

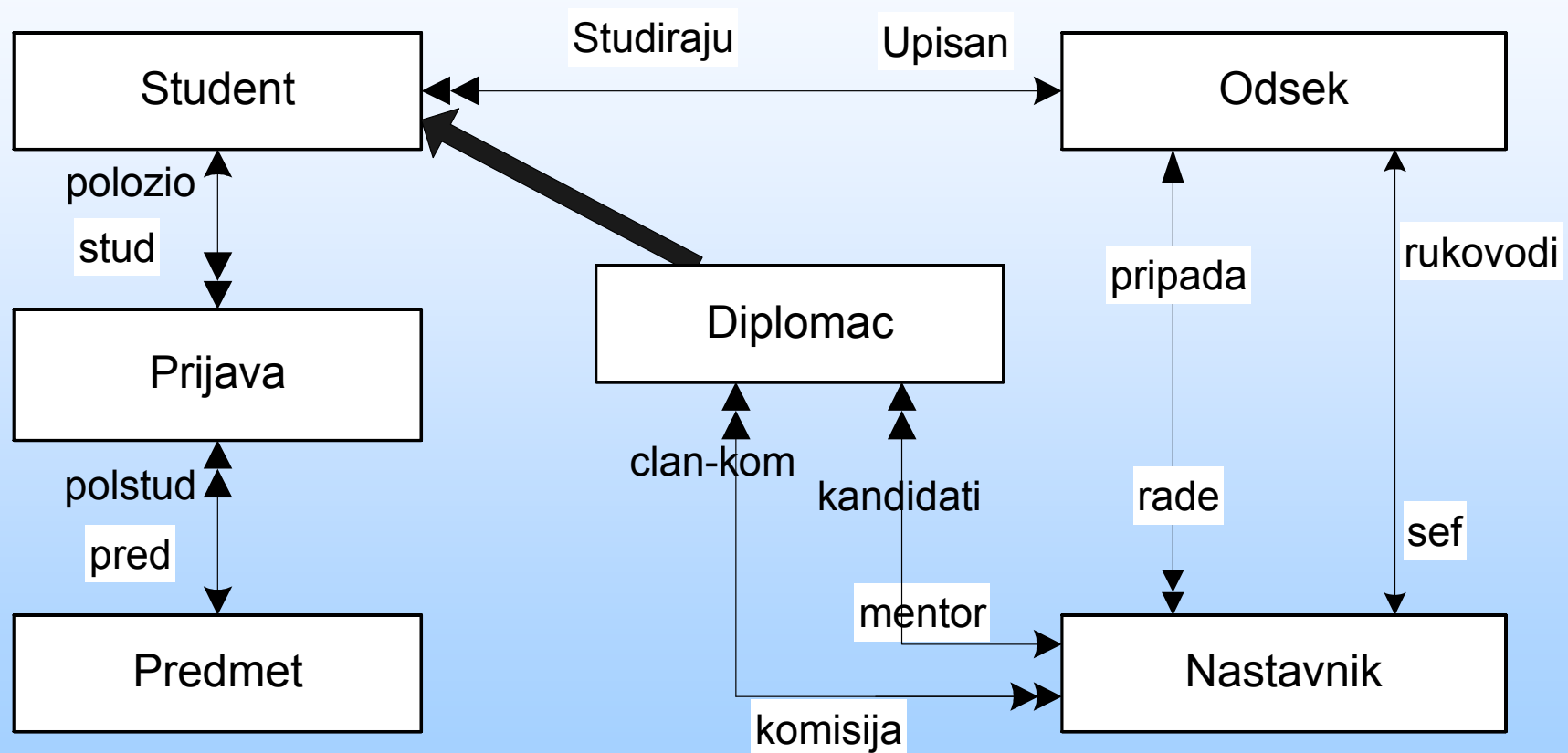
OBJEKTNI UPITNI JEZIK - OQL



OBJEKTNI UPITNI JEZIK - OQL

- POŠTO ODMG RASPOLAŽE SA MNOGO BOGATIJIM SKUPOM TIPOVA, OBJEKTNI UPITNI JEZIK TREBA DA BUDE ZNATNO MOĆNIJI OD RELACIONOG (SQL-A).
- OBJEKTNI UPITNI JEZIK (OQL) NIJE IZVEDEN IZ NEKOG STROGO TEORIJSKOG MODELA, NA PRIMER NEKOG OBJEKTNOG RAČUNA, VEĆ JE DEFINISAN KAO JEDAN JEZIK KOJI NE RADI SAMO SA SKUPOVIMA OBJEKATA, VEĆ I SA KONCEPTIMA STRUKTURE, LISTE I NIZA.
- OQL VEOMA LIČI NA SQL ALI GA ZNATNO PROŠIRUJE SA OO KONCEPTIMA: SLOŽENI OBJEKAT, DEFINICIJA PUTANJE, KORIŠĆENJE OPERACIJA, POLIMORFIZAM, KASNO POVEZIVANJE. KAO NI SQL ON NIJE RAČUNSKI KOMPLETAN.

OBJEKTNI UPITNI JEZIK - OQL: PRIMER OBJEKTNOG MODELA





PRIMER OBJEKTNOG MODELA

```
class Student
    (extent studenti key bi)
{ attribute string bi;
  attribute string ime;
  attribute short starost;
  attribute enum Polovi{M,Z} pol;
  relationship Odsek upisan inverse Odsek::Studiraju;
  relationship set<Prijava> položio inverse Prijava::
stud;
  float sred_ocena ( );
  void unesi_ocenu(in short sp; in Ocene ocena)
      raises (nekorektna_ocena, nekorektna_sp);};
```



PRIMER OBJEKTNOG MODELA

Class Odsek

(**extent** odseci **key** sod)

{

attribute short sod;

attribute string nazivod;

relationship set <Student> studiraju

inverse Student:: upisan;

relationship set <Nastavnik> rade

inverse Nastavnik:: pripada;

relationship Nastavnik sef

inverse Nastavnik:: rukovodi;

};



PRIMER OBJEKTNOG MODELA

class Predmet

(**extent** predmeti **key** sp)

{

