

BASE DI DATI

Esercizi

- Progettazione concettuale
- Progettazione logica

Informatica Umanistica
Università di Pisa

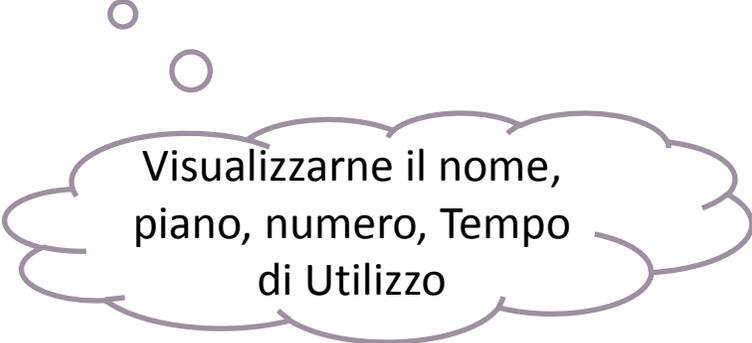
Esercizio: Parcheggi

Si consideri la seguente realtà riguardante una società di gestione dei parcheggi:

- La società gestisce vari parcheggi. Ogni parcheggio è caratterizzato da un nome, da una tipologia (es.: Multipiano, Sotterraneo, all'aperto, Custoditi, etc.), e da un indirizzo.
- Ogni parcheggio contiene vari posti auto. Ogni posto auto ha una posizione (ad esempio il piano ed il numero), può essere nello stato occupato, libero, o indisponibile. A seconda che sia occupato da una automobile, sia libero e disponibile, o indisponibile (ad esempio per motivi tecnici). Per ogni posto auto si vuole anche memorizzare il tempo di utilizzo, ossia il tempo globale in cui il posto auto si è trovato in stato occupato.
- I cittadini possono comprare l'abbonamento ad un parcheggio. Ogni cittadino può comprare abbonamenti diversi per parcheggi diversi. Di ogni cittadino si vuole memorizzare il Nome il Cognome, il codice fiscale, l'indirizzo. L'abbonamento ha un inizio ed una fine.

Interrogazioni

- Si risolvano le seguenti query facendo riferimento allo schema logico
 - Trovare i parcheggi a cui è abbonato Mario Rossi, ed il numero totale di posti auto del corrispondente parcheggio.
 - Trovare per ogni parcheggio, il posto auto meno utilizzato.
 - Trovare i parcheggio con più di 40 posti disponibili.



Visualizzarne il nome,
piano, numero, Tempo
di Utilizzo

Esercizio: Parcheggi (Identificazione classi)

Si consideri la seguente realtà riguardante una società di gestione dei parcheggi:

- La società gestisce vari **parcheggi**. Ogni parcheggio è caratterizzato da un nome, da una tipologia (es.: Multipiano, Sotterraneo, all'aperto, Custoditi, etc.), e da un indirizzo.
- Ogni parcheggio contiene vari **posti auto**. Ogni posto auto ha una posizione (ad esempio il piano ed il numero), può essere nello stato occupato, libero, o indisponibile. A seconda che sia occupato da una automobile, sia libero e disponibile, o indisponibile (ad esempio per motivi tecnici). Per ogni posto auto si vuole anche memorizzare il tempo di utilizzo, ossia il tempo globale in cui il posto auto si è trovato in stato occupato.
- I **cittadini** possono comprare l'abbonamento ad un parcheggio. Ogni cittadino può comprare abbonamenti diversi per parcheggi diversi. Di ogni cittadino si vuole memorizzare il Nome il Cognome, il codice fiscale, l'indirizzo. L'abbonamento ha un inizio ed una fine.

Esercizio: Parcheggi (Identificazione attributi)

Si consideri la seguente realtà riguardante una società di gestione dei parcheggi:

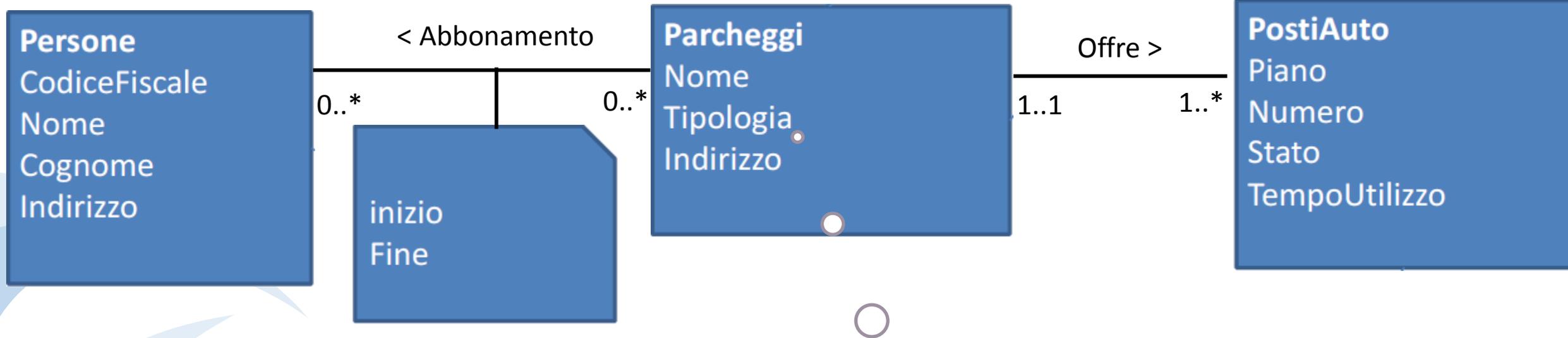
- La società gestisce vari **parcheggi**. Ogni parcheggio è caratterizzato da un **nome**, da una **tipologia** (es.: Multipiano, Sotterraneo, all'aperto, Custoditi, etc.), e da un **indirizzo**.
- Ogni parcheggio contiene vari **posti auto**. Ogni posto auto ha una posizione (ad esempio il **piano** ed il **numero**), può essere nello **stato** occupato, libero, o indisponibile. A seconda che sia occupato da una automobile, sia libero e disponibile, o indisponibile (ad esempio per motivi tecnici). Per ogni posto auto si vuole anche memorizzare il **tempo di utilizzo**, ossia il tempo globale in cui il posto auto si è trovato in stato occupato.
- I **cittadini** possono comprare l'abbonamento ad un parcheggio. Ogni cittadino può comprare abbonamenti diversi per parcheggi diversi. Di ogni cittadino si vuole memorizzare il **Nome**, il **Cognome**, il **codice fiscale**, l'**indirizzo**. L'abbonamento ha un **inizio** ed una **fine**.

Esercizio: Parcheggi (Identificazione associazioni)

Si consideri la seguente realtà riguardante una società di gestione dei parcheggi:

- La società gestisce vari **parcheggi**. Ogni parcheggio è caratterizzato da un **nome**, da una **tipologia** (es.: Multipiano, Sotterraneo, all'aperto, Custoditi, etc.), e da un **indirizzo**.
- Ogni parcheggio contiene vari **posti auto**. Ogni posto auto ha una posizione (ad esempio il **piano** ed il **numero**), può essere nello **stato** occupato, libero, o indisponibile. A seconda che sia occupato da una automobile, sia libero e disponibile, o indisponibile (ad esempio per motivi tecnici). Per ogni posto auto si vuole anche memorizzare il **tempo di utilizzo**, ossia il tempo globale in cui il posto auto si è trovato in stato occupato.
- I **cittadini** possono comprare l'abbonamento ad un parcheggio. Ogni cittadino può comprare abbonamenti diversi per parcheggi diversi. Di ogni cittadino si vuole memorizzare il **Nome**, il **Cognome**, il **codice fiscale**, l'**indirizzo**. L'abbonamento ha un **inizio** ed una **fine**.
- Quindi:
 - Persone – Parcheggi (abbonamento con attributi inizio e fine)
 - Parcheggi – Posti (offre)

Schema concettuale (senza generalizzazioni)



Progettazione logica

- I passo: traduzione iniziale delle classi non coinvolte in gerarchie
- II passo: traduzione iniziale delle gerarchie
- III passo: traduzione degli attributi multivalore
- IV passo: traduzione delle associazioni molti a molti
- V passo: traduzione delle associazioni uno a molti
- VI passo: traduzione delle associazioni uno a uno
- VII passo: introduzione di eventuali ulteriori vincoli
- VIII passo: progettazione degli schemi esterni

