

# 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	Company Name	PhoneNumber	FaxNumber
34	Prem Prabhu	(423) 555-0108	(423) 555-0102	NULL	NULL	NULL
35	Sunita Jadhav	(423) 555-0108	(423) 555-0101	NULL	NULL	NULL
36	Marcos Costa	(252) 555-0101	(252) 555-0106	NULL	NULL	NULL
37	Matheus Oliveira	(252) 555-0101	(252) 555-0107	NULL	NULL	NULL
38	Eliza Soderberg	(201) 555-0106	(201) 555-0101	NULL	NULL	NULL
39	Sara Karlsson	(201) 555-0106	(201) 555-0100	NULL	NULL	NULL
40	Donald Jones	(605) 555-0101	(605) 555-0101	NULL	NULL	NULL
41	Sharon Graham	(605) 555-0101	(605) 555-0106	NULL	NULL	NULL
42	Hai Dam	(218) 555-0108	(218) 555-0101	NULL	NULL	NULL
43	Thanh Dinh	(218) 555-0108	(218) 555-0104	NULL	NULL	NULL
44	Hubert Helms	(415) 555-0104	(415) 555-0103	NULL	NULL	NULL
45	Donald Small	(415) 555-0104	(415) 555-0105	NULL	NULL	NULL
46	Waldemar Fisar	(308) 555-0101	(308) 555-0100	Tailspin Toys (...)	(308) 555-01...	(308) 555-...
47	Laimonis Berzi...	(308) 555-0101	(308) 555-0100	Tailspin Toys (...)	(308) 555-01...	(308) 555-...
48	Lorena Cindric	(406) 555-0101	(406) 555-0100	Tailspin Toys (...)	(406) 555-01...	(406) 555-...
49	Hung Van Gro...	(406) 555-0101	(406) 555-0100	Tailspin Toys (...)	(406) 555-01...	(406) 555-...
50	Bhaargav Ram...	(480) 555-0101	(480) 555-0100	Tailspin Toys (...)	(480) 555-01...	(480) 555-...
51	Mudar Jevtic	(480) 555-0101	(480) 555-0100	Tailspin Toys (...)	(480) 555-01...	(480) 555-...
52	Daniel Roman	(316) 555-0101	(316) 555-0100	Tailspin Toys (...)	(316) 555-01...	(316) 555-...
53	Leyla Radnia	(316) 555-0101	(316) 555-0100	Tailspin Toys (...)	(316) 555-01...	(316) 555-...
54	Johanna Huiting	(212) 555-0101	(212) 555-0100	Tailspin Toys (...)	(212) 555-01...	(212) 555-...
55	Robert Ruutli	(212) 555-0101	(212) 555-0100	Tailspin Toys (...)	(212) 555-01...	(212) 555-...
56	Biswajeet Thak...	(701) 555-0101	(701) 555-0100	Tailspin Toys (...)	(701) 555-01...	(701) 555-...
57	Radha Shah	(701) 555-0101	(701) 555-0100	Tailspin Toys (...)	(701) 555-01...	(701) 555-...
58	Kalidas Nadar	(423) 555-0101	(423) 555-0100	Tailspin Toys (...)	(423) 555-01...	(423) 555-...
59	Filip Nedvidek	(423) 555-0101	(423) 555-0100	Tailspin Toys (...)	(423) 555-01...	(423) 555-...
60	Kanti Kotadia	(303) 555-0101	(303) 555-0100	Tailspin Toys (...)	(303) 555-01...	(303) 555-...

