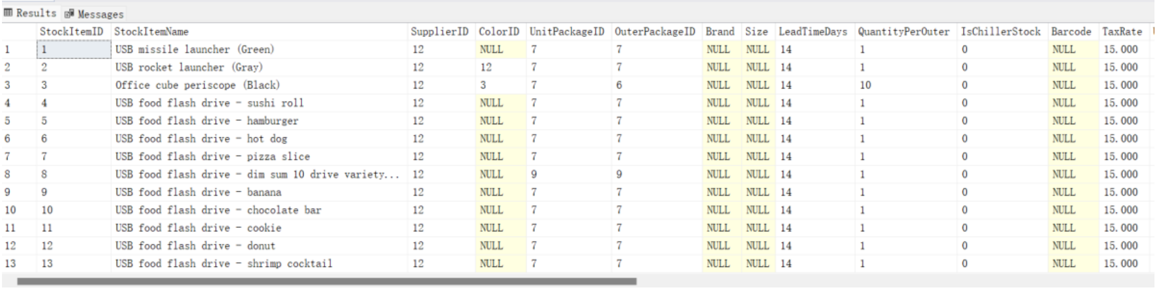
```
(79 rows affected)

Completion time: 2022-11-04T03:17:34.1484109-04:00
```



	orderid	orderdate	ordertotal	customerid
1	1	2013-01-01	2645	832
2	2	2013-01-01	465.75	803
3	3	2013-01-01	103.5	105
4	4	2013-01-01	511.98	57
5	5	2013-01-01	809.6	905
6	6	2013-01-01	494.5	976
7	7	2013-01-01	694.03	575
8	8	2013-01-01	296.7	964
9	9	2013-01-01	204.7	77
10	10	2013-01-01	119.6	191
11	11	2013-01-01	14.95	586
12	12	2013-01-01	263.35	529
13	13	2013-01-01	2794.5	473
14	14	2013-01-01	747.5	870

22. Create a new table called ods.StockItem. It has following columns: [StockItemID], [StockItemName] ,[SupplierID] ,[ColorID] ,[UnitPackageID] ,[OuterPackageID] ,[Brand] ,[Size] ,[LeadTimeDays] ,[QuantityPerOuter] ,[IsChillerStock] ,[Barcode] ,[TaxRate] ,[UnitPrice],[RecommendedRetailPrice] ,[TypicalWeightPerUnit] ,[MarketingComments] ,[InternalComments], [CountryOfManufacture], [Range], [Shelflife]. Migrate all the data in the original stock item table.



	StockItemID	StockItemName	SupplierID	ColorID	UnitPackageID	OuterPackageID	Brand	Size	LeadTimeDays	QuantityPerOuter	IsChillerStock	Barcode	TaxRate
1	1	USB missile launcher (Green)	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
2	2	USB rocket launcher (Gray)	12	12	7	7	NULL	NULL	14	1	0	NULL	15.000
3	3	Office cube periscope (Black)	12	3	7	6	NULL	NULL	14	10	0	NULL	15.000
4	4	USB food flash drive - sushi roll	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
5	5	USB food flash drive - hamburger	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
6	6	USB food flash drive - hot dog	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
7	7	USB food flash drive - pizza slice	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
8	8	USB food flash drive - dim sum 10 drive variety...	12	NULL	9	9	NULL	NULL	14	1	0	NULL	15.000
9	9	USB food flash drive - banana	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
10	10	USB food flash drive - chocolate bar	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
11	11	USB food flash drive - cookie	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
12	12	USB food flash drive - donut	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000
13	13	USB food flash drive - shrimp cocktail	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000

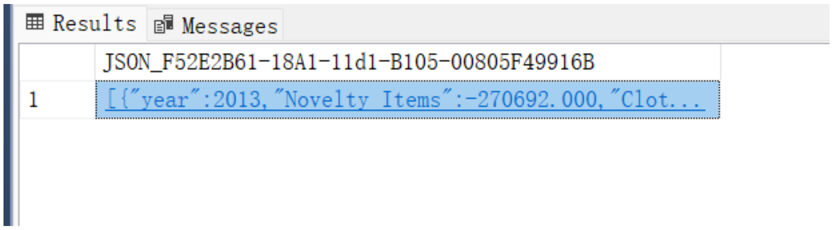
23. Rewrite your stored procedure in (21). Now with a given date, it should wipe out all the order data prior to the input date and load the order data that was placed in the next 7 days following the input date.

