

# Progettazione di database relazionali

# Progettazione concettuale

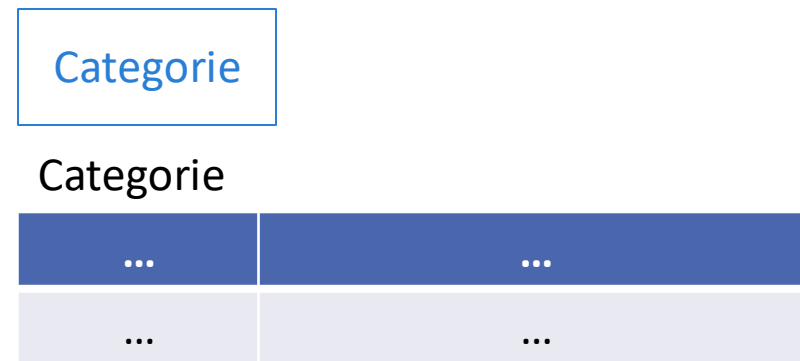
- L'obiettivo ultimo è la progettazione dello schema logico relativo al DBMS scelto.
- Progettare direttamente a livello logico non è la cosa più efficace
  - In fase di progettazione relazioni e attributi non sono ancora noti
  - Da dove parto?
- Serve un metodo incrementale per creare un modello concettuale dal quale ricavare il modello logico.
  - Permette di concentrarsi prima sugli aspetti più importanti per poi scendere nei dettagli
- Sono state create numerose metodologie di progetto concettuale.
- Vedremo gli aspetti più importanti della metodologia Entity - Relationship

# Modello Entity - Relationship

- Facilita la progettazione dello schema logico.
- E' indipendente dal DBMS
- Ha una rappresentazione grafica che ne facilita l'utilizzo e la comprensione
- Permette di procedere per passi successivi, partendo dal massimo livello di astrazione
- Permette di passare facilmente allo schema logico relazionale
- Si basa su alcuni concetti di base:
  - Entità
  - Associazione (Relationship)
  - Attributo
  - Vincoli di cardinalità

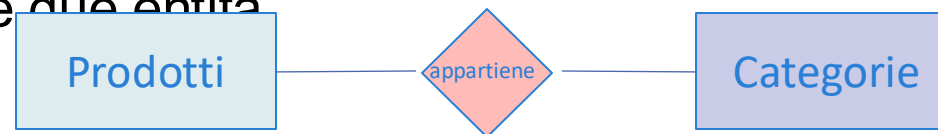
# Entità

- E' definita come un insieme di oggetti della realtà da modellare.
- Graficamente si rappresenta con un rettangolo
- Uno specifico oggetto di una entità è detto istanza
- Possiamo, in prima approssimazione, considerarle come delle tabelle (relazioni) delle quali non conosciamo ancora gli attributi.



# Associazioni

- Rappresentano un legame logico tra due entità. Il legame deve essere rilevante nella realtà da modellare.
- Graficamente si rappresentano con un rombo
- L'associazione crea una istanza di associazione tra le istanze delle entità che prendono parte all'associazione
- Possiamo, in prima approssimazione, considerarle come delle tabelle (relazioni) delle quali non conosciamo ancora gli attributi. Resta però il fatto che devono mettere in relazione oggetti delle due entità

