

# SKLADIŠTE PODATAKA BIBLIOTEKA

---

**Odsjek:** Razvoj Softvera

**Studenti:** Belma Muratović, Dženana Fejzić, Elhattab Yahia Aissa, Faris Hrvo, Sanjin Šabanović

**Predmetni nastavnik:** Doc. Dr. Damir Omerašević

## Sadržaj

<b>Uvod.....</b>	<b>3</b>
<b>Use-case dijagram .....</b>	<b>4</b>
<b>Activity dijagram.....</b>	<b>5</b>
<b>Relacijska baza podataka .....</b>	<b>6</b>
<b>ER Dijagram .....</b>	<b>6</b>
<b>Star schema .....</b>	<b>7</b>
<b>ETL proces.....</b>	<b>8</b>
<b>Kocka.....</b>	<b>15</b>
<b>Analiza podataka .....</b>	<b>21</b>
<b>Proces pravljenja projekta.....</b>	<b>22</b>

## Uvod

Naš projekat se bazira na optimizaciji biblioteka. Željeli smo da omogućimo lagan i pristupačan način rada radnicima biblioteke.

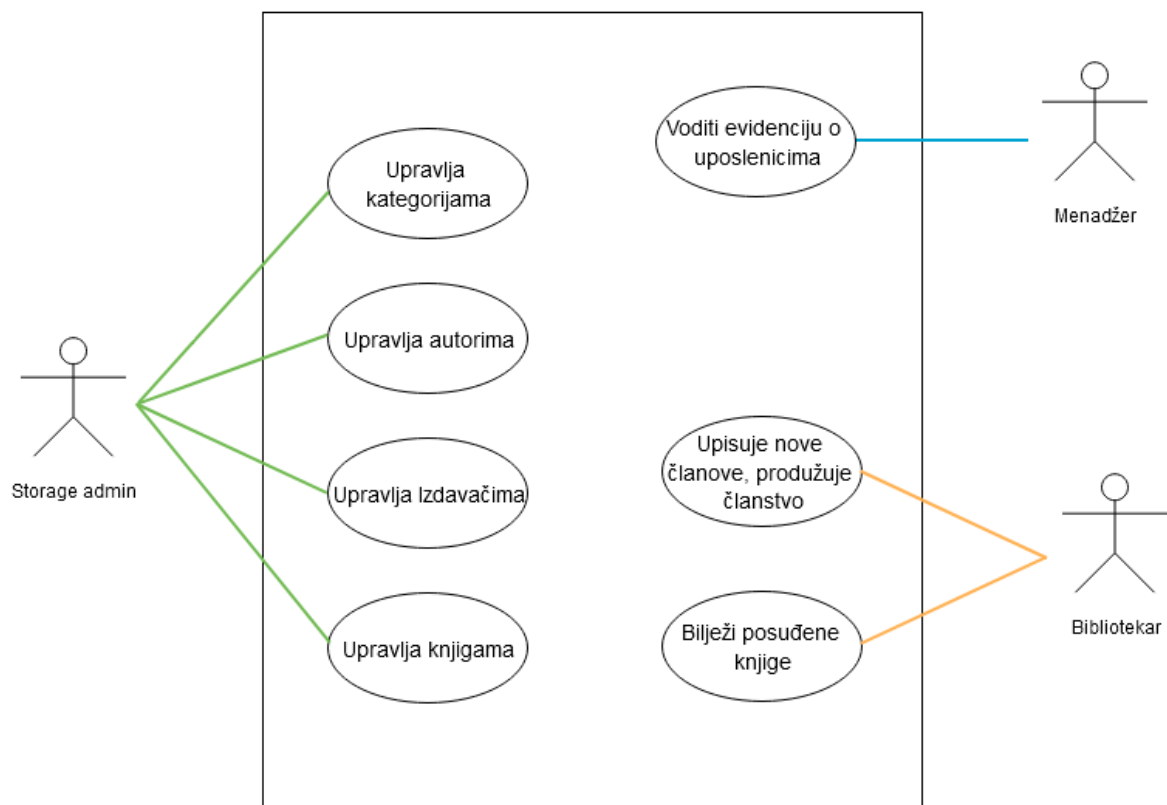
Cilj projekta je stvaranje organizovanog sistema u kojem su svi podaci povezani i sprečavanje suvišnosti, nedosljednosti i gubitka podataka.

Naš cilj je bio da napravimo skladište podataka za posuđivanje knjiga na čitanje. Inače je ta tabela velika pa bi se sama optimizovala, a zaposlenici biblioteke bi imali uvid u prethodno vremensko razdoblje.

U ovom izvještaju ćemo vam pokazati kako smo realizovali naš projekat.

## Use-case diagram

Use-case diagram predstavlja dijagram koji prikazuje interakciju korisnika sa bazom. U našem slučaju imamo 3 korisnika: menadžer, zaposlenik i admin skladišta. Menadžer vodi evidenciju o uposlenicima. Zaposlenik biblioteke može pisati, brisati, mijenjati sadržaj tabela korisnici biblioteke, članovi i posuđene knjige dok tabelu knjige može samo čitati. Admin skladišta upravlja knjigama, autorima, kategorijama i izdavačima.



## Activity dijagram

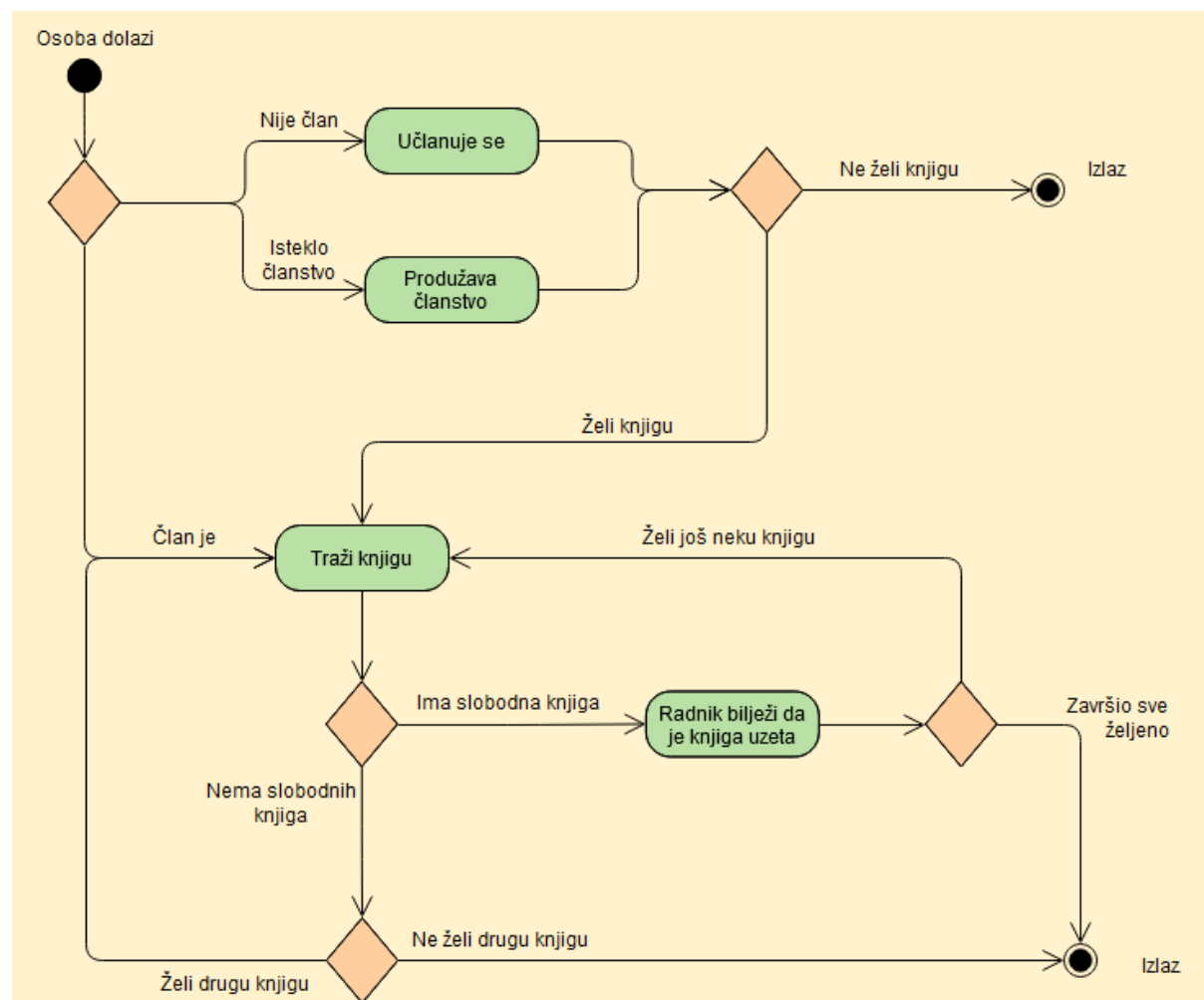
Activity dijagram u našem slučaju predstavlja opis radnji kada osoba dođe u biblioteku.

