

# Administracija baza podataka

---

## Pogledi - VIEW

(Kao objekti relacione baze podataka)

# Pogledi - VIEW

---

- ❑ Pogled se funkcionalno "ponaša" kao "prozor" kroz koji se vide podaci baze podataka
  - Pojednostavljuje korišćenje baze podataka
  - Može se koristiti za zaštitu podataka od neovlašćenog pristupa
  - Sa aspekta performansi – pogledi se čuvaju u kompiliranom obliku
- ❑ Pogled se može posmatrati kao podsHEMA neke šema Relacione baze podataka.
- ❑ Kreirani pogled se u okviru SUBP-a (njegovog rečnika) čuva kao plan izvršenja čijim se izvršavanjem dobijaju podaci koji se "vide" kroz/preko pogleda.

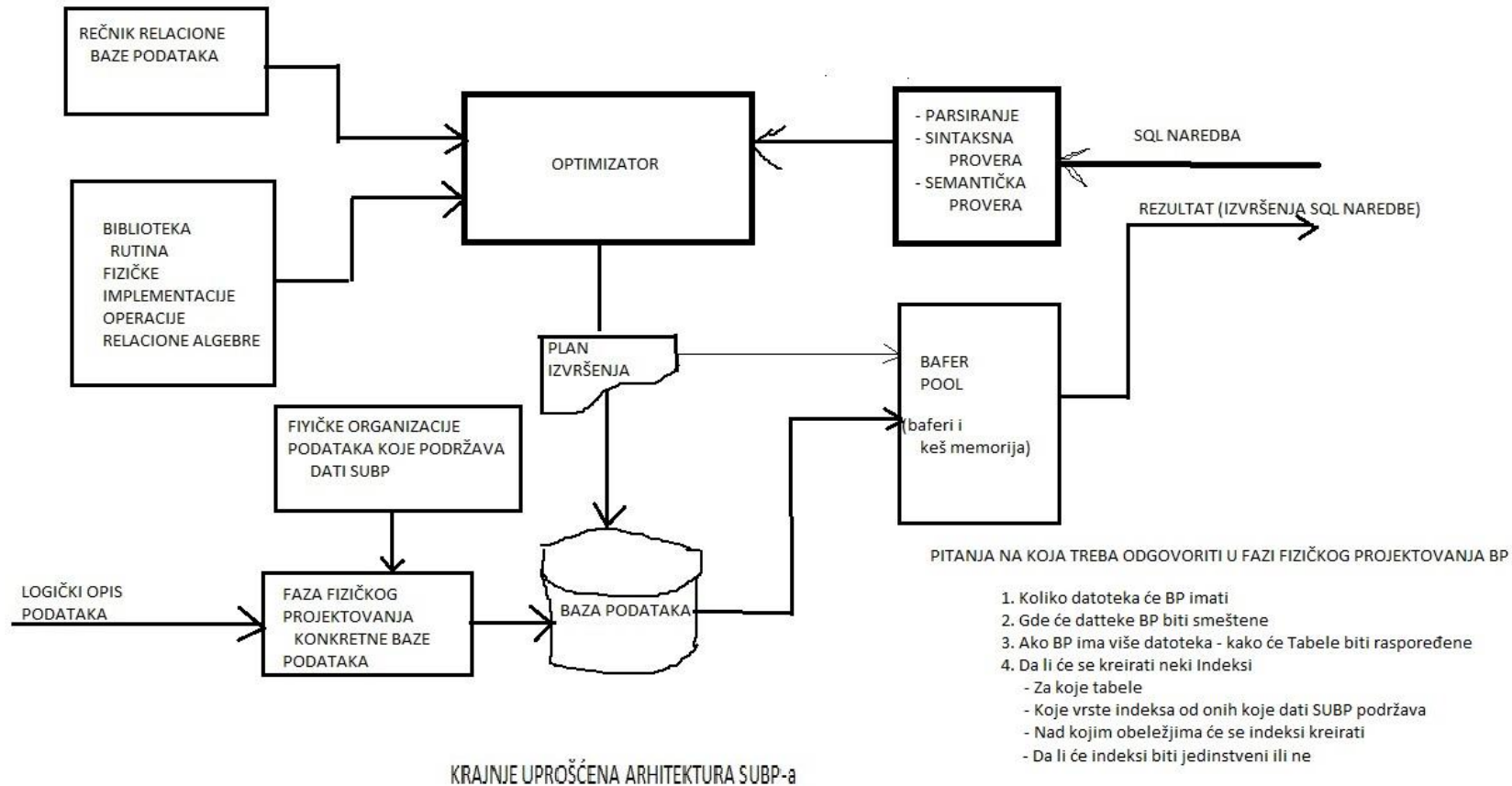
# Sintaksa naredbe za kreiranje pogleda

---

```
CREATE VIEW <naziv_pogleda>  
    [(naziv_obl1 [, naziv_obl2] , . . .)]  
AS Podupit
```

- Podupit – Standardna Select SQL naredba (koju smo obradili iz predmeta Osnove baza podataka)

# Pogled (VIEW) - šta je suštinski



# Pogledi (View) - primer

---

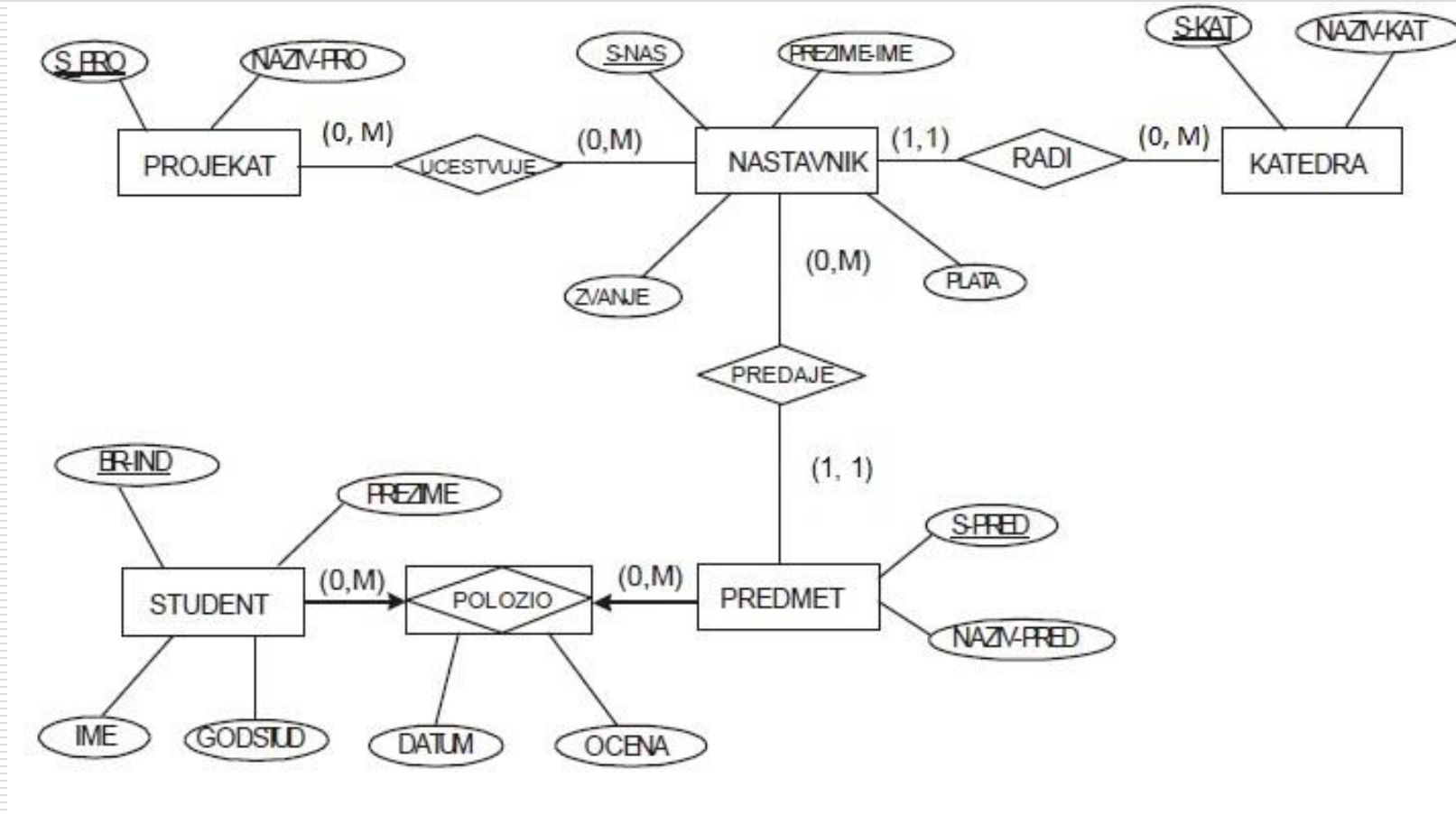
Kreirati pogled `NastavnikPredmet` kroz koji se "vide":

```
IdNas  Prezime_Ime  Naziv  FondCasova  
-----
```