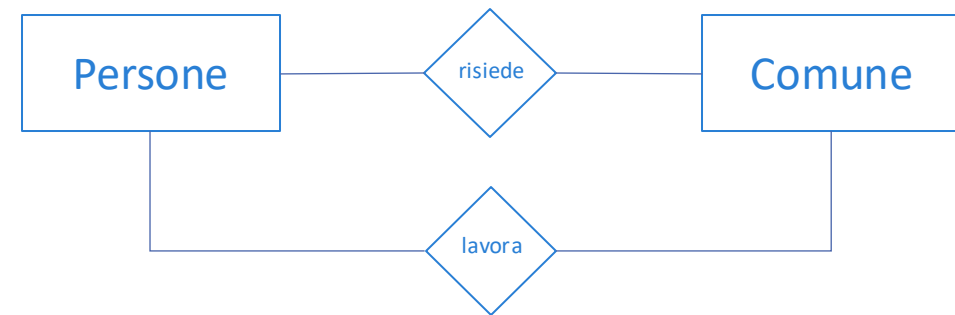
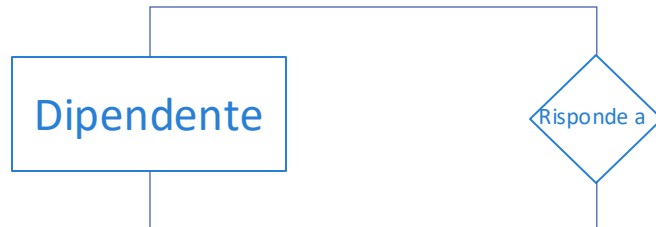


Appartiene

...	...	...	...
...	...	...	...

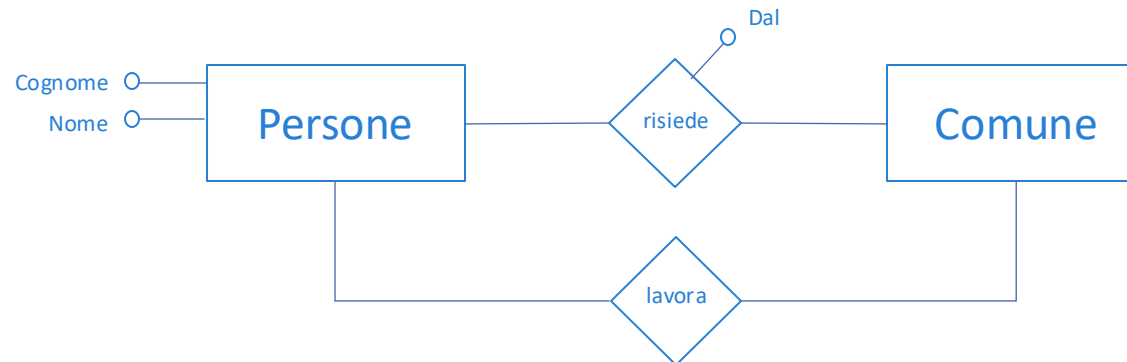
# Associazioni

- Si dice grado dell'associazione il numero di entità coinvolte.
- Possono esistere più associazioni tra le stesse entità
- Possono esistere associazioni che coinvolgono più volte una entità (associazioni ad anello)



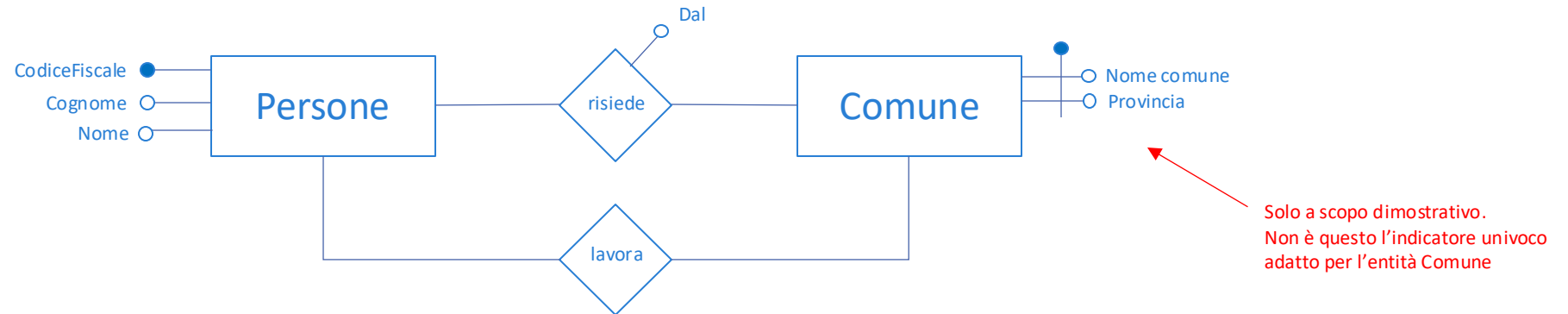
# Attributi

- Rappresentano le proprietà elementari delle entità e delle associazioni.
- Graficamente di rappresentano con un connettore
- Ogni attributo è definito su un dominio
- Possiamo considerare gli attributi come gli attributi della tabella associata all'entità o alla associazione
- **ATTENZIONE** a dove si specificano gli attributi