24. Consider the JSON file:

```
{
  "PurchaseOrders":[
    {
      "StockItemName":"Panzer Video Game",
      "Supplier":"7",
      "UnitPackageId":"1",
      "OuterPackageId":[
        6,
        7
      ],
      "Brand":"EA Sports",
      "LeadTimeDays":"5",
      "QuantityPerOuter":"1",
      "TaxRate":"6",
      "UnitPrice":"59.99",
      "RecommendedRetailPrice":"69.99",
      "TypicalWeightPerUnit":"0.5",
      "CountryOfManufacture":"Canada",
      "Range":"Adult",
      "OrderDate":"2018-01-01",
      "DeliveryMethod":"Post",
      "ExpectedDeliveryDate":"2018-02-02",
      "SupplierReference":"WWI2308"
    },
    {
      "StockItemName":"Panzer Video Game",
      "Supplier":"5",
      "UnitPackageId":"1",
      "OuterPackageId":"7",
      "Brand":"EA Sports",
      "LeadTimeDays":"5",
      "QuantityPerOuter":"1",
      "TaxRate":"6",
      "UnitPrice":"59.99",
      "RecommendedRetailPrice":"69.99",
      "TypicalWeightPerUnit":"0.5",
      "CountryOfManufacture":"Canada",
      "Range":"Adult",
      "OrderDate":"2018-01-025",
      "DeliveryMethod":"Post",
      "ExpectedDeliveryDate":"2018-02-02",
      "SupplierReference":"269622390"
    }
  ]
}
```

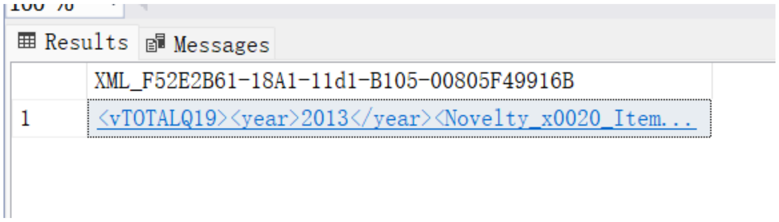
Looks like that it is our missed purchase orders. Migrate these data into Stock Item, Purchase Order and Purchase Order Lines tables. Of course, save the script.

25. Revisit your answer in (19). Convert the result in JSON string and save it to the server using TSQL FOR JSON PATH.



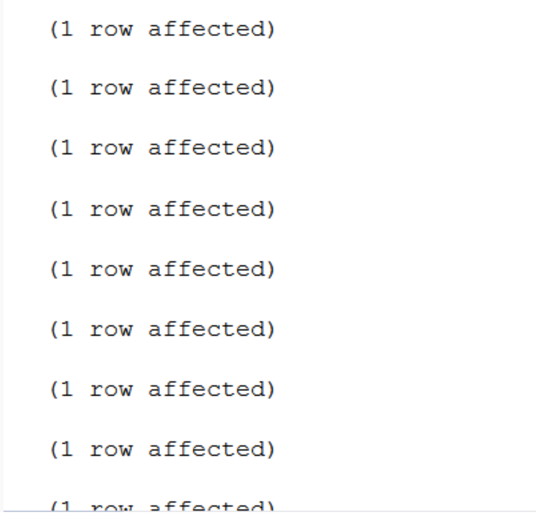
	JSON_F52E2B61-18A1-11d1-B105-00805F49916B
1	[{"year":2013,"Novelty Items":-270692.000,"Clot...

26. Revisit your answer in (19). Convert the result into an XML string and save it to the server using TSQL FOR XML PATH.



	XML_F52E2B61-18A1-11d1-B105-00805F49916B
1	<vTOTALQ19><year>2013</year><Novelty_x0020_Item...