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

	CustomerName	PhoneNumber	PhoneNumber
1	Tailspin Toys (Head Office)	(308) 555-0100	(308) 555-0100
2	Tailspin Toys (Sylvanite, MT)	(406) 555-0100	(406) 555-0100
3	Tailspin Toys (Peeples Valley, AZ)	(480) 555-0100	(480) 555-0100
4	Tailspin Toys (Medicine Lodge, KS)	(316) 555-0100	(316) 555-0100
5	Tailspin Toys (Gasport, NY)	(212) 555-0100	(212) 555-0100
6	Tailspin Toys (Jessie, ND)	(701) 555-0100	(701) 555-0100
7	Tailspin Toys (Frankewing, TN)	(423) 555-0100	(423) 555-0100
8	Tailspin Toys (Bow Mar, CO)	(303) 555-0100	(303) 555-0100
9	Tailspin Toys (Netcong, NJ)	(201) 555-0100	(201) 555-0100
10	Tailspin Toys (Wimbledon, ND)	(701) 555-0100	(701) 555-0100
11	Tailspin Toys (Devault, PA)	(215) 555-0100	(215) 555-0100
12	Tailspin Toys (Biscay, MN)	(218) 555-0100	(218) 555-0100
13	Tailspin Toys (Stonefort, IL)	(217) 555-0100	(217) 555-0100
14	Tailspin Toys (Long Meadow, MD)	(240) 555-0100	(240) 555-0100
15	Tailspin Toys (Batson, TX)	(210) 555-0100	(210) 555-0100
16	Tailspin Toys (Coney Island, MO)	(314) 555-0100	(314) 555-0100
17	Tailspin Toys (East Fultonham, OH)	(216) 555-0100	(216) 555-0100
18	Tailspin Toys (Goffstown, NH)	(603) 555-0100	(603) 555-0100
19	Tailspin Toys (Lemeta, AK)	(907) 555-0100	(907) 555-0100
20	Tailspin Toys (College Place, WA)	(206) 555-0100	(206) 555-0100
21	Tailspin Toys (Tresckow, PA)	(215) 555-0100	(215) 555-0100
22	Tailspin Toys (Ward Ridge, FL)	(239) 555-0100	(239) 555-0100
23	Tailspin Toys (Ikatan, AK)	(907) 555-0100	(907) 555-0100
24	Tailspin Toys (Dundarrach, NC)	(252) 555-0100	(252) 555-0100
25	Tailspin Toys (Avenal, CA)	(209) 555-0100	(209) 555-0100
26	Tailspin Toys (Hedrick, IA)	(319) 555-0100	(319) 555-0100
27	Tailspin Toys (Bowlus, MN)	(218) 555-0100	(218) 555-0100

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

CustomerID
------------

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

	Description	Quantity
1	"The Gu" red shirt XML tag t-shirt (White) XS	139407
2	"The Gu" red shirt XML tag t-shirt (Black) 4XL	124697
3	"The Gu" red shirt XML tag t-shirt (White) 5XL	109756
4	"The Gu" red shirt XML tag t-shirt (White) XXS	106377
5	"The Gu" red shirt XML tag t-shirt (Black) XL	59935
6	"The Gu" red shirt XML tag t-shirt (White) M	52584
7	Black and orange glass with care despatch tape 48...	40878
8	Tape dispenser (Red)	40874
9	USB food flash drive - dim sum 10 drive variety pack	204
10	USB food flash drive - dessert 10 drive variety pack	196
11	USB food flash drive - chocolate bar	111
12	RC vintage American toy coupe with remote control ...	106
13	USB food flash drive - sushi roll	104
14	USB food flash drive - cookie	102
15	USB food flash drive - pizza slice	102
16	USB food flash drive - donut	101
17	USB food flash drive - shrimp cocktail	99

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

	StockItemName	(No column name)
1	Chocolate frogs 250g	20
2	Tape dispenser (Red)	20
3	Tape dispenser (Blue)	21
4	Chocolate sharks 250g	21
5	Chocolate beetles 250g	22
6	Tape dispenser (Black)	22
7	Chocolate echidnas 250g	23
8	Air cushion machine (Blue)	26
9	USB rocket launcher (Gray)	26
10	Furry animal socks (Pink) L	27
11	Furry animal socks (Pink) M	27
12	Furry animal socks (Pink) S	27
13	Furry animal socks (Pink) XL	28
14	USB missile launcher (Green)	28
15	USB food flash drive - donut	28
16	USB food flash drive - cookie	29

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

	StockItemName
1	Chocolate frogs 250g
2	Novelty chilli chocolates 500g
3	Novelty chilli chocolates 250g
4	Chocolate echidnas 250g
5	Chocolate beetles 250g
6	White chocolate snow balls 250g
7	White chocolate moon rocks 250g
8	Chocolate sharks 250g

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

	StateProvinceName	Hours
1	New Hampshire	254
2	Wyoming	210
3	Indiana	181
4	Missouri	173
5	Arizona	166
6	Nebraska	166
7	Massachusetts[E]	163
8	Michigan	162
9	North Carolina	161
10	Virginia	160
11	South Carolina	156
12	West Virginia	154
13	California	152
14	Idaho	151
15	Puerto Rico (US Territory)	145