# Indicatori univoci

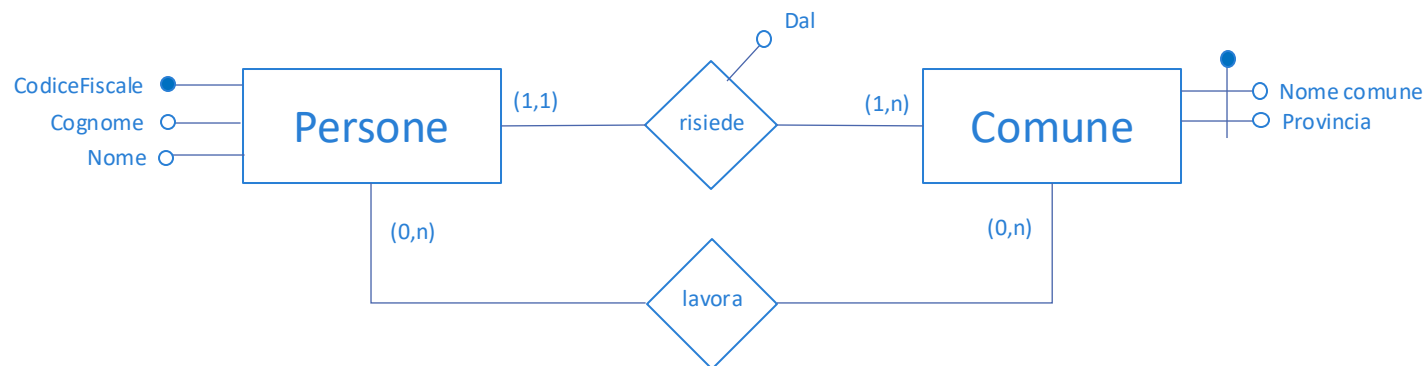
- Gli attributi possono rappresentare indicatori univoci (chiavi).
- Possono esistere indicatori univoci formati da un solo attributo o da più attributi





# Cardinalità delle associazioni

- Sono dei vincoli associati ad ogni entità che partecipa in una associazione
- Sono coppie di valori che rappresentano, rispettivamente, il numero minimo e il numero massimo di istanze dell'associazione a cui una istanza dell'entità può partecipare.
  - (1,1) Cardinalità uno-a-uno obbligatoria
  - (0,1) Cardinalità uno-a-uno opzionale
  - (1,n) Cardinalità uno-a-molti obbligatoria
  - (0,n) Cardinalità uno-a-molti opzionale
  - (m,n) Cardinalità molti-a-molti



# Dal modello E/R allo schema logico

Il passaggio dal modello E/R allo schema logico relazionale è piuttosto semplice:

- Ogni entità viene convertita in una relazione (tabella), nella quale:
  - Il nome della relazione è dato dal nome dell'entità
  - Gli attributi della relazione sono dati dagli attributi dell'entità
  - Le chiavi della relazione sono date dalle chiavi dell'entità