Ovaj dijagram se kreira preko radnji koji se mogu desiti kada osoba dođe u biblioteku sve dok je ta osoba ne napusti. Uključene su sve moguće opcije koje osoba može uraditi. Uslovi kada sljedeća radnja zavisi od nečega su predstavljeni oblicima baklava, dok su radnje predstavljene zaobljenim zelenim kvadratima.

Kada osoba dođe provjerava se da li je član. Ako nije član nudi se mogućnost učlanjenja te nakon učlanjenja može da traži knjigu koju želi da pozajmi na čitanje ili može otići u slučaju da trenutno ne želi knjigu. U slučaju da je osobi isteklo članstvo, ona može produžiti članstvo te onda može tražiti knjigu koju želi.

U slučaju kada je osoba član ona može da traži knjigu koju želi pozajmiti. Kada zaposlenik provjeri da ima slobodna knjiga koju osoba želi, bilježi da je osoba pozajmila knjigu. Zatim osoba može da traži još neku knjigu ili napustiti biblioteku.

U slučaju da nema slobodna knjiga koju osoba želi, osoba može tražiti neku drugu knjigu ili napustiti biblioteku.



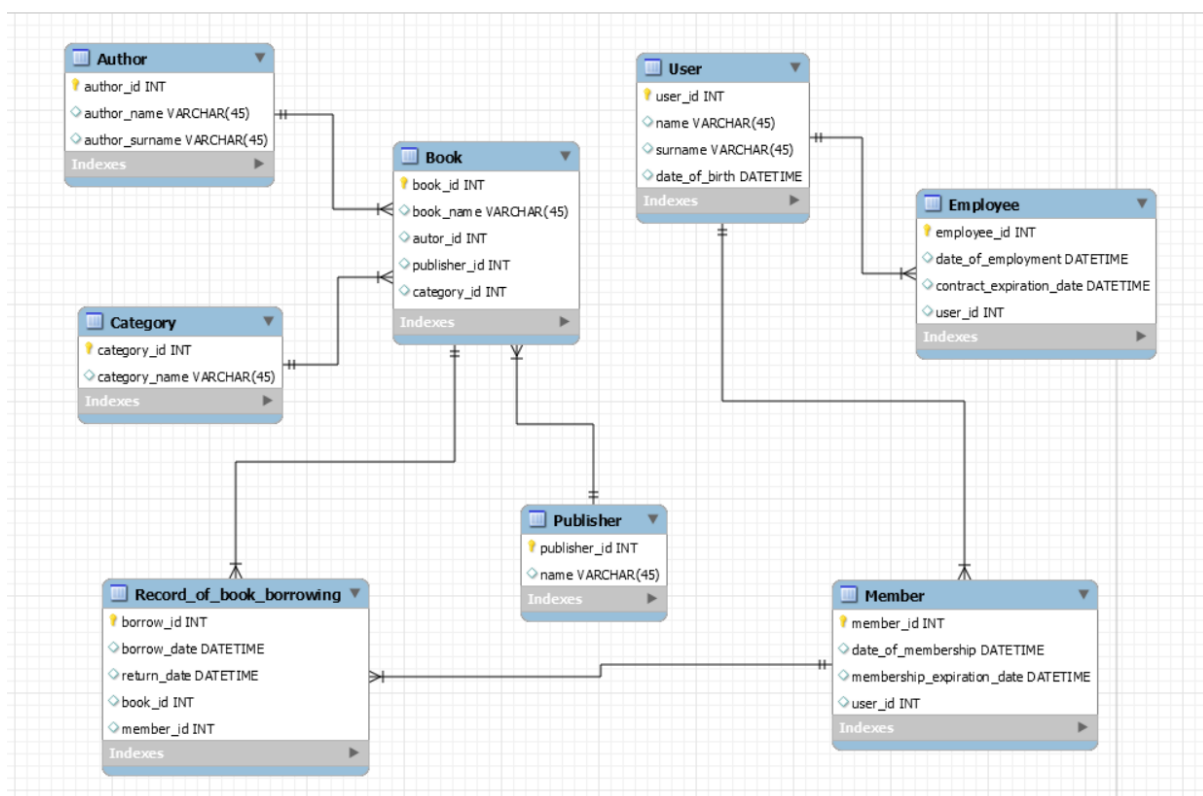
## Relacijska baza podataka

Što se tiče relacijske baze podataka, ona je tip baze podataka kod kojeg je organizacija podataka zasnovana na relacionom modelu. U ovim bazama, organizacija podataka se vrši u skup relacija između kojih se definišu određene veze.

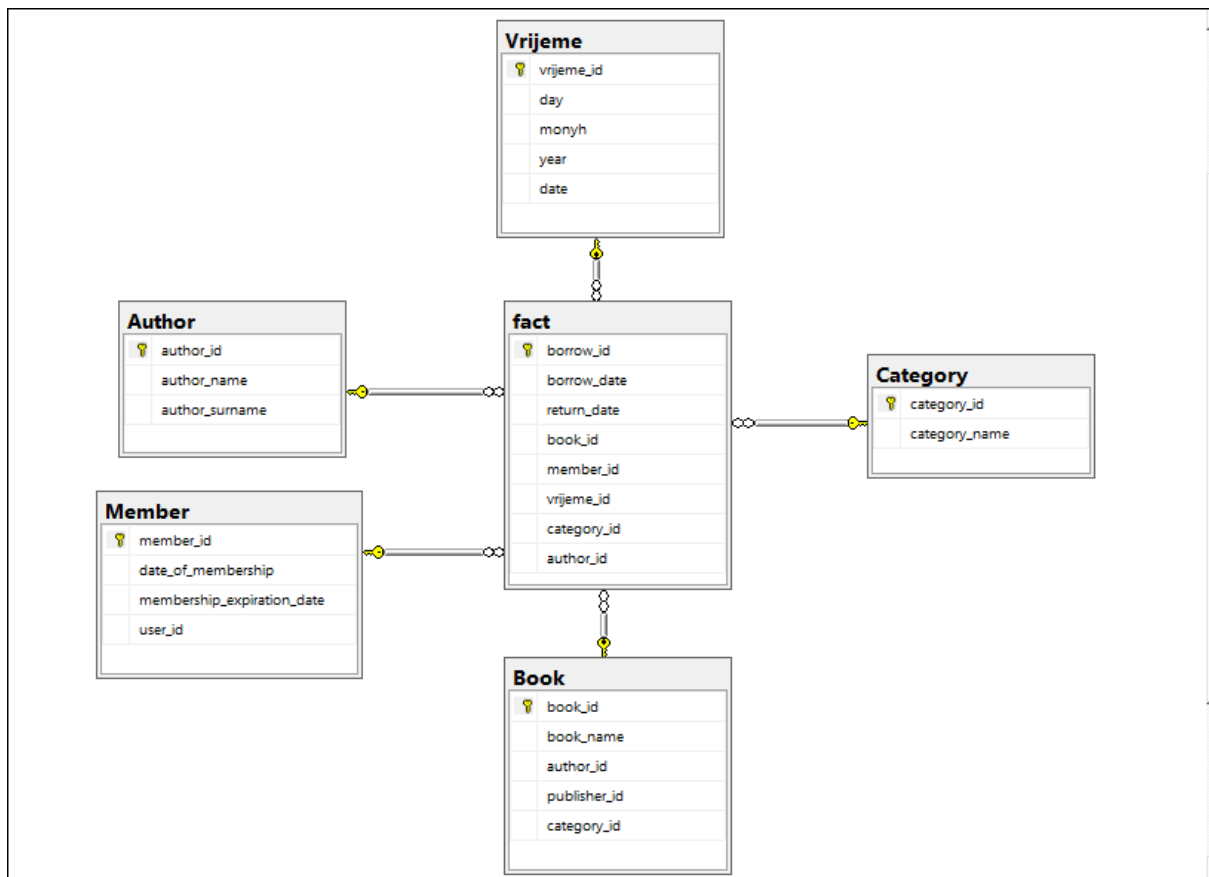
U relacionim bazama podataka, svaka relacija mora da ima definisan primary key, a pored njega može posjedovati i foreign key. U tabele koje smo kreirali unijeli smo vrijednosti za sve podatke koji se pojavljuju i na ER dijagramu.