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

	StateProvinceName	Hours	Month
1	Alabama	263	1
2	Alabama	62	2
3	Alabama	96	3
4	Alabama	117	4
5	Alabama	88	5
6	Alabama	68	6
7	Alabama	288	7
8	Alabama	34	8
9	Alabama	93	9
10	Alabama	142	10
11	Alabama	286	11
12	Alabama	132	12
13	Alaska	34	1
14	Alaska	35	2
15	Alaska	36	3
16	Alaska	61	4

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

	Description
1	"The Gu" red shirt XML tag t-shirt (White) M
2	Black and orange glass with care despatch tape 48...
3	Tape dispenser (Red)
4	"The Gu" red shirt XML tag t-shirt (White) 5XL
5	Shipping carton (Brown) 305x305x305mm
6	"The Gu" red shirt XML tag t-shirt (Black) 4XL
7	"The Gu" red shirt XML tag t-shirt (White) XS
8	"The Gu" red shirt XML tag t-shirt (Black) XL
9	"The Gu" red shirt XML tag t-shirt (White) XXS

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.

	CustomerName	PhoneNumber	Total Mugs Purchased
1	Aakriti Byrraju	(216) 555-0100	9
2	Abel Spirlea	(218) 555-0100	7
3	Abel Tatarescu	(217) 555-0100	6
4	Adrian Andreasson	(217) 555-0100	6
5	Agrita Abele	(206) 555-0100	4
6	Aive Petrov	(316) 555-0100	7
7	Akhil Nadar	(479) 555-0100	9
8	Allan Mannik	(218) 555-0100	6
9	Alvin Bollinger	(270) 555-0100	8
10	Amarasimha Vinjamuri	(339) 555-0100	6
11	Amet Shergill	(303) 555-0100	6
12	Amrita Ganguly	(787) 555-0100	4
13	Anand Mudaliyar	(206) 555-0100	2
14	Anca Gogean	(209) 555-0100	5
15	Andris Saulitis	(210) 555-0100	5
16	Anindya Ghatak	(219) 555-0100	8
17	Anna Gyarmathi	(252) 555-0100	9

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

	CityName	ValidFrom
1	Adrian	2015-07-01 16:00:00.0000000
2	Carlton	2015-07-01 16:00:00.0000000
3	East Smithfield	2015-07-01 16:00:00.0000000
4	Fairfax	2015-07-01 16:00:00.0000000
5	Laupahoehoe	2015-07-01 16:00:00.0000000
6	McWhorter	2015-07-01 16:00:00.0000000
7	Norborne	2015-07-01 16:00:00.0000000
8	North Granby	2015-07-01 16:00:00.0000000
9	Pondosa	2015-07-01 16:00:00.0000000
10	Richvale	2015-07-01 16:00:00.0000000
11	Springville	2015-07-01 16:00:00.0000000
12	Throop	2015-07-01 16:00:00.0000000
13	Urbancrest	2015-07-01 16:00:00.0000000

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.

