BASE DI DATI

Esercizi

- Progettazione concettuale
- Progettazione logica

Concetti avanzati SQL:

- Raggruppamento
- Nidificazione

Informatica Umanistica

Università di Pisa

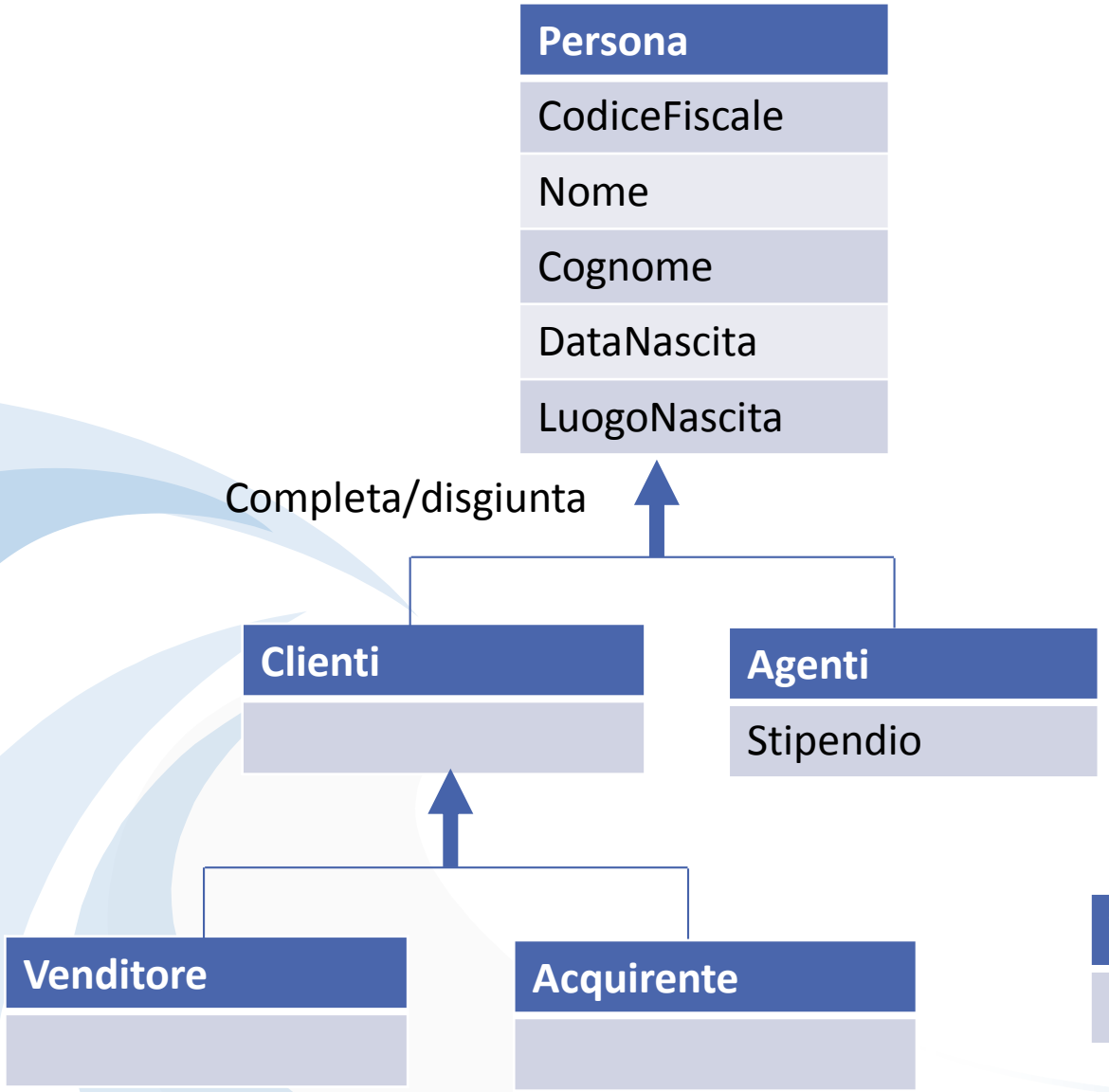
Esercizio: Agenzia immobiliare

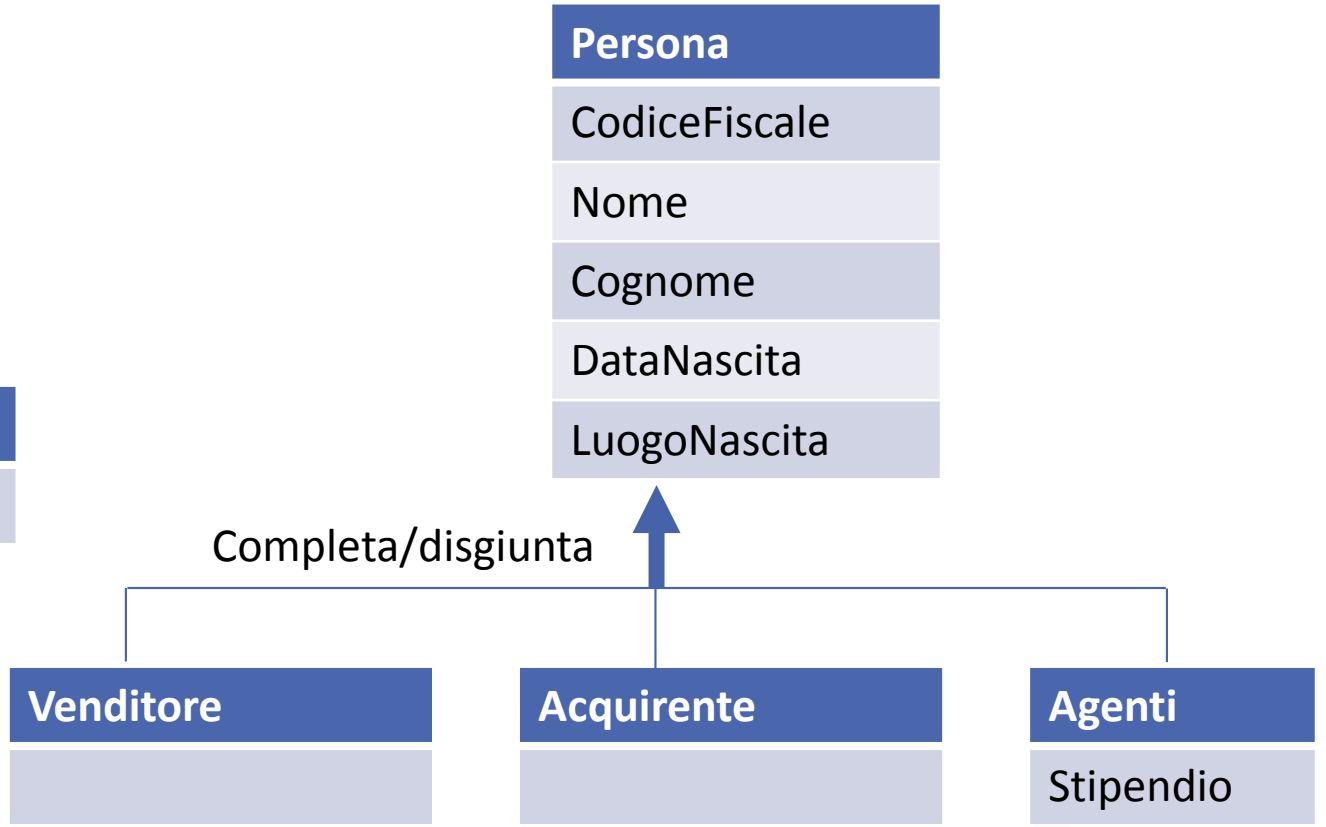
Si considerino i seguenti fatti riguardanti una base di dati relativa alla gestione di un'agenzia immobiliare.

- Nella base di dati figurano tre categorie di persone: gli agenti, gli acquirenti e i venditori. Si noti che una persona può essere una sola di queste figure. Le persone sono caratterizzati dai consueti dati anagrafici.
- Gli agenti immobiliari sono caratterizzati oltre che dagli usuali dati anagrafici anche dallo stipendio.
- I venditori mettono la loro proprietà a disposizione per la vendita.
- Le proprietà, individuate tramite un codice univoco, hanno una tipologia (ad esempio, terreno, appartamento, villino, villa, ….) ed una cubatura. Di una proprietà potrebbero essere indicati il piano e la presenza dell'ascensore (che può assumere valore "si" o "no") nel caso in cui tali informazioni siano conosciute.
- Una proprietà è di un solo venditore, un venditore può avere più proprietà.
- Gli acquirenti sono interessati a visionare degli immobili.
- Il lavoro degli agenti consiste nel mostrare ad uno o più acquirenti le proprietà in vendita durante un appuntamento.
- Di ogni appuntamento interessa la data e il punto di incontro, nel caso in cui l'agente e gli acquirenti si incontreranno in un indirizzo diverso da quello in cui è ubicato la proprietà.
- In una certa data, ovvero ad ogni appuntamento, una proprietà è mostrata da un singolo agente e un agente mostra una singola proprietà ad uno o più acquirenti e riporta, per ciascuno degli acquirenti, una nota. Ad esempio, durante l'appuntamento del "19 gennaio 2016" l'agente "Luigi Neri" fa visionare la proprietà "PalazzoBlu" a diversi acquirenti, fra cui l'acquirente "Mario Rossi", di cui viene riportata la nota: "è interessato alla proprietà ma ritiene il prezzo eccessivo".
- Si noti che un agente può mostrare una proprietà ad un acquirente più volte mediante diversi appuntamenti.

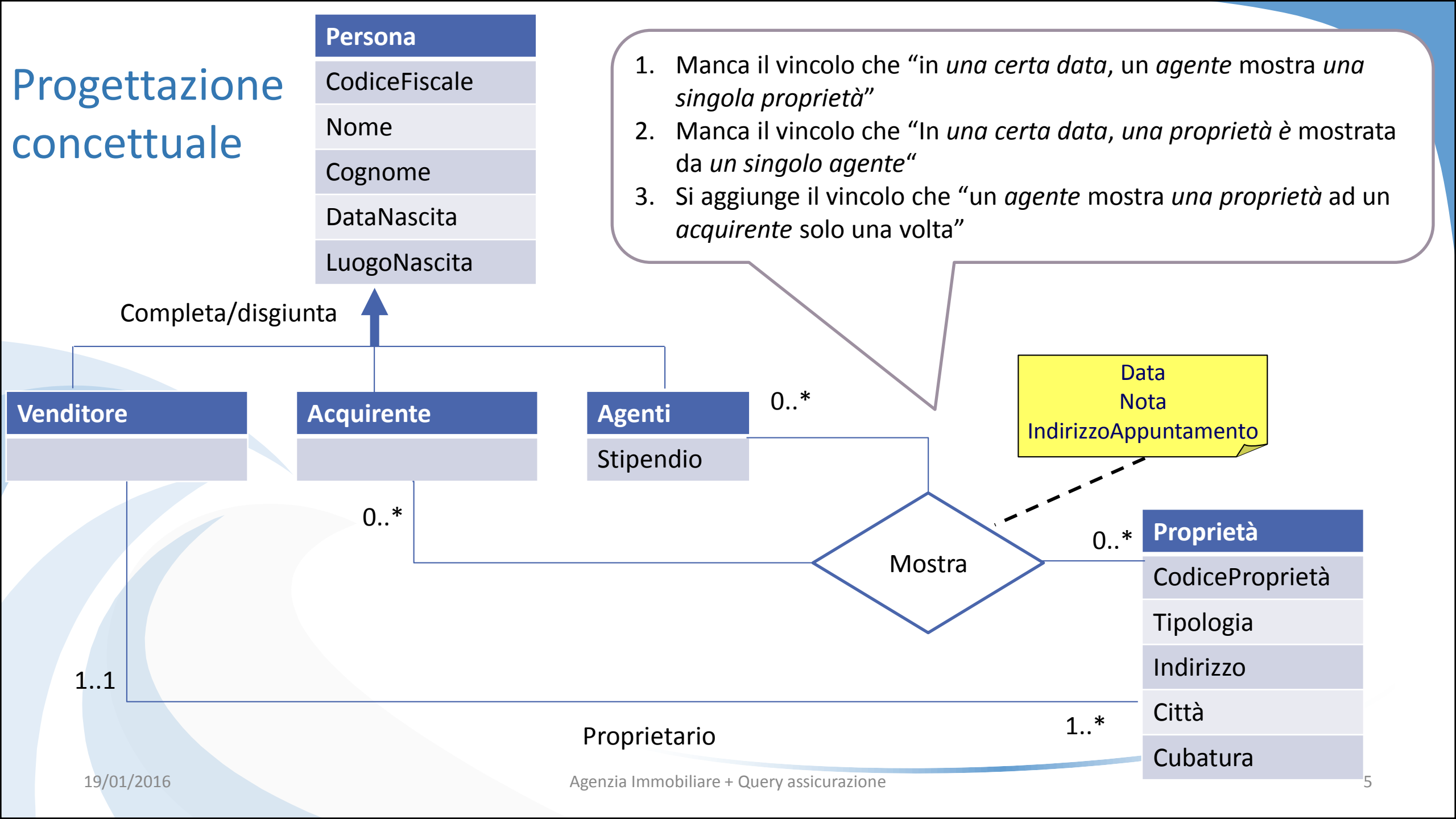
PROGETTAZIONE CONCETTUALE

Generalizzazione (due possibili alternative)



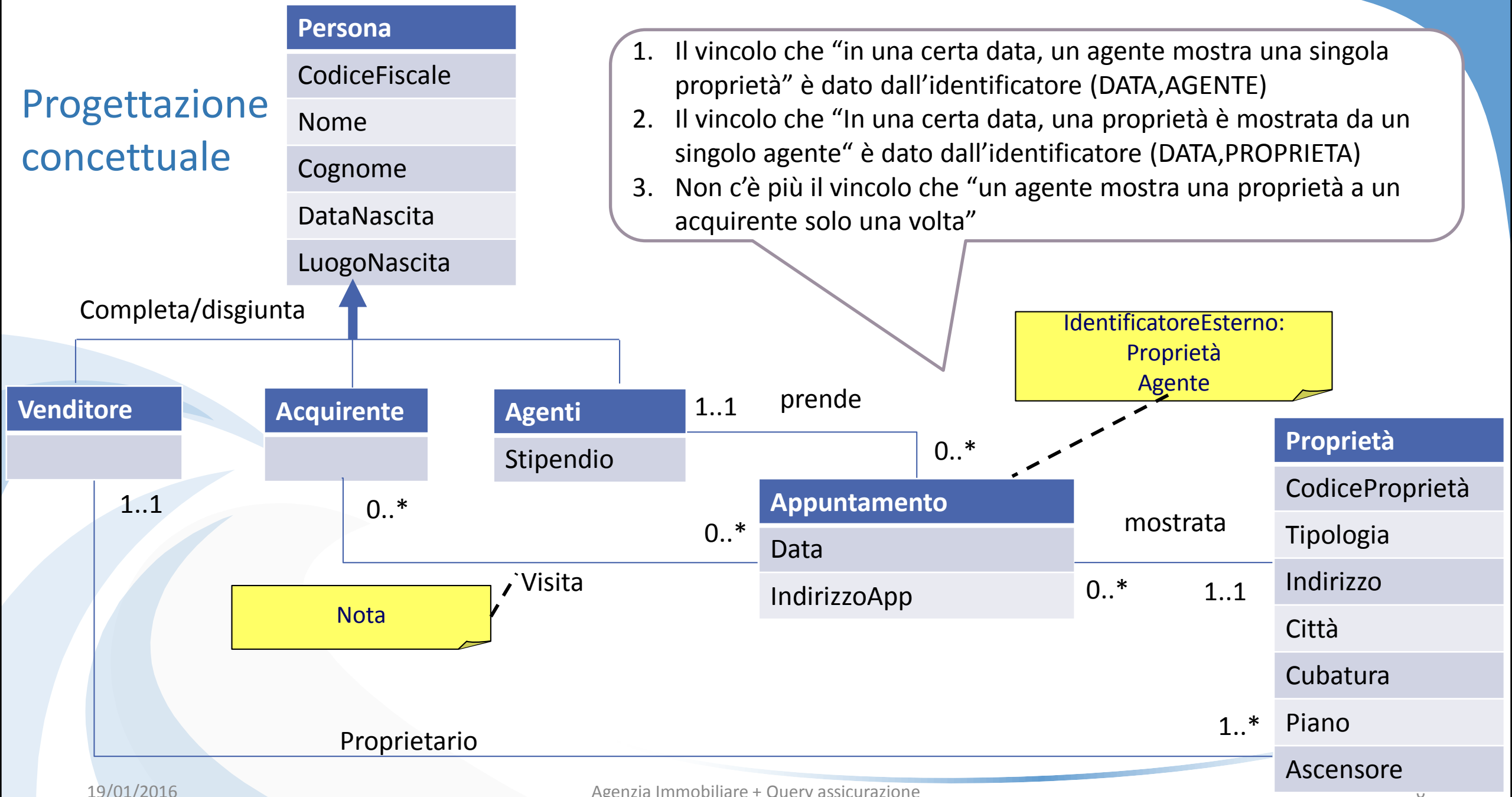


Progettazione concettuale



1. Manca il vincolo che "in *una certa data*, un *agente* mostra *una singola proprietà*"
2. Manca il vincolo che "In *una certa data*, *una proprietà è* mostrata da *un singolo agente*"
3. Si aggiunge il vincolo che "un *agente* mostra *una proprietà* ad un *acquirente* solo una volta"

Progettazione concettuale



1. Il vincolo che “in una certa data, un agente mostra una singola proprietà” è dato dall’identificatore (DATA,AGENTE)
2. Il vincolo che “In una certa data, una proprietà è mostrata da un singolo agente“ è dato dall’identificatore (DATA,PROPRIETA)
3. Non c’è più il vincolo che “un agente mostra una proprietà a un acquirente solo una volta”

PROGETTAZIONE LOGICA

I passo: traduzione iniziale delle classi non coinvolte in gerarchie e chiavi primarie

Proprietà
CodiceProprietà
Tipologia
Indirizzo
Città
Cubatura
Piano
Ascensore

Proprietà	T
codiceProprietà CHAR(16)	PK
Tipologia VARCHAR(16)	
Indirizzo VARCHAR(16)	
Citta VARCHAR(16)	
Cubatura INTEGER	
Piano INTEGER	
Ascensore CHAR(2)	

Il passo: traduzione iniziale delle gerarchie

Soluzione: Solo i figli

Si noti che una persona può essere una sola di queste figure.

Persona
CodiceFiscale
Nome
Cognome
DataNascita
LuogoNascita

Completa/disgiunta

Venditore

Acquirente

Agenti
Stipendio

Agenti	T
CodiceFiscale CHAR(16)	PK
Nome VARCHAR(16)	
Cognome VARCHAR(16)	
DataNascita DATE	
LuogoNascita VARCHAR(16)	
Stipendio INTEGER	

Venditore	T
CodiceFiscale CHAR(16)	PK
Nome VARCHAR(16)	
Cognome VARCHAR(16)	
DataNascita DATE	
LuogoNascita VARCHAR(16)	

Acquirente	T
CodiceFiscale CHAR(16)	PK
Nome VARCHAR(16)	
Cognome VARCHAR(16)	
DataNascita DATE	
LuogoNascita VARCHAR(16)	

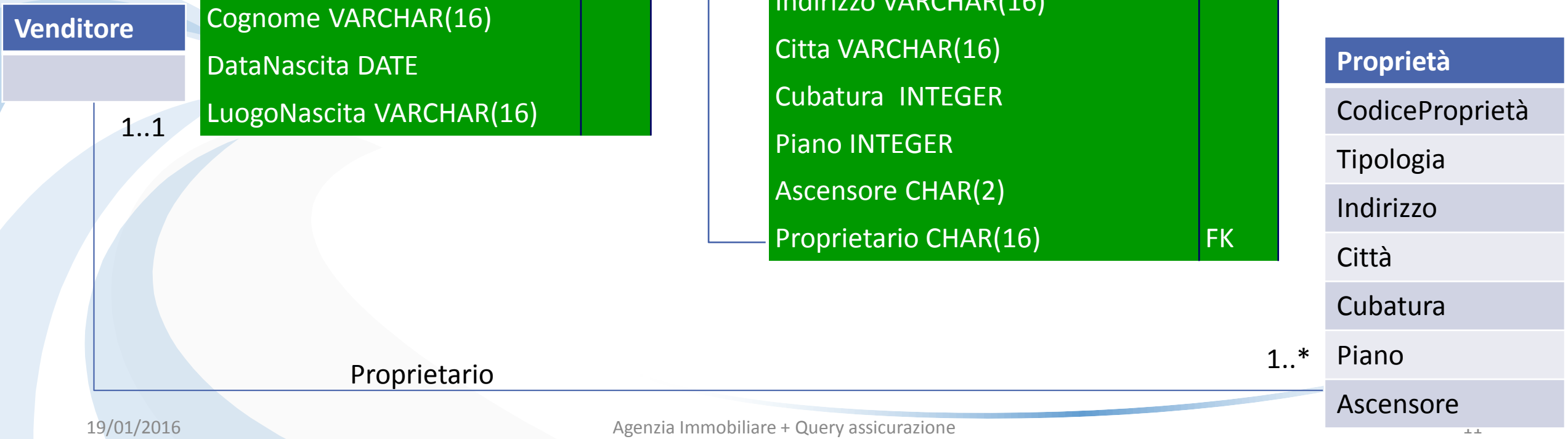
Si potrebbe anche mantenere Persone????

III passo: traduzione degli attributi multivalore

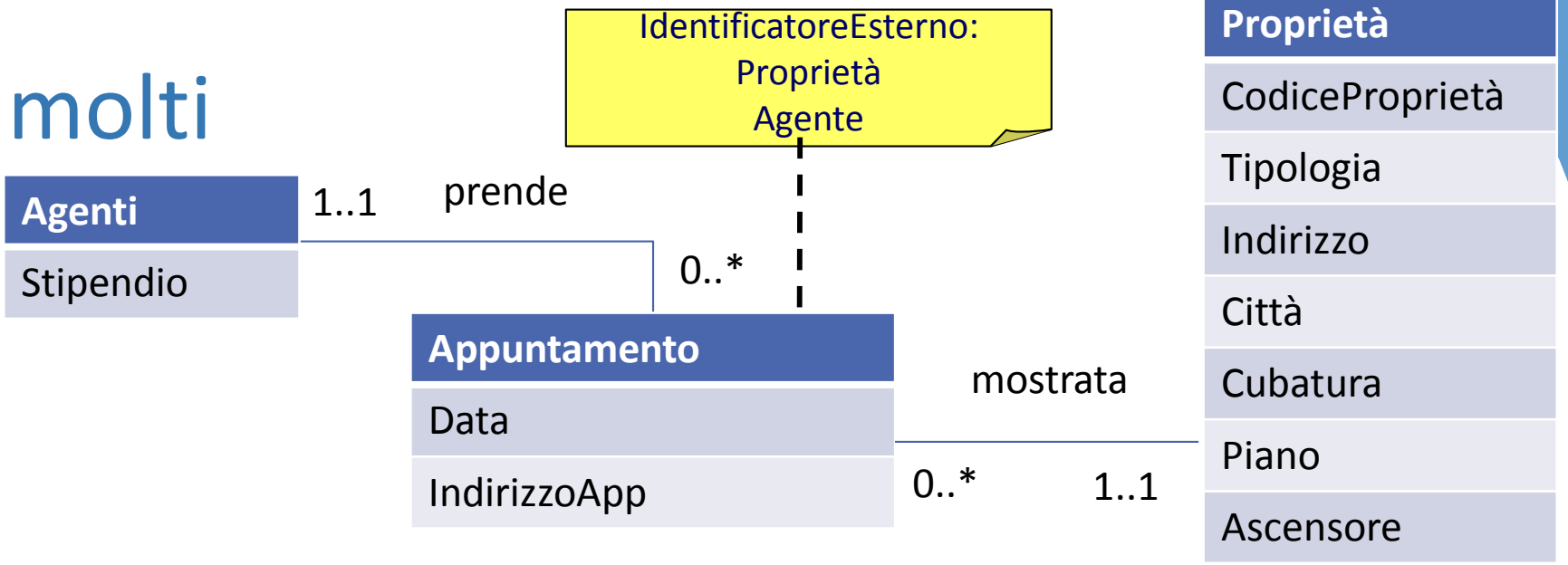
Associazione uno a molti

Venditore	T
CodiceFiscale CHAR(16)	PK
Nome VARCHAR(16)	
Cognome VARCHAR(16)	
DataNascita DATE	
LuogoNascita VARCHAR(16)	

Proprietà	T
codiceProprietà CHAR(16)	PK
Tipologia VARCHAR(16)	
Indirizzo VARCHAR(16)	
Citta VARCHAR(16)	
Cubatura INTEGER	
Piano INTEGER	
Ascensore CHAR(2)	
Proprietario CHAR(16)	FK



