Basi di dati

Giuseppe De Giacomo

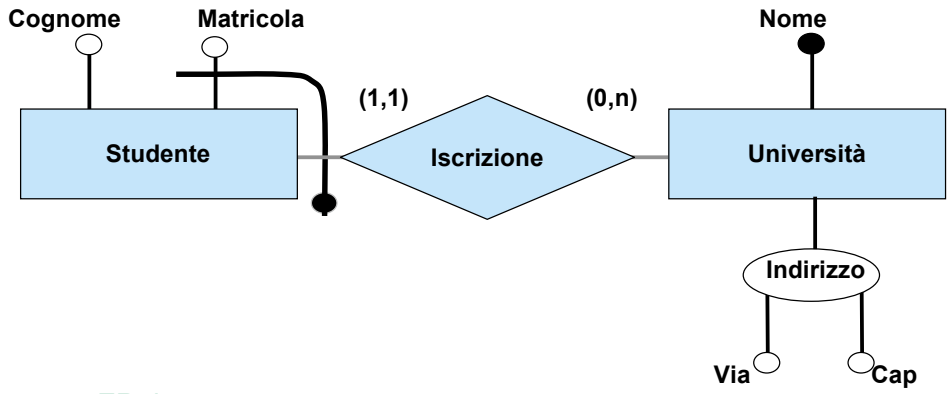
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
Università di Roma "La Sapienza"

Anno Accademico 2007/08
Canale M-Z

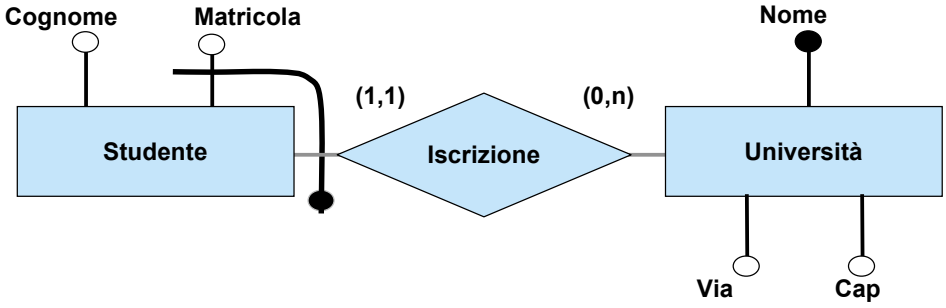
Tradizione di schemi ER ristrutturati in relazionale (parte2)