	Description	DeliveryAddressLine1	DeliveryAddressLine2	StateProvinceName	CityName	CountryName	CustomerName	Contact Person Name	PhoneNumb
1	Developer joke mug - understanding recursion requi...	Unit 212	1837 Vasiljevic Boulevard	Pennsylvania	Rose Tree	United States	Wingtip Toys (Rose Tree, PA)	Lan Chu	(215) 555-01
2	Developer joke mug - that's a hardware problem (Bl...	Unit 193	583 Alun Road	Puerto Rico (US Territory)	Rosa Sánchez	United States	Wingtip Toys (Rosa Sánchez, PR)	Selma Seppanen	(787) 555-01
3	Developer joke mug - there are 10 types of people in ...	Suite 258	771 Kidambi Road	Texas	Lavon	United States	Tailspin Toys (Lavon, TX)	Alba Ponce	(210) 555-01
4	DBA joke mug - two types of DBAs (White)	Suite 185	1492 Shah Road	Illinois	Stonefort	United States	Tailspin Toys (Stonefort, IL)	Razeena Hosseini	(217) 555-01
5	Developer joke mug - (hip, hip, array) (Black)	Shop 20	1046 Saucier Road	Ohio	East Fultonham	United States	Wingtip Toys (East Fultonham, OH)	Masa Buecek	(216) 555-01
6	DBA joke mug - mind if I join you? (White)	Unit 50	519 Jogi Street	Puerto Rico (US Territory)	Indios	United States	Tailspin Toys (Indios, PR)	Roxane Rastgu	(787) 555-01
7	DBA joke mug - you might be a DBA if (Black)	Shop 11	965 Horackova Lane	New York	Greig	United States	Kertu Sokk	Kertu Sokk	(212) 555-01
8	DBA joke mug - daaaaaa-ta (White)	Suite 258	771 Kidambi Road	Texas	Lavon	United States	Tailspin Toys (Lavon, TX)	Alba Ponce	(210) 555-01
9	Developer joke mug - when your hammer is C++ (W...	Unit 15	685 Pavel Road	Alabama	Southside	United States	Manca Hrastovsek	Manca Hrastovsek	(205) 555-01
10	Developer joke mug - when your hammer is C++ (Bl...	Shop 104	1889 Sminov Road	South Carolina	Sans Souci	United States	Tailspin Toys (Sans Souci, SC)	Coralie Emond	(803) 555-01
11	DBA joke mug - you might be a DBA if (White)	Suite 8	1761 Nastase Avenue	Pennsylvania	Bell Acres	United States	Wingtip Toys (Bell Acres, PA)	Abhra Thakur	(215) 555-01
12	DBA joke mug - SELECT caffeine FROM mug (Black)	Shop 18	1528 Dikshit Crescent	Oklahoma	Seiling	United States	Ebru Erdogan	Ebru Erdogan	(405) 555-01
13	Developer joke mug - that's a hardware problem (Bl...	Unit 148	1161 Chang Lane	New Jersey	Absecon	United States	Tailspin Toys (Absecon, NJ)	Sang Tran	(201) 555-01
14	Developer joke mug - fun was unexpected at this tim...	Unit 111	983 Cavalcante Street	New York	Lime Lake	United States	Wingtip Toys (Lime Lake, NY)	Tanja Penttila	(212) 555-01
15	IT joke mug - hardware: part of the computer that can...	Unit 32	917 Morgan Boulevard	California	Glen Avon	United States	Tailspin Toys (Glen Avon, CA)	Karie Mercier	(209) 555-01
16	Developer joke mug - when your hammer is C++ (Bl...	Suite 258	771 Kidambi Road	Texas	Lavon	United States	Tailspin Toys (Lavon, TX)	Alba Ponce	(210) 555-01
17	Developer joke mug - when your hammer is C++ (W...	Suite 187	1110 Esch Boulevard	Texas	Musand	United States	Tailspin Toys (Musand, TX)	Nikolaie Velaz	(210) 555-01

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

	StockGroupID	Quantity Purchased	Quantity Sold	Remaining
1	1	4638	1168276	-1163638
2	4	7594938	1800072	5794866
3	3	1442	244010	-242568
4	9	1313	121360	-120047
5	2	7595794	2842918	4752876
6	8	587	395849	-395262
7	6	7597975	2181299	5416676
8	10	2710170	5838043	-3127873
9	7	1580	81057	-79477

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”.

	Description	CityName	Quantity Sold
1	10 mm Double sided bubble wrap 10m	Abbotsburg	170
2	Black and orange handle with care despatch tape 48...	Absecon	168
3	Black and orange fragile despatch tape 48mmx100m	Accomac	360
4	Shipping carton (Brown) 356x229x229mm	Aceitunas	300
5	Shipping carton (Brown) 356x356x279mm	Airport Drive	225
6	Shipping carton (Brown) 480x270x320mm	Akhiok	425
7	32 mm Anti static bubble wrap (Blue) 20m	Alcester	110
8	Red and white urgent despatch tape 48mmx75m	Alden Bridge	360
9	Shipping carton (Brown) 356x356x279mm	Alstead	200
10	"The Gu" red shirt XML tag t-shirt (Black) XXL	Amado	108
11	"The Gu" red shirt XML tag t-shirt (Black) M	Amado	108
12	Shipping carton (Brown) 457x279x279mm	Amanda Park	225
13	Red and white urgent heavy despatch tape 48mmx1...	Andrix	240
14	Black and orange fragile despatch tape 48mmx75m	Annamoriah	360
15	3 kg Courier post bag (White) 300x190x95mm	Antares	250
16	"The Gu" red shirt XML tag t-shirt (Black) XXS	Antonito	168
17	Shipping carton (Brown) 356x229x229mm	Arbor Vitae	225

