

# Basi di Dati

*Corso di Laurea in Informatica*  
2012-2013

## Progetto: Introduzione

Mauro Conti

Department of Mathematics

University of Padua

conti@math.unipd.it

<http://www.math.unipd.it/~conti/>



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Slides credits to:  
Prof. Kristen Brent Venable

# Costruzione di una base di dati

---

- ▶ Analisi dei requisiti
- ▶ Progettazione
  - ▶ Progettazione concettuale, logica e fisica dei dati
  - ▶ Progettazione delle applicazioni
- ▶ Realizzazione
- ▶ Noi spesso considereremo l'analisi dei requisiti una parte della progettazione concettuale



# Il progetto di basi di dati

---

► Base di dati + Interfaccia web di una organizzazione a scelta dello studente. Comprende:

1. Abstract
2. Descrizione testuale dei requisiti e operazioni tipiche
3. Progettazione concettuale
4. Progettazione logica
5. Implementazione dello schema logico
6. Query, procedure, trigger e funzioni
7. Interfaccia web



# Abstract e analisi dei requisiti

---

- ▶ **Abstract:** descrizione generale e breve del caso di studio, (non piu' di 30 righe)
  - ▶ Che cosa viene modellato
  - ▶ In quale contesto
  - ▶ Quali sono le funzioni fondamentali



# Analisi dei requisiti

---

- ▶ Identificare le classi
- ▶ Per ogni classe dire quali
  - ▶ proprietà,
  - ▶ le operazioni,
  - ▶ e le relazioni con altre classi

si vogliono modellare

- ▶ Frasi del tipo
  - ▶ Di X interessa.....



# Progettazione concettuale

---

- ▶ Lista di tutte le classi :
  - ▶ breve descrizione della collezione che rappresenta
  - ▶ attributi con il loro tipo
  - ▶ Opzionale: eventuali vincoli di integrità'
- ▶ Descrizione delle associazioni e le loro proprietà strutturali
  - ▶ molteplicità'
  - ▶ totalità'
- ▶ Descrizione della gerarchia tra le classi



# Per essere accettabile

---

- ▶ Il progetto deve contenere :
  - ▶ Un numero adeguato di classi:  $\geq 5$
  - ▶ Almeno una gerarchia
  - ▶ Un esempio di associazione per ogni tipo di molteplicità





# Schema concettuale

---

- ▶ Oltre alle liste indicate prima, fornire lo schema concettuale, cioè la rappresentazione grafica delle nozioni indicate in precedenza
- ▶ Il formalismo grafico è quello visto a lezione (e descritto nel libro)
- ▶ Opzionale indicare i vincoli di integrità
- ▶ Se ci sono altri vincoli indicarli in modo testuale
- ▶ Editor grafico Gnome Dia



# SuperMarket Online : Abstract

---

- ▶ Abstract
  - ▶ Tale progetto consiste nella realizzazione di una semplice base di dati per la gestione di un negozio di prodotti alimentari e non. E' stata , inoltre, creata una interfaccia web che permette l'interazione dell'utente con la base di dati. A tal proposito sono state definite due categorie di utenti che possono accedere al database, ossia i clienti, i quali possono effettuare degli acquisti di prodotti e i membri dello staff del negozio che, invece, gestiscono i vari prodotti e gli ordini di acquisto. Questo negozio online e' stato chiamato "SuperMarket Online".
  - ▶ Osservazioni
  - ▶ L'abstract non deve parlare del progetto ma di cosa modella la base di dati
    - ▶ Mancano informazioni importanti
      - ▶ cosa si intende per gestione del negozio
    - ▶ Ci sono dettagli non adeguati a un abstract
  - ▶
    - ▶ ripartizione dei ruoli etc...
-

# SuperMarket Online: A. dei requisiti(1)

---

- ▶ **Negozi di alimentari**
  - ▶ Gestione ordinaria
  - ▶ Interfaccia web che permette di fare ordini online
- ▶ **Due tipi di utenti:**
  - ▶ clienti
  - ▶ personale
- ▶ **Classi:**
- ▶ **Prodotti:**
  - ▶ codice numerico identificativo
  - ▶ nome
  - ▶ categoria merceologica
  - ▶ marca
  - ▶ genere (alimentare /non-alimentare)
  - ▶ prezzo
  - ▶ quantità disponibile
  - ▶ data di scadenza ( se alimentare)
- ▶ Ogni prodotto può appartenere ad un solo tipo, marca
- ▶ Un prodotto è disponibile se quantità > 0, e , se alimentare se la data di scadenza è maggiore di quella corrente



