Basi di Dati

Leonardo Marro

23 marzo 2023

Indice

Parte I
Introduzione Alla Teoria

1 Sistema Informativo

Un Sistema Informativo è un componente di un'organizzazione che gestisce le informazioni di interesse (utili).

[Imm 1] Gestione delle info:

- Raccolta, acquisizione;
- Archiviazione;
- Elaborazione;
- Distribuzione.

Le informazioni vengono rappresentate attraverso i **Dati**.
Informazioni: Dati che assumono uno specifico significato in un determinato contesto. (Ovvero dati interpretati o correlati).

2 Basi di Dati

- Accezione metodologica, generica: **Insieme Organizzato** di un ente pubblico o privato.
- Accezione specifica, metodologica e tecnologica:
  - insieme di dati gestito da un DBMS (DataBase Management System).

Una **base di dati** è insieme di dati atomici strutturati e persistenti, raggruppati in **insiemi omogenei in relazione tra loro** organizzati con la minima ridondanza per essere utilizzati da applicazioni diverse in modo sicuro e controllato.

2.1 Caratteristiche di Basi di Dati

- Condivise
  - Organizzazioni divise in settori con diverse attività.
  - Ciascun settore/attività ha un sottosistema informativo (non per forza totalmente separati)

  [imm2]

Una base di dati è una risorsa integrata condivisa tra applicazioni con le seguenti conseguenze:

- Attività svolte da utenti che usano dati condivisi $->$ vengono richiesti **meccanismi di autorizzazione**.
- Accessi da diversi soggetti in unisono $->$ vengono richiesti dei controlli della **concorrenza**.

Di base il rapporto costo/ n°settori è esponenziale, questo comporta un eventuale **collasso**.
Criticità e costi:

- Eterogeneità dei sistemi: fornitori e tecnologie.
- Difficoltà nell'interscambio di dati incompatibili.
- Ridondanza e incoerenza delle informazioni.
- Assenza di una visione unitaria (Programmatori differenti).

Si usa allora lo sviluppo integrato per migliorare l'efficienza.

2.2 Sviluppo Integrato

1. Data la struttura organizzata si esegue un'**analisi** attenta del sistema informativo (come funziona).
2. Definiti schema e dati e info di interesse, si sviluppa l'applicazione.
3. L'app si interfaccia attraverso il DBMS in base alle sue esigenze.

$$\text{Prima i Dati, Poi le Applicazioni!} \tag{1}$$

3 Il modello relazionale

[tabella da copiare (ospedale)]
Il termine tabella è errato, sono chiamate **relazioni**. Ogni relazione contiene il suo **nome**, degli **attributi** (o campi o proprietà)(intestazioni delle singole colonne), l'insieme di tutti gli attributi rappresenta lo **schema** della relazione.
All'interno della tabella si trovano **tupla**(o n-upla o record) che rappresentano occorrenze dei medici (in questo esempio) descritti dagli attributi, l'insieme delle tuple si chiama **istanza** della relazione, in fine abbiamo il singolo

dato (il quale ottiene un significato grazie alla correlazione degli attributi).
In una relazione può essere presente un attributo sottolineato, chiamato **attributo identificatore** della relazione, nel nostro esempio attraverso quel dato possiamo identificare i singoli medici.
Attraverso l'attributo Primario/MATR possiamo correlare le relazioni.

3.1 Relazione

Una relazione è definita:

- da uno **schema** della relazione
- dalla **istanza** della relazione (o stato)

L'**attributo** di una relazione è definito come una coppia $A_i : T_i$

- T_i è il tipo dell'attributo;
- A_i è il nome dell'attributo.
- Tipi standard (o default):
  - Integer, Real, String...
- Tipi utente (definiti dall'utente sotto certe restrizioni); non si possono raggruppare diversi tipi.

Tipo T_i caratterizzato da:

- T_i nome identificativo del tipo;
- D_i dominio di valori;
- collezione di **operazioni** che agiscono su D_i $(+, -, *, /)$;
- operazioni di **confronto** su D_i $(<, >, \leq, \geq)$.