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

	InvoiceID
1	19
2	26
3	29
4	32
5	37
6	39
7	64
8	73
9	78
10	91
11	100
12	121
13	122
14	124
15	125
16	147
17	150

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

	StockItemName	Country of Manufacture
1	USB missile launcher (Green)	China
2	USB rocket launcher (Gray)	China
3	Office cube periscope (Black)	China
4	DBA joke mug - mind if I join you? (White)	China
5	DBA joke mug - mind if I join you? (Black)	China
6	DBA joke mug - daaaaaa-ta (White)	China
7	DBA joke mug - daaaaaa-ta (Black)	China
8	DBA joke mug - you might be a DBA if (White)	China
9	DBA joke mug - you might be a DBA if (Black)	China
10	DBA joke mug - it depends (White)	China
11	DBA joke mug - it depends (Black)	China
12	DBA joke mug - I will get you in order (White)	China
13	DBA joke mug - I will get you in order (Black)	China
14	DBA joke mug - SELECT caffeine FROM mug (White)	China
15	DBA joke mug - SELECT caffeine FROM mug (Black)	China
16	DBA joke mug - two types of DBAs (White)	China
17	DBA joke mug - two types of DBAs (Black)	China

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

	Country	Quantity
1	Japan	22365
2	China	2850885

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]

	GroupID	2013	2014	2015	2016	2017
1	1	276609	306077	328677	256913	NULL
2	3	65713	70384	77268	30645	NULL
3	4	486924	528096	558144	226908	NULL
4	9	32266	35403	38303	15388	NULL
5	2	767341	831573	889178	354826	NULL
6	8	107839	112845	125924	49241	NULL
7	6	588555	639315	677480	275949	NULL
8	10	1572415	1694778	1826433	744417	NULL
9	7	21328	23685	26048	9996	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	1	2	3	4	5	6	7	8	9	10
1	2013	276609	767341	65713	486924	NULL	588555	21328	107839	32266	1572415
2	2016	256913	354826	30645	226908	NULL	275949	9996	49241	15388	744417
3	2014	306077	831573	70384	528096	NULL	639315	23685	112845	35403	1694778
4	2015	328677	889178	77268	558144	NULL	677480	26048	125924	38303	1826433

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.

	InvoiceID	OrderID	OrderTotal
1	1	1	10
2	2	2	18
3	3	3	3
4	4	4	103
5	5	5	27
6	6	6	19
7	7	7	91
8	8	8	14
9	9	9	6
10	10	10	8
11	11	11	1
12	12	12	13
13	13	13	18
14	14	14	19

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.

	OrderID	OrderDate	OrderTotal	CustomerID
1	40790	2015-01-03	295	172
2	40791	2015-01-03	86	927
3	40792	2015-01-03	90	25
4	40793	2015-01-03	13	953
5	40794	2015-01-03	22	984
6	40795	2015-01-03	138	512
7	40796	2015-01-03	44	162
8	40797	2015-01-03	88	193
9	40798	2015-01-03	228	935
10	40799	2015-01-03	111	937
11	40800	2015-01-03	34	485
12	40801	2015-01-03	34	529
13	40802	2015-01-03	253	140
14	40803	2015-01-03	78	932
15	40804	2015-01-03	126	524

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	UnitPrice	Recommender
1	1	USB missile launcher (Green)	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	25.00	37.38
2	2	USB rocket launcher (Gray)	12	12	7	7	NULL	NULL	14	1	0	NULL	15.000	25.00	37.38
3	3	Office cube periscope (Black)	12	3	7	6	NULL	NULL	14	10	0	NULL	15.000	18.50	27.66
4	4	USB food flash drive - sushi roll	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
5	5	USB food flash drive - hamburger	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
6	6	USB food flash drive - hot dog	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
7	7	USB food flash drive - pizza slice	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
8	8	USB food flash drive - dim sum 10 drive variety ...	12	NULL	9	9	NULL	NULL	14	1	0	NULL	15.000	240.00	358.80
9	9	USB food flash drive - banana	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
10	10	USB food flash drive - chocolate bar	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
11	11	USB food flash drive - cookie	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
12	12	USB food flash drive - donut	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
13	13	USB food flash drive - shrimp cocktail	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
14	14	USB food flash drive - fortune cookie	12	NULL	7	7	NULL	NULL	14	1	0	NULL	15.000	32.00	47.84
15	15	USB food flash drive - dessert 10 drive variety pa...	12	NULL	9	9	NULL	NULL	14	1	0	NULL	15.000	240.00	358.80
16	16	DBA joke mug - mind if I join you? (White)	5	35	7	7	NULL	NULL	12	1	0	NULL	15.000	13.00	19.44