## ER Dijagram

ER dijagram prikazuje entitete i veze kreirane baze podataka. Na slici je prikazano osam tabela koje se sadrže od podataka na osnovu kojih smo napravili našu relacijsku bazu podataka. Zapravo, ER dijagram nam je poslužio kao temelj za dalji rad na projektu.



## Star schema



Ova star shema nam služi za analizu koje knjige se najviše čitaju po raznim vremenskim intervalima.

**Dimenzije:** Member, Author, Book, Category, Vrijeme

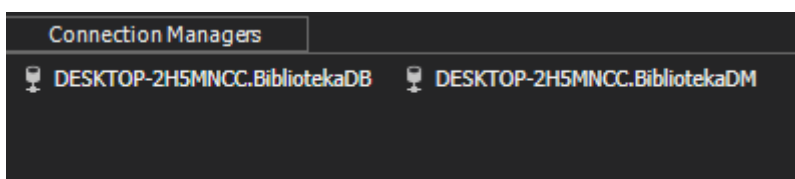
**Fact tabela:** Fact

## ETL proces

Skraćenica ETL predstavlja (extraction, transformation i load) odnosno ekstrakciju/izdvajanje podataka iz različitih OLTP-ova. On predstavlja tri funkcije objedinjene u jedan alat koje služe za izvlačenje podataka iz produkcijske (u našem slučaju relacije) baze podataka i njihovo učitavanje u DW.

ETL proces smo napravili koristeći program Microsoft Visual Studio 2019, tačnije njegovu ekstenziju SQL Server Integration Services Projects.

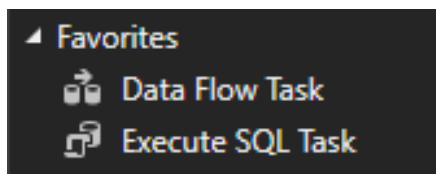
Kreirali smo novi SSIS package unutar kojeg smo napravili konekcije sa bazom skladišta podataka i bazom izvora podataka (relacione baze) nakon čeka smo u tabu Control Flow napravili naše skladište podataka.



slika 1.Prikaz uspješno ostvarene konekcije sa bazama

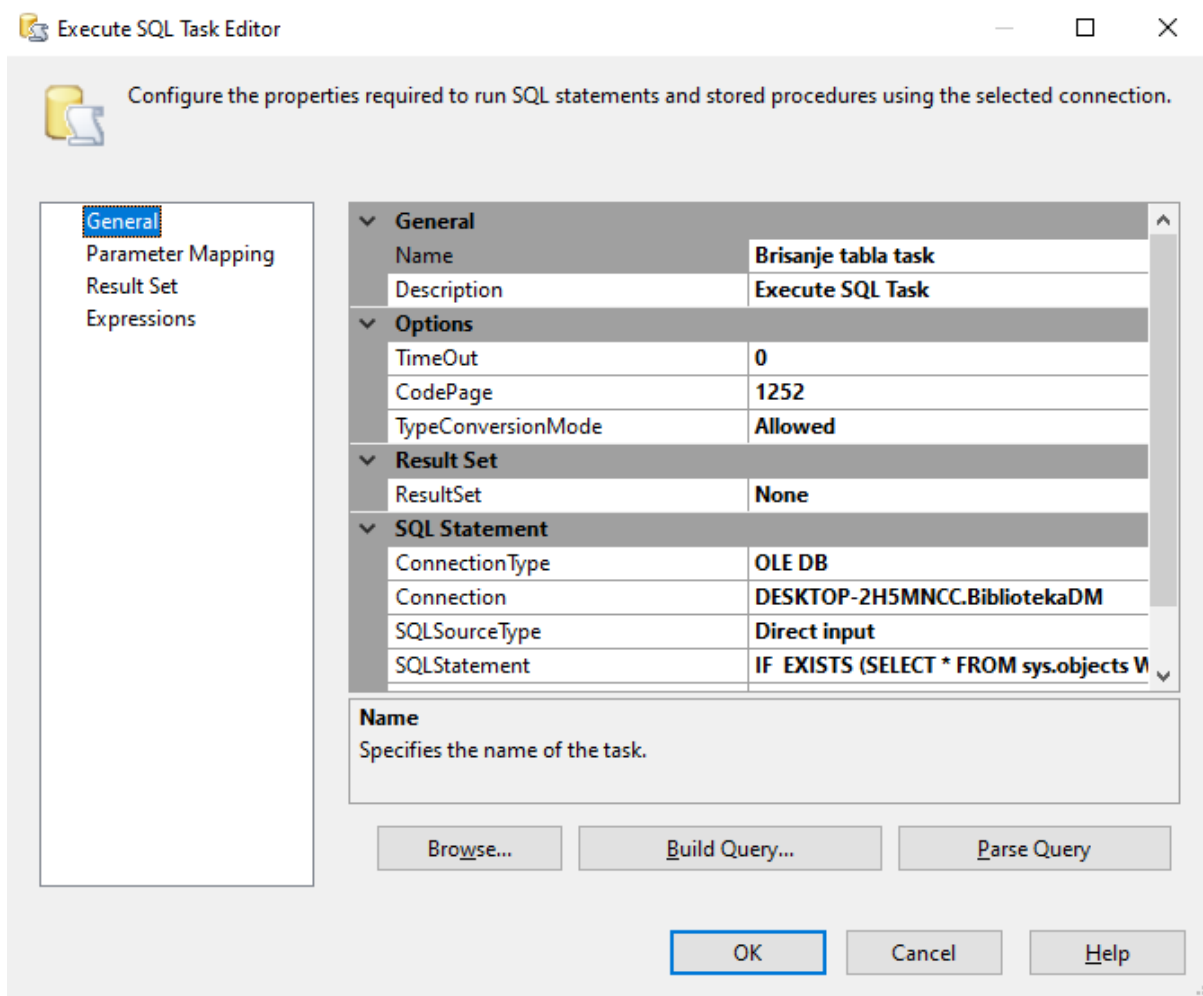
Detaljan opis kreiranja:

U Control Flow smo ubacili Execute SQL Task (slika 2) i obrisali tabele unutar baze skladišta podataka (način pomoću kojeg se osiguravamo da ne bi došlo do konflikta ukoliko tabele i/ili podaci već postoje).