3.2 Dominio di un attributo

Un tipo è **sempre associato** ad un dominio. Questo avviene attraverso la funzione ***dom*** che associa A_i con il suo dominio D_i.

$$D_i = dom(A_i)$$

3.3 Valore nullo

Se alla creazione di un elemento non conosco un suo dato, esiste un valore speciale chiamato **valore nullo** o **NULL** con la seguente proprietà:

$$\forall T_i, \text{NULL} \in D_i$$

ovvero il valore NULL appartiene al dominio D_i di qualsiasi tipo T_i.

3.4 Schema di una relazione

Lo schema di una relazione è un insieme di attributi con la seguente notazione:

$$A_1 : T_1, ..., A_n : T_n$$

Sono insiemi di attributi tutti con **nomi distinti**; possono avere tipi uguali all'interno della stessa relazione, mentre l'ordine degli attributi è **irrilevante** (nella definizione formale di Codd).

3.5 Schemi con nomi

Lo schema di una relazione **con nome** ha la seguente notazione:

$$R(A_1 : T_1, ..., A_n : T_n)$$

dove R è un generico nome. In certi casi il nome si può sotto intendere, ma nei DBMS è **obbligatorio specificarlo**. Esempio:

Pazienti(COD, Cognome, Nome....)

3.5.1 Notazione Generale

Definendo l'insieme di attributi A, posso usare la notazione $R(A)$ per indicare l'insieme degli attributi per sintetizzare la rappresentazione, dove R è il nome dello schema.

3.6 Grado di una relazione

La **cardinalità** di $|A|$ di uno schema è il n° degli attributi dello schema A. L'insieme di attributi di uno schema di una relazione è sempre un insieme **non vuoto**, ovvero $|A| > 1$.

3.7 Istanza (o stato) di una relazione

Dato uno schema A, l'**istanza** o **stato** r di una relazione è un insieme di tuple $< v_1, ..., v_n$ aventi la proprietà:

$$\forall i, v_i \in dom(A_i)$$

3.8 Cardinalità di una relazione

La cardinalità $|r|$ di una relazione è il numero di tuple$\in$ r, detta **cardinalità di una relazione**. Con il simbolo t, si indica la singola tupla.

3.9 Notazioni ibride

Utilizziamo lettere maiuscole per indicare le relazione, mentre uso le minuscole per indicare le istanze delle relazioni.

3.10 Notazione tabellare

[imm tabella]

3.11 Relazione (notazione)

La relazione è definita da una coppia

$$< R, r >$$

dove R è lo schema e R è l'istanza.

3.11.1 Tupla

Data una relazione r, per indicare una singola tuple, si usa:

$$t =< v_1, ..., v_n >$$

, quindi $t \in r(A_1, ..., A_n)$.

3.11.2 Valori di una tupla

Per indicare il valore di una tupla t si usa la notazione $t[A_i]$, quindi $t[A_i] = v_i$. Ricordarsi che l'ordine non è importante. Esiste anche la dot-notation $(t.A)$ in aggiunta alla notazione standard.

4 Relazione matematica e Relazione di Codd

In matematica $A \times B$ è diverso da $B \times A$, per cui le relazioni sono dipendenti dall'ordine, questo si oppone al modello relazionale il quale considera l'ordine irrilevante. In oltre in matematica una relazione è è un sottoinsieme di un prodotto cartesiano: $s \subset \{dom(A_1) \times dom(A_2) \times ...\}$.
Una **funzione totale** $f : X \to Y$ può essere vista come un insieme di tuple.
Il dominio sono gli attributi e i codominio sono i valori degli attributi corrispondenti. Avremo la funzione $t : \{A_1, A_2, ..., A_n\} \to D$ con $D = \{dom(A_1) \cup dom(A_2) \cup ...\}$.
Una tupla, in quanto una funzione, si può anche rappresentare anche sotto forma di tabella: [imm tabella].
N.B. Tutte le tuple (quindi anche funzioni) t_i sono **tutte distinte**, in quanto due tuple identiche non possono esistere.
Una tupla t si dice **compatibile** con lo schema A se:

- t è **funzione totale** su A;
- ogni valore di t∈D;
- vale il vincolo $\forall A_i, t[A_i] \in dom(A_i)$

5 Vincoli

5.1 Vincoli locali

I **vincoli locali** sono caratterizzazioni dei valori che le tuple posso ottenere per farle rispettare la realtà da rappresentare. Si possono considerare come delle restrizioni sui valori delle tuple.