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",
  "UnitPackageld": "1",
  "OuterPackageld": [
    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",
  "UnitPackageld": "1",
  "OuterPackageld": "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,"1":276609,"2":767341,"3":65713,"4":486924,"5":588555,"7":21328,"8":107839,"9":32266,"10":1572415},{"Year":2016,"1":256913,"2":354826,"3":30645,"4":226908,"5":2759...
```

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 <Warehouse_YearQuantityStockGroup Year="2013" x0031 ="276609" x0032 ="767341" x0033 ="65713" x0034 ="486924" x0036 ="588555" x0037 =...
```

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.

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

Transactions is a unit of work performed within a database. For example, it can be a group of queries or SQL code bundled into one transaction.

Transactions follow the ACID principle, which stand for atomicity, consistency, isolation, and durability.

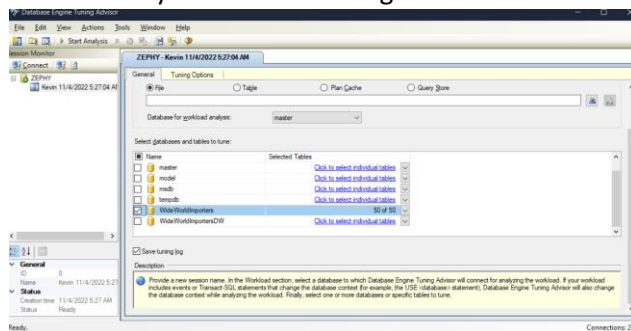
Locks are used to control concurrency transactions, which makes sure the action of one user of an database wouldn't affect another user.

Isolation levels are used to control the degree to which one transaction should be isolated from others. The five isolation levels are

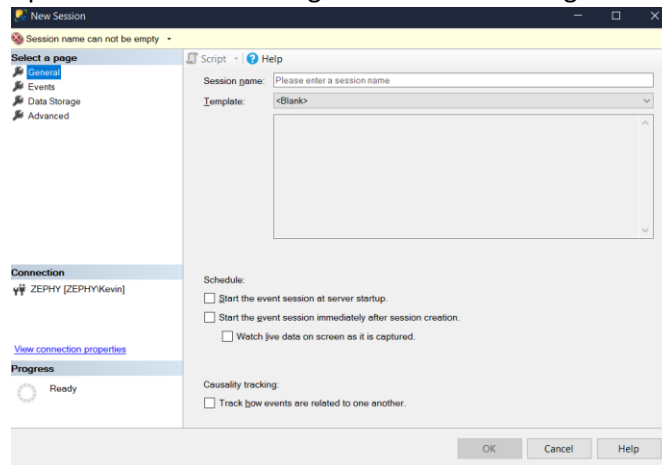
1. Read Uncommitted
2. Read Committed
3. Repeatable Read
4. Serializable
5. Snapshot

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: analyzes database and gives recommendation on improvements



## Extended events: performance monitoring and trouble shooting



DMV

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.

-- In the begining of the merge, we have to insert our bike products' info into the Warehouse.StockItems table. And then, we can assign these bike

-- products into their group in StockGroups table.

```
INSERT INTO Application.People (
    FullName, PreferredName, SearchName, IsPermittedToLogon,
    LogonName, PhoneNumber, EmailAddress, FaxNumber, LastEditedBy,
    ValidFrom, ValidTo)
SELECT
    aau.FullName, aau.PreferredName, aau.SearchName, aau.IsPermittedToLogon,
    aau.LogonName, aau.PhoneNumber, aau.EmailAddress, aau.FaxNumber,
    aau.LastEditedBy, aau.ValidFrom, aau.ValidTo
FROM [Adventure_works].Application.Users aau
```