slika 2.Execute SQL Task u SSIS Toolbox-u

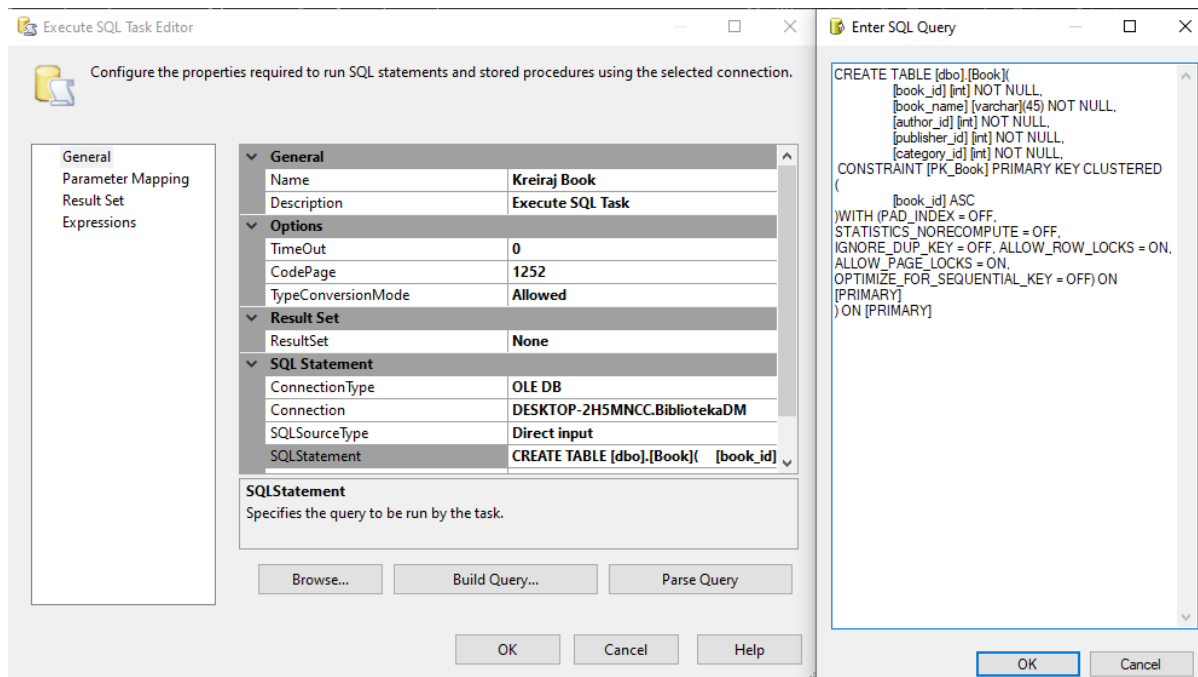




Slika 3. Prikaz Edit-a Execute SQL Task

Na slici 3. vidimo prikaz Execute SQL Task-a gdje je potrebno odabrati za konekciju bazu skladišta i napisati SQLStatement koji će izvršiti provjeru da li tabele postoje, ukoliko su prisutne izvršit će njihovo brisanje.

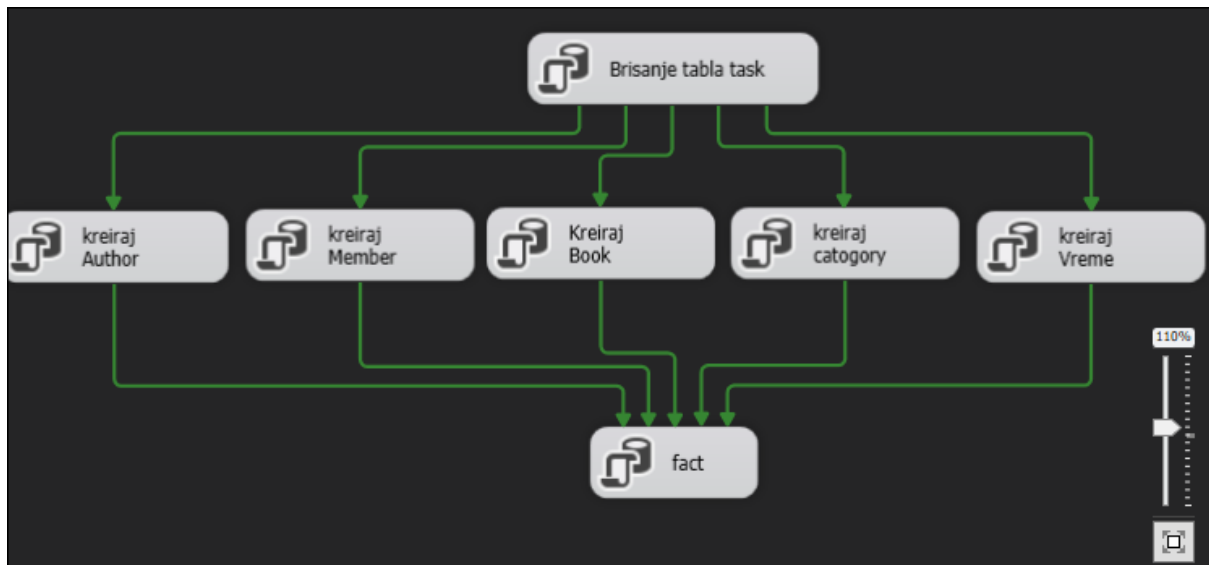
Drugi korak je sačinjen od kreiranog dimenzija star schema. Način na koji smo to postigli je sličan kao u primjeru iznad razlikuje se samo u SQLStatmantu gdje smo kreirali nove tabele (prikaz procesa kreiranja jedne od dimenzija star schema se nalazi ispod, na sličan nači smo kreirali i ostale tabele dimenzija ).



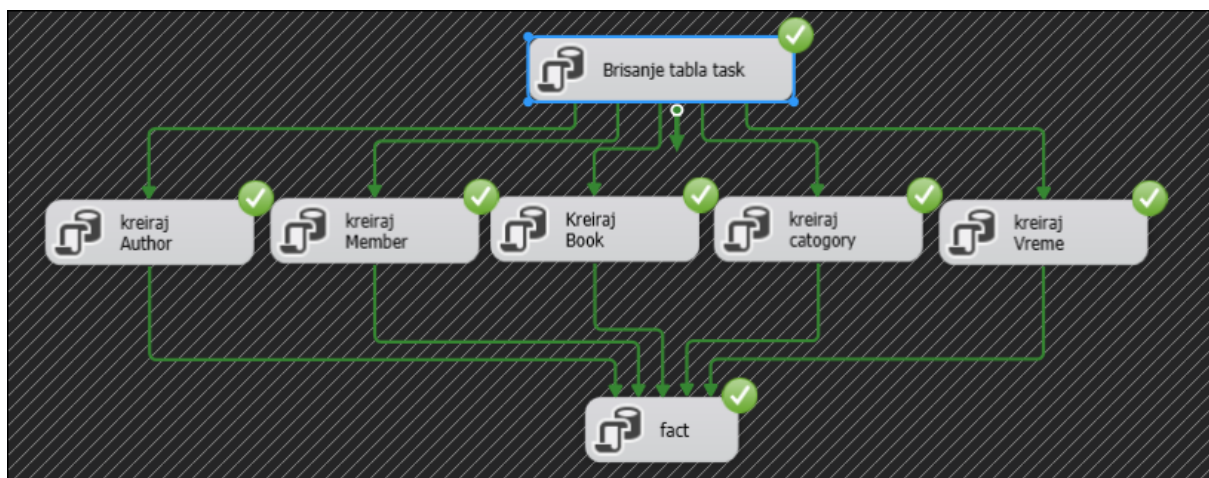
Slika 4. SQLStatement za kreiranje Book dimenzije (Book)

U narednom koraku smo kreirali Tabele Činjenica na način kao i dimenzije. Potom smo kreirali Alter Tabela Činjenica postupak koji se mora izvršiti da bi se dodali strani ključevi.

Nakon kreiranja svih potrebnih tabela izvršili smo njihovo povezivanje. Prikaz povezane šeme se nalazi ispod.

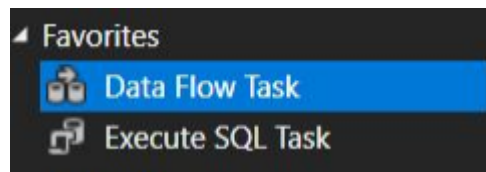


Slika 5. Control Flow ETL Process-a Kreiranja Skladišta podataka



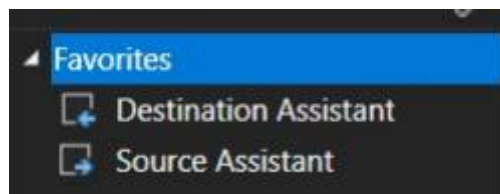
Slika 6. Sve uspješno radi

U sljedećem koraku kreirali smo novi SSISPackage te smo umjesto execute SQLTask koristili Data Flow Task

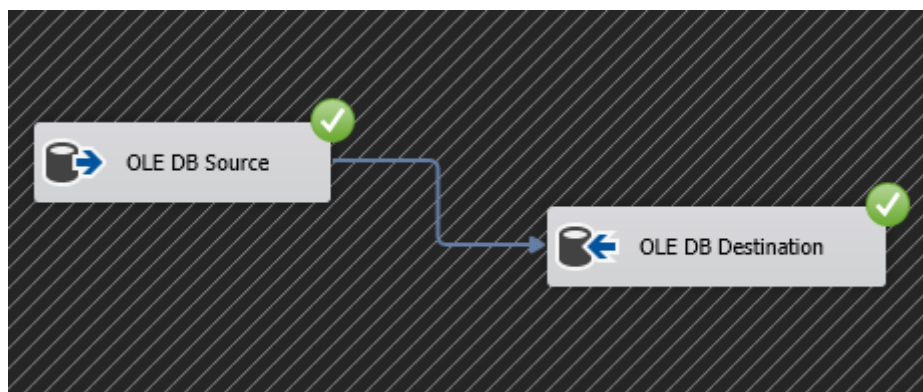


Slika 7. Data Flow Task

Nakon toga smo pomoću Source Assistant i Destination Assistant alata odabrali izvornu i odredišnu tabelu ovisno o dimenzijama tih tabela.