5.1.1 Vincolo locale di dominio

Per ogni attributo ho un **insieme ben definito** di valori possibili, quindi qualsiasi valore v_i del dominio D è **atomico**(singolo carattere, singola stringa, etc.), se invece il valore dell'attributo è una relazione, non può essere atomico. **NON PERMESSO**.

5.2 Vincoli su valori nulli

Un vincolo NOT NULL è soddisfatto se il valore bersaglio $\forall t \in r$, il valore $t[A_i]$ **non è nullo**.

5.3 Altri vincoli

Possiamo introdurre **restrizioni sui domini** (una persona non può avere più di 120 anni) per obbligare i valori ad avere un certo senso, oppure introduciamo **confronti sui domini** (la data di inizio deve essere antecedente alla data di conclusione).

5.4 Vincolo di identificazione

Questo vincolo viene applicato attraverso il vincolo di **chiave relazionale**, indica il contesto applicativo (ad esempio la matricola di uno studente), questi vincoli possono essere composti da **più valori**. Nelle relazioni di Codd tutti gli elementi dell'istanza devono essere distinti tra di loro.

5.5 Superchiave

Data una relazione $r(A)$, un sottoinsieme di attributi $sk \subseteq A$ è una **superchiave** se:

$$\forall i, j(t_i[sk] = t_j[sk]) \to (t_i[A] = t_j[A])$$

5.6 Verifica dell'implicazione

$$\alpha \to \beta \text{ equivalente a} (\neg\alpha \vee \beta)$$

Verifico l'unicità con i diversi valori della stessa tabella sotto lo stesso parametro, un dato deve essere **SOLO uguale a se stessa**; se questa condizione non è valida per almeno una tupla, allora la tupla **non è superchiave**.

5.6.1 Da quantificazione universale ad esistenziale

$$\alpha \to \beta \equiv (\neg\alpha \vee \beta)$$

negandola

$$\neg(\alpha \to \beta) \equiv \neg(\neg\alpha \vee \beta)$$

da cui

$$\nexists i, j(t_i[sk] = t_j[sk]) \to (t_i \neq t_j)$$

5.7 Chiave candidata

$k \subseteq A$ si dice **chiave candidata** se k è **superchiave** di R e k è **superchiave minimale**, ovvero una superchiave il cui sottoinsieme non è a sua volta una superchiave.

5.7.1 Proprietà delle superchiavi

Se sk è una superchiave allora $sk \subseteq w \subseteq A$ è una **superchiave** per cui A è necessariamente una superchiave e di conseguenza A contiene almeno una chiave candidata. Sarà a discrezione del progettista scegliere una sola **Chiave principale** (o primaria). Tutti gli attributi di una chiave principale devono essere **NON NULLI**, mentre una chiave candidata può averne.

5.7.2 Correttezza di una relazione

Un'istanza di una relazione $r(A)$ è corretta se per ogni tupla $t \in r$ è compatibile con lo schema A e sono soddisfatti tutti i vincoli locali.

6 Relazione in prima forma normale

Una relazione si dice in **prima forma normale** quando tutti gli attributi hanno domini di valori atomici. Verrà approfondito nell'argomento della **normalizzazione**. Questo serve a verificare la compatibilità della tupla e a controllare che gli eventuali valori non siano nulli.

6.1 Schema di una basi di dati

Lo schema di una base di dati è un insieme di schemi con nome

$$S = \{R_1, R_2, ..., R_n\}$$

identificati dalle seguenti proprietà:

- I nomi sono distinti;
- Ogni schema possiede i rispettivi vincoli locali;
- Ogni relazione R_i possiede una chiave principale.

Ogni relazione va successivamente definita.

6.2 Vincoli globali su S

Su ogni relazione R si può definire un vincolo globale **interrelazionali** su uno schema S, il più importante è il **vincolo di integrità referenziale** (foreign key).

6.2.1 Vincolo di integrità referenziale

Sotto esempio: Se in RICOVERI c'è una tupla $t.[PAZ] = X$ allora necessariamente in PAZIENTI dovrà esistere una tupla $t.[COD] = X$. Questo criterio non per forza deve essere inverso.