# SuperMarket Online: A. dei requisiti(2)

---

- ▶ Ogni cliente registrato ha la possibilità di fare acquisti
- ▶ Cliente:
  - ▶ e-mail (identifica univocamente)
  - ▶ password
  - ▶ nome
  - ▶ cognome
  - ▶ città
  - ▶ indirizzo
  - ▶ provincia (solo due disponibili per la consegna)
  - ▶ CAP
  - ▶ numero telefonico
- ▶ Ad ogni cliente viene associata una “carta amica”
- ▶ accumula punti: 0.2 punti x 1Euro
- ▶ se punti > 15 → sconto 5% per spese >50 Euro, 10% per spese >100Euro
- ▶ Carta
  - ▶ e-mail
  - ▶ saldo punti



# SuperMarket Online: A. dei requisiti(3)

---

- ▶ Ogni cliente puo' effettuare ordini di acquisto di prodotti disponibili e per quantita' minori o uguali alla disponibilita'
- ▶ Ordini sono: conclusi o non conclusi
- ▶ Se non conclusi possono essere disdetti
- ▶ Se inserito nel database l'ordine e' concluso
- ▶ Ordine:
  - ▶ codice numerico identificativo
  - ▶ cliente
  - ▶ punti
  - ▶ data di completamento (solo se concluso)
  - ▶ membro dello staff responsabile (solo se concluso)



# SuperMarket Online: A. dei requisiti(4)

---

- ▶ I membri dello staff si occupano degli ordini e della gestione del database
- ▶ Possono aumentare le quantità disponibili dei prodotti
- ▶ Modificare la data di scadenza (in caso di ritiro e sostituzione)



# SuperMarket Online: Progettazione concettuale (1) CLASSI

---

- ▶ Lista delle classi con tipi e attributi
- ▶ **Utenti:** rappresenta gli utenti che possono accedere alla base di dati per fare acquisti o per la gestione dei prodotti e degli ordini d'acquisto
  - ▶ e-mail : string <<PK>>
  - ▶ pwd : string
  - ▶ nome : string
  - ▶ cognome : string
  - ▶ **clienti**
    - ▶ città' : string
    - ▶ indirizzo :string
    - ▶ provincia : string
    - ▶ CAP : int
    - ▶ numtel : string
  - ▶ **Staff**
    - ▶ data assunzione: string    Non serve l'indirizzo dei membri dello staff?



# SuperMarket Online: Progettazione concettuale (2) CLASSI

---

- ▶ Carte

- ▶ Punti: int

- ▶ Acquisti

- ▶ Id: int <<PK>>
  - ▶ Punti acquisto : int
  - ▶ Conclusi
    - ▶ Data di conclusione : date
    - ▶ Concluso da : string
  - ▶ Non conclusi
    - ▶ nessun attributo





# SuperMarket Online: Progettazione concettuale (3) CLASSI

---

## ▶ **Prodotti**

- ▶ Id : int <<PK>>
- ▶ Nome : string
- ▶ Prezzo : real
- ▶ Quantita' : int

### ▶ **Alimentari**

- ▶ Data di scadenza : date

### ▶ **Non alimentari**

- ▶ nessun attributo proprio

## ▶ **Tipi (categorie merceologiche)**

- ▶ Nome : string <<PK>>

## ▶ **Marche**

- ▶ Nome : string <<PK>>
- 



# SuperMarket Online: Progettazione concettuale (4) ASSOCIAZIONI

---

## ► Associazioni

### ► Clienti-Carte: possiede

- Ogni cliente ha una carta , ogni carta e' di un cliente
- Molteplicita': 1:1
- Totalita':
  - totale da Clienti a Carte parziale
  - totale da Carte a Clienti

### ► Acquisti-Clienti: e' effettuato da

- Ogni cliente puo' fare 0 o piu' ordini, ogni ordine e' effettuato da un unico cliente
- Molteplicita': M:1
- Totalita':
  - totale da Acquisti a clienti
  - parziale da Clienti a Acquisti

### ► Acquisti-staff: concluso da

- Ogni membro dello staff puo' concludere uno o piu' acquisti, Ogni acquisto puo' essere completato da un unico membro dello staff
- Molteplicita': (M:1)
- Totalita':
  - totale da Acquisti a Staff
  - parziale da Staff a Acquisti

►

---



# SuperMarket Online: Progettazione concettuale (5) ASSOCIAZIONI

---

## ▶ Acquisti-Prodotti: riguarda

- ▶ Ogni acquisto puo' riguardare quantita' diverse di uno o piu' prodotti e ogni prodotto puo' essere contenuto in 0 o piu' acquisti in quantita' diverse. Questa associazione ha come attributo: Quantita'.
- ▶ Molteplicita': M:N
- ▶ Totalita':
  - ▶ totale verso Prodotti
  - ▶ parziale verso Acquisti

