

Լաբորատոր աշխատանք 9

UNION, EXCEPT, INTERSECT գործողությունների կիրառումը

Աշխատանքի նպատակը՝ ուսումնասիրել մի քանի հարցումների միավորումը մեկ հարցման մեջ **UNION, EXCEPT, INTERSECT** գործողությունների կիրառմամբ: Իրականացնել հարցումների միավորում երկու և ավելի աղյուսակների հիման վրա:

Ուսումնասիրել աղյուսակից ստացված տվյալների հավաքածուի շրջման, կամ պտտման **PIVOT** և **UNPIVOT** հրամանները:

Տեսական մաս

UNION, EXCEPT, INTERSECT գործողությունները միավորում են երկու կամ ավելի հարցումների արդյունքները մի հավաքածուի մեջ, որը ներառում է միավորման բոլոր հարցումներին պատկանող բոլոր տողերը:

UNION, EXCEPT, INTERSECT գործողություն հիմնական կանոններն են՝

- ✓ Բոլոր հարցումներում պունակների քանակը և հերթականությունը պետք է լինի նույնը:
- ✓ Տվյալների տիպերը պետք է լինեն համատեղելի:

UNION, EXCEPT, INTERSECT գործողությունների քերականությունը հետևյալն է՝

Հարցում

UNION [ALL] / EXCEPT / INTERSECT

Հարցում

UNION-ի դեպքում դուրս են բերվում հարցման արդյունքում ստացված բազմության չկրկնվող տվյալները:

UNION ALL -ի դեպքում դուրս են բերվում հարցման արդյունքում ստացված բազմության բոլոր տվյալները:

EXCEPT -ի դեպքում հարցման առաջին բազմությունից հեռացնում է երկրորդ հարցման բազմությունը:

Այս դեպքում հաշվի է առնվում **SELECT** հրամանում նշված բազմությունների հերթականությունը:

INTERSECT -ի դեպքում վերադարձնում է բոլոր հարցումների բազմություններում առկա նույն բազմությունը:

Այս դեպքում էլ է հաշվի առնվում **SELECT** հրամանում նշված բազմությունների հերթականությունը:

PIVOT-ը **Transact-SQL** հրաման է, որը շրջում է է ստացված տվյալները: Այս հրամանում օգտագործվում են ագրեգատային ֆունկցիաները, և տվյալները, համապատասխանաբար, խմբավորվում են:

Այլ կերպ ասած, այն արժեքները, որոնք գտնվում են ուղղահայաց, դասավորվում են հորիզոնական:

PIVOT հրամանի քերականությունը հետևյալն է՝

SELECT <non-pivoted column>, [first pivoted column] AS <column name>,

[second pivoted column] AS <column name>, ... [last pivoted column] AS <column name>
FROM
(<SELECT query that produces the data>
AS <alias for the source query>

PIVOT
(
<aggregation function>(<column being aggregated>
FOR
[<column that contains the values that will become column headers>]
IN ([first pivoted column], [second pivoted column], ... [last pivoted column])) AS <alias for the pivot table>

<optional ORDER BY clause>;

UNPIVOT հրամանը կատարում է PIVOT- հրամանի հակառակ գործողությունը, այսինքն՝ տողերում գրած տվյալները ներկայացնում է մի սյունակով:
UNPIVOT հրամանում չեն օգրագործվում ագրեգատային ֆունկցիաներ:

Առաջադրանքներ՝

9.1 Կիրառելով UNION գործողությունը դուրս բերել Selles և Customers աղյուսակներից վաճառողների (Sellers.name_sel, Sellers.city_sel) և պատվիրատուների (Customers.name_cust, Customers.city_cust) անունները և քաղաքները:

9.2 Կիրառելով UNION ALL գործողությունը դուրս բերել Selles և Customers աղյուսակներից վաճառողների (Sellers.city_sel) և պատվիրատուների (Customers.city_cust) քաղաքները դասակարգված քաղաքների անուններով նվազման կարգով: Ապա նույն հարցումները միավորել UNION գործողության օգնությամբ: Բացատրել տարբերություններ:

9.3 Ստեղծել հետևյալ կառուցվածքով աղյուսակներ՝

Products (Ապրանքներ)

