

### Сортирање података

Према теорији рада база података, скупови података не укључују дефиницију редоследа података.

Зато, ако је потребно са резултујући сет буде приказан у одређеном редоследу, потребно је да се наредби SELECT дода ORDER BY клаузула.

### ORDER BY клаузула

Она се користи за сортирање резултујућих података, у растућем или опадајућем редоследу, на основу вредности једне или више колона.

Део је SELECT наредбе и пише се на њеном крају:

ORDER BY lista\_kolona [ASC | DESC]

Пошто се ова клаузула извршава последња код евалуације SELECT наредбе, могуће је користити и алијасе у листи колона.

Редослед сортирања се може контролисати и са употребом кључне речи ASC (за растући редослед) или DESC (за опадајући редослед), и пишу се после сваке наведене колоне.

Пример:

```
SELECT KreditLimit, MusterijaIme
FROM dbo.Musterije
ORDER BY KreditLimit DESC;
```

Упит ће приказати две колоне (КредитЛимит и МуштеријеИме) али сортиране према колони КредитЛимит у опадајућем редоследу.

То значи да ће се муштерије са највећим кредитним лимитом појавити на врху сортиране листе.

На дну сортиране листе ће се појавити муштерије које немају податак о лимиту, тј који имају вредност NULL.

Пример:

```
SELECT KreditLimit, MusterijaIme
FROM dbo.Musterije
ORDER BY KreditLimit DESC, MusterijaIme ASC;
```

Овај упит користи две колоне за сортирање резултата упита и то у опадајућем редоследу по кредитном лимиту а у оквиру кредитног лимита у растућем редоследу по називу муштерије.

То значи да ако се појави више муштерија са истим износом кредитног лимита, оне ће бити сортиране према абecedном редоследу њихових имена.

### WHERE клаузула

Ова клаузула се користи да ограничи број редова које враћа упит и за то користи логичке услове. Логички услов може да врати TRUE, FALSE или UNKNOWN.

Прихватају се редови за које логички услов даје вредност TRUE.

У писању ове клаузуле не могу се користити алијаси за колоне пошто се WHERE извршава пре SELECT.

WHERE uslov\_za\_filtriranje

Услов за филтрирање може да има и из више логичких услова који се спајају са AND, OR, NOT.

У већини случајева: WHERE naziv\_kolone operator vrednost

Пример:

```
SELECT MusterijaID, MusterijaIme, KreditLimit
FROM dbo.Musterije
WHERE MusterijaKategorijaID = 5
ORDER BY KreditLimit DESC;
```

Упит враћа све купце који припадају категорији 5 а резултати су сортирани у опадајућем редоследу према кредитном лимиту.

Пример:

```
SELECT f.FakturaID, f.MusterijaID, i.FakturaDatum, m.MusterijaIme
FROM dbo.Fakture AS i
INNER JOIN dbo.Musterije AS m ON f.MusterijaID = m.MusterijaID
WHERE i.FakturaDatum > '20160101'
ORDER BY i.FakturaDatum;
```

Упит враћа податке о фактурама креираним после 1.1.2016. године.

Подаци о фактурама су у табели dbo.Fakture.

Пошто се колона са називом муштерије налази у табели dbo.Musterije, потребно је спојити ове табеле користећи колону MusterijaID.

Датумска константа се пише између апострофа у језички неутралном облику: година месец дан.

Пример:

```
SELECT OsobaID, ImePrezime, Email
FROM dbo.Osobe
WHERE ImePrezime LIKE 'K%'
ORDER BY ImePrezime;
```

Да би се добили подаци о собама у бази, чије пуно име и презиме почињу са К, потребно је користити оператор LIKE, који ради са карактерима.

Потребно је да се вредност за поређење напише између апострофа, а % указује на стринг било које дужине.

Пример:

```
SELECT OsobaID, ImePrezime, JesteZaposleni, JesteProdavac
FROM dbo.Osobe
WHERE JesteProdavac = 1 AND JesteZaposleni = 1
ORDER BY OsobaID;
```