## ▶ Prodotti-Tipi: appartiene

- ▶ Ogni prodotto appartiene a un solo tipo, ogni tipo puo' essere associato a piu' prodotti
- ▶ Molteplicita': M:1 **non M:N ?**
- ▶ Totalita':
  - ▶ totale verso tipi
  - ▶ totale verso prodotti

## ▶ Prodotti-Marche: e' associato

- ▶ Ogni prodotto e' associato a una marca e ogni marca ha piu' prodotti nella propria linea
  - ▶ Molteplicita': M:1
  - ▶ Totalita':
    - ▶ totale verso marche
    - ▶ totale verso prodotti
- 



# SuperMarket Online: Progettazione concettuale (6) NOTE DI MODELLAZIONE

---

- ▶ Costo totale di un acquisto (=somma prezzi dei prodotti) non viene mantenuto separatamente (evita ridondanza)
- ▶ Punti dell'acquisto vengono invece mantenuti per una scelta implementativa



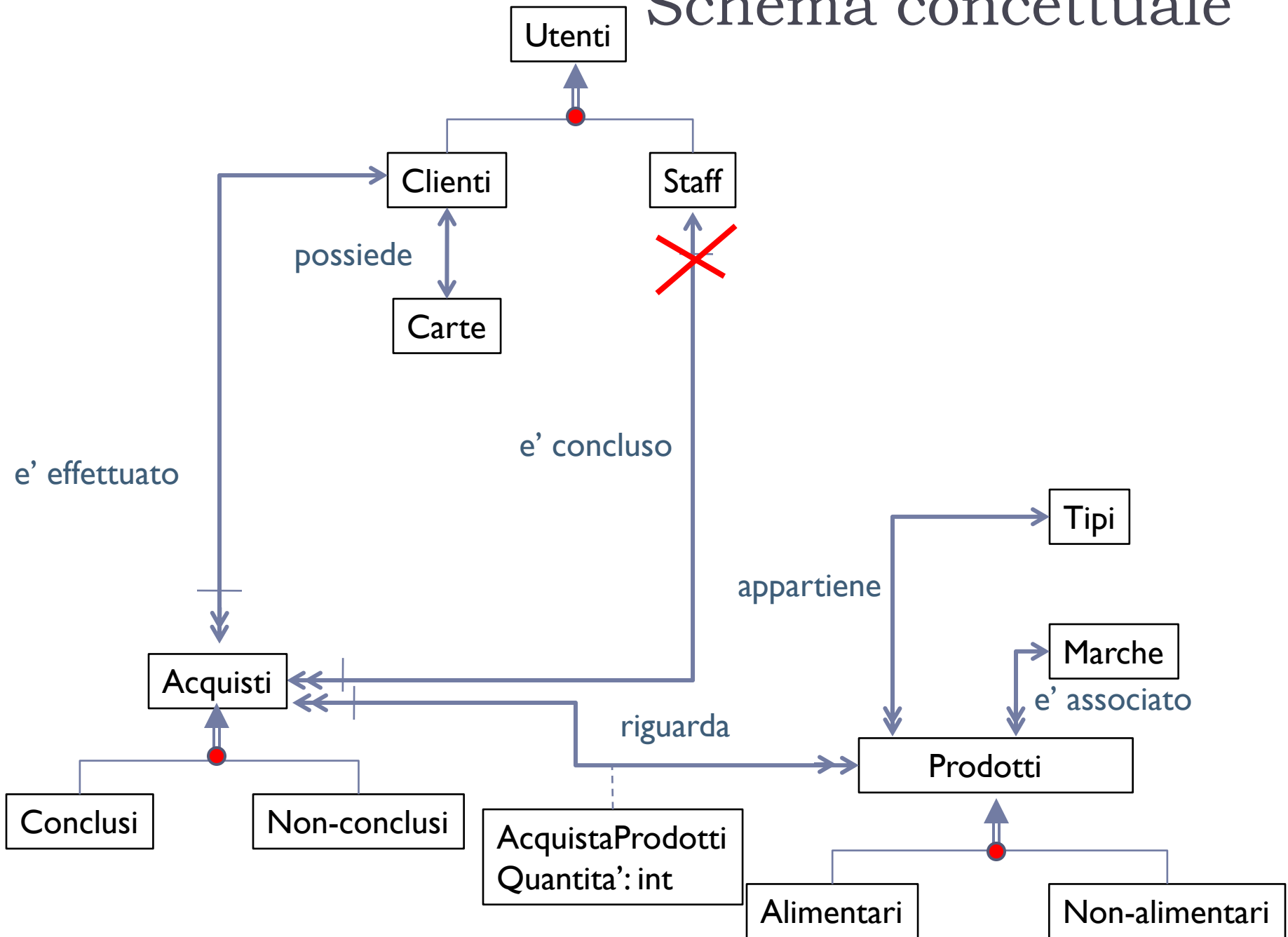
# SuperMarket Online: Progettazione concettuale (7): GERARCHIA delle CLASSI

---

- ▶ Gerarchia Utenti: Clienti/Staff
  - ▶ Classi disgiunte **ragionevole?**
  - ▶ L'unione e' una copertura
- ▶ Gerarchia Acquisti: Conclusi/Non conclusi
  - ▶ Classi disgiunte
  - ▶ L'unione e' una copertura
- ▶ Gerarchia Prodotti: Alimentari/Non Alimentari
  - ▶ Classi disgiunte
  - ▶ L'unione e' una copertura



# Schema concettuale



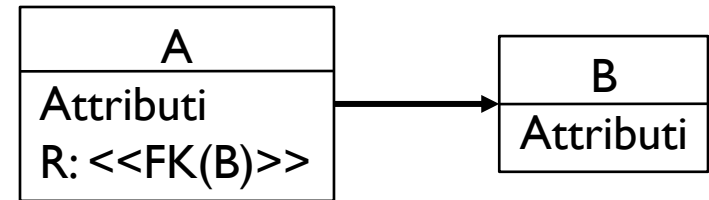
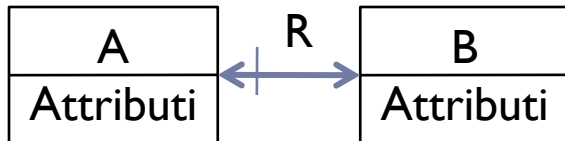
# SuperMarket online: progettazione logica(1)

---

- ▶ Descrizione testuale della rappresentazione delle associazioni
  - ▶ Chiavi esterne
- ▶ Descrizione della rappresentazione delle gerarchie di inclusione
  - ▶ tecnica di partizionamento
- ▶ Rappresentazione grafica dello schema logico

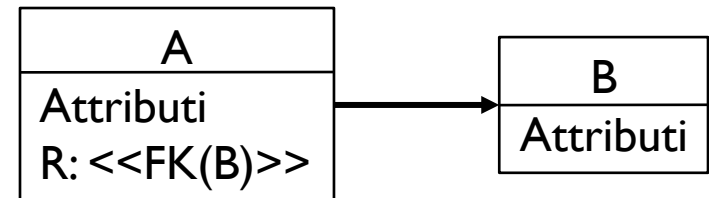
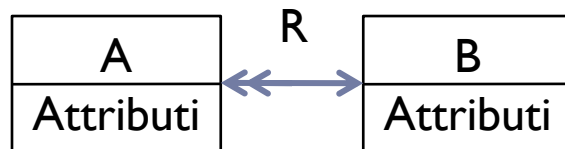


# Ripasso: concettuale → logico



univocità dell'inversa di R con vincolo di chiave  
(<<unique>>, <<key>>) sulla chiave esterna  
possibile solo se R è totale

la direzione di R scelta in modo che sia totale, se possibile

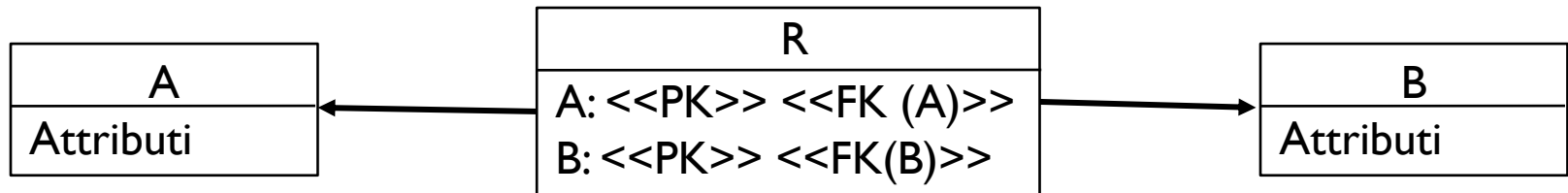
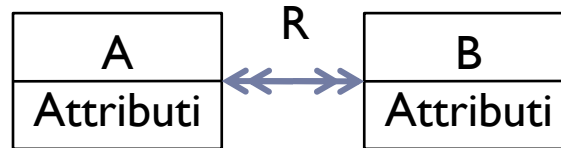


totalità di R con vincolo not-null sulla chiave esterna  
totalità dell'inversa non rappresentabile  
eventuali attributi dell'associazione si possono inserire in A