U radu s pogledima uočavamo sledeća 3 koraka:

1. Formiranje i testiranje upita
2. Kreiranje pogleda
3. Korišćenje pogleda

# Konceptualna šema Baze podataka - Fakultet



# Kreiranje pogleda - NastavnikPredmet

---

## 1. Kreiranje podupita

```
Select A.S_Nas IdNas, Prezime_Ime, Naziv_Pred, Casova As 'Fond casova'  
From Nastavnik A,Predmet B  
Where A.S_Nas = B.S_Nas
```

# Kreiranje pogleda - NastavnikPredmet

---

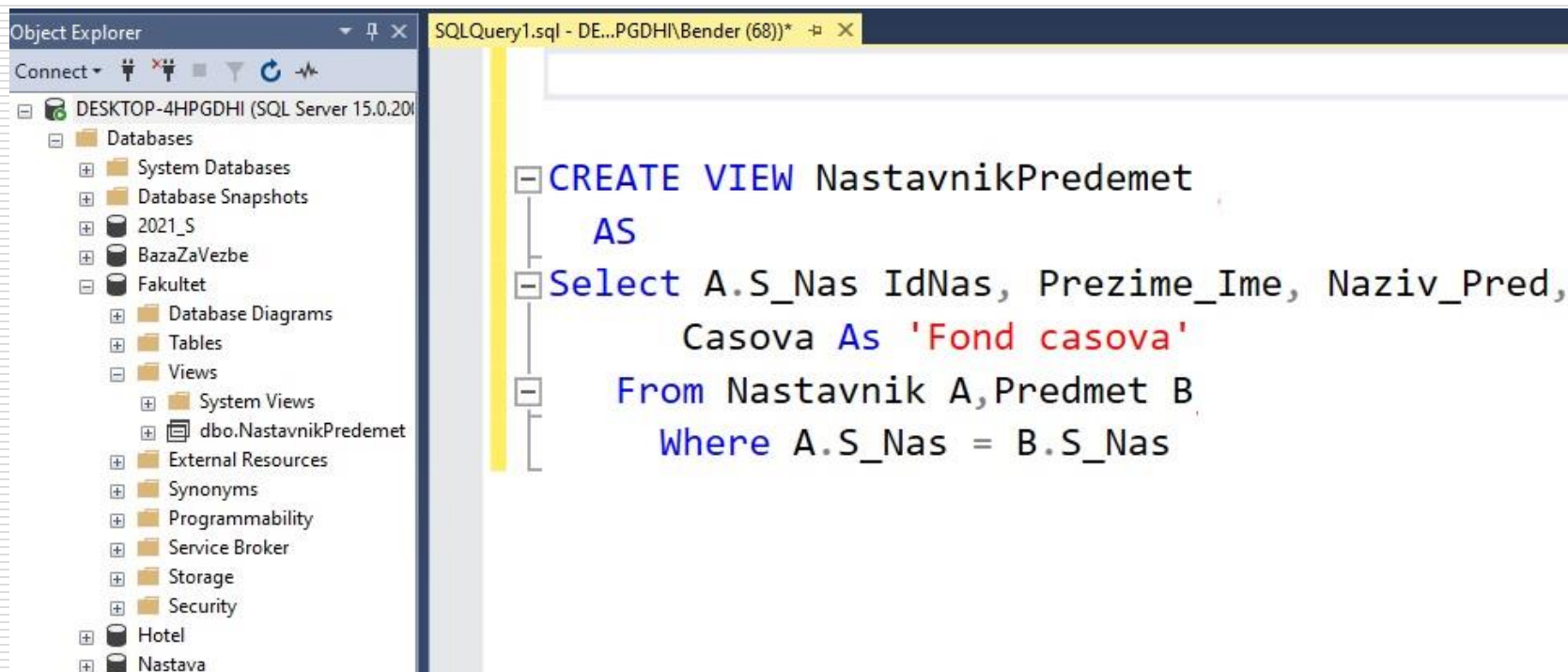
## 2. Kreiranje pogleda

```
CREATE VIEW NastavnikPredmet
AS
Select A.S_Nas IdNas, Prezime_Ime, Naziv_Pred,
       Casova As 'Fond casova'
From Nastavnik A, Predmet B
Where A.S_Nas = B.S_Nas
```