27. Create a new table called ods.ConfirmedDeviveryJson with 3 columns (id, date, value) . Create a stored procedure, input is a date. The logic would load invoice information (all columns) as well as invoice line information (all columns) and forge them into a JSON string and then insert into the new table just created. Then write a query to run the stored procedure for each DATE that customer id 1 got something delivered to him.



```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

28. Write a short essay talking about your understanding of transactions, locks and isolation levels.

Transactions: The unit contains one or more SQL logical statements with ACID principles (Atomicity, Consistency, Isolation, Durability). It is the save/undo button in SQL. Transaction only has two outcome: committed or rolled back. SQL has multiple kinds of transactions, including autocommit transaction, implicit transaction, explicit transaction, batch-scoped transactions (which is not very relevant to DE)

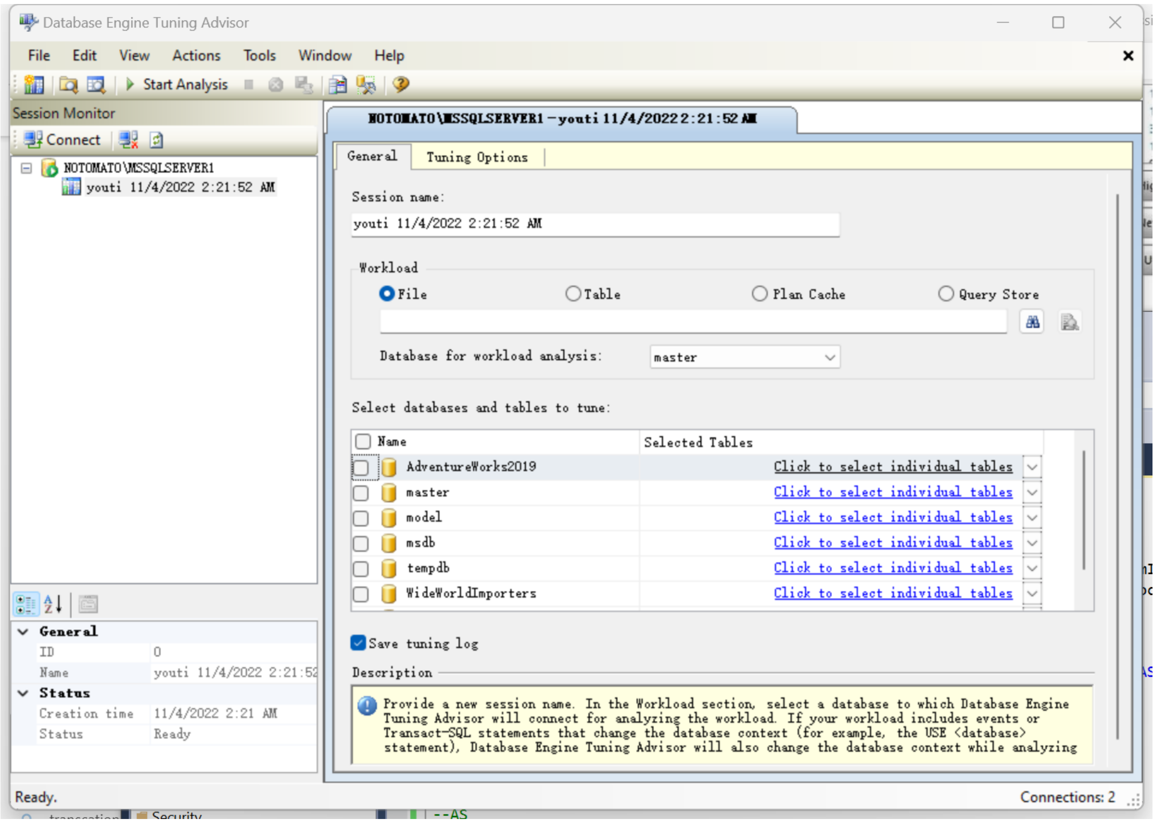
Locks: Used to prevent database from concurrency issues when user is modifying data and affect others like dirty read. Applied lock can make user wait in line until the one transaction is over. There are multiple kinds of locks in SQL server, including Exclusive (X) lock, usually applied in data modify process like insert, update or delete to make sure the data will not have many updates at the same time; Shared (S) lock usually applied in read only, like ‘SELECT’; Update (U) lock used in the source that can be updated, similar to Exclusive (X) lock but more flexible.

Isolation levels: It determine the order when more than one users are processing the same target database. There are Read Uncommitted, Read Committed, Repeatable Read, Serializable, Snapshot total five kinds of isolation levels, and the level increases one by one.

29. Write a short essay, plus screenshots talking about performance tuning in SQL Server. Must include Tuning Advisor, Extended Events, DMV, Logs and Execution Plan.

Tuning Advisor

- Find Database Engine Tuning Advisor at Tools



NOTOMATO\MSSQLSERVER1
youti 11/4/2022 2:21:52 AM

NOTOMATO\MSSQLSERVER1 – youti 11/4/2022 2:21:52 AM

Session name: youti 11/4/2022 2:21:52 AM

Workload

● File ○ Table ○ Plan Cache ○ Query Store

Database for workload analysis: master

Select databases and tables to tune:

□ Name	Selected Tables
□ AdventureWorks2019	Click to select individual tables
□ master	Click to select individual tables
□ model	Click to select individual tables
□ msdb	Click to select individual tables
□ tempdb	Click to select individual tables
□ WideWorldImporters	Click to select individual tables

☑ Save tuning log

Description

Provide a new session name. In the Workload section, select a database to which Database Engine Tuning Advisor will connect for analyzing the workload. If your workload includes events or Transact-SQL statements that change the database context (for example, the USE <database> statement), Database Engine Tuning Advisor will also change the database context while analyzing