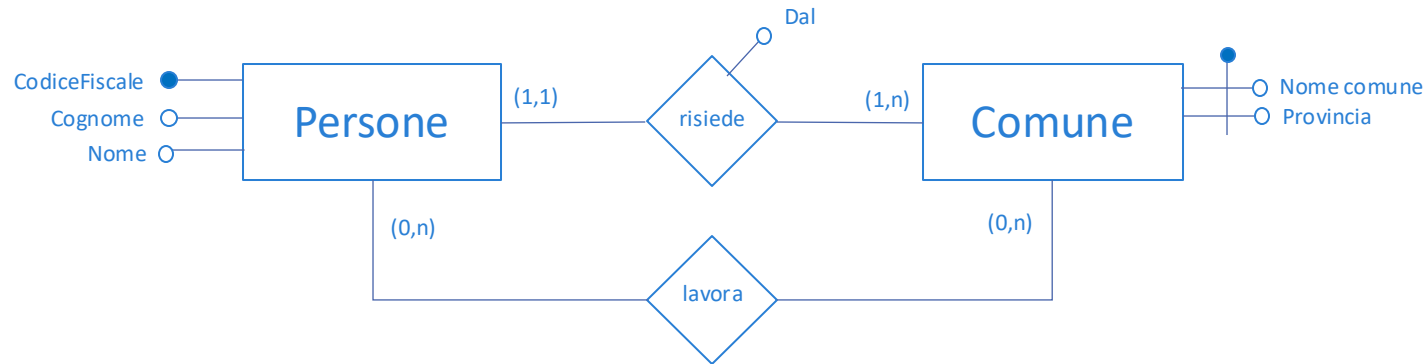
Persone

<u>CodiceFiscale</u>	Cognome	Nome
...	...	...

# Dal modello E/R allo schema logico

Il passaggio dal modello E/R allo schema logico relazionale è piuttosto semplice:

- Ogni associazione viene convertita in una relazione (tabella), nella quale:
  - Il nome della relazione è dato dal nome dell'associazione
  - Gli attributi della relazione sono dati dagli attributi dell'associazione
  - Le chiavi della relazione dipendono dalla cardinalità dell'associazione.



Risiede

<u>CodiceFiscale</u>	Dal	NomeComune	Provincia
...	...	...	

Lavora

<u>CodiceFiscale</u>	<u>NomeComune</u>	<u>Provincia</u>
...	...	

# Esercizio: Progettazione di database relazionali

1. Modellare lo schema E/R di base per un gestionale per una azienda che produce abiti.  
Il database deve permettere di gestire:
  - Gli ordini che l'azienda riceve dai rivenditori, caratterizzati da dati del cliente, articoli ordinati e stato di avanzamento
  - Gli abiti, disponibili in diverse taglie e caratterizzati da una categoria
  - La produzione degli articoli ordinati, considerando articoli realizzati solo con l'utilizzo di filati
2. Realizzare lo schema logico relativo allo schema E/R appena creato
3. Procedere con la normalizzazione in, almeno, seconda forma normale

# Le forme normali

# Le forme normali

La realizzazione di uno schema logico relazionale può dare origine a relazioni (tabelle) che presentano problemi (ridondanza dei dati e anomalie di aggiornamento/cancellazione):

Persone

<u>CodiceFiscale</u>	Cognome	Nome	DataNascita	ComuneNascita	ProvNascita
...	...	...			

- Ridondanza: I dati dei comuni sono riportati più volte.
- Anomalia di aggiornamento: se un comune cambia provincia devo aggiornare tutti i record
- Anomalia di cancellazione: se cancello tutte le persone di un comune, perdo anche le informazioni del comune stesso

# Le forme normali

- Una forma normale è una proprietà dello schema relazionale che ne garantisce l'assenza di alcuni difetti. ➔ uno schema è in forma normale.
- L'attività di trasformazione dello schema in forma normale è detta normalizzazione e tipicamente consiste in un intervento a livello di tabelle e/o colonne
- La normalizzazione è una tecnica di verifica e non di progettazione
- Esistono molteplici forme normali, ognuna in grado di garantire un determinato livello di «qualità»
- Le forme normali si basano su concetti abbastanza articolati, cercheremo di semplificare. Vedremo:
  - Prima forma normale (1FN)
  - Seconda forma normale (2FN)
  - Terza forma normale (3FN)
  - Forma normale di Boyce-Codd (BCNF)

# Prima forma normale (1NF)

- Una relazione è in 1NF se e solo se ogni attributo di ogni tupla ha un valore atomico, non ulteriormente decomponibile.
- Non solo pertanto ammesse liste, array, relazioni.
- Eventuali attributi non atomici vanno decomposti in una serie di attributi atomici.