# Ripasso: concettuale logico

---



eventuali attributi dell'associazione si  
inseriscono in R (in questo caso possono far  
parte della chiave primaria)  
totalità non rappresentabile



# SuperMarket online: progettazione logica(2)

---

- ▶ Rappresentazione Associazioni
  - ▶ **Clienti-Carte: possiede**
    - ▶ Ogni cliente ha una carta , ogni carta e' di un cliente
    - ▶ Molteplicita': 1:1
    - ▶ Totalita':
      - ▶ totale da Clienti a Carte **parziale**
      - ▶ totale da Carte a Clienti
    - ▶ **Chiave esterna su carte**
  - ▶ **Acquisti-Clienti: e' effettuato da**
  - ▶ Ogni cliente puo' fare 0 o piu' ordini, ogni ordine e' effettuato da un unico cliente
  - ▶ Molteplicita': M:1
  - ▶ Totalita':
    - ▶ totale da Acquisti a clienti
    - ▶ parziale da Clienti a Acquisti
  - ▶ **Chiave esterna su Acquisti**
  - ▶ **Acquisti-staff: concluso da**
  - ▶ Ogni membro dello staff puo' concludere uno o piu' acquisti, Ogni acquisto puo' essere completato da un unico membro dello staff
  - ▶ Molteplicita': (M:1)
  - ▶ Totalita':
    - ▶ totale da Acquisti a Staff
    - ▶ parziale da Staff a Acquisti
  - ▶ **Chiave esterna su Acquisti**
- 



# SuperMarket Online: Progettazione logica (3)

---

## ▶ Prodotti-Tipi: appartiene

- ▶ Ogni prodotto appartiene a un solo tipo, ogni tipo puo' essere associato a piu' prodotti
- ▶ Molteplicita': M:1    non M:N ?
- ▶ Totalita':
  - ▶ totale verso tipi
  - ▶ totale verso prodotti
- ▶ Chiave esterna su Prodotti

## ▶ Prodotti-Marche: e' associato

- ▶ Ogni prodotto e' associato a una marca e ogni marca ha piu' prodotti nella propria linea
- ▶ Molteplicita': M:1
- ▶ Totalita':
  - ▶ totale verso marche
  - ▶ totale verso prodotti
- ▶ Chiave esterna su Prodotti



# SuperMarket Online: Progettazione logica (4)

---

## ► Acquisti-Prodotti: riguarda

- Ogni acquisto puo' riguardare quantita' diverse di uno o piu' prodotti e ogni prodotto puo' essere contenuto in 0 o piu' acquisti in quantita' diverse. Questa associazione ha come attributo: Quantita'.
- Molteplicita': M:N
- Totalita':
  - totale verso Prodotti
  - parziale verso Acquisti
- Nuova relazione: AcquistiProdotti
  - Chiave esterna a Acquisti
  - Chiave esterna a Prodotti
  - Nuovo attributo Quantita'
- Acquisti-Prodotti:
  - Acquisto: int <<PK>> <<FK(Acquisti)>>
  - Prodotto : int <<PK>> <<FK(Prodotti)>>
  - Quantita' : int <<not null>>



# SuperMarket Online: Progettazione concettuale

---

- ▶ Gerarchia Utenti: Clienti/Staff
    - ▶ Classi disgiunte **ragionevole?**
    - ▶ L'unione e' una copertura
    - ▶ **Partizionamento orizzontale**
    - ▶ **ok perche' non c'e' una associazione entrante nella superclasse**
    - ▶ **ogni sottoclasse ha invece una relazione che la coinvolge sepratamente**
  - ▶ Gerarchia Acquisti: Conclusi/Non conclusi
    - ▶ Classi disgiunte
    - ▶ L'unione e' una copertura
    - ▶ **Tabella unica**
    - ▶ **Attributo discriminatore multivalore Stato con valori "concluso" e non concluso"**
    - ▶ **Ok visto che gli attributi delle sottoclassi sono pochi**
    - ▶ **Gli attributi di Acquisti conclusi (ConclusoInData,ConclusoDa) possono ora avere valore NULL**
  - ▶ Gerarchia Prodotti: Alimentari/Non Alimentari
    - ▶ Classi disgiunte
    - ▶ L'unione e' una copertura
    - ▶ **Tabella unica**
    - ▶ **Attributo discriminatore multivalore Genere con valori "alimentare" e "non alimentare"**
    - ▶ **Ok visto che gli attributi delle sottoclassi sono pochi**
    - ▶ **L'attributo di Prodotti Alimentari , DataScadenza puo' ora avere valore NULL**
- 