	Column Name	Data Type	Allow Nulls
🔑	prod_id	int	<input type="checkbox"/>
	name	nvarchar(50)	<input checked="" type="checkbox"/>
	price	int	<input checked="" type="checkbox"/>
	description	nvarchar(50)	<input checked="" type="checkbox"/>

Sells_2015 (2015 թվականի վաճառքներ)

🔑	id_sel	int	<input type="checkbox"/>
	prod_id	int	<input type="checkbox"/>
	count	int	<input checked="" type="checkbox"/>
	sel_date	date	<input checked="" type="checkbox"/>

Ապա կիրառելով **Select * into <new_table> from <exist_table>** հրամանը **Sells_2015** աղյուսակի հիման վրա ստեղծել ևս 2 աղյուսակ 2016 և 2017 թվականի վաճառքների աղյուսակներ Sells_2016 և Sells_2017 անվամբ համապատասխանաբար:

Ստեղծել նոր Database Diagram և կապել ստեղծված 4 աղյուսակները համապատասխան դաշտերով:

9.4 Բոլոր 4 աղյուսակները լրացնել հետևյալ տվյալներով:

Products

	prod_id	name	price	description
1	1	Notebook Asser 15.6" intel core i5	500	ddr 500
2	2	Notebook Asser 13.3" intel core i5	600	ssd 125
3	3	Notebook Asser 13.3" intel core i7	750	ssd 250
4	4	Notebook HP 13.3" intel core i7	800	ssd 250
5	5	Notebook HP15.6" intel core i7	600	ddr 1 trb
6	6	Mouse laser	10	Geniuse
7	7	Mouse bluetooth	15	HP
8	8	Mouse pen	20	8X6

Sells_2015

	id_sel	prod_id	count	sel_date
1	1	1	1	2015-04-01
2	2	3	1	2015-08-25
3	3	6	5	2015-10-10
4	4	2	1	2015-10-25
5	5	7	3	2015-12-23

Sells_2016

	id_sel	prod_id	count	sel_date
1	1	2	1	2016-02-12
2	2	3	2	2016-02-26
3	3	4	1	2016-04-12
4	4	5	2	2016-06-20
5	5	8	4	2016-09-23

Sells_2017

	id_sel	prod_id	count	sel_date
1	1	1	1	2017-04-02
2	2	4	1	2017-05-14
3	3	5	1	2017-07-02
4	4	7	5	2017-09-30

9.5 Կիրառելով **UNION**, ապա **UNION ALL** գործողությունները դուրս բերել 2015, 2016 և 2017 թվականներին վաճառված ապրանքների անունները, կատարված վաճառքների համարները (sel_id) և քանակները (count) դասակարգված ըստ քանակների աճման:

9.6 Կիրառելով **UNION**, ապա **UNION ALL** դուրս բերել 2016 և 2017 թվականներին վաճառված ապրանքներ քանակները և տարին դասակարգված ըստ տարիների նվազման կարգով:

9.7 Կիրառելով **UNION** գործողությունը դուրս բերել յուրաքանչյուր տարվա (2015, 2016 և 2017) կատարված վաճառքի մաքսիմալ քանակը (count):

9.8 Կիրառելով **EXCEPT** գործողությունը դուրս բերել այն ապրանքների անվանումները և համարները, որոնք վաճառվել են 2015 թվականին և չեն վաճառվել 2016-ին:

9.9 Կիրառելով **INTERSECT** գործողությունը դուրս բերել այն ապրանքների անվանումները և համարները, որոնք վաճառվել են 2015, 2016 և 2017 թվականներին:

9.10 Կատարել դասախոսության մեջ բերված PIVOT և UNPIVOT հրամանների օրինակները:

9.11 Կատարել հետևյալ առաջադրանքը, հետևյալ քայլերով՝

- Ստեղծել նոր աղյուսակ "TOTAL_SELLS" անվամբ, որը կունենա Sells_2015 աղյուսակի կառուցվածքը,
- կիրառելով INSERT INTO հրամանը այդ աղյուսակում հավաքագրել բոլոր երեք աղյուսակների տվյալները Sells_2015, Sells_2016, Sells_2017.
- կիրառելով PIVOT հրամանը շրջել տվյալները և դուրս բերել ապրանքների անունները և վաճառքների քանակների գումարները sum(count) ըստ տարիների՝ 2015, 2016, 2017: