

Селектовање података из више табела

За писање упита кјима се врши селектовање података из више табела, користе се JOIN оператори. Они омогућавају преузимање података из две или више табела, најчешће на основу постојећих, успостављених релација између табела.

За писање упита над више табела, битно је разумети концепт Декартовог (Картезијанског) производа.

У математици, Декартов производ је директни производ скупова; производ скупа са 2 елемента и скупа са 6 елемената је скуп са 12 елемената (2 x 6).

Код база података, Декартов производ је резултат придруживања сваког реда једне улазне табеле, сваком реду друге улазне табеле.

Нпр, производ табеле са 10 редова и табеле са 100 редова је резултујући скуп (табела) који садржи 1000 редова.

Најчешће као циљ упита, Декартов производ није жељени резултат.

Ако се две улазне табеле, које немају међусобну логичку везу, придруже Декартовим производом, добија се огромна табела са нелогичним, неповезаним подацима.

Једино место где се користи Декартов производ јесте креирање табела бројева или табела са подацима за тестирање.

За претраживање више табела, SQL језик подржава две синтаксе за писање упита:

1. Према ANSI SQL-89 синтакси, табеле се наводе иза FROM клаузуле, при чему се само раздвајају зарезом; ако је потребно, филтрирање редова се обавља преко услова у WHERE клаузули

```
SELECT ...
FROM Tabela1, Tabela2
WHERE uslov_za_filtiranje
```

2. Према ANSI SQL-92 синтакси, имена табела се наводе иза FROM клаузуле а табеле се спајају помоћу одговарајућег JOIN оператора.
Логички односи између табела представљају се преко ON клаузуле, која врши филтрирање редова за резултујућу виртуелну табелу.
У пракси се најчешће врши коришћење основног кључа једне табеле и страног кључа друге табеле.

```
SELECT ...
FROM Tabela1 JOIN Tabela2
ON on_predikat
```

Оба начина су исправна али се препоручује други пошто са првим начином се може остварити нежељени Декартов производ, ако се изостави WHERE клаузула.

INNER JOIN

Када се у SQL упиту користи оператор INNER JOIN, SQL ће из спојених табела вратити искључиво редове који одговарају постављеном услову.

У теорији скупова, аналогија за оператор INNER JOIN је пресек два скупа.

```
FROM t1 INNER JOIN t2 ON t1.c1 = t2.c2
```

Иза кључне речи FROM наведено је име табеле t1 а затим се наводи тип оператора JOIN, у овом случају то је INNER JOIN, па назив друге табеле t2.

Услов за спајање табела је наведен после ON кључне речи: колона c1 из табеле t1 мора да исту вредност као колона c2 из табеле t2.

Наођење речи INNER није неопходно пошто се исти резултат добија и само са JOIN.

За рад са JOIN операторима, креирају се две табеле са бројчаним вредностима:

Update Script File: dbo.Tabla.sql

Name	Data Type	Allow Nulls	Default
ID1	int	<input type="checkbox"/>	
Vrednost1	nvarchar(10)	<input type="checkbox"/>	

Keys (1)
<unnamed> (Primary Key, Clustered: ID1)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Design T-SQL

```

1 CREATE TABLE dbo.Tabela1
2 (
3     ID1 INT PRIMARY KEY, Vrednost1 nvarchar(10) NOT NULL
4 );
5

```

ID1	Vrednost1
1	Jedan
2	Dva
3	Tri
4	Cetiri
5	Pet
NULL	NULL

SELECT * FROM dbo.Tabela1;

Друга табела:

SQLQuery2.sql * dbo.Tabela2 [Data] dbo.Tabela2 [Design] SQLQuery1.sql dbo.Tabela1 [Data] dbo.Tabela

Update Script File: dbo.Table_1.sql

Name	Data Type	Allow Nulls	Default
ID2	int	<input type="checkbox"/>	
Vrednost2	nvarchar(10)	<input type="checkbox"/>	

Design T-SQL

```

1 CREATE TABLE dbo.Tabela2
2 (
3     ID2 INT PRIMARY KEY, Vrednost2 nvarchar(10) NOT NULL
4 );
5

```

SQLQuery2.sql * **dbo.Tabela2 [Data]**

Max Rows: 1000

	ID2	Vrednost2
▶	1	Jedan
	2	Dva
	3	Tri
	6	Sest
	7	Sedam
⌵	NULL	NULL

```
SELECT * FROM dbo.Tabela2;
```

Пример: За издвајање заједничких редова из обе табеле користи се INNER JOIN оператор, који у ON клаузули садржи услов да колона ID1 табеле t1 (алијас за Tabela1) буде једнака колони ID2 табеле t2 (алијас за Tabela2).

```
SELECT t1.*, t2.*
FROM dbo.Tabela1 t1 INNER JOIN dbo.Tabela2 t2
ON t1.ID1 = t2.ID2;
```

Tabela1 ID1 Vrednost1 Tabela1TableAdapter Fill,GetData ()	Tabela2 ID2 Vrednost2 Tabela2TableAdapter Fill,GetData ()
--	--

ЕР дијаграм се добија десним кликом на Database4DataSet.xsd у панелу Solution Explorer.

Даје:

ID1	Vrednost1	ID2	Vrednost2
1	Jedan	1	Jedan
2	Dva	2	Dva
3	Tri	3	Tri

Извршавање написаног упита, уз постављени услов, враћа заједничке редове из задатих табела.

У упиту се може појавити следећи услов: Ако кредитни лимит није дефинисан, приказати 0, а ако јесте, приказати износ кредитног лимита.

За овакав упит се може користити логичка функција IIF.

Ова функција се пише у следећем облику:

```
IIF(uslov, vrednost_kada_je_uslov_tacan, vrednost_kada_je_uslov_netacan)
```

```
IIF(KreditniLimit IS NULL, 0, KreditniLimit)
```

Задаци за самосталан рад

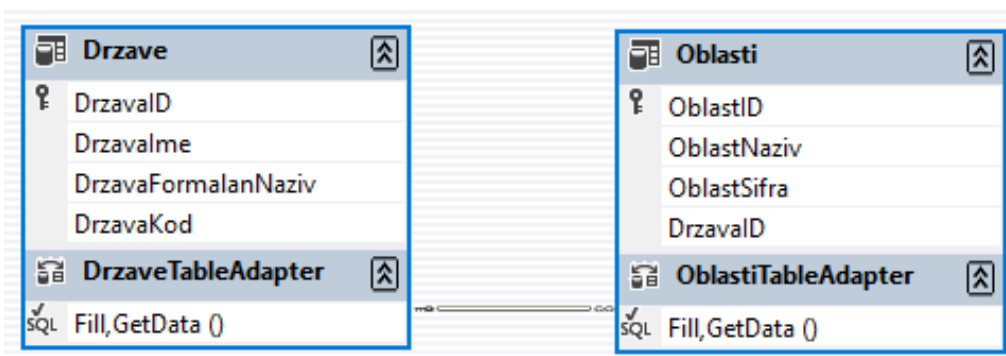
1. Креирати две табеле `dbo.Drzave` и `dbo.Oblasti`. Табела `Drzave` има колоне `DrzavaID`, `Drzavalme`, `DrzavaFormalanNaziv`, `DrzavaKod`. Табела `Oblasti` има колоне `OblastID`, `OblastNaziv`, `OblastSifra`, `DrzavaID`.

The screenshot displays the SQL Server Enterprise Designer interface. The top pane shows the 'Design' view for the `Oblasti` table, with columns: `OblastID` (int, primary key), `OblastNaziv` (nchar(20)), `OblastSifra` (nchar(3)), and `DrzavaID` (int, foreign key). The bottom pane shows the T-SQL script for creating the `Oblasti` table:

```
1 CREATE TABLE [dbo].[Oblasti]
2 (
3     [OblastID] INT NOT NULL PRIMARY KEY,
4     [OblastNaziv] NCHAR(20) NOT NULL,
5     [OblastSifra] NCHAR(3) NOT NULL,
6     [DrzavaID] INT NOT NULL
7 )
```

The middle pane shows the 'Design' view for the `Drzave` table, with columns: `DrzavaID` (int, primary key), `Drzavalme` (varchar(50)), `DrzavaFormalanNaziv` (nvarchar(50)), and `DrzavaKod` (char(3)). The bottom pane shows the T-SQL script for creating the `Drzave` table:

```
1 CREATE TABLE [dbo].[Drzave] (
2     [DrzavaID] INT NOT NULL,
3     [Drzavalme] VARCHAR (50) NOT NULL,
4     [DrzavaFormalanNaziv] NVARCHAR (50) NOT NULL,
5     [DrzavaKod] CHAR (3) NOT NULL,
6     PRIMARY KEY CLUSTERED ([DrzavaID] ASC)
7 );
```



Користећи `INNER JOIN`, написати `SELECT` упит који ће вратити колоне `OblastID`, `OblastNaziv`, `OblastSifra` и `Drzavalme`. Које колоне би требало користити за повезивање ове две табеле преко `ON` клаузуле и зашто?

Решење: Табела `dbo.Drzave` и `dbo.Oblasti` су повезане преко колоне `DrzavaID`, због чега се у `ON` клаузули ставља услов: вредност колоне `DrzavaID` из табеле `dbo.Drzave` треба да буде једнака вредности колоне `DrzavaID` из табеле `dbo.Oblasti`.

```

1 SELECT
2     o.OblastID,
3     o.OblastSifra,
4     o.OblastNaziv,
5     d.DrzavaIme
6 FROM dbo.Oblasti AS o
7 INNER JOIN dbo.Drzave AS d
8 ON d.DrzavaID = o.DrzavaID

```

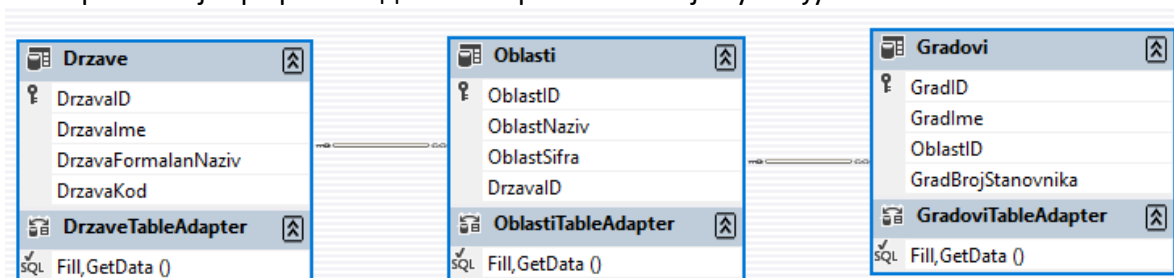
100 % No issues found

T-SQL Results Message

	OblastID	OblastSifra	OblastNaziv	Drzavalme
1	1	PUS	Pustinja	Avganistan
2	2	PLA	Planina	Avganistan
3	3	SUM	Sumadija	Srbija
4	4	BAN	Banat	Srbija
5	5	MAC	Macva	Srbija

2. Претходно написани упит модификовати тако да приказује и колону GradIme из табеле dbo.Gradovi. На који начин се то може урадити?

Решење: Приказан је графички однос све три табеле које су међусобно повезане.



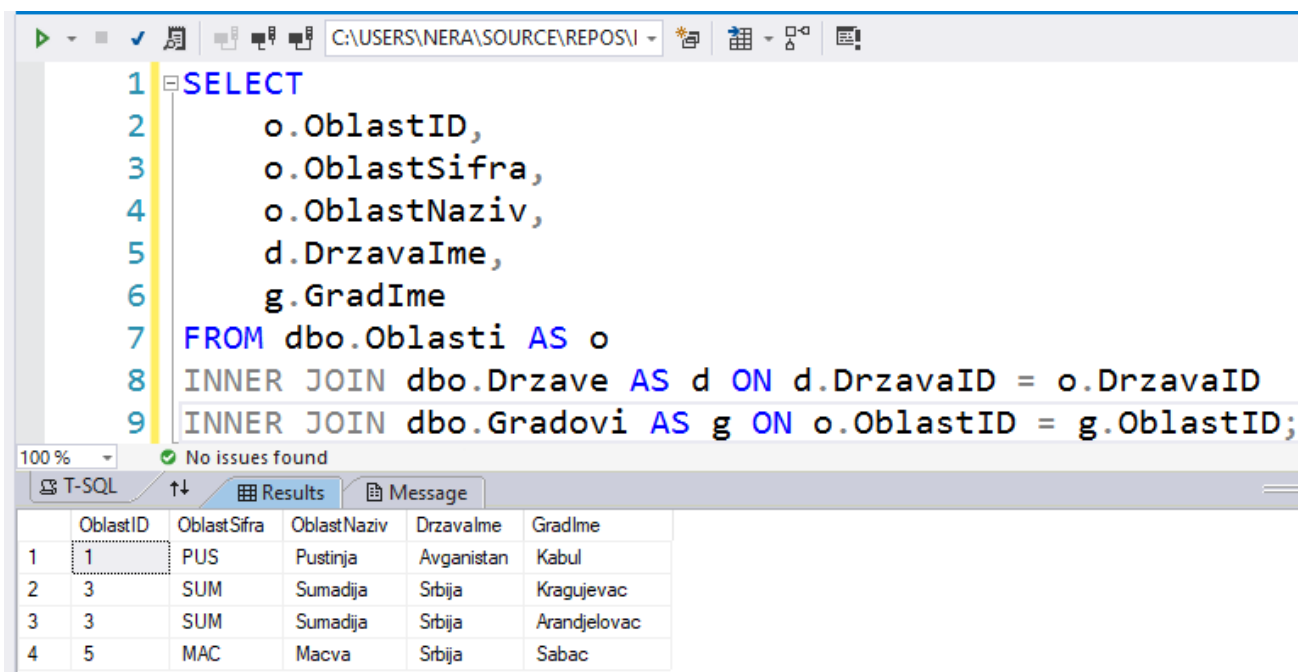
У табели Gradovi налази се колона OblastID, помоћу које се ова табела повезује са табелом Oblasti.

Види се да табела Drzave је родитељ табели Oblasti и ове табеле су повезане преко колоне DrzavaID.

Табела Oblasti је родитељ табели Gradovi и ове две табеле су повезане колоном OblastID.

Кључ означава родитељску а знак бесконачно дете табелу.

Једна земља може да има 0, 1 или више области а у свака област може имати 0, 1 или више градова.



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a T-SQL query with line numbers 1 through 9. The query selects columns from three tables: Oblasti, Drzave, and Gradovi. The bottom pane shows the results of the query in a table format with 6 columns: OblastID, OblastSifra, OblastNaziv, DrzavaIme, and GradIme. The results table contains 4 rows of data.

```
1 SELECT
2     o.OblastID,
3     o.OblastSifra,
4     o.OblastNaziv,
5     d.DrzavaIme,
6     g.GradIme
7 FROM dbo.Oblasti AS o
8 INNER JOIN dbo.Drzave AS d ON d.DrzavaID = o.DrzavaID
9 INNER JOIN dbo.Gradovi AS g ON o.OblastID = g.OblastID;
```

100 % No issues found

	OblastID	OblastSifra	OblastNaziv	DrzavaIme	GradIme
1	1	PUS	Pustinja	Avganistan	Kabul
2	3	SUM	Sumadija	Srbija	Kragujevac
3	3	SUM	Sumadija	Srbija	Arandjelovac
4	5	MAC	Macva	Srbija	Sabac