I passo: traduzione iniziale delle classi non coinvolte in gerarchie

Persone
CodiceFiscale
Nome
Cognome
Indirizzo

Parcheggi
Nome
Tipologia
Indirizzo

PostiAuto
Piano
Numero
Stato
TempoUtilizzo

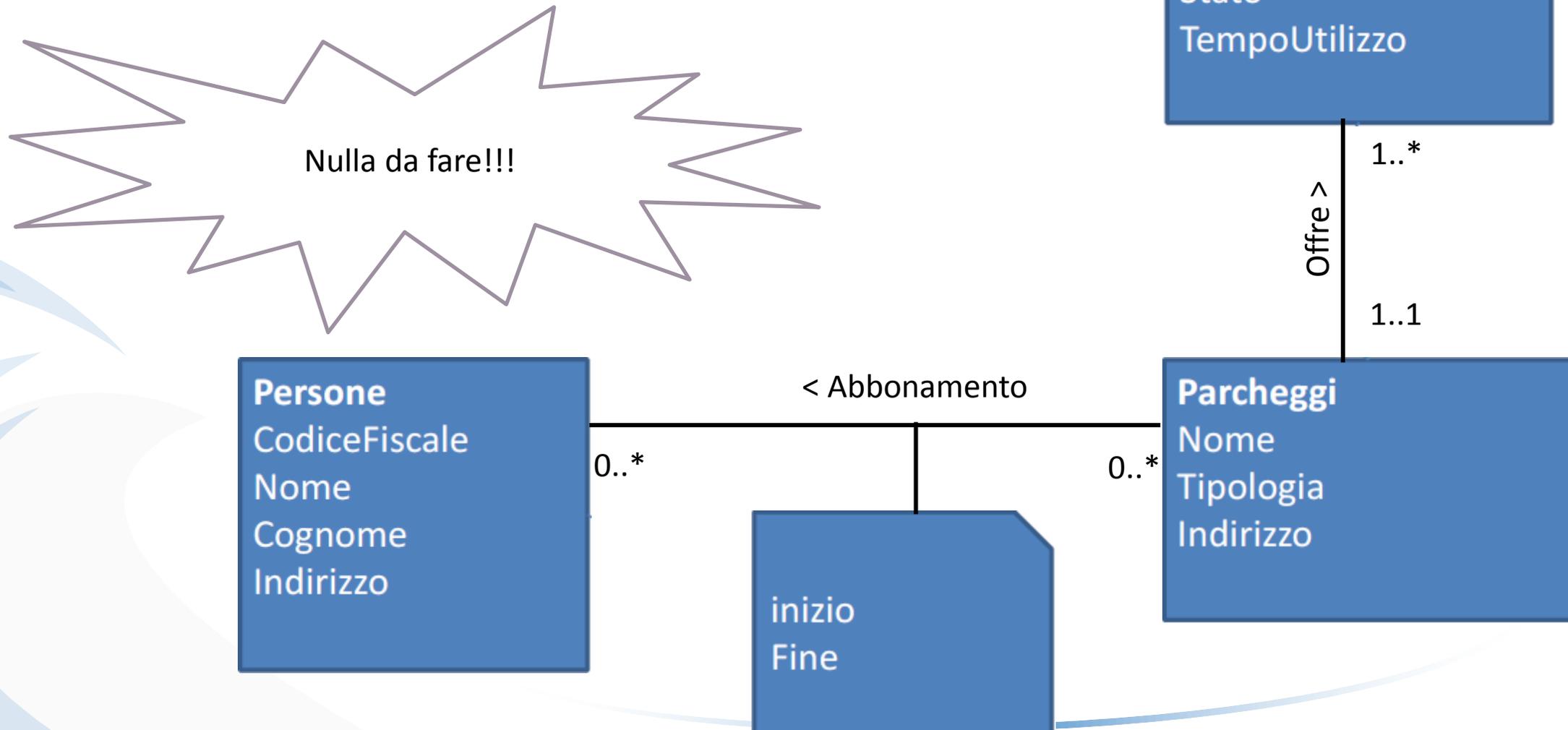
Persone	T
codice CHAR(16)	PK
Nome CHAR(15)	
Cognome CHAR(15)	
Indirizzo CHAR(45)	

Parcheggi	T
idParcheggio INTEGER	PK
Nome CHAR(10)	
Tipologia CHAR(45)	
Indirizzo CHAR(45)	

PostiAuto	T
codice CHAR(16)	PK
Piano INTEGER	
Numero INTEGER	
Stato CHAR(10)	
TempoUtilizzo INTEGER	