http://www.dis.uniroma1.it/~degiacomo/didattica/basidati/

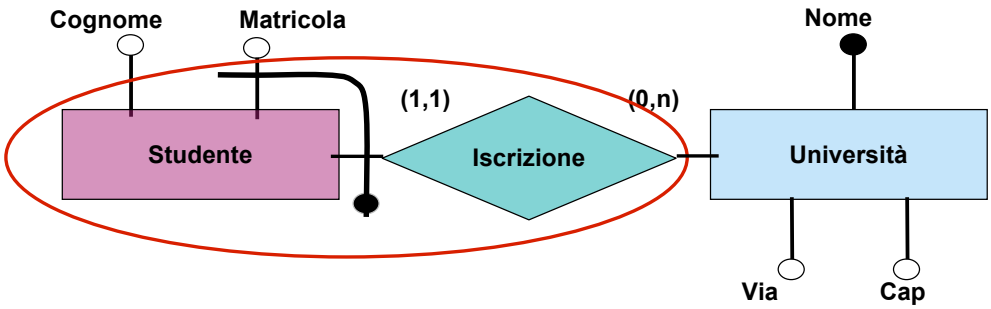
ER



ER ristrutturato



ER ristrutturato



SQL

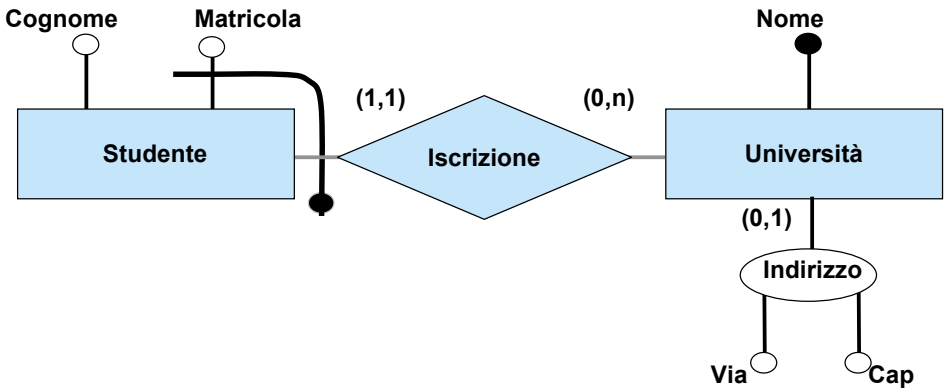
```
create table StudenteIscrizione(
  matricola varchar(20) not null,
  cognome varchar (30) not null,
  universita varchar(30) not null,
  primary key(matricola, universita),
  foreign key (universita) references
                          Universita(nome)
)
create table Universita(
  nome varchar(30) primary key,
  via varchar(40) not null,
  cap varchar(10) not null,
)
```

Schema Relazionale

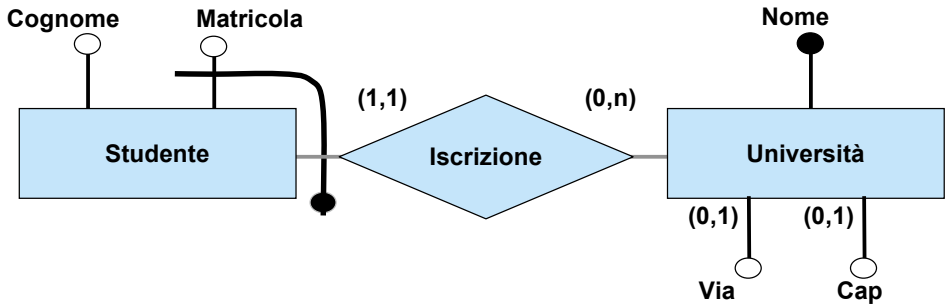
StudenteIscrizione (matricola, cognome, universita)
FK: StudenteIscrizione[universita] ⊆ Universita[nome]

Universita(nome,nabitanti)

ER

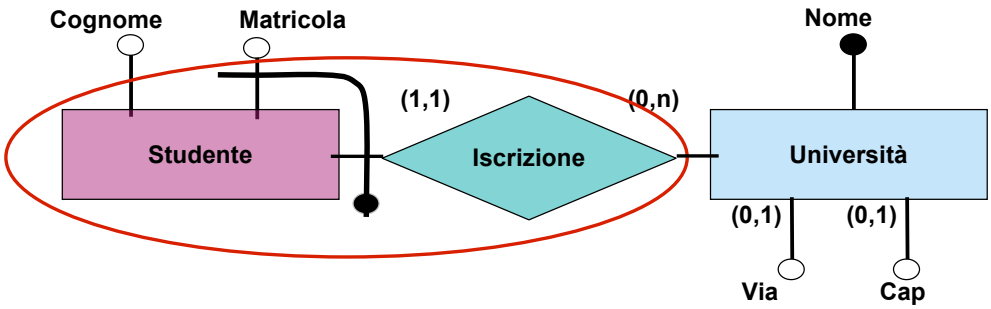


ER ristrutturato



Vincolo Estreno: ***Via e Cap o sono presenti entrambi o mancano entrambi***

ER ristrutturato



SQL

```
create table StudenteIscrizione(
  matricola varchar(20) not null,
  cognome varchar (30) not null,
  universita varchar(30) not null,
  primary key(matricola, universita),
  foreign key (universita) references
                        Universita(nome)
)

create table Universita(
  nome varchar(30) primary key,
  via varchar(40),
  cap varchar(10),
  check ((nome is null and cap is null) or
          (via is not null and cap is not null)
         )
)
```