Colori

<u>Nome</u>	RGB
Rosso	[255,0,0]
Verde	[0,255,0]
Blu	[0,0,255]



Colori

<u>Nome</u>	RGB_red	RGB_green	RGB_blue
Rosso	255	0	0
Verde	0	255	0
Blu	0	0	255



## Seconda forma normale (2NF)

- Una relazione è in 2NF se e solo se ogni attributo con non partecipa alla chiave (attributo non primo) dipende completamente dalla chiave primaria.
- Magazzini(Articolo, Magazzino, Quantità, Indirizzo) non è in 2FN in quanto l'attributo Indirizzo dipende solo da una parte della chiave primaria
- La normalizzazione in 2FN consiste nel creare una nuova relazione per rappresentare la dipendenza:
  - ArticoliMagazzini(Articolo, Magazzino, Quantità)
  - IndirizziMagazzini(Magazzino, Indirizzo)
- Uno schema in 1NF nel quale tutte le chiavi sono composte da un solo attributo è automaticamente anche in 2NF

## Terza forma normale (3NF)

- Una relazione è in 3NF se e solo se ogni attributo con non partecipa alla chiave (attributo non primo) non ha dipendenze con altri attributi non primi. Deve cioè dipendere solo dalla chiave primaria.
- Impiegati(Codice, Nome, Ufficio, CapoUfficio) è in 2FN ma non in 3FN in quanto l'attributo CapoUfficio dipende dall'attributo non primo Ufficio.
- La normalizzazione in 3FN consiste nel creare una nuova relazione per rappresentare la dipendenza:
  - Impiegati(Codice, Nome, Ufficio)
  - Uffici(Ufficio, CapoUfficio)
- Uno schema in 3NF è anche in 2FN

# Forma normale di Boyce-Codd (BCNF)

- Una relazione è in BCNF se e solo se non esistono campi che dipendono da attributi non primi.
- Insegnamenti(Studente, Corso, Docente) è in 3FN ma non in BCFN in quanto esiste la dipendenza  $\text{Corso} \rightarrow \text{Docente}$  nel quale Docente non è primo.
- Anche a normalizzazione in BCFN consiste nel creare una nuova relazione per rappresentare la dipendenza. Possono però nascere schemi più complessi da leggere e mantenere:
  - Docenze(Studente, Docente)
  - Corsi(Studente, Corso)
- Uno schema in BCNF è anche in 3FN
- Nella pratica si accetta la normalizzazione in 3FN, senza procedere ulteriormente

# Esercizio: Normalizzazione di database relazionali

Normalizzare in 3FN il seguente schema:

`AthletesFull(Id, IdAthlete, Name, Sex, Age, Height, Weight, NOC, Games, Year, Season, City, Sport, Event, Medal)`

<u>Id</u>	IdAthlete	Name	Sex	Age	Height	Weight	NOC	Games	Year	Season	City	Sport	Event	Medal
184738	92854	Federica Pellegrini	F	16	177	65	ITA	2004 Summer	2004	Summer	Athina	Swimming	200 metres Freestyle, Women	Silver
184741	92854	Federica Pellegrini	F	20	177	65	ITA	2008 Summer	2008	Summer	Beijing	Swimming	200 metres Freestyle, Women	Gold

# Esercizio: Normalizzazione di database relazionali / soluzione

Normalizzare in 3FN il seguente schema:

AthletesFull(Id, IdAthlete, Name, Sex, Age, Height, Weight, NOC, Games, Year, Season, City, Sport, Event, Medal)

Partecipations(Id, IdAthlete, Age, NOC, IdGame, IdEvent, Medal)

Athletes(IdAthlete, Name, Sex, Height, Weight)

Games(Id, Games, Year, Season, City)

Events(Id, Event, Sport)