# Kreiranje pogleda - Izvršena DDL naredba kreiranja pogleda

---



# Korišćenje pogleda

---

## 3. Korišćenje pogleda

- Pogledi se koriste kao i bazne tabele.
- Korisnik ne zna da li koristi baznu tabelu ili pogled (odnosno izvedenu tabelu).

Na primer:

```
Select *  
From  
    NastavnikPredemet
```

# Uklanjanje pogleda

---

```
DROP VIEW <Naziv_Pogleda>
```

Ukloniti iz baze pogled pod nazivom NastavnikPredmet.

```
Drop View NastavnikPredmet
```

# Pogledi – Zadaci za vezbu

---

**Z2:** Kreirati pogled PolozeniIspiti kroz koji se vidi:

BrInd	Prezime	Ime	NazivPredmeta	Datum	Ocena
-----	-----	---	-----	-----	-----

1. Koristeći pogled PolozeniIspiti prikazati sve ispite koje je položio student s Brojem Indeksa 'E 7398'
2. Prikazati Broj indeksa i Prezime ime studenata i ocenu kojom je ispit položio, za studente koji su položili ispit i predmeta 'Osnove baza podataka'.

# Pogledi – Zadaci za vezbu

---

```
Create View PolozeniIspiti (BrInd, Prezime, Ime,  
    NazivPredmeta, DatumIspita,Ocena) As  
Select B.Br_Ind, Prezime, Ime, Naziv, Datum, Ocena  
    From Student A, Polozio B, Predmet C  
    Where A.Br_Ind = B.Br_Ind And  
        B.S_Pred = C.S_Pred
```

# Pogledi – Zadaci za vezbu

---

## z2.1:

```
Select NazivPredmeta, DatumIspita, Ocena  
      From StudentPolozio  
      Where BrInd = 'E 7398'
```

# Pogledi – Zadaci za vezbu

---

**z2.2:**

```
Select BrInd, Prezime, Ime, Ocena  
      From PolozeniIspiti  
      Where NazivPredmeta = 'Osnove baza podataka'
```