Associazione 1 a molti

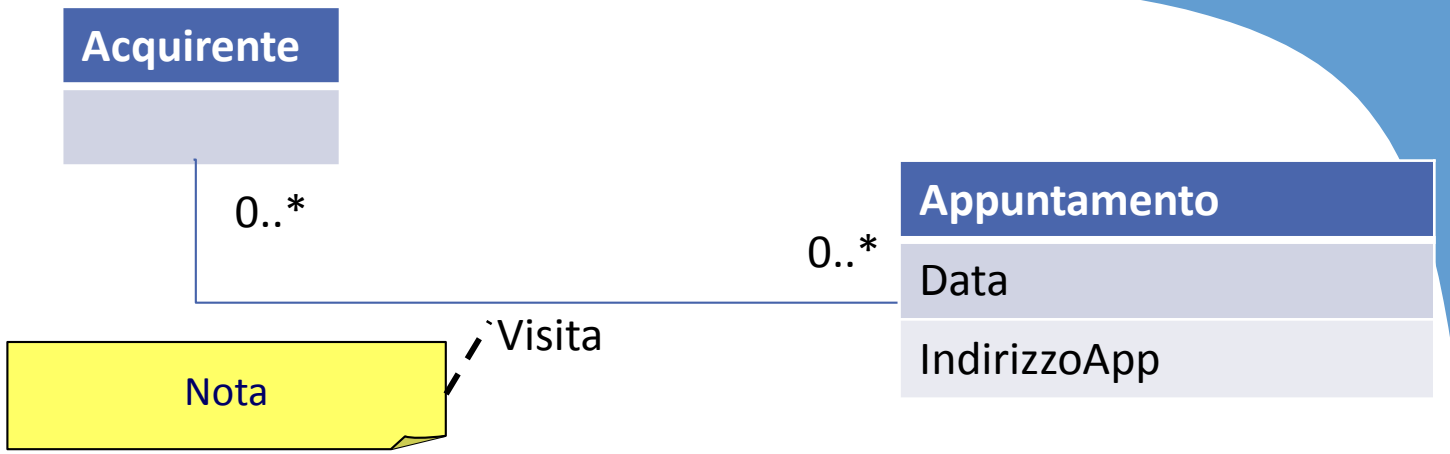


Agenti	T
CodiceFiscale CHAR(16)	PK
Nome VARCHAR(16)	
Cognome VARCHAR(16)	
DataNascita DATE	
LuogoNascita VARCHAR(16)	
Stipendio INTEGER	

Appuntamento	T
Data DATE	PK
IndirizzoApp VARCHAR(16)	
Agenti CHAR(16)	PK, FK
codiceProprietà CHAR(16)	PK, FK

Proprietà	T
codiceProprietà CHAR(16)	PK
Tipologia VARCHAR(16)	
Indirizzo VARCHAR(16)	
Citta VARCHAR(16)	
Cubatura INTEGER	
Piano INTEGER	
Ascensore CHAR(2)	
Proprietario CHAR(16)	FK