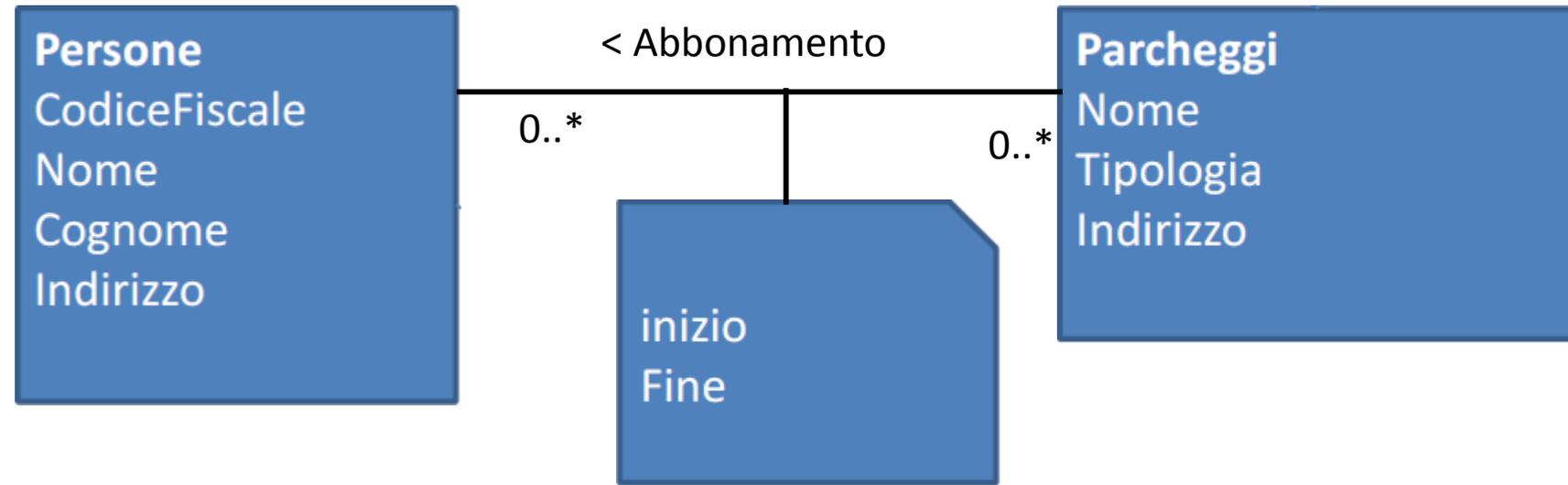
Il passo: traduzione iniziale delle gerarchie

III passo: traduzione degli attributi multivalore



IV passo: traduzione delle associazioni molti a molti

- Ogni associazione molti a molti **genera una tabella**
 - riferimenti (chiavi esterne) alle tabelle che traducono le classi coinvolte
 - eventuali attributi dell'associazione
 - la chiave della tabella deve includere le chiavi esterne



Persone	T
codice CHAR(16)	PK
Nome CHAR(15)	
Cognome CHAR(15)	
Indirizzo CHAR(45)	

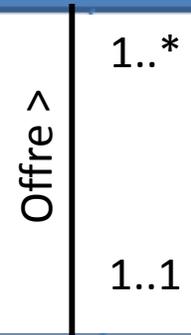
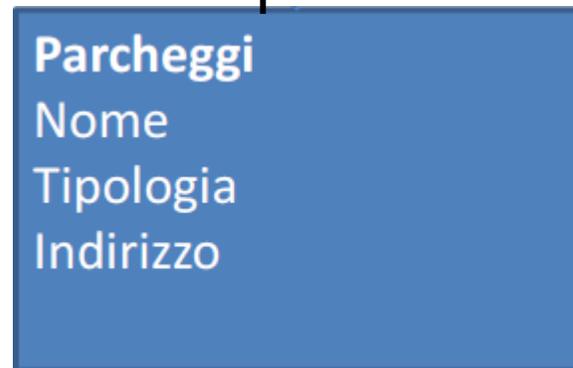
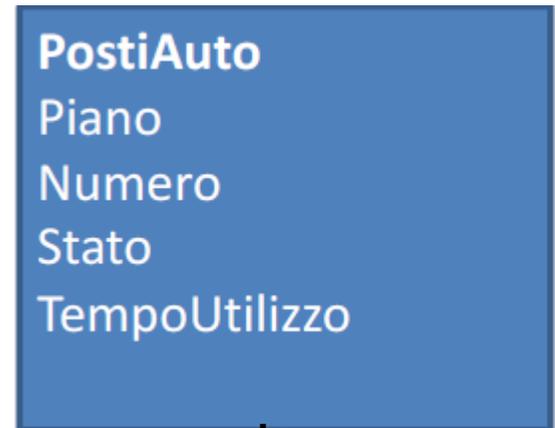
PersonaAbbonataAParcheggio	T
codiceFiscale CHAR(16)	PK FK
idParcheggio INTEGER	PK FK
Inizio DATE	
Fine DATE	

Parcheggi	T
idParcheggio INTEGER	PK
Nome CHAR(10)	
Tipologia CHAR(45)	
Indirizzo CHAR(45)	

V passo: traduzione delle associazioni uno a molti

- **Generano chiavi esterne**

- ciascuna istanza dell'associazione è identificata dall'oggetto dal lato 1
- chiave esterna della tabella dal lato 1 nella tabella corrispondente alla classe dal lato m