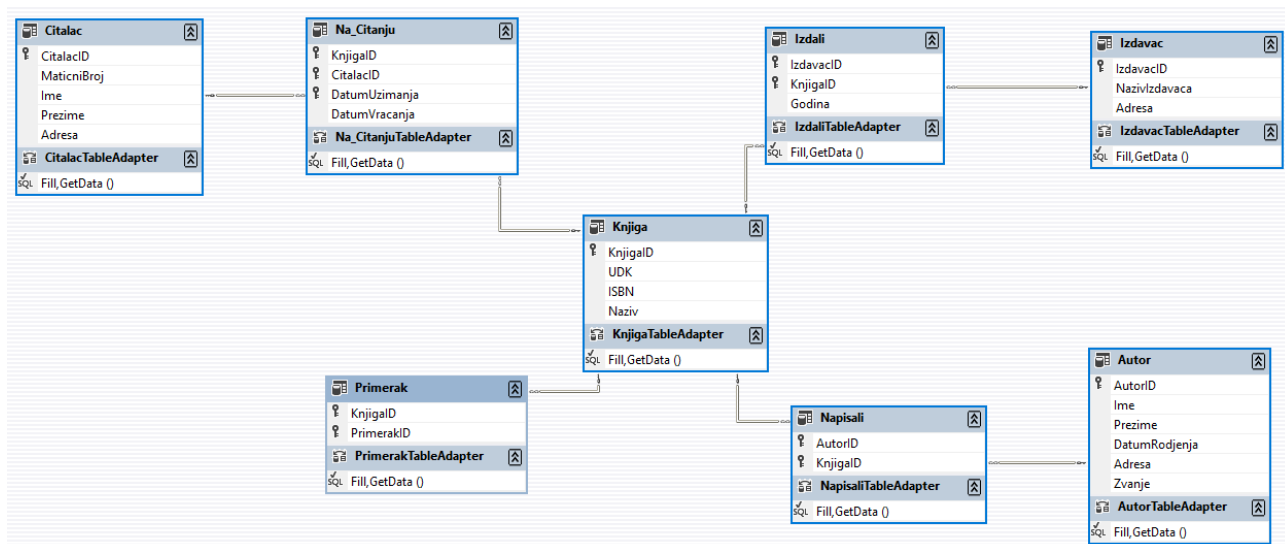
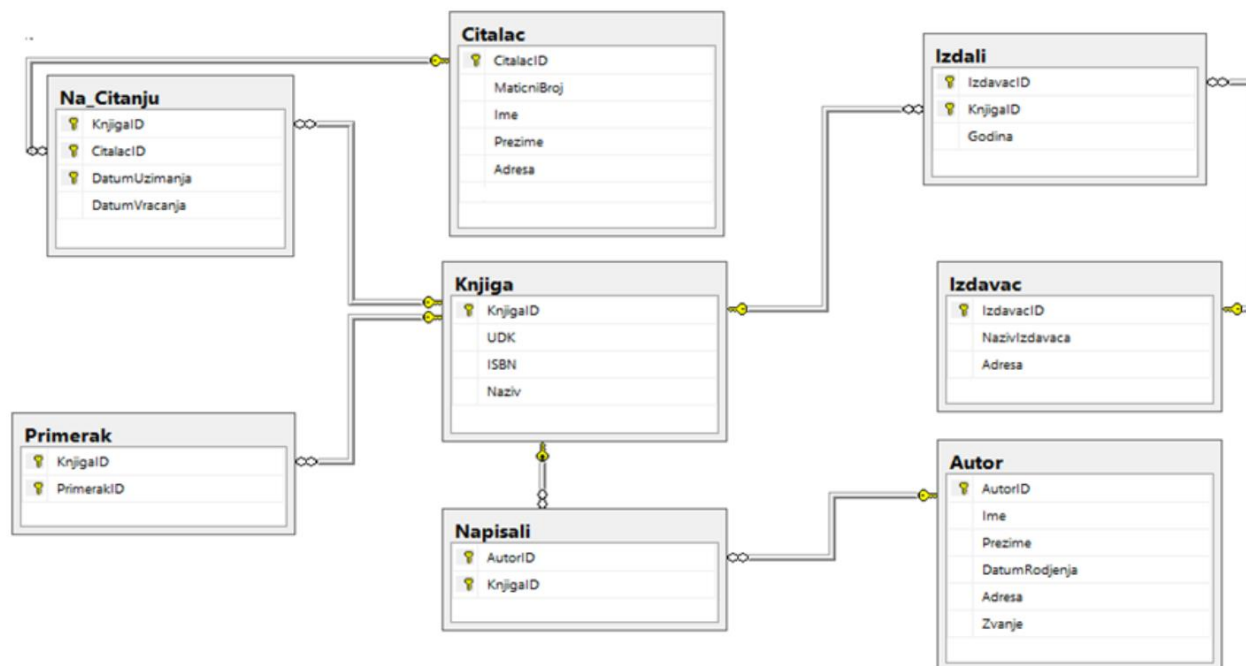
Овај упит враћа све особе које су истовремено и продавци и запослени у компанији, а редослед приказа је према броју OsobaID по дифолту од мањих ка већим вредностима.