Associazione molti a molti

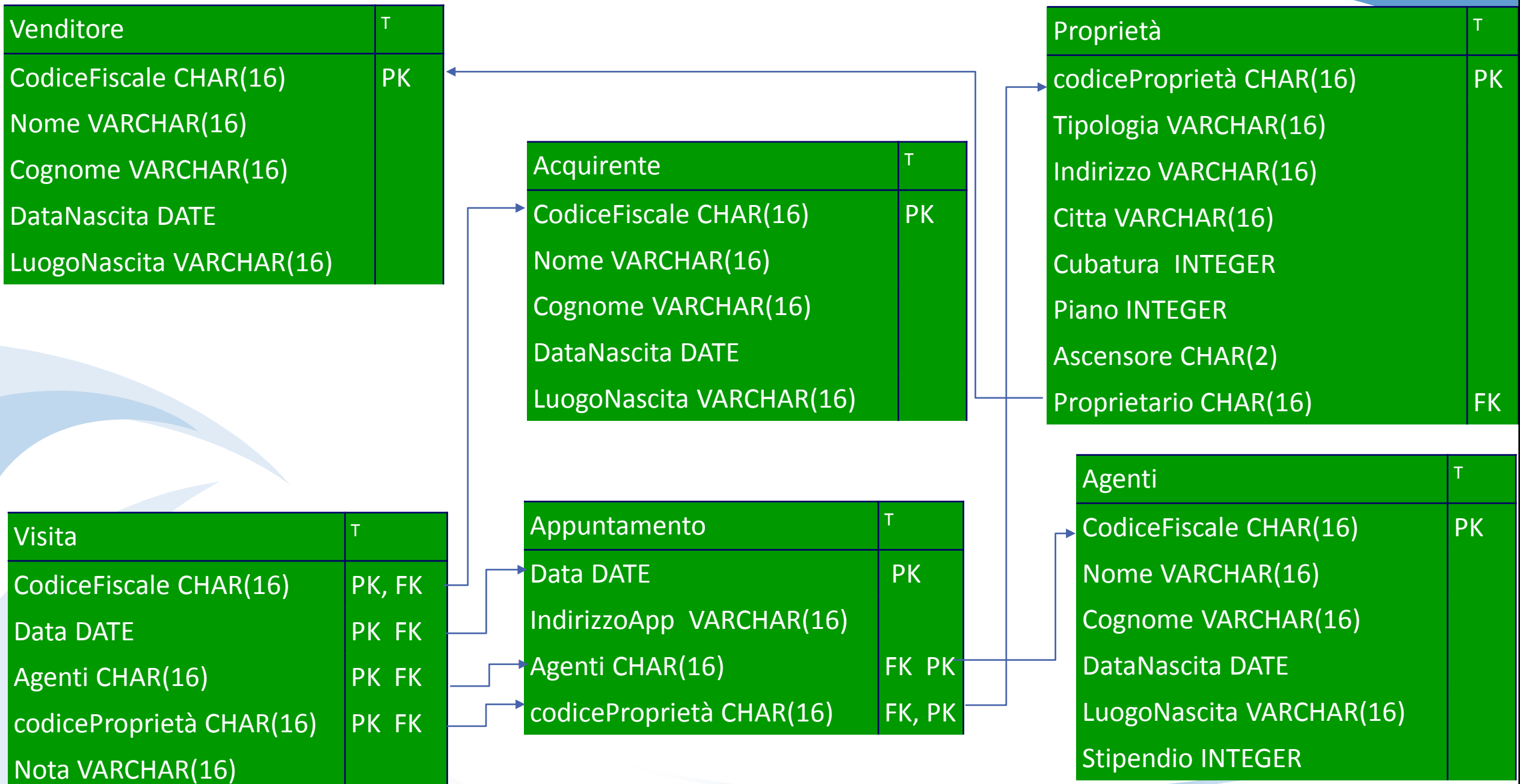


Acquirente	T
CodiceFiscale CHAR(16)	PK
Nome VARCHAR(16)	
Cognome VARCHAR(16)	
DataNascita DATE	
LuogoNascita VARCHAR(16)	

Visita	T
CodiceFiscale CHAR(16)	PK, FK
Data DATE	PK FK
Agenti CHAR(16)	PK FK
codiceProprietà CHAR(16)	PK FK
Nota VARCHAR(16)	

Appuntamento	T
Data DATE	PK
IndirizzoApp VARCHAR(16)	
Agenti CHAR(16)	FK PK
codiceProprietà CHAR(16)	FK, PK

E se ci fosse stato un CodiceAppuntamento???


Venditore
T
CodiceFiscale CHAR(16)
PK
Nome VARCHAR(16)
Cognome VARCHAR(16)
DataNascita DATE
LuogoNascita VARCHAR(16)
Acquirente
T
CodiceFiscale CHAR(16)
PK
Nome VARCHAR(16)
Cognome VARCHAR(16)
DataNascita DATE
LuogoNascita VARCHAR(16)
Proprietà
T
codiceProprietà CHAR(16)
PK
Tipologia VARCHAR(16)
Indirizzo VARCHAR(16)
Citta VARCHAR(16)
Cubatura INTEGER
Piano INTEGER
Ascensore CHAR(2)
Proprietario CHAR(16)
FK
Visita
T
CodiceFiscale CHAR(16)
PK, FK
Data DATE
PK FK
Agenti CHAR(16)
PK FK
codiceProprietà CHAR(16)
PK FK
Nota VARCHAR(16)
Appuntamento
T
Data DATE
PK
IndirizzoApp VARCHAR(16)
Agenti CHAR(16)
FK PK
codiceProprietà CHAR(16)
FK, PK
Agenti
T
CodiceFiscale CHAR(16)
PK
Nome VARCHAR(16)
Cognome VARCHAR(16)
DataNascita DATE
LuogoNascita VARCHAR(16)
Stipendio INTEGER

VII passo: introduzione di eventuali ulteriori vincoli

- NOT NULL
- CHECK
- ….

Esempi:

- Di una proprietà potrebbero essere indicati il piano e la presenza dell'ascensore (che può assumere valore "si" o "no") nel caso in cui tali informazioni siano conosciute.
  - Il vincolo NOT NULL non va su piano e ascensore
- Di ogni appuntamento interessa la data e il punto di incontro, nel caso in cui l'agente e gli acquirenti si incontreranno in un indirizzo diverso da quello in cui è ubicato la proprietà.
  - Il vincolo NOT NULL non va su punto di incontro

CONCETTI AVANZATI SQL

Query 1:

Scrivere una vista che elenca i veicoli coinvolti in più di due sinistri. Di ogni veicolo visualizzare la Targa, la Cilindrata, la ragione sociale dell'assicurazione e l'importo totale dei danni subiti nei vari sinistri in cui tale veicolo è stato coinvolto

TABLE Proprietari
{ *CodiceFiscale* char(16) **PRIMARY KEY,**
Nome varchar(20) **NOT NULL,**
Cognome varchar(20) **NOT NULL,**
*Indirizzo var*char(20) **NOT NULL**,
}

TABLE Assicurazioni
{ *CodiceAssic var*char(20) **PRIMARY KEY,**
RagioneSociale varchar(20) **NOT NULL,**
Citta varchar(20) **NOT NULL**
}