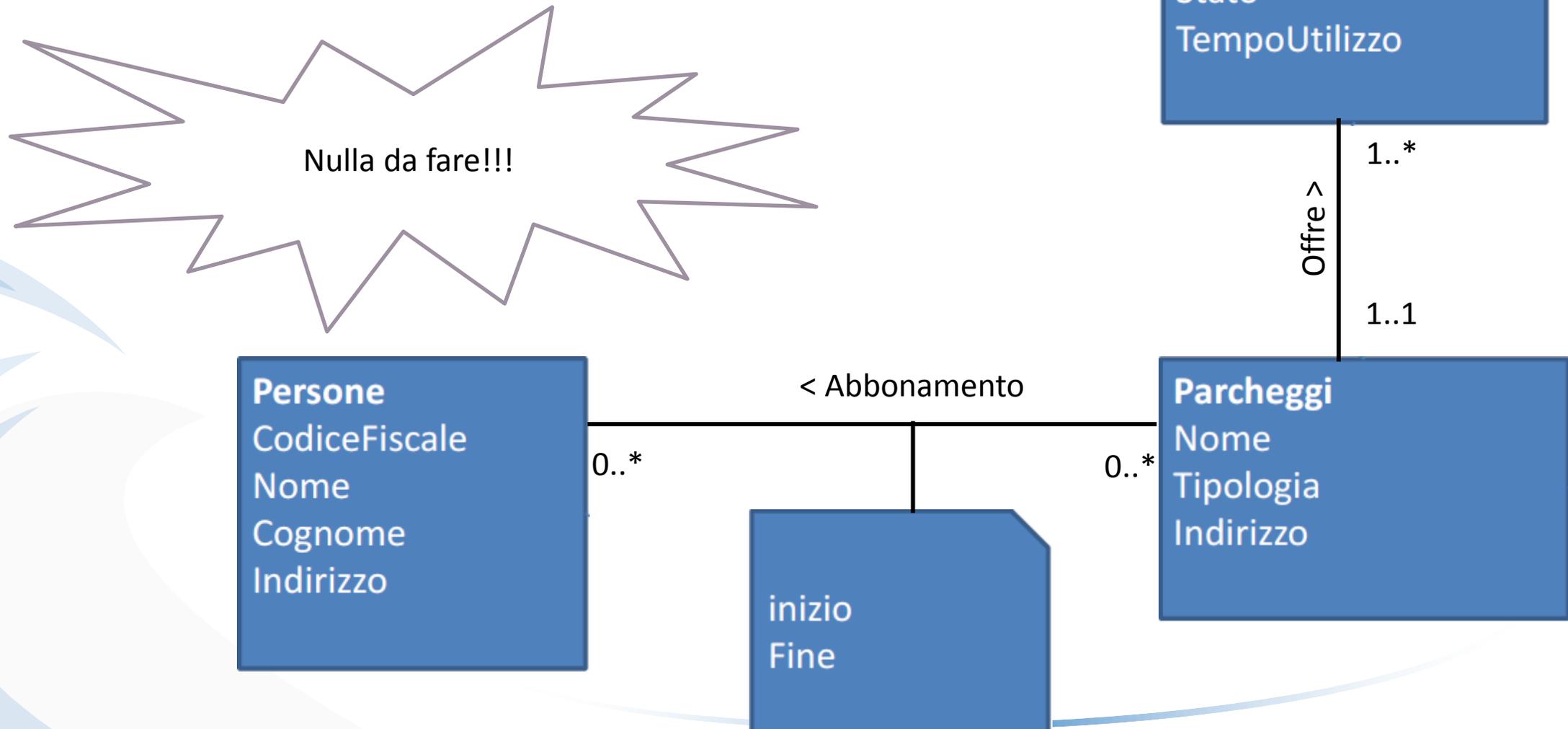


Parcheggi	T
idParcheggio INTEGER	PK
Nome VARCHAR (10)	
Tipologia VARCHAR(45)	
Indirizzo VARCHAR(45)	

PostiAuto	T
codice INT	PK
Piano INTEGER	
Numero INTEGER	
Stato CHAR(10)	
TempoUtilizzo INTEGER	
idParcheggio INTEGER	FK



VI passo: traduzione delle associazioni uno a uno



VII passo: introduzione di eventuali ulteriori vincoli

Vincoli NOT NULL, CHECK ecc.