--Insert basic new product info into WWI database

```
INSERT INTO Warehouse.StockItems (
    StockItemID, ColorID, StockItemName, Size, TaxRate, UnitPrice, Tags)
```

SELECT

```
aap.Product_id, aap.ColorID, aap.product_name,
aap.size, aap.TaxRate, aap.UnitPrice, aap.Category
FROM [Adventure_works].Application.Products aap
```

-- Assign product group into WWI stock\_group

-- Firstly, a new StockGroupID, Stockname, LastEditedBy, ValidFrom, ValidTo info need in the [WWI].Warehouse.StockGroups needed

```
INSERT INTO Warehouse.StockGroups(StockGroupID, StockGroupName, LastEditedBy,
ValidFrom, ValidTo)
```

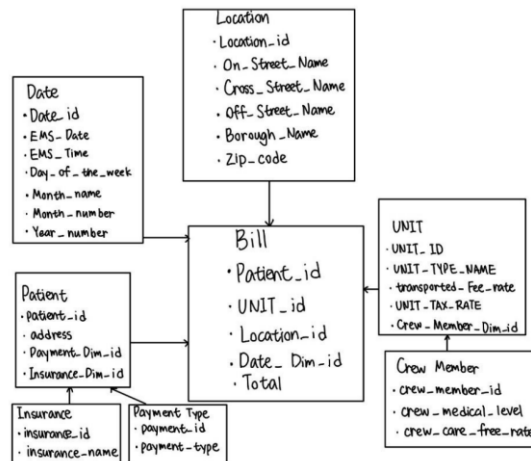
```
VALUE (11, 'Bike', 1, '2022-08-03 00:00:00.0000000', '9999-12-31 23:59:59.9999999')
```

--Then we can assign different bike products into their groups in the StockItemStockGroup table

```
INSERT INTO Warehouse.StockItemStockGroups
```

```
SELECT DISTINCT aap.Product_id, 11, 1, '2022-08-03 00:00:00.0000000'
FROM [Adventure_works].Application.Products aap
```

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.

```
ALTER TABLE WideWorldImportersDW.Dimension.[Stock Item]
ADD [Country of Manufacture] NVARCHAR(20)
UPDATE WideWorldImportersDW.Dimension.[Stock Item]
SET [Country of Manufacture] =
JSON_VALUE(SI.CustomFields, '$.CountryOfManufacture')
FROM WideWorldImporters.Warehouse.StockItems AS SI
WHERE [Stock Item Key] = SI.StockItemID
SELECT [Country of Manufacture] FROM WideWorldImportersDW.Dimension.[Stock Item]
CREATE PROCEDURE dbo.ExtractOrder
AS
SELECT
C.DeliveryCityID,
O.CustomerID,
OL.StockItemID ,
O.OrderDate,
CONVERT(DATE,O.PickingCompletedWhen) AS [Picked Date Key],
O.SalespersonPersonID,
O.PickedByPersonID,
O.OrderID ,
O.BackorderOrderID ,
SI.StockItemName,
PT.PackageTypeName,
OL.Quantity,
OL.UnitPrice,
OL.TaxRate,
IL.TaxAmount
FROM WideWorldImporters.Sales.Orders AS O
JOIN WideWorldImporters.Sales.OrderLines AS OL
ON O.OrderID = OL.OrderID
JOIN WideWorldImporters.Sales.Invoices AS I
ON I.OrderID = O.OrderID
JOIN WideWorldImporters.Sales.InvoiceLines AS IL
ON IL.InvoiceID = I.InvoiceID AND IL.StockItemID = OL.StockItemID
JOIN WideWorldImporters.Warehouse.StockItems AS SI
ON SI.StockItemID = OL.StockItemID
JOIN WideWorldImporters.Warehouse.PackageTypes AS PT
ON PT.PackageTypeID = OL.PackageTypeID
JOIN WideWorldImporters.Sales.Customers AS C
ON C.CustomerID = O.CustomerID
GO
CREATE TABLE WideWorldImportersDW.Integration.ExtractOrder_Staging(
DeliveryCityID INT,
CustomerID INT,
StockItemID INT ,
OrderDate DATE ,
[Picked Date Key] DATE ,
SalespersonPersonID INT ,
PickedByPersonID INT ,
OrderID INT,
BackorderOrderID INT ,
StockItemName NVARCHAR(MAX),
PackageTypeName NVARCHAR(50),
Quantity INT,
UnitPrice DECIMAL(18,2),
TaxRate DECIMAL(18,3),
TaxAmount DECIMAL(18,2)
```

```

);
INSERT INTO WideWorldImportersDW.Integration.ExtractOrder_Staging
EXEC dbo.ExtractOrder ;
CREATE PROCEDURE dbo.TrasformOrder
AS
SELECT
DeliveryCityID,
CustomerID,
StockItemID,
OrderDate,
[Picked Date Key],
SalespersonPersonID,
PickedByPersonID,
OrderID,
BackorderOrderID,
StockItemName,
PackageTypeName,
Quantity,
UnitPrice,
TaxRate,
Quantity*UnitPrice AS [Total Excluding Tax],
TaxAmount,
Quantity*UnitPrice + TaxAmount AS [Total Including Tax]
FROM WideWorldImportersDW.Integration.ExtractOrder_Staging
GO
CREATE TABLE WideWorldImportersDW.Integration.TransformOrder_Staging(
DeliveryCityID INT,
CustomerID INT,
StockItemID INT ,
OrderDate DATE ,
[Picked Date Key] DATE ,
SalespersonPersonID INT ,
PickedByPersonID INT ,
OrderID INT,
BackorderOrderID INT ,
StockItemName NVARCHAR(MAX),
PackageTypeName NVARCHAR(50),
Quantity INT,
UnitPrice DECIMAL(18,2),
TaxRate DECIMAL(18,3),
[Total Excluding Tax] DECIMAL(18,3),
TaxAmount DECIMAL(18,2),
[Total Including Tax] DECIMAL(18,3)
);
INSERT INTO WideWorldImportersDW.Integration.TransformOrder_Staging
EXEC dbo.TrasformOrder;
DROP TABLE WideWorldImportersDW.Integration.ExtractOrder_Staging;
CREATE PROCEDURE dbo.LoadOrder
AS
INSERT INTO WideWorldImportersDW.Fact.[Order](
[City Key],
[Customer Key],
[Stock Item Key],
[Order Date Key],
[Picked Date Key],
[Salesperson Key],
[Picker Key],
[WWI Order ID],