TABLE Sinistro
{ *CodiceSinistro var*char(20) **PRIMARY KEY,**
Luogo varchar(20) **NOT NULL**
Data Date **NOT NULL,**
}

TABLE Veicoli
{*Targa* char(7) **PRIMARY KEY**,
Cilindrata INTEGER **NOT NULL**,
Potenza INTEGER **NOT NULL**,
CodFiscale char(16) **REFERENCES** Proprietari*(CodiceFiscale)*
NOT NULL,
*CodAssic var*char(20) **REFERENCES** Assicurazioni *(CodiceAssic)*
NOT NULL,
}

TABLE VeicoliCoinvolti
{
TargaVeicolo **REFERENCES** *Veicoli(Targa)*,
*CodSinistro var*char(20) **REFERENCES** Sinistro *(CodiceSinistro)*
NOT NULL,
ImportoDanni DOUBLE ***NOT NULL,***
PRIMARY KEY (targaVeicolo, codSinistro)
}

```
Create view vistaVeicoliAssicurati as
Select V.targa, V.cilindrata, A.ragioneSociale, sum(importoDanni)
From Veicoli V JOIN VeicoliCoinvolti VC      ON      V.targa=VC.targaVeicoli JOIN
                Assicurazioni A              ON      V.CodiceAssic = A.CodiceAssic
Group by V.targa, V. cilindrata, A.RagioneSociale
Having count(*) > 2
```

Query 2:

Elenco che riporta il codice dell'assicurazione e la città per cui tutti i veicoli assicurati da tale assicurazione hanno una cilindrata maggiore di 1300.

TABLE Proprietari
{ *CodiceFiscale* char(16) **PRIMARY KEY,**
Nome varchar(20) **NOT NULL,**
Cognome varchar(20) **NOT NULL,**
Indirizzo varchar(20) **NOT NULL**,
}

TABLE Assicurazioni
{ *CodiceAssic* varchar(20) **PRIMARY KEY,**
RagioneSociale varchar(20) **NOT NULL,**
Citta varchar(20) **NOT NULL**
}

TABLE Sinistro
{ *CodiceSinistro* varchar(20) **PRIMARY KEY,**
Luogo varchar(20) **NOT NULL**
Data Date **NOT NULL,**
}

TABLE Veicoli
{*Targa* char(7) **PRIMARY KEY**,
Cilindrata INTEGER **NOT NULL**,
Potenza INTEGER **NOT NULL**,
CodFiscale char(16) **REFERENCES** Proprietari*(CodiceFiscale)*
NOT NULL,
CodAssic varchar(20) **REFERENCES** Assicurazioni *(CodiceAssic)*
NOT NULL,
}

TABLE VeicoliCoinvolti
{
TargaVeicolo **REFERENCES** *Veicoli(Targa)*,
CodSinistro varchar(20) **REFERENCES** Sinistro *(CodiceSinistro)*
NOT NULL,
ImportoDanni DOUBLE ***NOT NULL,***
PRIMARY KEY (targaVeicolo, codSinistro)
}

```
SELECT A.codiceAssic, A.citta
FROM Assicurazioni A
WHERE A.codiceAssic  <> all (
        SELECT codAssic
        FROM Veicoli V JOIN VeicoliCoinvolti VC ON V.targa = VC.TargaVeicolo
        AND V.cilindrata < 1300)
```

Query 3:

Elenco che, per ogni assicurazione della città di Pisa riporta il codice dell'assicurazione e il numero di veicoli assicurati.

TABLE Proprietari
{ *CodiceFiscale* char(16) **PRIMARY KEY,**
Nome varchar(20) **NOT NULL,**
Cognome varchar(20) **NOT NULL,**
Indirizzo varchar(20) **NOT NULL**,
}

TABLE Assicurazioni
{ *CodiceAssic varchar*(20) **PRIMARY KEY,**
RagioneSociale varchar(20) **NOT NULL,**
Citta varchar(20) **NOT NULL**
}

TABLE Sinistro
{ *CodiceSinistro varchar*(20) **PRIMARY KEY,**
Luogo varchar(20) **NOT NULL**
Data Date **NOT NULL,**
}

TABLE Veicoli
{*Targa* char(7) **PRIMARY KEY**,
Cilindrata INTEGER **NOT NULL**,
Potenza INTEGER **NOT NULL**,
CodFiscale char(16) **REFERENCES** Proprietari*(CodiceFiscale)*
NOT NULL,
CodAssic varchar(20) **REFERENCES** Assicurazioni *(CodiceAssic)*
NOT NULL,
}

TABLE VeicoliCoinvolti
{
TargaVeicolo **REFERENCES** *Veicoli(Targa)*,
CodSinistro varchar(20) **REFERENCES** Sinistro *(CodiceSinistro)*
NOT NULL,
ImportoDanni DOUBLE ***NOT NULL,***
PRIMARY KEY (targaVeicolo, codSinistro)
}

```
Select A.codAssic, count(*)
From Veicoli V JOIN Assicurazioni A      ON      V.CodiceAssic = A.CodiceAssic
Where V.citta = 'Pisa'
Group by codAssic
```