Check su data inizio/Fine

PostiAuto	T
codice INT	PK
Piano INTEGER NOT NULL	
Numero INTEGER NOT NULL	
Stato CHAR(10) NOT NULL	
TempoUtilizzo INTEGER NOT NULL	
idParcheggio INTEGER	FK

Persone	T
codice CHAR(16)	PK
Nome VARCHAR(15) NOT NULL	
Cognome VARCHAR(15) NOT NULL	
Indirizzo VARCHAR(45)	

PersonaAbbonataAParcheggio	T
codiceFiscale CHAR(16)	PK FK
idParcheggio INTEGER	PK FK
Inizio DATE NOT NULL	
Fine DATE NOT NULL	

Parcheggi	T
idParcheggio INTEGER	PK
Nome VARCHAR(10) NOT NULL	
Tipologia VARCHAR(15)	
Indirizzo VARCHAR(45)	

Tabelle in SQL: persone - parcheggi

```
CREATE TABLE persone (  
    CodiceFiscale CHAR(16) PRIMARY KEY ,  
    Nome VARCHAR(15) NOT NULL ,  
    Cognome VARCHAR(15) NOT NULL ,  
    Indirizzo VARCHAR(45) ,  
)
```

```
CREATE TABLE Parcheggi (  
    idParcheggio INT PRIMARY KEY,  
    Nome VARCHAR(10) NOT NULL ,  
    Tipologia VARCHAR(15),  
    Indirizzo VARCHAR(45),  
)
```

Persone	T
codice CHAR(16)	PK
Nome VARCHAR(15) NOT NULL	
Cognome VARCHAR(15) NOT NULL	
Indirizzo VARCHAR(45)	

Parcheggi	T
idParcheggio INTEGER	PK
Nome VARCHAR(10) NOT NULL	
Tipologia VARCHAR(15)	
Indirizzo VARCHAR(45)	

Tabelle in SQL: PersonaAbbonataAParcheggio

Persone	T
codice CHAR(16)	PK
Nome VARCHAR(15) NOT NULL	
Cognome VARCHAR(15) NOT NULL	
Indirizzo VARCHAR(45)	

PersonaAbbonataAParcheggio	T
codiceFiscale CHAR(16)	PK FK
idParcheggio INTEGER	PK FK
Inizio DATE NOT NULL	
Fine DATE NOT NULL	

Parcheggi	T
idParcheggio INTEGER	PK
Nome VARCHAR(10) NOT NULL	
Tipologia VARCHAR(15)	
Indirizzo VARCHAR(45)	

```

CREATE TABLE PersonaAbbonataAParcheggio (
    CodiceFiscale CHAR(16) NOT NULL ,
    idParcheggio INT NOT NULL ,
    Inizio DATE NOT NULL ,
    Fine DATE NOT NULL ,
    PRIMARY KEY (CodiceFiscale, idParcheggio) ,
    FOREIGN KEY (CodiceFiscale ) REFERENCES persone (CodiceFiscale )
    FOREIGN KEY (idParcheggio ) REFERENCES Parcheggi (idParcheggio )
)
    
```

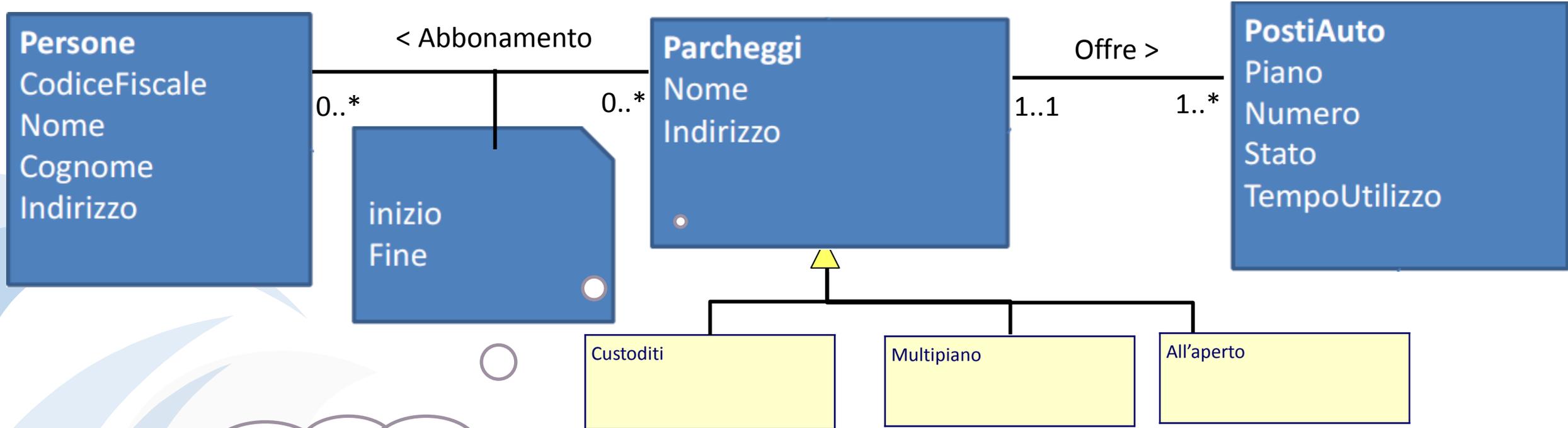
Tabelle in SQL: Posti auto

```
CREATE TABLE PostiAuto (  
  codice INT PRIMARY KEY,  
  Piano INT NOT NULL ,  
  Numero INT NOT NULL ,  
  Stato VARCHAR(10) NOT NULL ,  
  TempoUtilizzo INT NOT NULL ,  
  idParcheggio INT NOT NULL ,  
  FOREIGN KEY (idParcheggio ) REFERENCES Parcheggi (idParcheggio )  
)
```