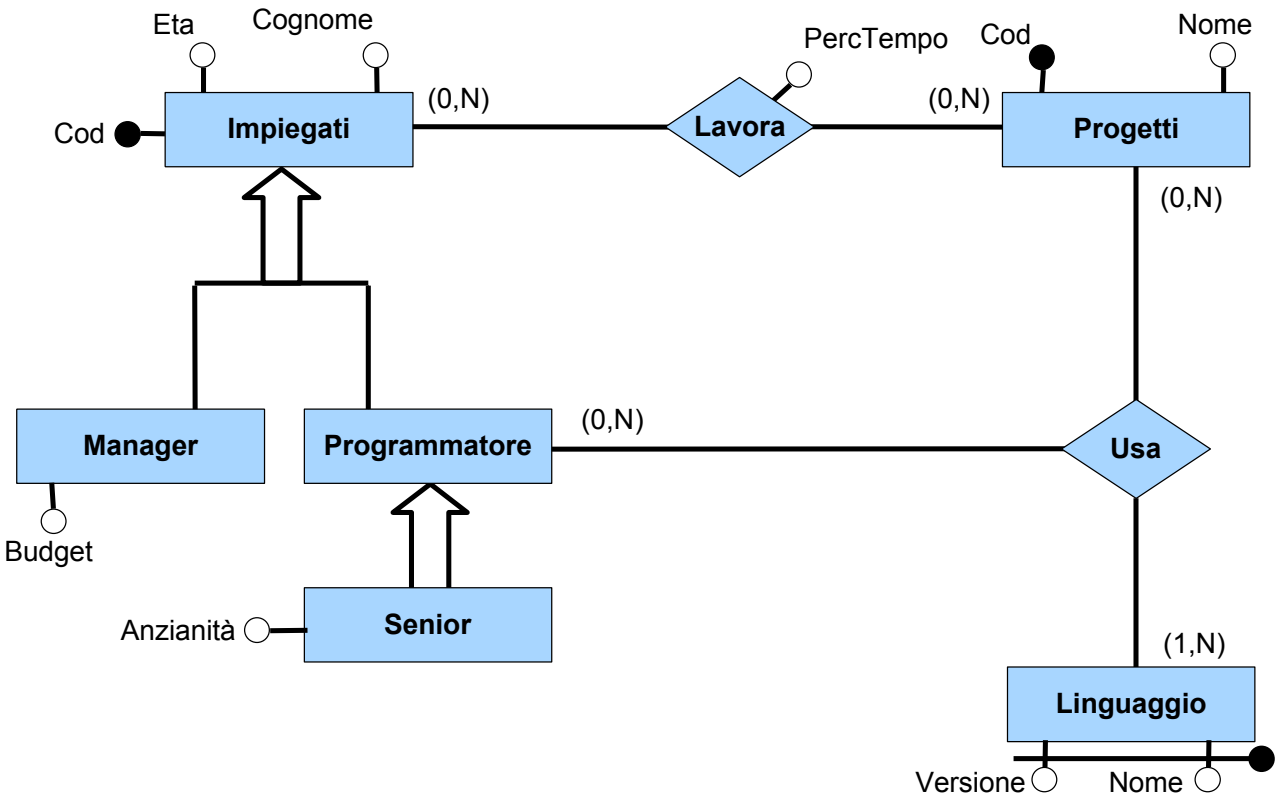
Schema Relazionale

StudenteIscrizione (matricola, cognome, universita)
FK: StudenteIscrizione[universita] ⊆ Universita[nome]

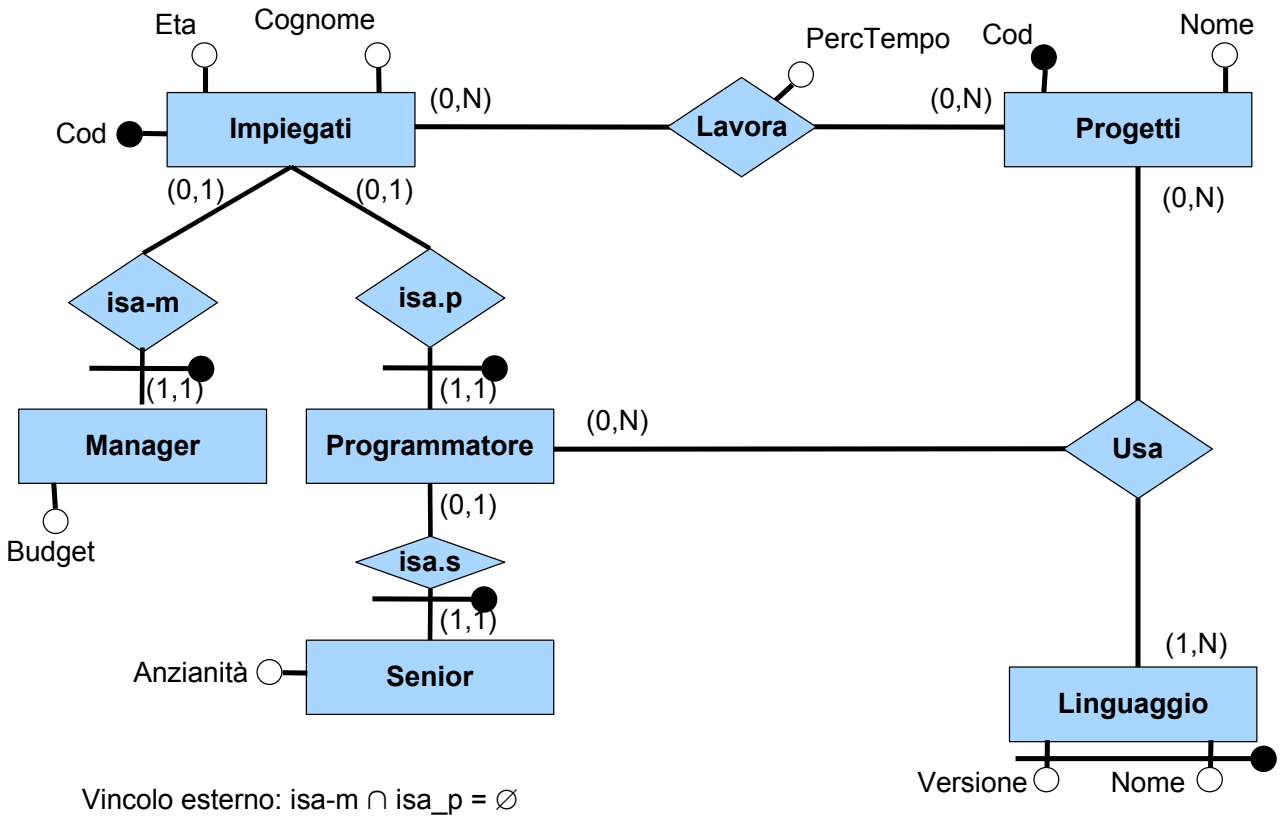
Universita(nome,nabitanti)