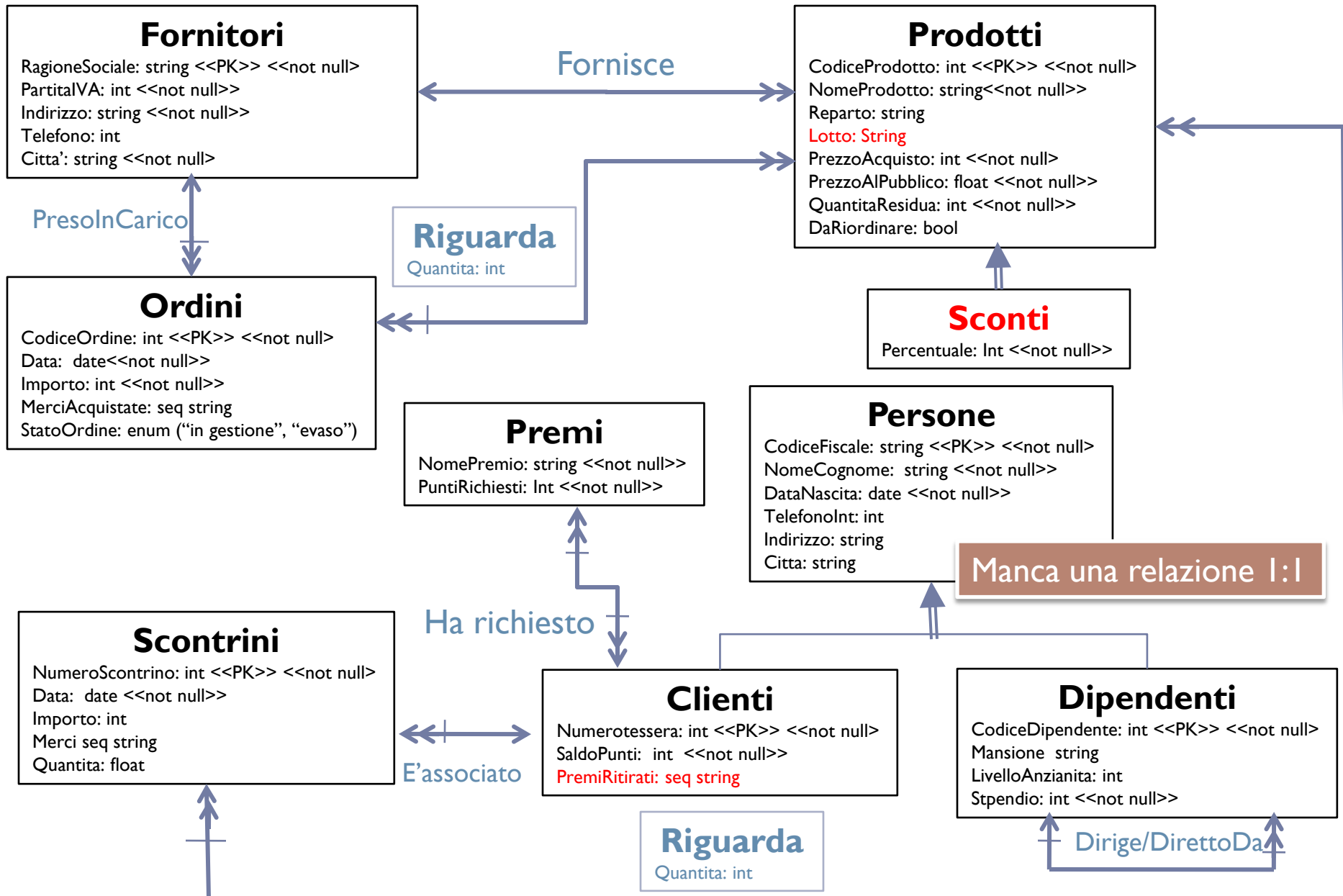
# Esempio: Supermercato

---

- ▶ Il progetto riguarda la gestione di un supermercato ed e' suddiviso in due parti:
  - ▶ la prima parte definisce la struttura della basi di dati;
  - ▶ la seconda implementa un'interfaccia web che permette l'interazione tra l'utente e il database
- 
- ▶ Come caso di studio si e' scelto di prendere in considerazione la gestione di un piccolo supermercato, cercando di rendere le operazioni tipiche il piu' automatizzate possibile. Durante lo sviluppo abbiamo cercato di attenerci a quelle che potrebbero essere le problematiche reali risolvibili con l'aiuto degli strumenti studiati a lezione.