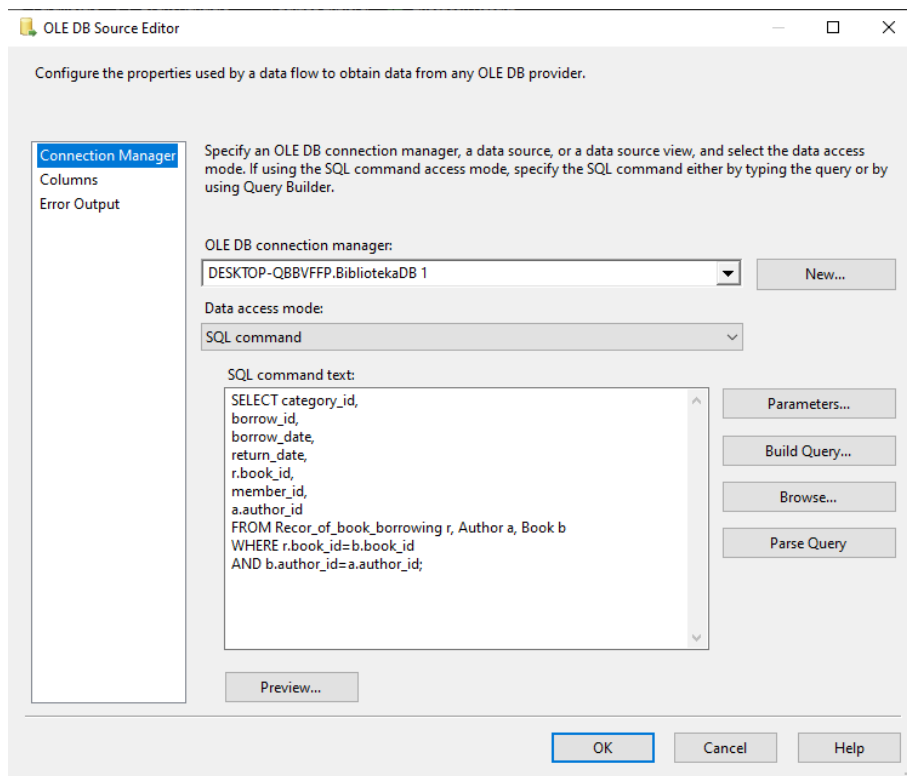
Slika 8. Assistant alati



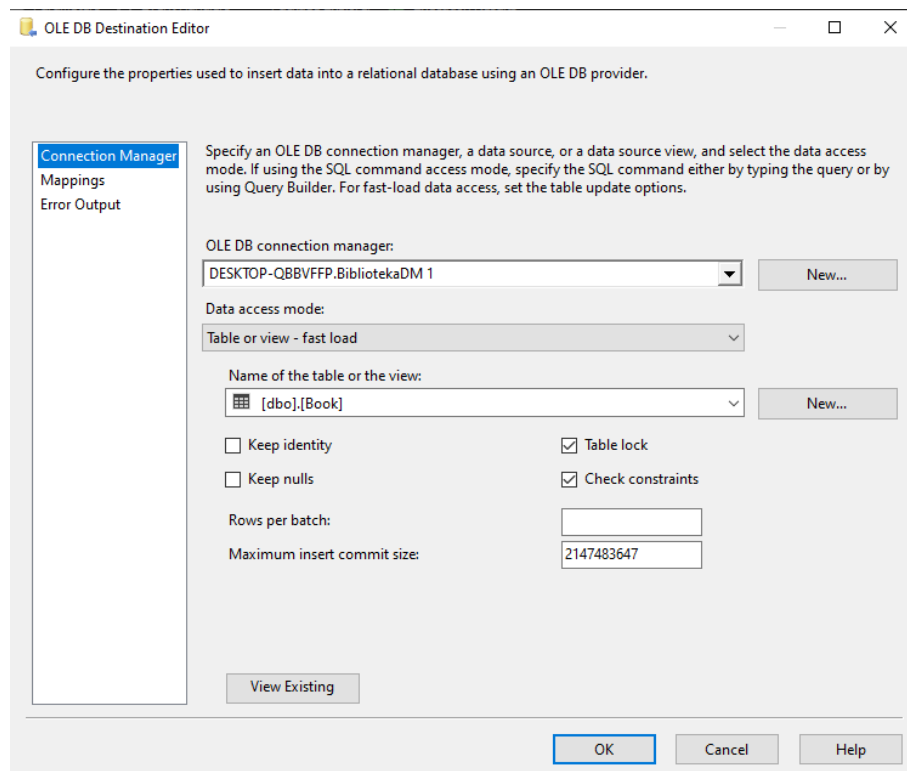
Slika 9. Primjer uspješno odabranih izvorišne i odredišne tabele

Za izvorišnu tabelu smo napisali upit koji kupi podatke koje treba uzeti iz te tabele.

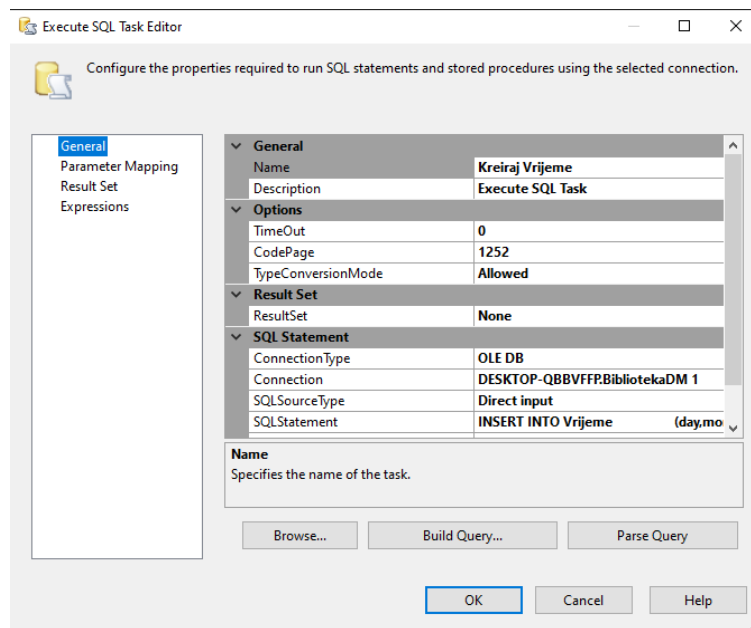
U source Assistantu smo odredili odredišnu tabelu u koju ćemo eksportovati vrijednosti iz izvorne tabele.