attribute short sp;

attribute string nazivpr;

attribute string brcas;

relationship set <Prijava> polstud

inverse Prijava:: pred;

float sred_oc_pr()

};



PRIMER OBJEKTNOG MODELA

```
class Prijava
  (extent prijave)
{
  attribute enum Ocene {6,7,8,9,10} ocena;
  relationship Student stud inverse Student:: polozio;
  relationship Predmet pred inverse Predmet:: polstud;};

class Diplomac extends Student
  (extent diplomci)
  {relationship Nastavnik mentor
    inverse Nastavnik:: kandidati;
  relationship set<Nastavnik> komisija
    inverse Nastavnik:: clan_kom;
  };
```



PRIMER OBJEKTNOG MODELA

class Nastavnik

(**extent** nastavnici **key** snast)

{

attribute short snast;

attribute string imenast;

attribute string zvanje;

relationship Odsek pripada **inverse** Odsek:: rade;

relationship set<Diplomaci> kandidati **inverse** Diplomac ::
mentor;

relationship set<Diplomac> clan_kom **inverse** Diplomac::
komisija;

relationship Odsek rukovodi **inverse** Odsek:: sef;

void izbor (**in string** novo_zvanje);

};



OBJEKTNI UPITNI JEZIK - OQL

- OQL JE FUNKCIONALNI JEZIK U KOME SE OPERATORI SLOBODNO KORISTE, VODEĆI RAČUNA DA OPERAND POŠTUJE ODGOVARAJUĆI TIP. DRUGIM REČIMA REZULTAT UPITA JE NEKI ODMG TIP I MOŽE BITI PONOVO PODVRGNUT UPITU.
- “ULAZNE TAČKE” U BAZU PODATAKA MOGU DA BUDU BILO KOJI OPSEG (EXTENT) NEKE KLASSE ILI BILO KOJI PERZISTENTNI OBJEKAT ČIJE JE IME DEFINISANO PREKO OPERACIJE **bind** OBJEKTA **Database** .
- NA PRIMER “ULAZNA TAČKA” **odseci** UKAZUJE NA PERZISTENTNU KOLEKCIJU OBJEKATA.