## Esempio: SuperMercato schema concettuale



# Esempio: Document manager

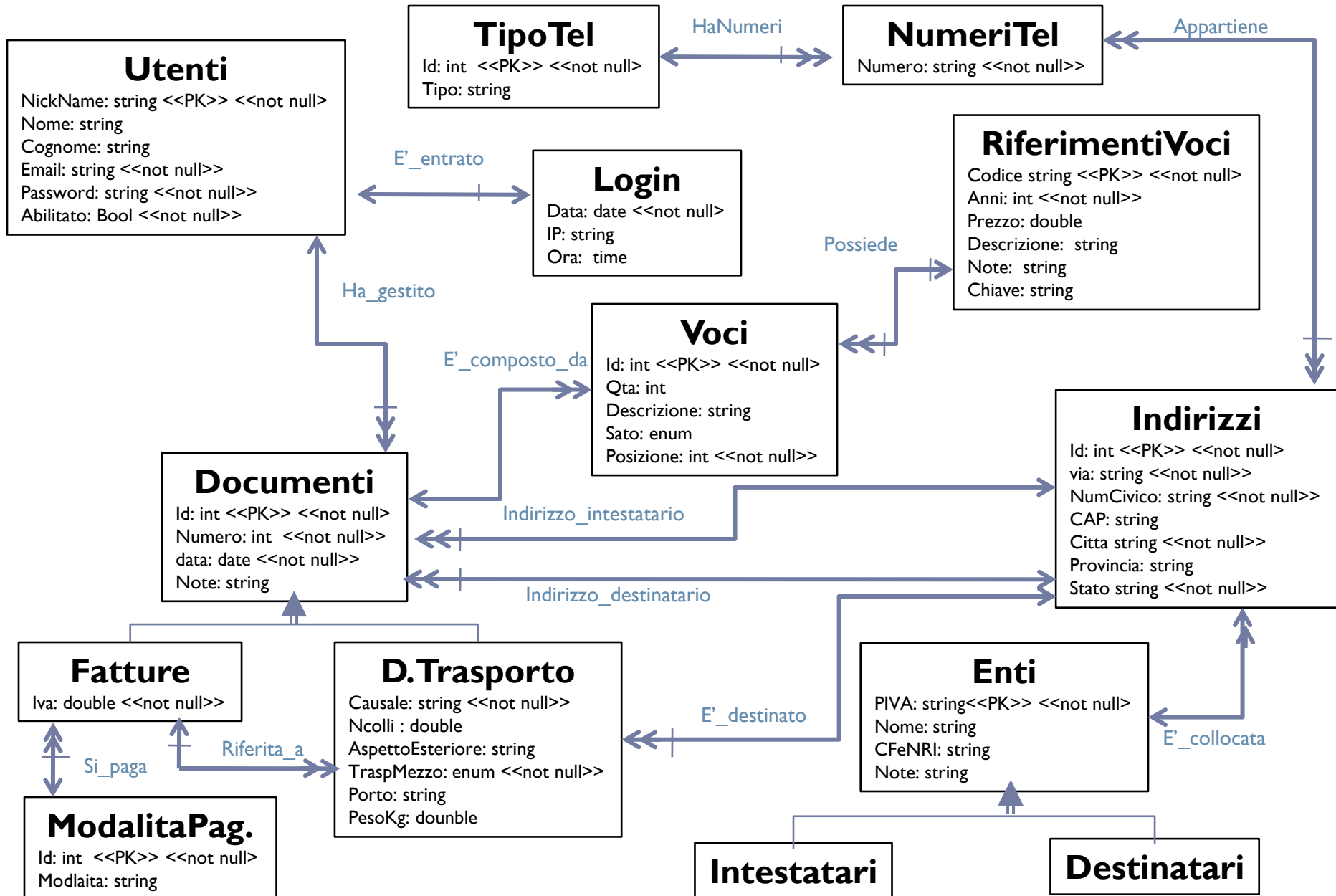
---

- ▶ Abstract
- ▶ L'applicativo web realizzato e' indirizzato ad un'azienda che necessita di uno strumento per la creazione, gestione e archiviazione di documenti, in particolar modo per l'amministrazione di fatture e documenti di trasporto. Si vuole fornire uno strumento per la memorizzazione dei dati relativi all'azienda stessa e delle aziende che hanno rapporti con essa in modo da facilitarne l'inserimento qualora siano di interesse per un documento.