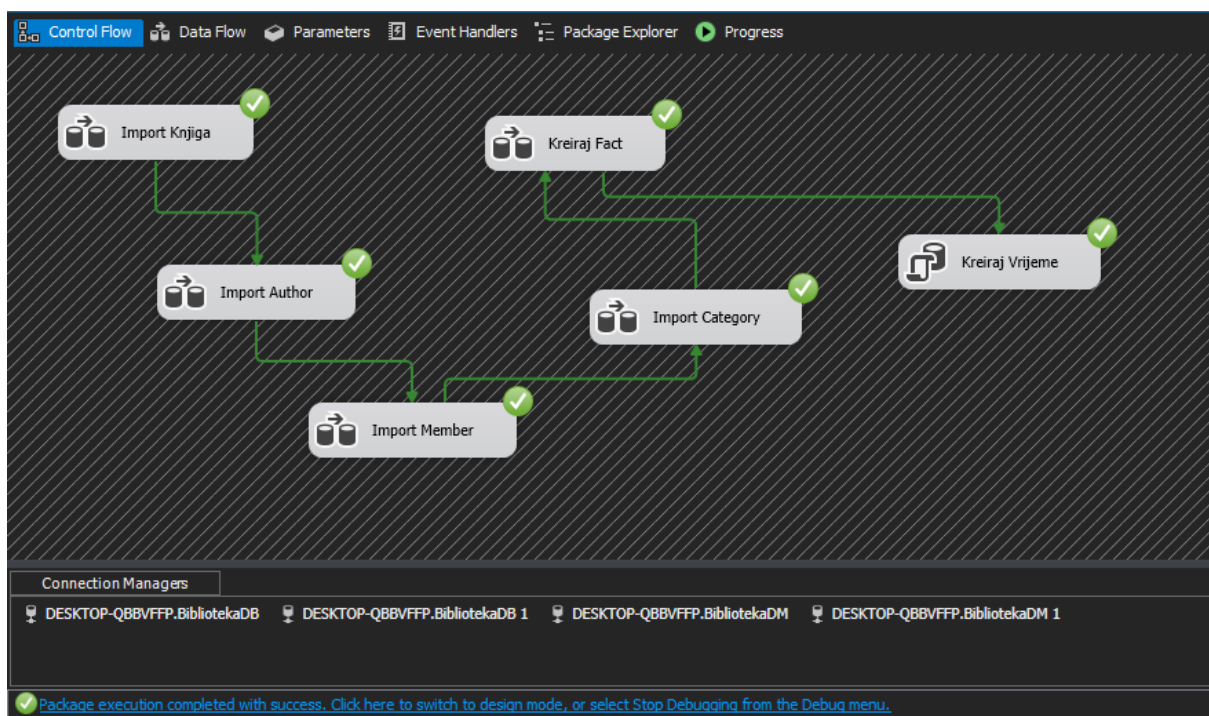
Slika 10. Source Assistant za izvornu tabelu



Slika 11. Source Assistant za odredišnu tabelu



Slika 12. Source Assistant za kreiranje tabele Vrijeme



slika 13. Finalni prikaz uspješno odrađenog ETL Procesu.

## Kocka

OLAP je pristup analizi i izvještavanju koji omogućuje korisniku da lako i selektivno izdvaja i pregledava podatke s različitih stajališta temeljeno na multidimenzijskoj strukturi podataka zvanoj kocka (engl. cube). Kocka tipično odgovara jednom zvjezdastom spoju i tipično se unutar OLAP sistem definira uzevši postojeći zvjezdasti spoj kao izvor podataka.

Prvo smo kreirali novi projekat u Microsoft Visual Studiu, zatim u Solution Explorer-u kliknemo desnim klikom miša na Data Sources te odaberemo opciju New Data Source te nam se pojavljuje wizard za kreiranje konekcije:

**Data Source Wizard**

**Select how to define the connection**  
You can select from a number of ways in which your data source will define its connection string.

☐ Create a data source based on another object

☒ Create a data source based on an existing or new connection

Data connections:

DESKTOP-2H5MNCC.BibliotekaDM
------------------------------

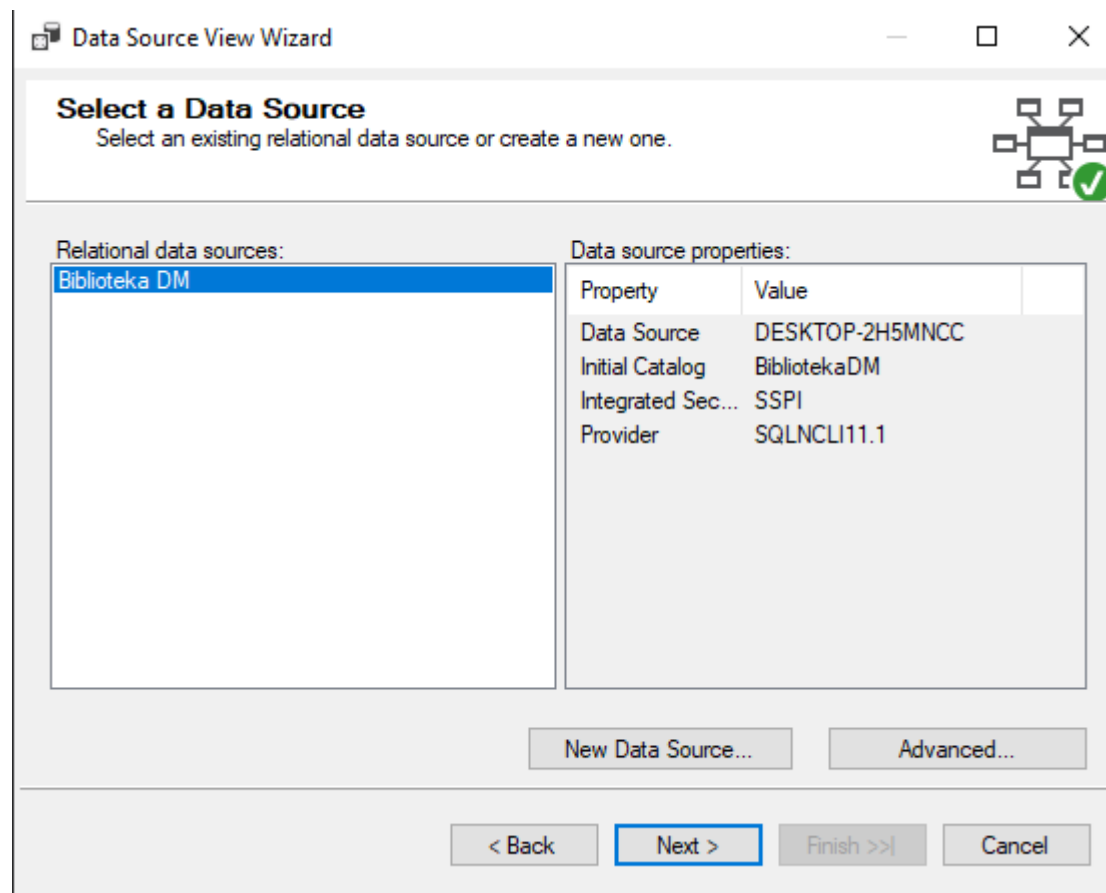
Data connection properties:

Property	Value
Data Source	DESKTOP-2H5MNCC
Initial Catalog	BibliotekaDM
Integrated Sec...	SSPI
Provider	SQLNCLI11.1

New... Delete


< Back Next > Finish >>| Cancel

Potom odabiremo izvor podataka:






Nakon što odaberemo izvor podataka, onda biramo koje tabele želimo uključiti u kocku. Mi smo izabrali tabelu činjenica koja je neophodna, te tabele vrijeme, knjige i članovi.




 Data Source View Wizard — □ ×

---

**Select Tables and Views**  
Select objects from the relational database to be included in the data source view.







Available objects:

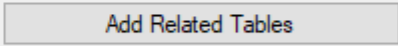
Name	Type
 Author (dbo)	Table
 Category (dbo)	Table
 sysdiagrams (dbo)	Table





Filter:

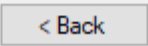
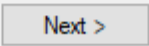
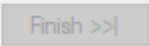
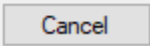
☐ Show system objects

Included objects:

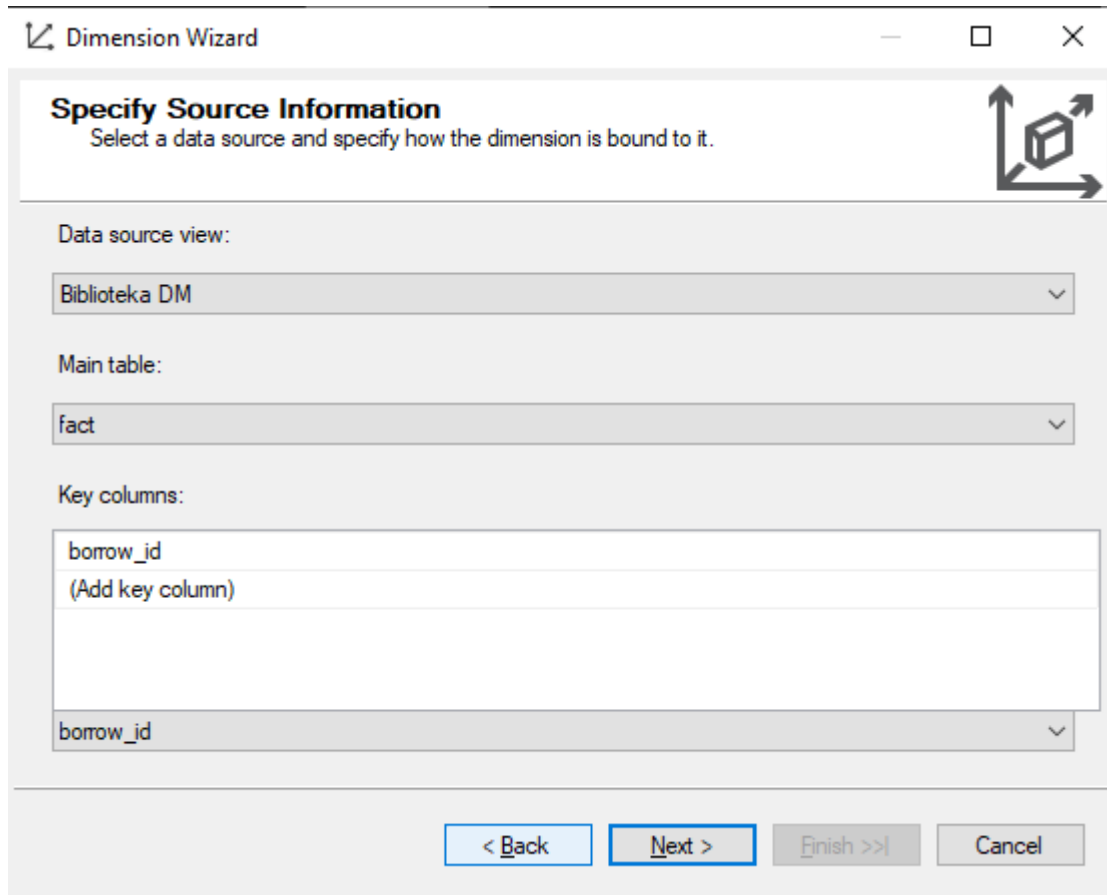
Name	Type
 fact (dbo)	Table
 Vrijeme (dbo)	Table
 Book (dbo)	Table
 Member (dbo)	Table



Preostaje nam definiranje dimenzija kocke, a to radimo desnim klikom na Dimensions te odabirom opcije New dimension, čime nam se pojavljuje čarobnjak za kreiranje nove dimenzije. Potom odabiremo izvor podataka koji smo definirali ranije.



The screenshot shows the 'Dimension Wizard' dialog box, specifically the 'Specify Source Information' step. The window has a title bar with the text 'Dimension Wizard' and standard Windows window controls (minimize, maximize, close). The main area is titled 'Specify Source Information' with a subtitle 'Select a data source and specify how the dimension is bound to it.' and a 3D cube icon. The 'Data source view:' dropdown is set to 'Biblioteka DM'. The 'Main table:' dropdown is set to 'fact'. The 'Key columns:' section contains a list box with 'borrow\_id' and '(Add key column)', and a dropdown at the bottom also set to 'borrow\_id'. At the bottom of the dialog are four buttons: '< Back', 'Next >', 'Finish >>', and 'Cancel'.

Dimension Wizard

**Specify Source Information**  
Select a data source and specify how the dimension is bound to it.

Data source view:  
Biblioteka DM

Main table:  
fact

Key columns:  
borrow\_id  
(Add key column)  
borrow\_id

< Back   Next >   Finish >>   Cancel

Preostaje nam još samo dati naziv:

**Completing the Wizard**  
Type a name for the new dimension, verify the dimension structure, and then click Finish to save the dimension.

Name:

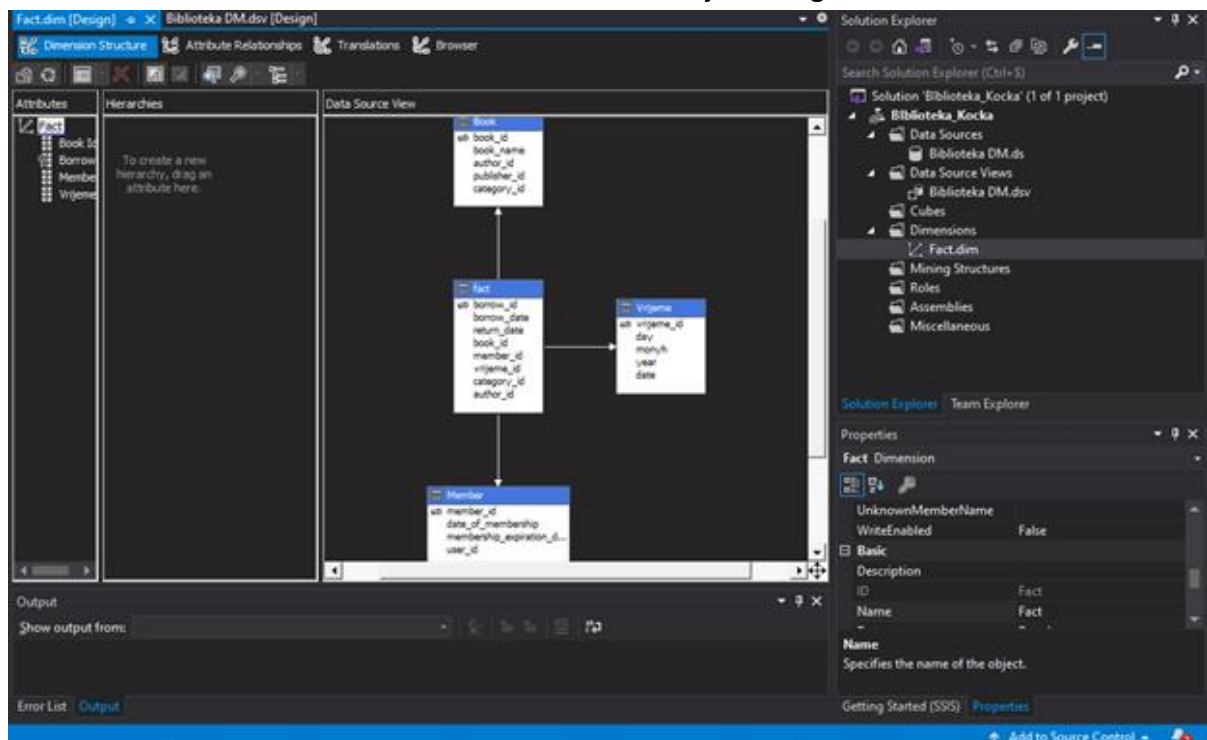
Fact

Preview:

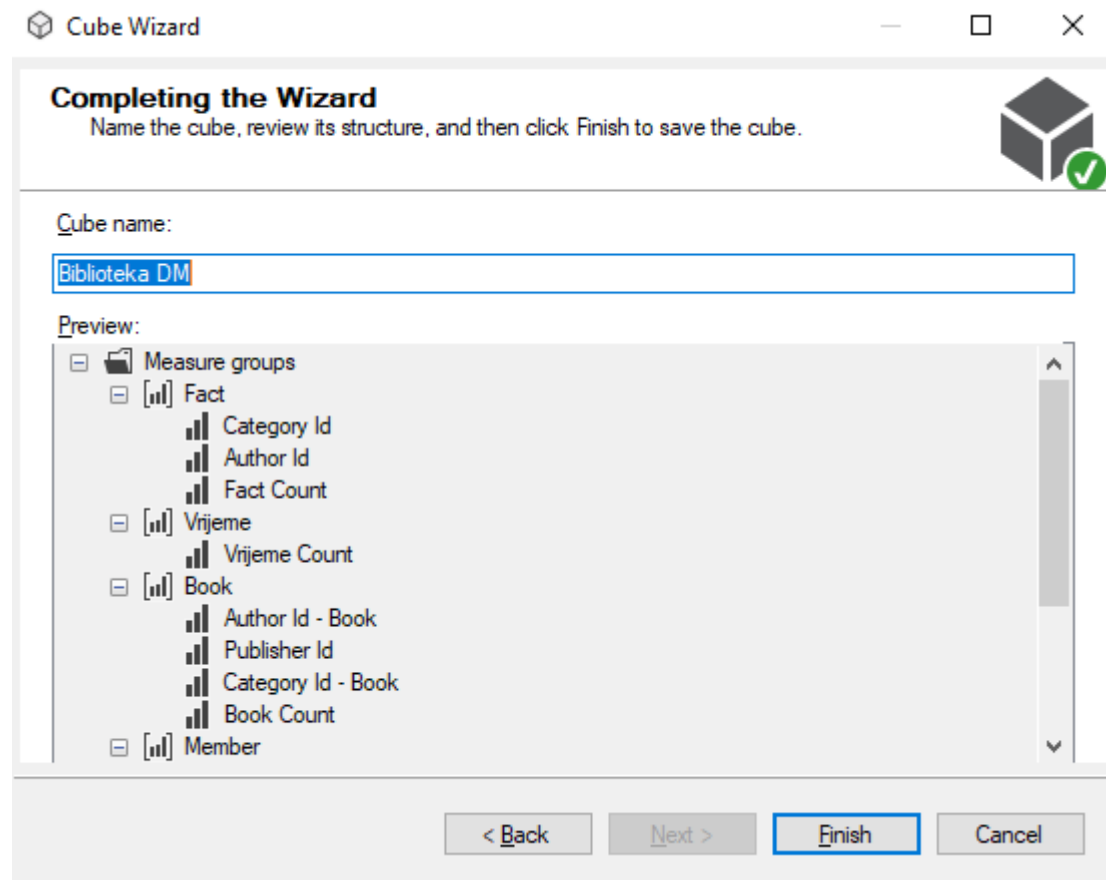
- Fact
  - Attributes
    - Borrow Id
    - Book Id
    - Member Id
    - Vrijeme Id