## Приказ базе података за пројекат Библиотека A1

Подела пројеката ученицима.

Сваки ученик бира редни број пројекта од 2 до 30 (29 пројеката).

### ПРИЛОГ:



Update Script File: dbo.Table.sql

Name	Data Type	Allow Nulls	Default
KnjigaID	int	<input type="checkbox"/>	
UDK	nvarchar(50)	<input type="checkbox"/>	
ISBN	nvarchar(50)	<input type="checkbox"/>	
Naziv	nvarchar(50)	<input type="checkbox"/>	

Design T-SQL

```

1 CREATE TABLE [dbo].[Knjiga]
2 (
3     [KnjigaID] INT NOT NULL PRIMARY KEY,
4     [UDK] NVARCHAR(50) NOT NULL,
5     [ISBN] NVARCHAR(50) NOT NULL,
6     [Naziv] NVARCHAR(50) NOT NULL
7 )
8

```

Update Script File: dbo.Table\_3.sql

Name	Data Type	Allow Nulls	Default
KnjigaID	int	<input type="checkbox"/>	
PrimerakID	int	<input type="checkbox"/>	

Design T-SQL

```

1 CREATE TABLE [dbo].[Primerak]
2 (
3     [KnjigaID] INT NOT NULL,
4     [PrimerakID] INT NOT NULL,
5     CONSTRAINT [PK_Primerak] PRIMARY KEY ([KnjigaID], [PrimerakID]),
6     CONSTRAINT [FK_Primerak_KnjigaID] FOREIGN KEY ([KnjigaID]) REFERENCES [Knjiga]([KnjigaID])
7 );
8

```

**Keys (1)**  
 PK\_Primerak (Primary Key, Clustered: KnjigaID, PrimerakID)  
**Check Constraints (0)**  
**Indexes (0)**  
**Foreign Keys (1)**  
 FK\_Primerak\_KnjigaID (KnjigaID)

Update Script File: dbo.Table\_2.sql

Name	Data Type	Allow Nulls	Default
CitalacID	int	<input type="checkbox"/>	
MaticniBroj	nvarchar(10)	<input type="checkbox"/>	
Ime	nvarchar(30)	<input type="checkbox"/>	
Prezime	nvarchar(30)	<input type="checkbox"/>	
Adresa	nvarchar(50)	<input type="checkbox"/>	

Design T-SQL

```

1 CREATE TABLE [dbo].[Citalac]
2 (
3     [CitalacID] INT NOT NULL PRIMARY KEY,
4     [MaticniBroj] NVARCHAR(10) NOT NULL,
5     [Ime] NVARCHAR(30) NOT NULL,
6     [Prezime] NVARCHAR(30) NOT NULL,
7     [Adresa] NVARCHAR(50) NOT NULL
8 )
9

```

Update Script File: dbo.Table\_1.sql

Name	Data Type	Allow Nulls	Default
KnjigaID	int	<input type="checkbox"/>	
CitalacID	int	<input type="checkbox"/>	
DatumUzimanja	date	<input type="checkbox"/>	
DatumVracanja	date	<input type="checkbox"/>	

**Keys (1)**  
 PK\_Na\_Citanju (Primary Key, Clustered: KnjigaID, CitalacID, DatumUzimanja)  
**Check Constraints (0)**  
**Indexes (0)**  
**Foreign Keys (2)**  
 FK\_Na\_Citanju\_CitalacID (CitalacID)  
 FK\_Na\_Citanju\_KnjigaID (KnjigaID)  
**Triggers (0)**

Design T-SQL

```

1 CREATE TABLE [dbo].[Na_Citanju]
2 (
3     [KnjigaID] INT NOT NULL,
4     [CitalacID] INT NOT NULL,
5     [DatumUzimanja] DATE NOT NULL,
6     [DatumVracanja] DATE NOT NULL,
7     CONSTRAINT [PK_Na_Citanju] PRIMARY KEY ([KnjigaID], [CitalacID], [DatumUzimanja]),
8     CONSTRAINT [FK_Na_Citanju_CitalacID] FOREIGN KEY ([CitalacID]) REFERENCES [Citalac]([CitalacID])
9     CONSTRAINT [FK_Na_Citanju_KnjigaID] FOREIGN KEY ([KnjigaID]) REFERENCES [Knjiga]([KnjigaID])
10 );