General

ID	0
Name	youti 11/4/2022 2:21:52

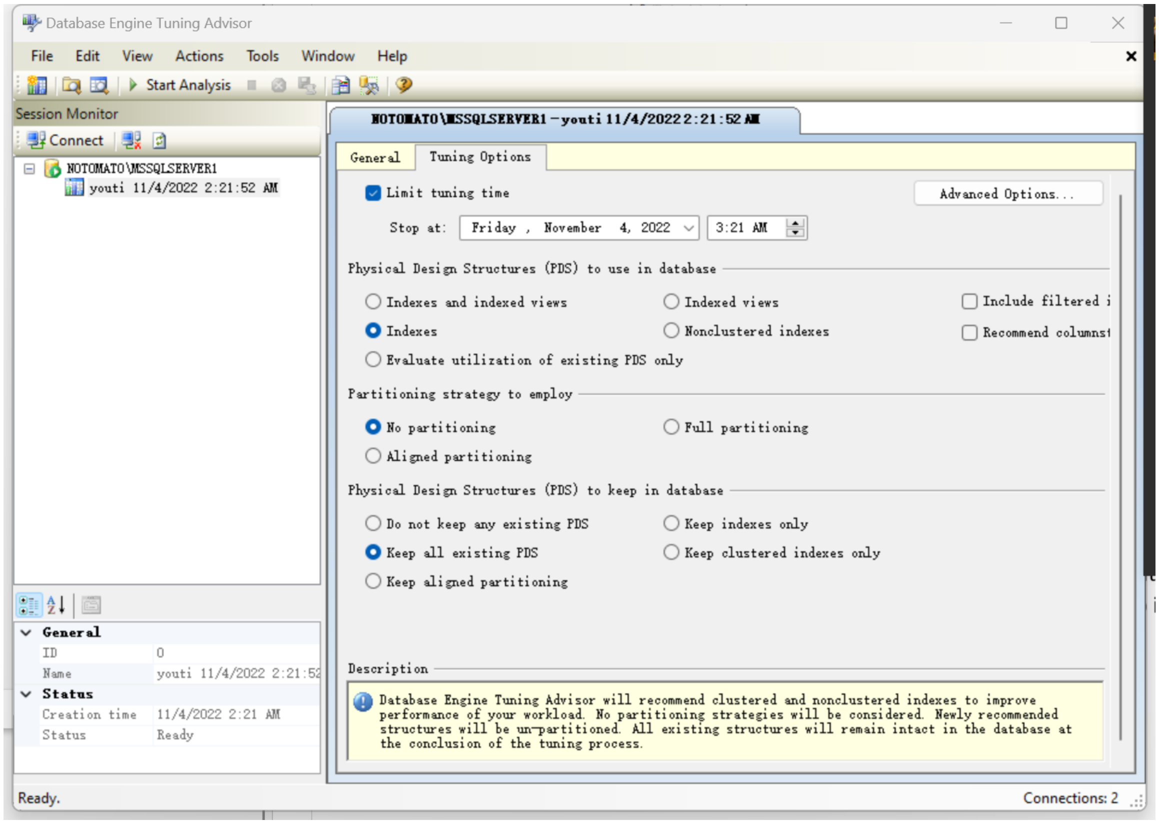
Status

Creation time	11/4/2022 2:21 AM
Status	Ready

Ready.

Connections: 2

- Tuning Options can be edited



NOTOMATO\MSSQLSERVER1
youti 11/4/2022 2:21:52 AM

General
ID 0
Name youti 11/4/2022 2:21:52
Status
Creation time 11/4/2022 2:21 AM
Status Ready

NOTOMATO\MSSQLSERVER1 – youti 11/4/2022 2:21:52 AM

General | Tuning Options

☑ Limit tuning time

Stop at: Friday , November 4, 2022 3:21 AM

Physical Design Structures (PDS) to use in database

○ Indexes and indexed views ○ Indexed views □ Include filtered i
● Indexes ○ Nonclustered indexes □ Recommend columnst
○ Evaluate utilization of existing PDS only

Partitioning strategy to employ

● No partitioning ○ Full partitioning
○ Aligned partitioning

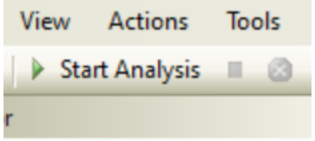
Physical Design Structures (PDS) to keep in database

○ Do not keep any existing PDS ○ Keep indexes only
● Keep all existing PDS ○ Keep clustered indexes only
○ Keep aligned partitioning

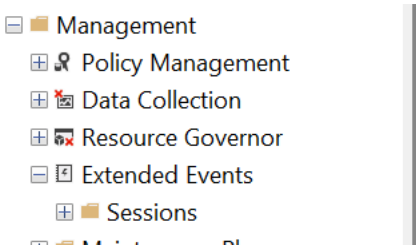
Description

Database Engine Tuning Advisor will recommend clustered and nonclustered indexes to improve performance of your workload. No partitioning strategies will be considered. Newly recommended structures will be un-partitioned. All existing structures will remain intact in the database at the conclusion of the tuning process.

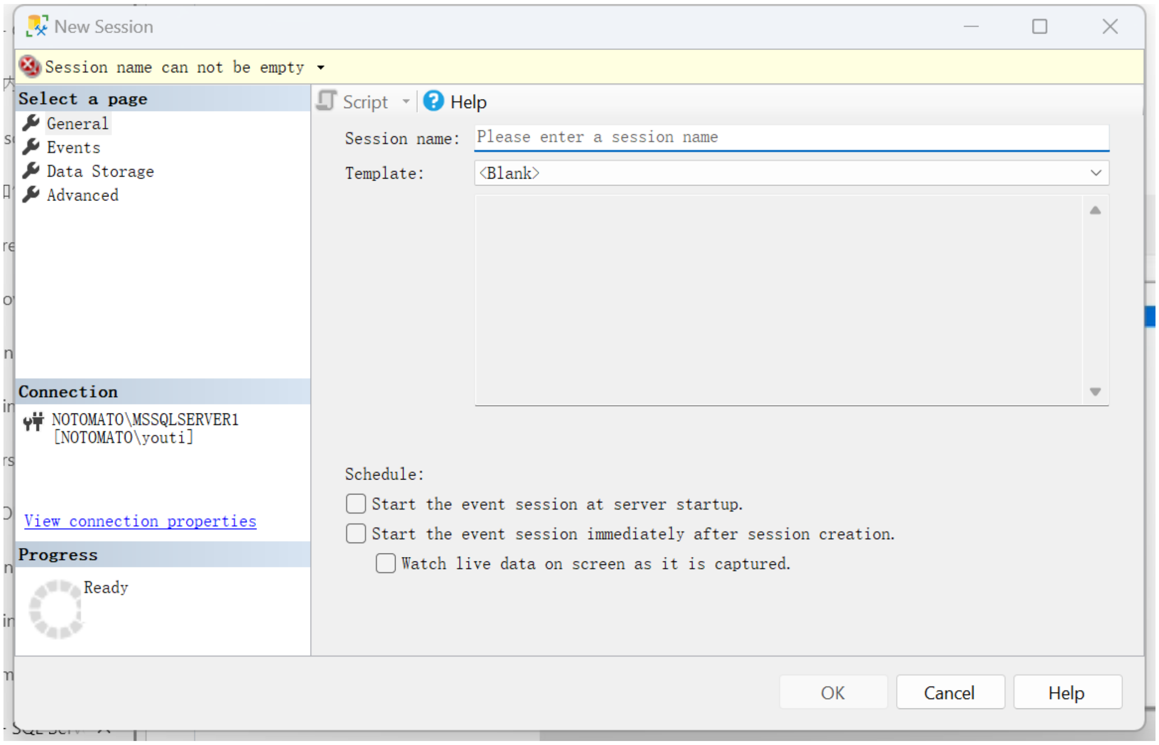
- And then can click Start Analysis