< Back   Next >   **Finish**   Cancel

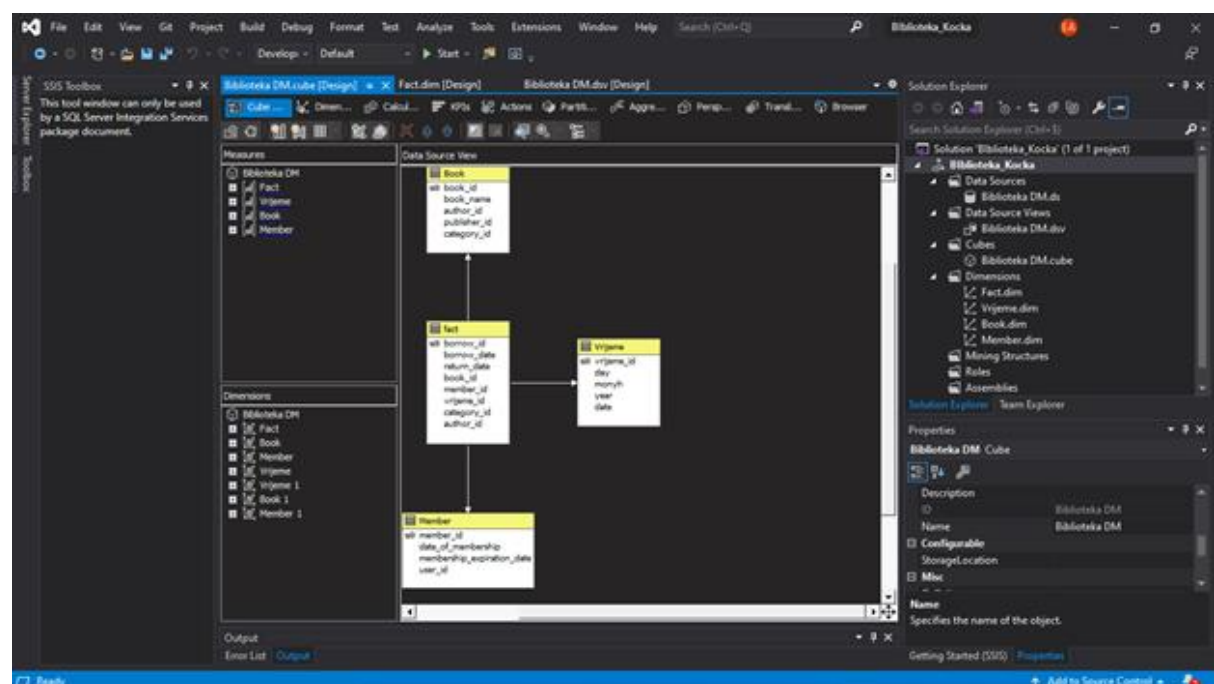
Klikom na Finish smo završili sa wizardom te imamo sljedeći izgled:



Kreiranje OLAP kocke se izvodi desnim klikom na Cubes te odabirom New Cube. Dimenzije će same prepoznati jer smo ih kreirali prije samog kreiranja kocke.



Finalni izgled kocke imamo na sljedećoj slici:

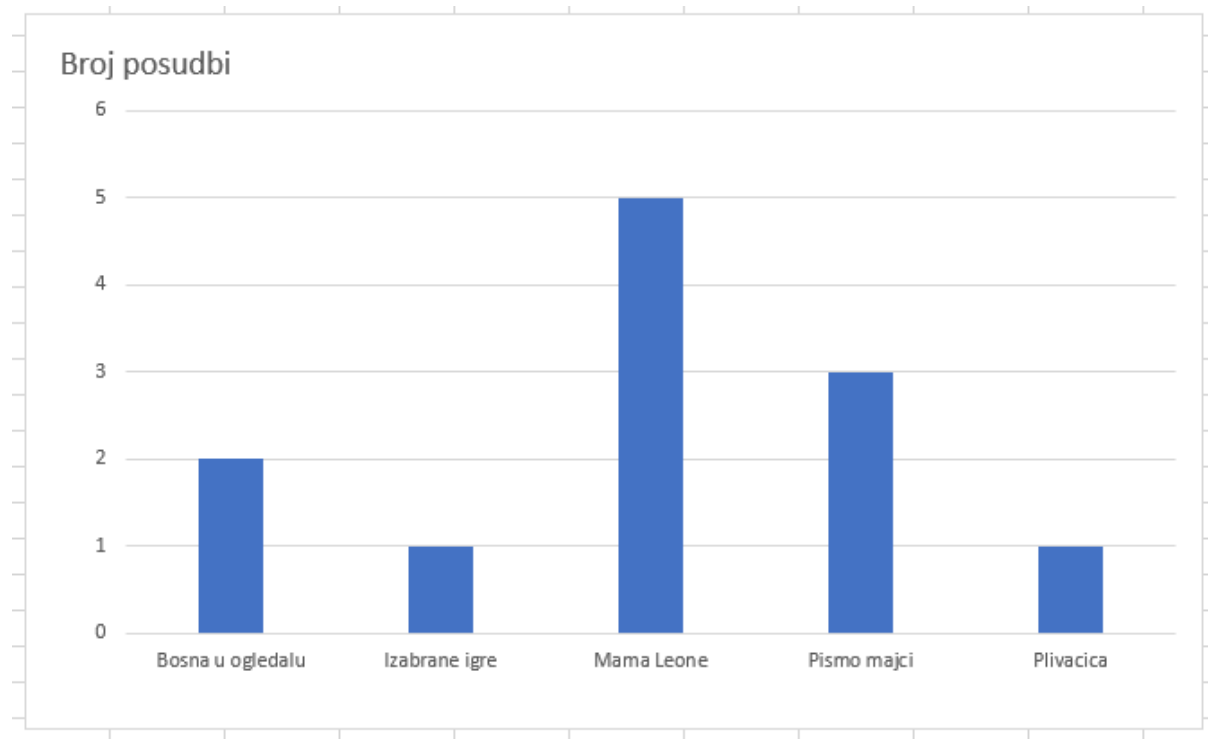


## Analiza podataka

Analiza podataka je proces u kojem se određeni podaci obrađuju. U našem slučaju bitne/potrebne podatke uzimamo putem upita.

Najbitnija stvar za biblioteku je da zna koje knjige se najviše posuđuju na čitanje u kojim vremenskim intervalima. Knjige koje su traženije se mogu naručiti da bi više imali istih. Očekivano je da u vremenu kada učenici imaju tu knjigu za lektiru da je taj mjesec ona jedna od najtraženijih, tako da bibliotekar može taj mjesec davati nešto kraći rok, osoba zadužena za skladište prije tog mjeseca nabaviti par novih knjiga ili posuditi od druge biblioteke i slično.

Primjer analize podataka:



Upit:

```
SELECT b.book_name AS "Ime knjige",count(rb.borrow_id) AS "Broj ponavljanja"

FROM Book b, Fact rb

WHERE b.book_id = rb.book_id

GROUP BY b.book_name;
```

## Proces pravljenja projekta