Vincolo Estreno: ***Via e Cap o sono peresenti entrambi o mancano entrambi***

Esercizio: ristrutturare e tradurre in relazionale



Esercizio: soluzione - ER ristrutturato



Vincolo esterno: isa-m ∩ isa_p = ∅

SQL

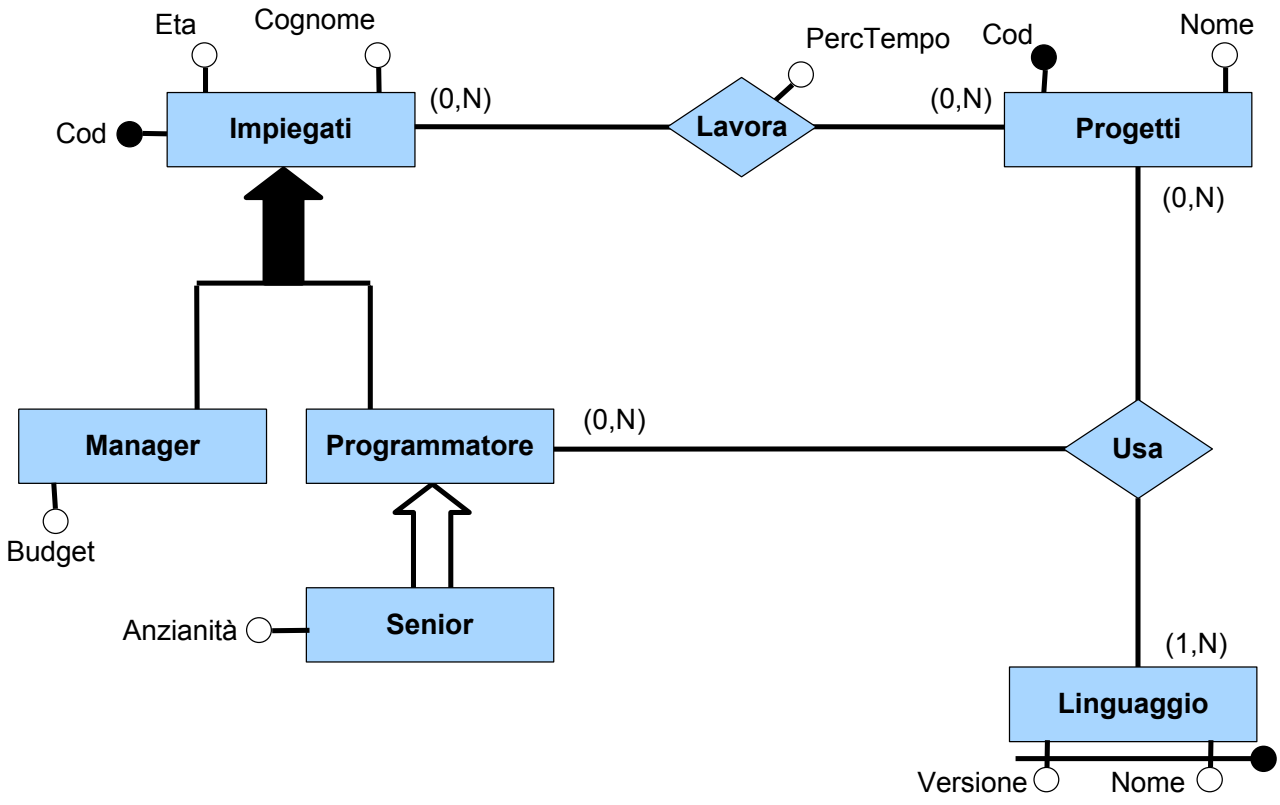
```
create table Impiegati(
  cod varchar(20) primary key,
  eta int not null,
  nome varchar(30) not null,
  cognome varchar(30) not null,
)

create table Progetti (
  cod varchar(10) primary key,
  nome varchar(30) not null
)

create table Lavora(
  impiegato varchar(20) not null,
  progetto varchar(10) not null,
  perctempo real not null,
  foreign key (impiegato) references Impiegati(cod),
  foreign key (progetto) references Progetti(cod),
  primary key (impiegato, progetto)
)

…
```