Extended Events:



- Then adjust the new session settings



Session name can not be empty

Session name: Please enter a session name

Template: <Blank>

Schedule:

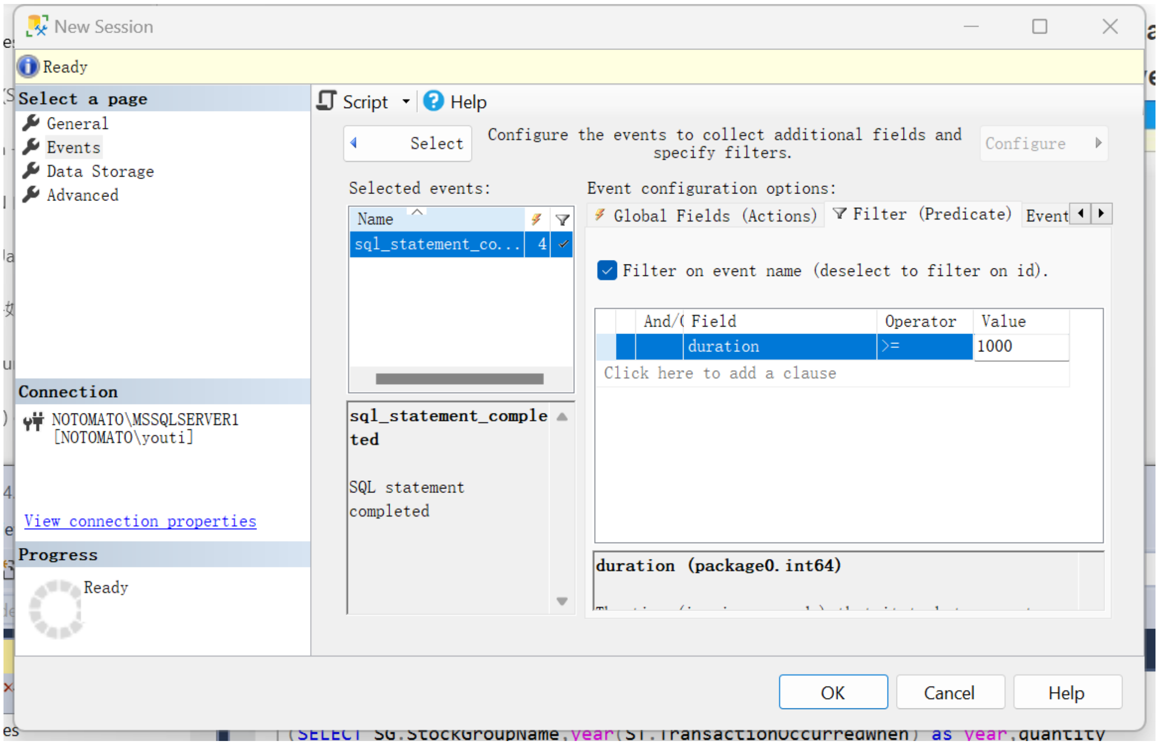
□ Start the event session at server startup.

□ Start the event session immediately after session creation.

□ Watch live data on screen as it is captured.

NOTOMATO\MSSQLSERVER1 [NOTOMATO\youti]

Ready



Ready

Configure the events to collect additional fields and specify filters.

Selected events:

Name		
sql_statement_co...	4	✓

sql_statement_completed

SQL statement completed

Event configuration options:

Global Fields (Actions) | Filter (Predicate) | Event

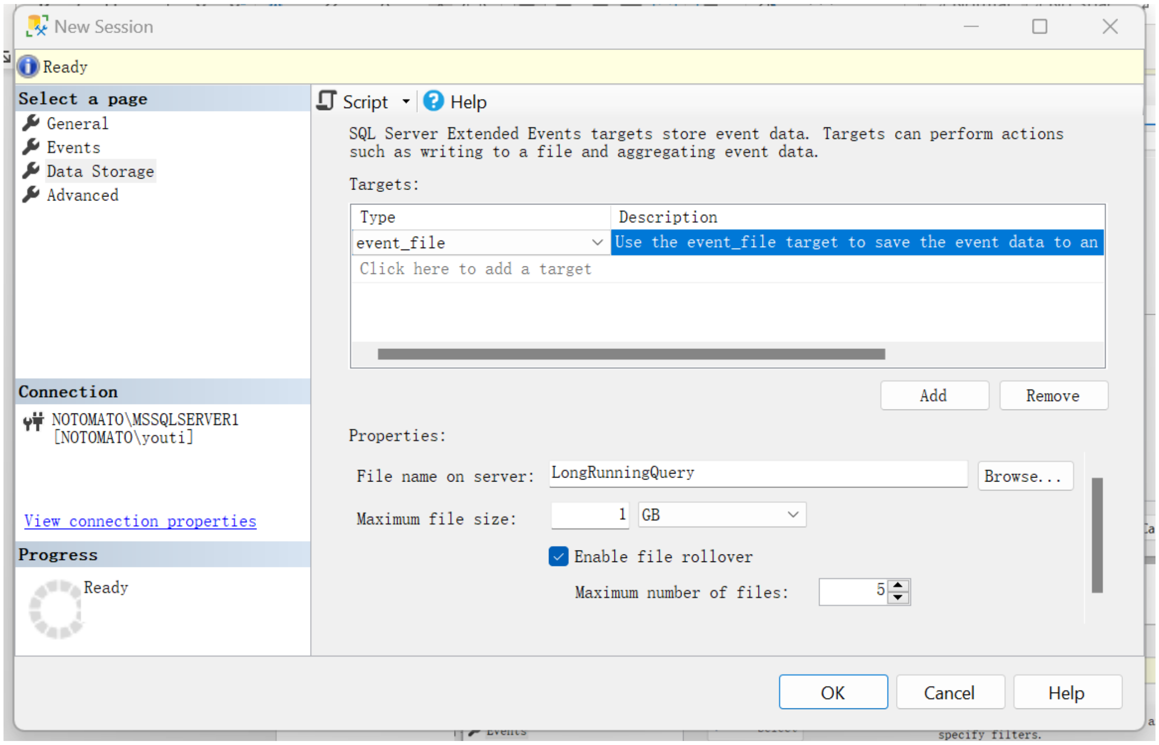
☑ Filter on event name (deselect to filter on id).

And/(	Field	Operator	Value
	duration	>=	1000
Click here to add a clause			

duration (package0.int64)

NOTOMATO\MSSQLSERVER1 [NOTOMATO\youti]

Ready



Ready

NOTOMATO\MSSQLSERVER1
[NOTOMATO\youti]

Ready

SQL Server Extended Events targets store event data. Targets can perform actions such as writing to a file and aggregating event data.

Targets:

Type	Description
event_file	Use the event_file target to save the event data to an
Click here to add a target	

Properties:

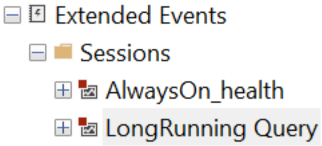
File name on server: LongRunningQuery

Maximum file size: 1 GB

☑ Enable file rollover

Maximum number of files: 5

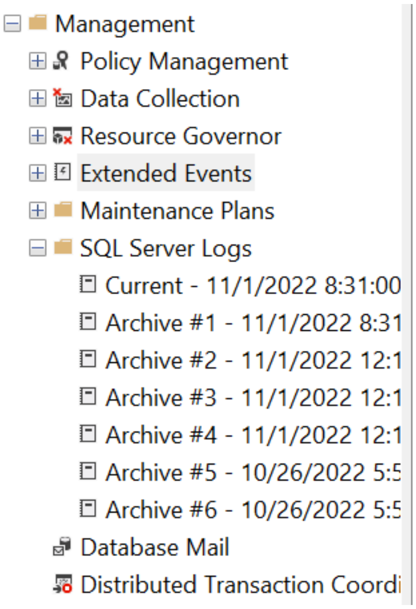
- Click 'OK' now you can see this session at 'Extended Event'>'Session'