```

```

[WWI Backorder ID],
[Description],
[Package],
Quantity,
[Unit Price],
[Tax Rate],
[Total Excluding Tax],
[Tax Amount],
[Total Including Tax],
[Lineage Key])
SELECT
City.[City Key],
ISNULL(C.[Customer Key],0) AS [Customer Key],
SI.[Stock Item Key],
OrderDate AS [Order Date Key],
[Picked Date Key],
E.[Employee Key] AS [Salesperson Key],
EE.[Employee Key] AS [Picker Key],
OrderID AS [WWI Order ID],
BackorderOrderID AS [WWI Backorder ID],
StockItemName AS [Description],
PackageTypeName AS [Package],
Quantity,
UnitPrice AS [Unit Price],
TaxRate AS [Tax Rate],
[Total Excluding Tax],
TaxAmount AS [Tax Amount],
[Total Including Tax],
9
FROM WideWorldImportersDW.Integration.TransformOrder_Staging AS A
LEFT JOIN WideWorldImportersDW.Dimension.Customer AS C
ON A.CustomerID = C.[WWI Customer ID] AND C.[Valid To]='9999-12-31
23:59:59.9999999'
LEFT JOIN WideWorldImportersDW.Dimension.[Stock Item] AS SI
ON SI.[WWI Stock Item ID] = A.StockItemID AND SI.[Valid To] = '9999-12-31
23:59:59.9999999'
LEFT JOIN WideWorldImportersDW.Dimension.Employee AS E
ON E.[WWI Employee ID] = A.SalespersonPersonID AND E.[Valid To] = '9999-12-
31 23:59:59.9999999'
LEFT JOIN WideWorldImportersDW.Dimension.Employee AS EE
ON EE.[WWI Employee ID] = A.PickedByPersonID AND EE.[Valid To] = '9999-12-
31 23:59:59.9999999'
LEFT JOIN WideWorldImportersDW.Dimension.City AS City
ON City.[WWI City ID] = A.DeliveryCityID AND City.[Valid To] = '9999-12-31
23:59:59.9999999'
EXEC dbo.LoadOrder
DROP TABLE WideWorldImportersDW.Integration.TransformOrder_Staging;

```