PostiAuto	T
codice INT	PK
Piano INTEGER NOT NULL	
Numero INTEGER NOT NULL	
Stato CHAR(10) NOT NULL	
TempoUtilizzo INTEGER NOT NULL	
idParcheggio INTEGER	FK

Parcheggi	T
idParcheggio INTEGER	PK
Nome VARCHAR(10) NOT NULL	
Tipologia VARCHAR(15)	
Indirizzo VARCHAR(45)	

Schema concettuale (con generalizzazione)



Si noti che non appare la tipologia di parcheggio

I passo: traduzione iniziale delle classi non coinvolte in gerarchie

Persone
CodiceFiscale
Nome
Cognome
Indirizzo

Parcheggi
Nome
Indirizzo

PostiAuto
Piano
Numero
Stato
TempoUtilizzo

Persone	T
codice CHAR(16)	PK
Nome CHAR(15)	
Cognome CHAR(15)	
Indirizzo CHAR(45)	

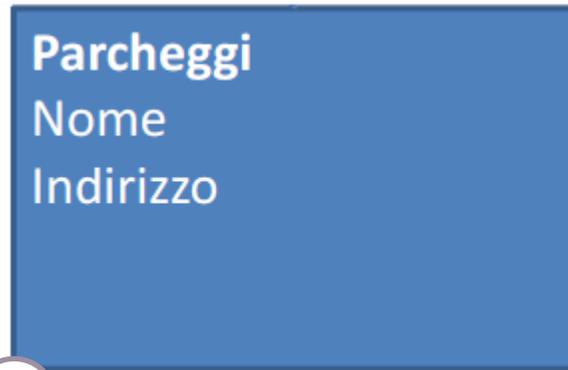
Parcheggi	T
idParcheggio INTEGER	PK
Nome CHAR(10)	
Indirizzo CHAR(45)	

PostiAuto	T
codice CHAR(16)	PK
Piano INTEGER	
Numero INTEGER	
Stato CHAR(10)	
TempoUtilizzo INTEGER	

Il Passo: Traduzione delle Gerarchie

- **I Soluzione: Solo il padre**

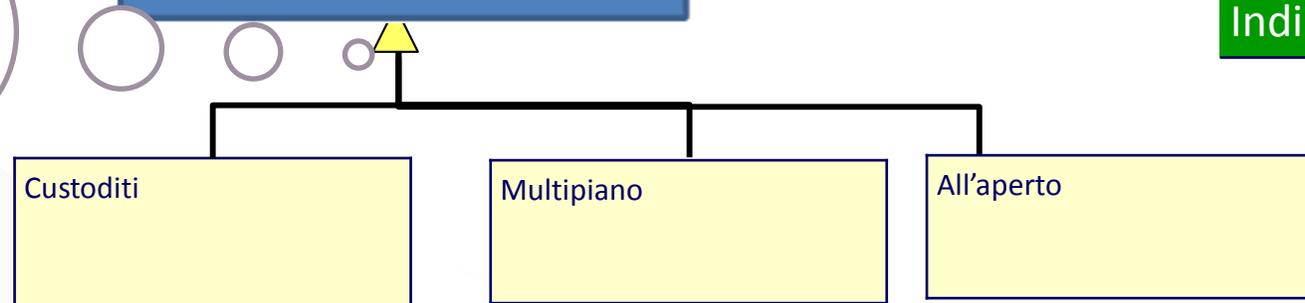
- un'unica tabella con il nome del padre
- la tabella deve avere tutti gli attributi di padre e figli
- serve un ulteriore attributo (es: tipo) per distinguere le istanze dei figli
- conveniente se le operazioni sui figli non sono particolarmente rilevanti nell'applicazioni
- genera valori nulli