# Pogledi – Zadaci za vezbu

---

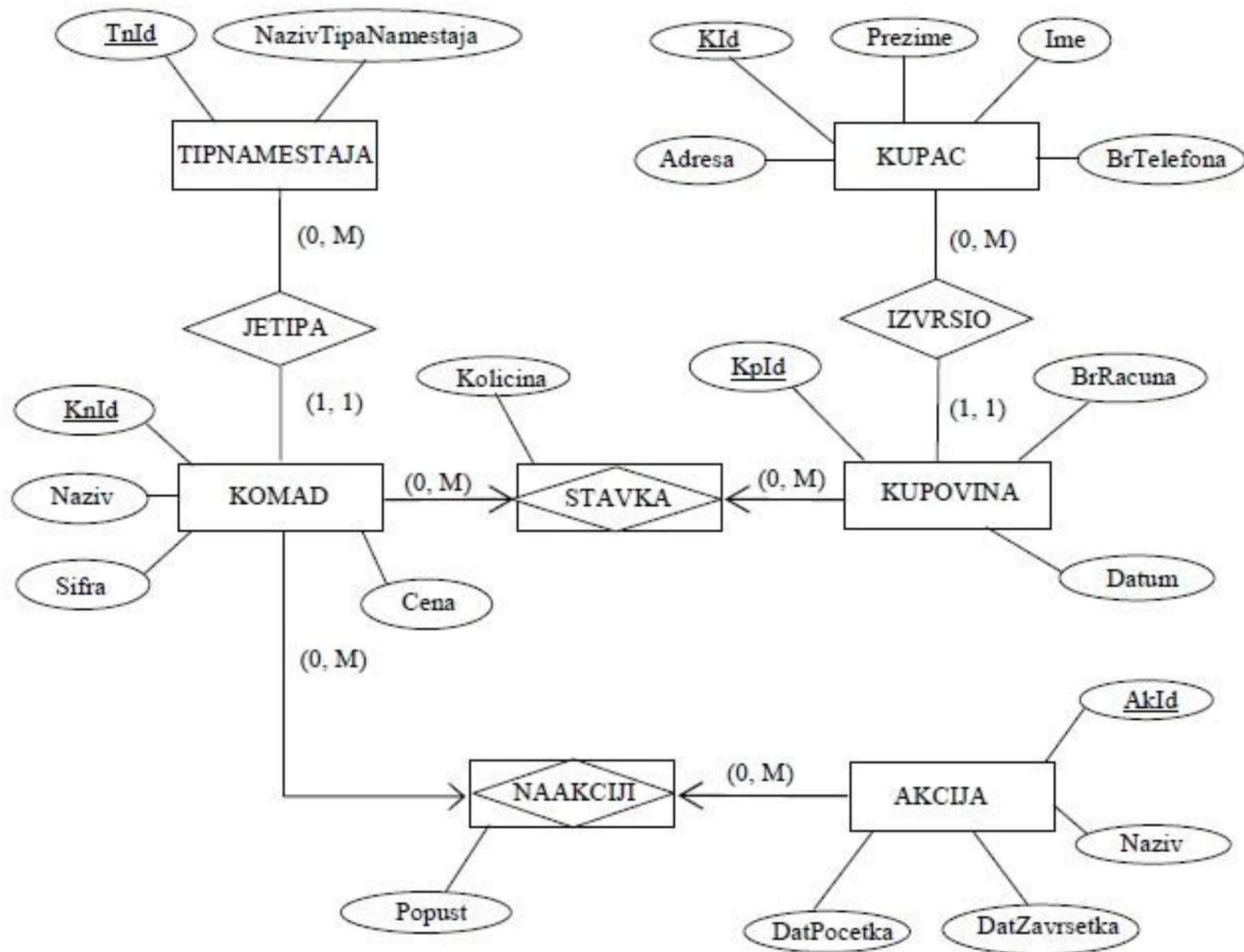
**Z3.** Kreirati pogled Cene kroz koji se vidi:

`NazivTipaSobe (VrstaSobe), NazivTipaIznajmljivanja (Najam) i Cena`

**Z4.** Kreirati pogled Posete kroz koji se vidi:

`Gid, Prezime, Ime gosta, BrSobe, TsId, NazivTipa Sobe, TiId, NazivNajma,  
BrNocenja i Datum dolaska u hotel`