```
⊟ Extended Events
  ⊟ Sessions
    ⊞ AlwaysOn_health
    ⊞ LongRunning Query
```

Logs:

- Will help user troubleshooting the issue

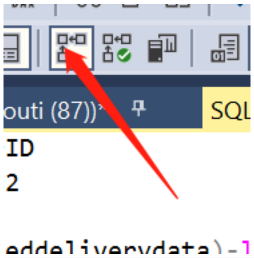


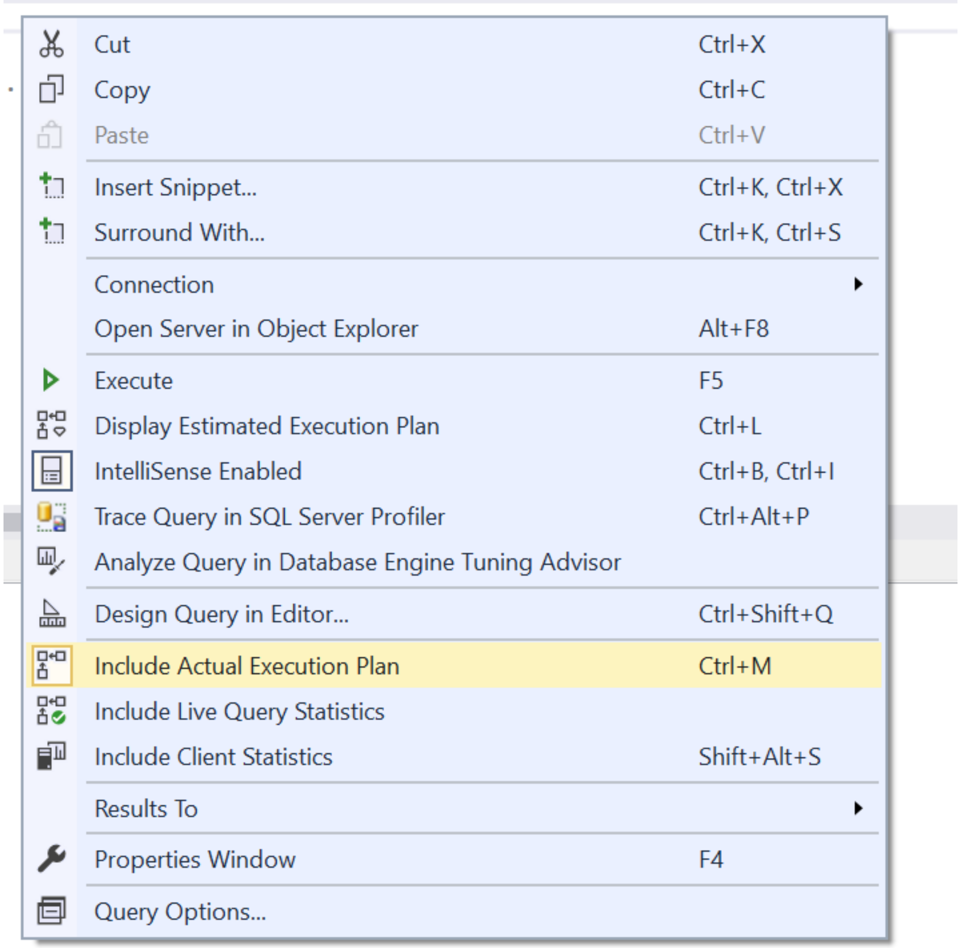
```
⊟ Management
  ⊞ Policy Management
  ⊞ Data Collection
  ⊞ Resource Governor
  ⊞ Extended Events
  ⊞ Maintenance Plans
  ⊟ SQL Server Logs
      Current - 11/1/2022 8:31:00
      Archive #1 - 11/1/2022 8:31
      Archive #2 - 11/1/2022 12:1
      Archive #3 - 11/1/2022 12:1
      Archive #4 - 11/1/2022 12:1
      Archive #5 - 10/26/2022 5:5
      Archive #6 - 10/26/2022 5:5
    Database Mail
    Distributed Transaction Coordi
```

DMV:

Dynamic Management Views allow for quick access to the internals for SQL Server. Usually starts with 'sys.'

Execution Plan:




Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Insert Snippet... Ctrl+K, Ctrl+X
Surround With... Ctrl+K, Ctrl+S
Connection
Open Server in Object Explorer Alt+F8
Execute F5
Display Estimated Execution Plan Ctrl+L
IntelliSense Enabled Ctrl+B, Ctrl+I
Trace Query in SQL Server Profiler Ctrl+Alt+P
Analyze Query in Database Engine Tuning Advisor
Design Query in Editor... Ctrl+Shift+Q
Include Actual Execution Plan Ctrl+M
Include Live Query Statistics
Include Client Statistics Shift+Alt+S
Results To
Properties Window F4
Query Options...

Assignments 30 - 32 are group assignments.

30. Write a short essay talking about a scenario: Good news everyone! We (Wide World Importers) just brought out a small company called "Adventure works"! Now that bike shop is our sub-company. The first thing of all works pending would be to merge the

user logon information, person information (including emails, phone numbers) and products (of course, add category, colors) to WWI database. Include screenshot, mapping and query.

- First, add new people in [WideWorldImporters].[Application].[People]:

  ```
  SELECT * INTO person_info
  FROM(
  SELECT a.FirstName,a.LastName,b.PhoneNumber,c.EmailAddress,d.PasswordHash
  FROM AdventureWorks2019.Person.Person a
  left join AdventureWorks2019.Person.PersonPhone b
  on a.BusinessEntityID = b.BusinessEntityID
  left join AdventureWorks2019.Person.EmailAddress c
  on a.BusinessEntityID =c.BusinessEntityID
  left join AdventureWorks2019.Person.Password d
  on a.BusinessEntityID = d.BusinessEntityID) AS tt

  SELECT * FROM person_info

  SELECT  * INTO product_info
  FROM(
  SELECT a.color,b.ProductSubcategoryID
  FROM AdventureWorks2019.Production.Product a
  left join AdventureWorks2019.Production.ProductSubcategory b
  on a.ProductSubcategoryID = b.ProductSubcategoryID) AS tb

  SELECT * FROM product_info
  ```

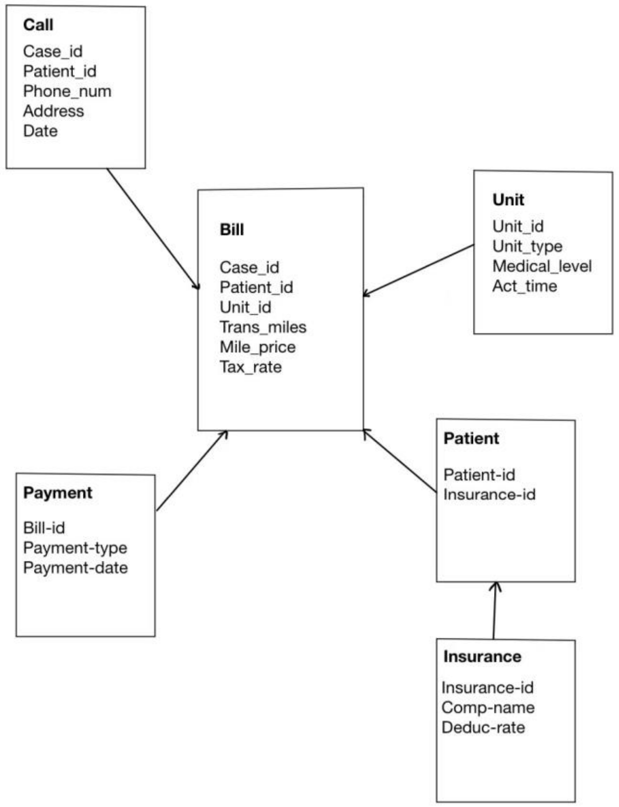
- Add product info:

  ```
  INTERT INTO [WideWorldImporters].[Warehouse].[StockItems](
  [StockItemID]
      ,[StockItemName]
      ,[SupplierID]
      ,[ColorID]
     )
  SELECT * FROM [AdventureWorks2019].[Production].[Product]
  ```

- Also update in [WideWorldImporters].[Warehouse].[StockGroups]:

  ```
  INSERT INTO [WideWorldImporters].[Warehouse].[StockGroups]
  VALUE( 11, ‘Bike’, 0,’ 2022-11-01 00:00:00.0000000’,’ 9999-12-31
  23:59:59.9999999’)
  ```

31. Database Design: OLTP db design request for EMS business: when people call 911 for medical emergency, 911 will dispatch UNITs to the given address. A UNIT means a crew on an apparatus (Fire Engine, Ambulance, Medic Ambulance, Helicopter, EMS supervisor). A crew member would have a medical level (EMR, EMT, A-EMT, Medic). All the treatments provided on scene are free. If the patient needs to be transported, that's where the bill comes in. A bill consists of Units dispatched (Fire Engine and EMS Supervisor are free), crew members provided care (EMRs and EMTs are free), Transported miles from the scene to the hospital (Helicopters have a much higher rate, as you can image) and tax (Tax rate is 6%). Bill should be sent to the patient insurance company first. If there is a deductible, we send the unpaid bill to the patient only. Don't forget about patient information, medical nature and bill paying status.



32. Remember the discussion about those two databases from the class, also remember, those data models are not perfect. You can always add new columns (but not alter or drop columns) to any tables. Suggesting adding Ingested DateTime and Surrogate Key

columns. Study the Wide World Importers DW. Think the integration schema is the ODS. Come up with a TSQL Stored Procedure driven solution to move the data from WWI database to ODS, and then from the ODS to the fact tables and dimension tables. By the way, WWI DW is a galaxy schema db. Requirements:

a. Luckly, we only start with 1 fact: Purchase. Other facts can be ignored for now.

b. Add a new dimension: Country of Manufacture. It should be given on top of Stock Items.

c. Write script(s) and stored procedure(s) for the entire ETL from WWI db to DW.

```
SELECT
        JSON_VALUE(CustomFields, '$.CountryOfManufacture') AS country
FROM
        Warehouse.StockItems
```