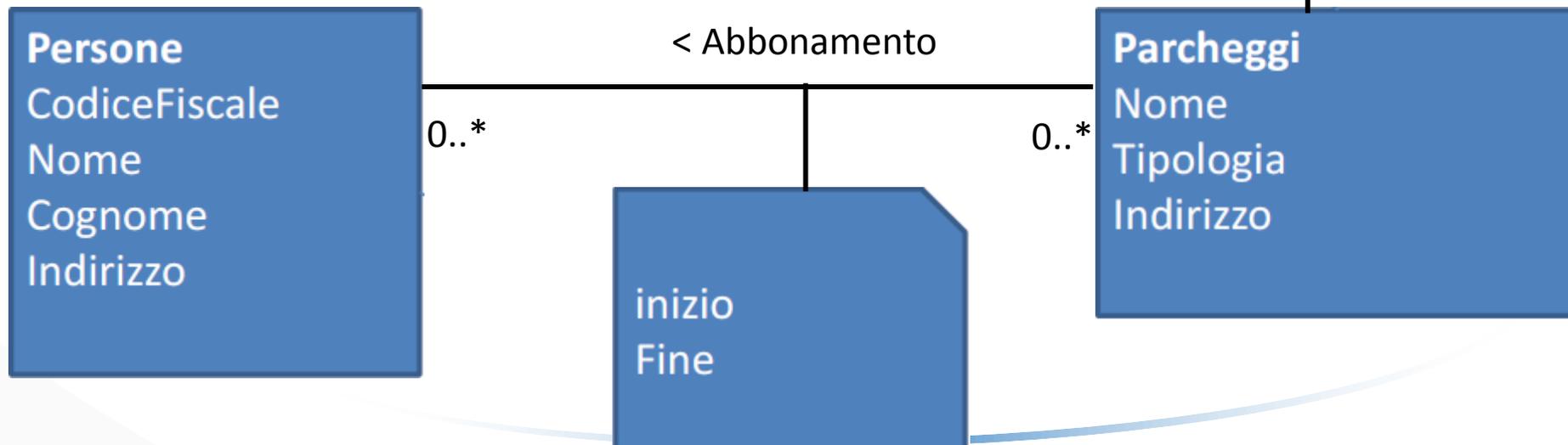
Parcheggi	T
idParcheggio INTEGER	PK
Nome CHAR(10)	
Tipologia CHAR(45)	
Indirizzo CHAR(45)	

Si noti che:

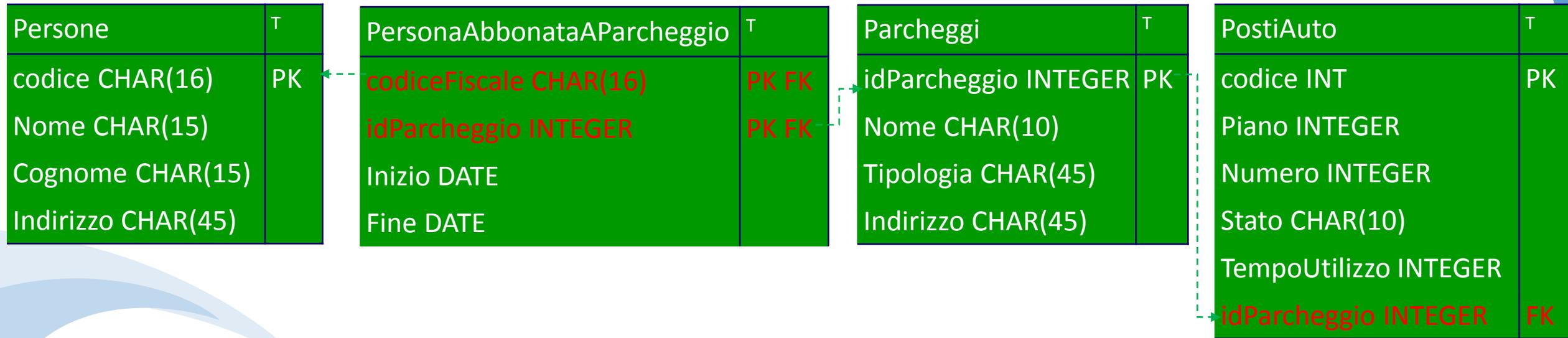
- Non ci sono attributi nei figli
- E' completa
- E' disgiunta



III passo: traduzione degli attributi multivalore



Query in SQL



- Si risolvano le seguenti query facendo riferimento allo schema logico
 - Trovare i parcheggi a cui è abbonato Mario Rossi, ed il numero totale di posti auto del corrispondente parcheggio.
 - Trovare per ogni parcheggio, il posto auto meno utilizzato.
 - Trovare i parcheggio con più di 40 posti disponibili.

Visualizzarne il nome, piano, numero, Tempo di Utilizzo