# Pogledi – Zadaci za vezbu

---

## Z3. Kreirati pogled Cene.

```
Create View Cene
```

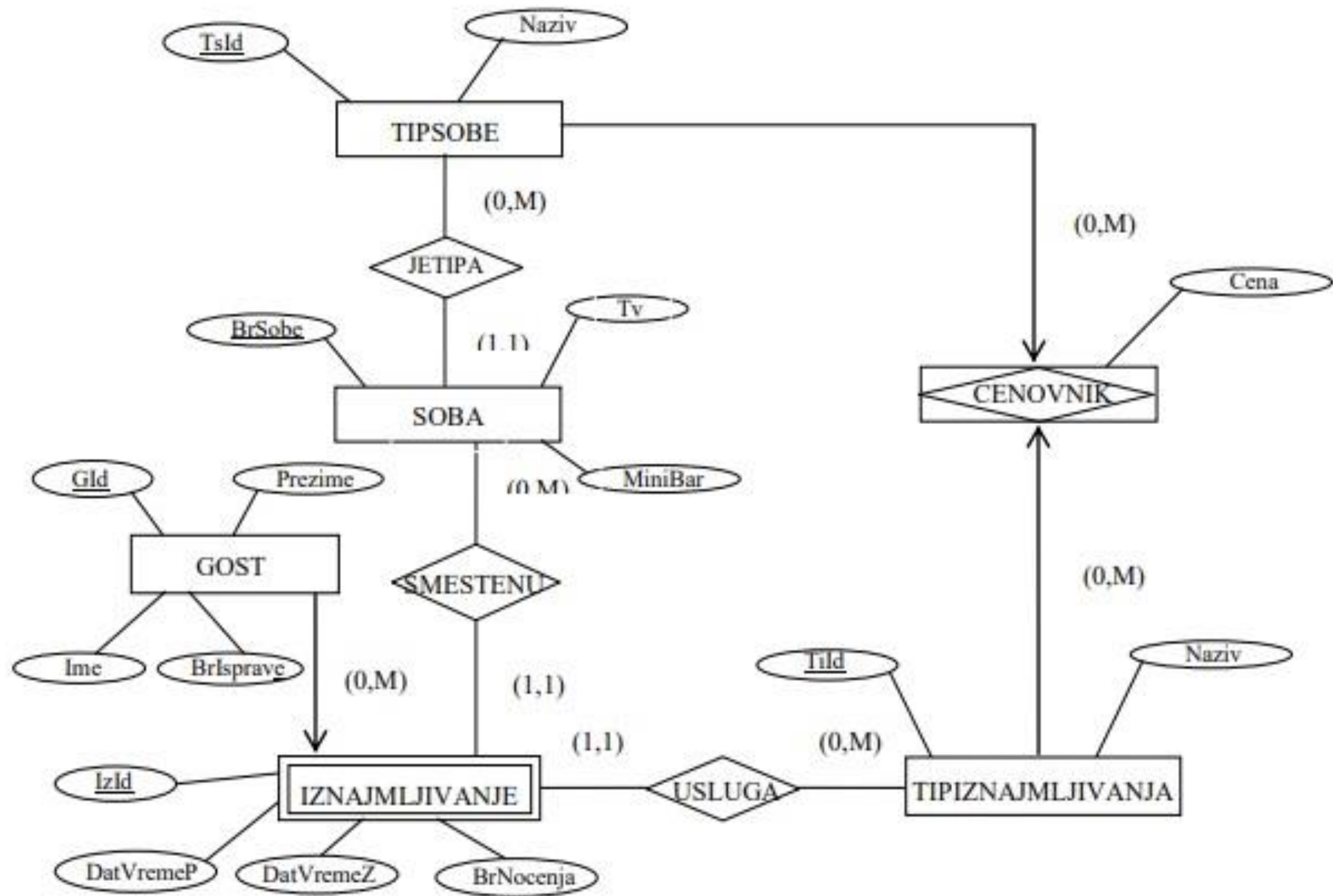
```
(VrstaSobe, Najam, Cena)
```

```
As
```

```
Select NazivTipaSobe, NazivTipaIznajmljivanja, Cena
```

```
From TipSobe A, Cenovnik B, TipIznajmljivanja C
```

```
Where A.TsId=B.TsId And B.TiId=C.TiId
```