SQL

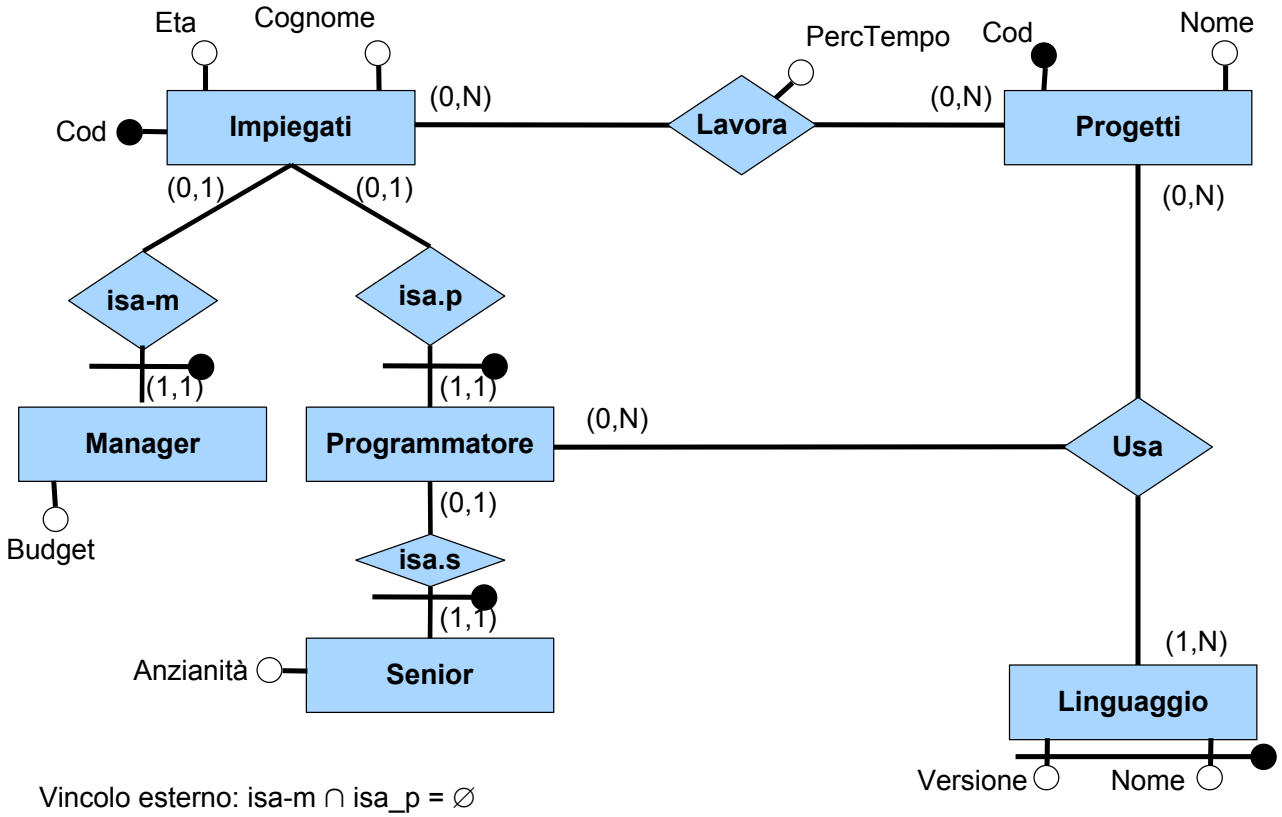
```
…
create table Manager(
  cod varchar(20) primary key,
  budget real not null,
  foreign key (cod) references Impiegato(cod)
)

create table Programmatore(
  cod varchar(20) primary key,
  foreign key (cod) references Impiegato(cod)
)

create table Senior(
  cod varchar(20) primary key,
  anzinita int not null,
  foreign key (cod) references Programmatore(cod)
)

create table Usa(
  programmatore varchar(20) not null,
  progetto varchar(20) not null,
  nomeL varchar(30) not null,
  versioneL varchar(30) not null,
  primary key
        (progammatore, progetto,nomeL,verisoneL),
  foreign key (programmatore)
              references Programmatore(cod),
  foreign key (progetto) references Progetto(cod),
  foreign key (nomeL,versioneL)
              references Linguaggio(nome,veisione)
```

```
create table Linguaggio(
  nome varchar(30) not null,
  versione varchar(30) not null,
  primary key (nome, versione),
  check ((nome,versione) in (
              select nomeL,versioneL
              form Usa
            )
          )
)

create assertion DisjointISA check (
  not exists (
     select *
     from Manager m,Programmatore p
     where m.cod = p.cod
  )
)
```