OBJEKTNI UPITNI JEZIK - OQL

- KAD GOD SE KOLEKCIJA REFERENCIRA U OQL-u, NEOPHODNO JE NAD NJOM DEFINISATI PROMENLJIVU (TZV "ITERATORSKA" PROMENLJIVA) KOJA UZIMA VREDNOST IZ TE KOLEKCIJE. OVA PROMENLJIVA SE DEFINIŠE NA ISTI NAČIN KAO U SQL-u, u "from" DELU UPITA. UZ "select" DEO SE OPISUJE REZULTAT, A "where" DEFINIŠE USLOV ZA SELEKCIJU ELEMENATA KOLEKCIJE.
- PROMENLJIVU **x** JE MOGUĆE DEFINISATI NA SLEDEĆE NAČINE:

studenti **x**
x in studenti
studenti **as x**



OBJEKTNI UPITNI JEZIK - OQL

```
select x.ime  
from studenti x  
where x.pol = "M";
```

PRIKAZANI UPIT KAO ODGOVOR
VRAĆA TIP **bag** < string>

```
select distinct x.ime  
from studenti x  
where x.pol = "M";
```

PRIKAZANI UPIT KAO ODGOVOR
VRAĆA TIP **set** < string>



OBJEKTNI UPITNI JEZIK - OQL

- NIJE NEOPHODNO DA SE UPIT POSTAVLJA PREKO "select...from...where" KLAUZULE. U NAJPROSTIJEM SLUCAJU JE DOVOLJNO NAVESTI NAZIV PERZISTENTNOG OBJEKTA. SVAKI PERZISTENTNI OBJEKAT SAM PO SEBI SE U OQL-u TRETIRA KAO UPIT. NA PRIMER, UPIT

odseci;

VRAĆA REFERENCU NA KOLEKCIJU PERZISTENTNIH OBJEKATA TIPA Odsek. ILI, AKO JE DEFINISANO IME PERZISTENTNOG OBJEKTA "odsekZaIS" TADA UPIT

odsekZaIS;

VRAĆA REFERENCU NA TAJ POSEBAN OBJEKAT

OQL- PUTANJA

- "IZRAZ PUTANJE" (PATH EXPRESSION) OMOGUĆAVA DA SE, POLAZEĆI OD NEKE "ULAZNE TAČKE" DOSPE DO ŽELJENOG OBJEKTA ILI ATRIBUTA
- SPECIFIKOVANJE PUTANJE OBIČNO POČINJE OD IMENOVANOG PERZISTENTNOG OBJEKTA ILI OD PROMENLJIVE DEFINISANE NAD KOLEKCIJOM OBJEKATA. ONA JE ISTOVREMENO I UPIT. PRIMER:

```
✓ odsekZaIS.sef; // vra}a objekat tipa Nastavnik  
✓ odsekZaIS.imenast; //vra}a literal tipa string  
✓ odsekZaIS.rade; //vra}a objekat tipa set<Nastavnik>  
odsekZaIS.rade.imenast; //nije dozvoljeno
```



OQL- PUTANJA

- NAD KOLEKCIJOM odsek.rade TREBA DEFINISATI PROMENLJIVU I PREKO NJE POSTAVITI UPIT:

```
select f.imenast  
from odsekZaS.rade f;
```

```
select distinct f.imenast  
from odsekZaS.rade f;
```

PRVI UPIT KAO REZULTAT DAJE TIP **bag<string>**
DRUGI UPIT KAO REZULTAT DAJE TIP **set<string>**

OQL- PUTANJA

- OQL MOŽE DA VRATI KAO REZULTAT KOMPLEKSNU STRUKTURU DEFINISANU U SAMOM UPITU KORIŠĆENJEM KLJUČNE REČI **struct**.

```
select struct (osn_pod: struct (brind : x.bi, imestud : x.ime,  
    ocene: (select struct (predmet:  
        element (select z.nazivpr  
            from y.pred z), ocena: y.ocena)  
    from x.polozio y), srednjaoc: x.sred_ocena)  
from studenti x;
```