# Pogledi – Zadaci za vezbu - (Kreiranje pogleda POSETE)

---

## **24.** Kreiranje pogled Posete

Create View Posete

(Gid, Prezime, Ime, BrSobe, TsId, TipSobe, TiId, TipNajma,  
BrNocenja, Datum)

As

Select A.Gid, Prezime, Ime, E.BrSobe, B.TsId, NazivTipaSobe,  
D.TiId, NazivTipaIznajmljivanja, BrNocenja, DatVremeP

From Gost A, TipSobe B, Soba C, TipIznajmljivanja  
D, Iznajmljivanje E

Where A.Gid = E.Gid And C.BrSobe=E.BrSobe And C.TsID =  
B.TsID And D.TiId = E.TiID

# Zadatak – Koriscenje pogleda POSETE

---

**ZADATAK 5:** Koristeci pogled POSETE za Goste koji su imali vise od dve posete hotelu prikazati Prezime, Ime, Ukupan broj poseta hotlu (BrPoseta), Ukupan Broj nocenja (BrNocenja) i UkupanIznos novca placenog za boravak u Hotelu (za sve posete). Rezultate urediti po opadajucem ukupnom broju nocenja.

# Korišćenje pogleda POSETE

---

## ZADATAK 5 - Rešenje:

```
Select Prezime, Ime, Count(*) As BrPoseta, Sum(BrNocenja) As  
    UkupnoNocenja, Sum(BrNocenja*Cena) As UkupanIznos  
From Cenovnik C, Posete P  
    Where C.TsId = P.TsId And C.TiId = P.TiId  
Group By Prezime, Ime Having Count(*) > 2  
Order By Sum(BrNocenja) Desc
```

# Pogledi - Zadaci

---

## ZADATAK 6:

Kreirati pogled (VIEW) pod nazivom KomadiNaAkciji kroz koji se za sve akcije vide:  
NazivAkcije, BrojKomada - nameštaja na akciji i prosecan procenat (ProsecanPopust)  
- popusta zaokruzen na ceo broj.

# Pogledi - Zadaci

---

## ZADATAK 6 - Rešenje:

```
Create View KomadiNaAkciji
As
Select NazivAkcije, Count(*) As BrKomada,
       Round(Avg(Popust),0) ProsecanPopust
From Akcija A, NaAkciji B
Where A.AkId = B.AkId
Group By NazivAkcije
```



# Pogledi - Zadaci

---

**ZADATAK 7:** Kreirati pogled (VIEW) pod nazivom ProdataKolicina kroz koji se vide:  
Identifikator komada namestaja (KnId), Naziv (Komada namestaja) i  
UkupnaProdataKolicina - svakog Komada, za komade namestaja koji  
su prodani u kolicini vecoj od jednog komada.

# Pogledi - Zadaci

---

## ZADATAK 7 - Rešenje:

Create View

ProdataKolicina(S\_Komada, NazivKomada, ProdataKolicina)

AS

Select A.KnId, Naziv, Sum(Kolicina)

From Komad A, Stavke B

Where A.KnId = B.KnId

Group By A.KnId, Naziv Having Sum(Kolicina) > 1

# Pogledi - Zadaci

---

## ZADATAK 8:

Kreirati pogled (VIEW) pod nazivom KupovineKupca kroz koji se vide: Prezime, Ime, Ukupan broj kupovina i Ukupan iznos novca koji je potrosen na sve njihove kupovine.