Esercizio: ristrutturare e tradurre in relazionale



Esercizio: soluzione - ER ristrutturato



Vincolo esterno: isa-m ∩ isa_p = ∅
Vincolo esterno: ogni impiegato partecipa a isa-m o a isa_p

SQL

```
create table Impiegati(
  cod varchar(20) primary key,
  eta int not null,
  nome varchar(30) not null,
  cognome varchar(30) not null,
)

create table Progetti (
  cod varchar(10) primary key,
  nome varchar(30) not null
)

create table Lavora(
  impiegato varchar(20) not null,
  progetto varchar(10) not null,
  perctempo real not null,
  foreign key (impiegato) references Impiegati(cod),
  foreign key (progetto) references Progetti(cod),
  primary key (impiegato, progetto)
)
…
```

SQL

```
…
create table Manager(
  cod varchar(20) primary key,
  budget real not null,
  foreign key (cod) references Impiegato(cod)
)

create table Programmatore(
  cod varchar(20) primary key,
  foreign key (cod) references Impiegato(cod)
)

create table Senior(
  cod varchar(20) primary key,
  anzinita int not null,
  foreign key (cod) references Programmatore(cod)
)

create table Usa(
  programmatore varchar(20) not null,
  progetto varchar(20) not null,
  nomeL varchar(30) not null,
  versioneL varchar(30) not null,
  primary key
        (progammatore, progetto,nomeL,verisoneL),
  foreign key (programmatore)
              references Programmatore(cod),
  foreign key (progetto) references Progetto(cod),
  foreign key (nomeL,versioneL)
              references Linguaggio(nome,veisione)
```

```
create table Linguaggio(
  nome varchar(30) not null,
  versione varchar(30) not null,
  primary key (nome, versione),
  check ((nome,versione) in (
              select nomeL,versioneL
              form Usa
            )
          )
)

create assertion DisjointISA check (
  not exists (
     select *
     from Manager m,Programmatore p
     where m.cod = p.cod
  )
)

create assertion CompleteISA check (
  not exists (
     select cod
     from Impiegato
     except
     (   select cod
         from Manager
         union
         select cod
         from Programmatore
     )
  )
```

SQL

```
create table Impiegati(
  cod varchar(20) primary key,
  eta int not null,
  nome varchar(30) not null,
  cognome varchar(30) not null,
  check ( cod in (select cod from Manager)
            or
            cod in (select cod from Programmatore)
          )
)

…
```

Soluzione alternativa

```
…

create assertion DisjointISA check (
  not exists (
     select *
     from Manager m,Programmatore p
     where m.cod = p.cod
  )
)

create assertion CompleteISA check (
  not exists (
     select cod
     from Impiegato
     except
     (   select cod
         from Manager
         union
         select cod
         from Programmatore
     )
  )
)
```