# Document Manager: Schema Concettuale



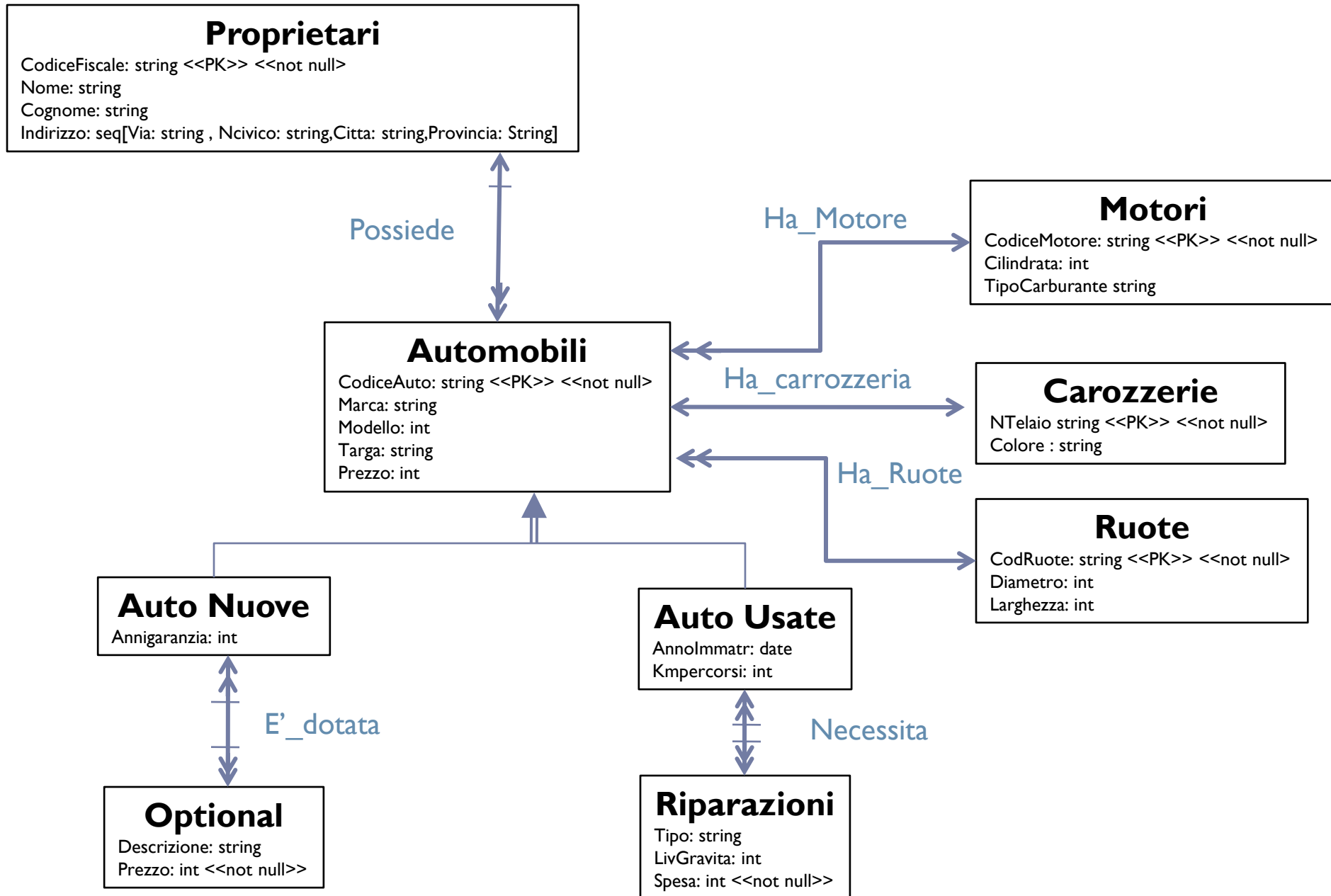
# Esempio: Concessionaria automobili

---

- ▶ Abstract
  - ▶ Si vogliono gestire con un database le informazioni riguardanti l'attività di una concessionaria di automobili multimarca; nel database verranno memorizzate le informazioni riguardanti:
    - ▶ I proprietari delle automobili
    - ▶ Le Automobili presenti in concessionaria (nuove o usate)
    - ▶ Gli optional disponibili per le auto nuove
    - ▶ Le riparazioni da effettuare alle auto usate
    - ▶ le componenti principali delle auto (motori, carrozzerie, ruote)
- 



# Concessionaria: Schema concettuale



<http://creately.com/>  
<https://www.lucidchart.com/>