# Pogledi - Zadaci

---

## ZADATAK 8 - Rešenje:

```
Create View KupovineKupca
```

```
(KupacId, PrezimeKupca, ImeKupca, BrojKupovina, PotroseniIznos)
```

```
AS
```

```
Select A.Kid, Prezime, Ime, Count(Distinct B.KpId), Sum(Cena *Kolicina)
```

```
From Kupac A, Kupovina B, Stavke C, Komad D
```

```
Where A.Kid = B.Kid And B.KpId = C.KpId And C.KnId = D.KnId
```

```
Group By A.Kid, Prezime, Ime
```

# Pogledi - Zadaci

---

## ZADATAK 9:

Kreirati pogled (VIEW) pod nazivom ProdatiKomadi kroz koji se vide: Identifikator komada nameštaja (KnId), Naziv (Komada nameštaja) , Cena, Ukupna prodana kolicinu po svakom komadu i Ukupan iznos novca dobijen od prodaje svakog komada nameštaja.

# Pogledi - Zadaci

---

## ZADATAK 9 - Rešenje:

```
Create View ProdatiKomadi
    (KomadId, NazivKomada, CenaKomada, ProdataKolicina, IznosProdaje)
AS
    Select A.KnId, Naziv, Cena, Sum(Kolicina), Sum(Kolicina*Cena)
        From Komad A, Stavke B
            Where A.KnId = B.KnId
Group By A.KnId, Naziv, Cena
```

# Information Schema (1/1)

---

- ☐ Šta je Information Schema?
- ☐ Za čega se koristi Information Schema?
- ☐ Zbog čega je uveden koncept INFORMATION SCHEMA?

## Information Schema (2/2)

---

- ❑ **INFORMATION\_SCHEMA** - je skup sistemskih pogleda.
- ❑ Omogućava korisnicima da pristupe metapodacima o strukturi i sadržaju baze podataka.
- ❑ Nalazi se u većini modernih Relacionih SUBP. Ona omogućava korisnicima da pristupe metapodacima o strukturi i sadržaju baze podataka.
- ❑ Koristi za prikazivanje informacija o samoj bazi i objektima baze podataka kao što su tabele, pogledi, ograničenja, funkcije, indeksi itd.
- ❑ Obezbeđuje se da se, skrivajući implementacione detalje same Systemske baze konkretnog SUBP-a, rečniku pristupa na potpuno isti funkcionalan način.
- ❑ Pristup sistemskim podacima preko pogleda Information Schema je potpuno isti kod svih SUBP-ova koji je podržavaju.



# Information Schema – Zadaci (1/3)

---

**ZADATAK:** Prikazati sve relacije (Tabele) u bazi podataka Hotel.

```
Select TABLE_CATALOG, TABLE_NAME, TABLE_TYPE  
      From INFORMATION_SCHEMA.Tables  
Order By TABLE_NAME
```

## Information Schema - Zadaci (2/3)

---

- ❑ **ZADATAK:** Prikazati sva obeležja svih tabela baze podataka FAKULTET za koja je prilikom kreiranja šeme te baze podataka definisano da ne smeju poprimiti *NULL* vrednost.

```
Select Column_Name, Table_Name, Is_Nullable, Data_Type
      From Information_Schema.Columns
      Where Is_Nullable = 'NO'
      Order by Column_Name
```

## Information Schema - Zadaci (3/3)

---

- ☐ **ZADATAK 1:** Prikazati sva obeležja relacije (tabele) IZNAJMLJIVANJE u bazi podataka Hotel.
- ☐ **ZADATAK 2:** Koristeci INFORMATION\_SCHEMA prikazati sve tabele u bazi podataka Fakultet.
- ☐ **ZADATAK 3:** Koristeci INFORMATION\_SCHEMA prikazati sva obelezja tabele Nastavnik iz baze podataka Fakultet.
- ☐ **ZADATAK 4:** Koristeci INFORMATION\_SCHEMA prikazati sve pogledе u bazi podataka Fakultet.