OQL- STRUKTURA REZULTATA

PRETHODNI UPIT DAJE KAO REZULTAT STRUKTURU KOJA SE MOŽE PREDSTAVITI KAO NENORMALIZOVANA TABELA

brind	imest	ocene		srednjaoc

OQL-KREIRANJE POGLEDA

- KAO I U SQL-u, I U OQL-u SE MOGU KREIRATI POGLEDI (VIEWS) KOJI SE NAZIVAJU I "IMENOVANI UPITI". NA PRIMER:

```
define studenti_odseka( ime_odseka)
select x
from studenti x
where x.upisan.nazivod = ime_odseka;
```

ISKAZ studenti_odseka(InfSist) DAĆE SKUP
STUDENATA OVOG ODSEKA I NAD NJIM SE MOGU
POSTAVLJATI UPITI.

PRETPOSTAVLJENO JE DA U KLASI STUDENT NIJE
DEFINISANA VEZA **upisan**, VEĆ ATRIBUT

attribute Odsek upisan,
ODNOSNO SAMO JEDNOSMERNNA VEZA STUDENT -> ODSEK.
GORNJA DEFINICIJA POGLEDA POKAZUJE KAKO SE U OVOM SLUČAJU
KREIRA INVERZNA VEZA



OQL - UREĐENJE ODGOVORA I NASLEĐIVANJE

- ATRIBUTI I OPERACIJE NADTIPA KORISTE SE U PODTIPU JER IH OVAJ NASLEĐUJE
- UREĐENJE ODGOVORA VRŠI SE NA NAČIN EKVIVALENTAN SQL-u

```
select struct (nastav: struct (imen : x.imenast,  
rang: x. zvanje, mentor_stud :  
    (select struct (imes: y. ime, prosek: y. sred_ocena)  
    from x.kandidati y  
    order by prosek desc))  
from studenti x  
order by rang asc, imen asc;
```


OQL - OPERACIJE NAD KOLEKCIJAMA

- OPERACIJE NAD KOLEKCIJAMA SU:

- (1) AGREGIRANE FUNKCIJE (**min, max, count, sum, avg**)
- (2) ISPITIVANJE ČLANSTVA U KOLEKCIJI (**in**)
- (3) KVANTIFIKATORSKI IZRAZI (**exists, for all**)

```
count( select x  
      from prijava x  
      where x.stud.bi = "33/99");
```

```
select x.imenast  
      from nastavnici x  
      where count (x.clan_kom) > 100);
```



```
select x.ime, x.upisan. nazivod
from student x
where "Baze podataka" in
      (select y.pred. nazivpr
       from x.polozio);
```


Pripadnost kolekciji

(exists v in c:b)

Egzistencijalni kvantifikator
vraća **true** ako postoji barem jedan
element u c koji zadovoljava b.

(for all v in c:b)

Univerzalni kvantifikator (**za svako**)
vraća **true** ako svi elementi
kolekcije c zadovoljavaju b.



```
exists g in (select x  
             from studenti x  
             where x.upisan = "InfSist")  
:g. sred_oc_pred = 10;
```

(Da li postoji student Odseka za InfSist koji ima prose~nu ocenu 10.)

```
for all g in (select x  
             from diplomci x  
             where x.mentor.pripada = "InfSist")  
: g. sred_oc_pred > 8.5;
```

Da li svi diplomci ~iji su mentori sa odseka za InfSist imaju prosek ve}i od 8.5



OQL-OPERACIJE AŽURIRANJA BAZE

- ODL KLASSE NISU POTPUNO UČAURENE, POŠTO SE POJEDINIM KOMPONENTAMA NE PRISTUPA ISKLJUČIVO PREKO OPERACIJA, VEĆ I PREKO UPITNOG JEZIKA.
- OQL ZBOG TOGA **NEMA OPERACIJE AŽURIRANJA**, JER SAMO PRETRAŽIVANJE BAZE NIJE BITNO NARUŠAVANJE UČAURENJA.