```

Update Script File: dbo.Table\_4.sql

Name	Data Type	Allow Nulls	Default
AutorID	int	<input type="checkbox"/>	
Ime	nvarchar(30)	<input type="checkbox"/>	
Prezime	nvarchar(30)	<input type="checkbox"/>	
DatumRodjenja	date	<input checked="" type="checkbox"/>	
Adresa	nvarchar(50)	<input checked="" type="checkbox"/>	
Zvanje	nvarchar(30)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Design T-SQL

```

1 CREATE TABLE [dbo].[Autor]
2 (
3     [AutorID] INT NOT NULL PRIMARY KEY,
4     [Ime] NVARCHAR(30) NOT NULL,
5     [Prezime] NVARCHAR(30) NOT NULL,
6     [DatumRodjenja] DATE NULL,
7     [Adresa] NVARCHAR(50) NULL,
8     [Zvanje] NVARCHAR(30) NULL
9 );
10

```

Update Script File: dbo.Table\_5.sql

Name	Data Type	Allow Nulls	Default
AutorID	int	<input type="checkbox"/>	
KnjigaID	int	<input type="checkbox"/>	

**Keys (1)**  
 PK\_Napisali (Primary Key, Clustered: AutorID, KnjigaID)  
**Check Constraints (0)**  
**Indexes (0)**  
**Foreign Keys (2)**  
 FK\_Napisali\_KnjigaID (KnjigaID)  
 FK\_Napisali\_AutorID (AutorID)

Design T-SQL

```

1 CREATE TABLE [dbo].[Napisali]
2 (
3     [AutorID] INT NOT NULL,
4     [KnjigaID] INT NOT NULL,
5     CONSTRAINT [PK_Napisali] PRIMARY KEY ([AutorID], [KnjigaID]),
6     CONSTRAINT [FK_Napisali_KnjigaID] FOREIGN KEY ([KnjigaID]) REFERENCES [Knjiga]([KnjigaID]),
7     CONSTRAINT [FK_Napisali_AutorID] FOREIGN KEY ([AutorID]) REFERENCES [Autor]([AutorID])
8 );

```

Update Script File: dbo.Table\_6.sql

Name	Data Type	Allow Nulls	Default
IzdavacID	int	<input type="checkbox"/>	
NazivIzdavaca	nvarchar(50)	<input type="checkbox"/>	
Adresa	nvarchar(50)	<input checked="" type="checkbox"/>	

Design T-SQL

```

1 CREATE TABLE [dbo].[Izdavac]
2 (
3     [IzdavacID] INT NOT NULL PRIMARY KEY,
4     [NazivIzdavaca] NVARCHAR(50) NOT NULL,
5     [Adresa] NVARCHAR(50) NULL
6 );
7

```

Update Script File: dbo.Table\_7.sql

Name	Data Type	Allow Nulls	Default
IzdavacID	int	<input type="checkbox"/>	
KnjigaID	int	<input type="checkbox"/>	
Godina	date	<input type="checkbox"/>	

**Keys (1)**  
 PK\_Izdali (Primary Key, Clustered: IzdavacID, KnjigaID)  
**Check Constraints (0)**  
**Indexes (0)**  
**Foreign Keys (2)**  
 FK\_Izdali\_KnjigaID (KnjigaID)  
 FK\_Izdali\_IzdavacID (IzdavacID)  
**Triggers (0)**

Design T-SQL

```

1 CREATE TABLE [dbo].[Izdali]
2 (
3     [IzdavacID] INT NOT NULL,
4     [KnjigaID] INT NOT NULL,
5     [Godina] DATE NOT NULL,
6     CONSTRAINT [PK_Izdali] PRIMARY KEY ([IzdavacID], [KnjigaID]),
7     CONSTRAINT [FK_Izdali_KnjigaID] FOREIGN KEY ([KnjigaID]) REFERENCES [Knjiga]([KnjigaID]),
8     CONSTRAINT [FK_Izdali_IzdavacID] FOREIGN KEY ([IzdavacID]) REFERENCES [Izdavac]([IzdavacID])
9 );
10

```

Задаци за самосталан рад

1. Приказати шифру и назив само муштерије које припадају категорији 'IT prodavnica'.  
Резултат сортирати по растућем редоследу, према називу муштерије.
2. Приказати све шифре и називе градова, чије име почиње са 'Novi'.
3. Приказати све шифре и називе купаца, који се налазе у градовима 'Beograd' или 'Nis'.