Definizione:
*Considerando la relazione $R_h(PK, ..)$ dove PK è l'insieme di attributi della chiave principale e la relazione $R_S(...B...)$ dove B sono un insieme di attributi, **Esiste un vincolo di integrità referenziale** di B rispetto a R_h se:*

$$\forall t_i(t_i \in r(R_j)) \to \exists t_j(t_j \in r(R_h)) \wedge (t_i[B] = t_j[PK])$$

Ovviamente i vincoli devono essere compatibili.

6.3 Stato o istanza di una base di dati

Lo stato di una base di dati è definito come

$$DB = (< R_1, r_1 >, ..., < R_n, r_n)$$

dove ogni r_i deve essere corretta e devono soddisfare i vincoli globali.

Si può concludere l'argomento specificando che il modello relazionale è un **modello di dati orientato ai valori**, mentre in altri modello le corrispondenze avvengono attraverso riferimenti.

Parte II

Algebra Relazionale

7 Modello interrogazione di Codd

Interrogare una base di dati significa ottenere dalla base di dati una relazione particolare. Bisogna passare dal linguaggio naturale ad un'espressione algebrica (combinazione di operatori con relative specifiche e parametri)

7.1 Algebra Relazionale

Codd utilizza il paradigma algebrico per formalizzare l'interrogazione che è una costruzione procedurale che indica i passaggi necessari per risolvere lo specifico problema. Vengono definiti degli operatori di base e operatori derivati la cui combinazione permette di comporre qualsiasi quesito:

- Operatori di base:
  - σ
  - Proiezione Π
  - Prodotto Cartesiano
  - Unione $\cup$
  - Differenza
  - Ridenominazione
- Operatori derivati:
  - Intersezione $\cap$
  - Join (e le varie forme) θ
  - Quoziente

Gli *operatori algebrici* ricevono molteplici relazioni in entrata producendo una singola relazione chiamata **Virtuale**.

7.1.1 Operatore di Selezione

Data una relazione r su uno schema $A, \sigma_p(r(A))$ è **l'operatore di selezione** con p predicato e $r(A)$ è l'argomento.

Il predicato p è un'espressione booleana di predicati atomici, ne esistono due tipi:

- $A_i\theta$ confronto costante;
- $A_i\theta$ confronto A_j

con A_i, A_j attributi di A e θ come operatore di confronto.

Otterrà come output una selezione di Tuple che soddisfano il predicato, ovvero una relazione **senza nome** con lo stesso schema dell'argomento, ma con come istanza un insieme delle Tuple che soddisfano il predicato.

Operatore di selezione: Una relazione r su uno schema A e un predicato p, l'operatore di selezione $\sigma(r(A))$ produce una relazione senza nome che ha l'identico schema A della relazione argomento, ma solo tuple che soddisfano il predicato.

7.2 Cardinalità della selezione

La **Cardinalità** della relazione è

$$0 \leq |\sigma_p(r(A))| \leq |r(A)|$$

che sarà 0 se il predicato è sempre falso e $|r(A)|$ se sempre vero.

7.3 Operatore di Proiezione

L'operatore di proiezione

$$\prod_{A_i, A_j \dots} (r(A))$$

produce come risultato lo schema $\{A_i, A_j .....A_k\}$ con istanza tutte le tuple della relazione argomento, ma solo rispetto ai campi $A_i, A_j .....A_k$. (In parole povere: prendo solo le colonne che mi interessano, i campi A_x).

7.3.1 Cardinalità della proiezione

La cardinalità della proiezione **NON è** uguale alla cardinalità della relazione stessa, invece la cardinalità della proiezione è uguale a:

$$0 \leq |\prod_{A_i, A_j, \dots, A_k}| \leq |r(A)|$$

, in quanto accetto "doppioni di dati" (quindi non chiavi), posso trovare delle ripetizioni, ma visto che non posso avere tuple uguali, le ripetizioni vengono **omesse**. Se si proietta su un insieme di attributi contenente una superchiave, la proiezione avrà la stessa cardinalità della relazione stessa.

7.4 Operatori Insiemistici

Gli operatori insiemistici sono l'unione $\cup$ e la differenza $-$, richiedono per definizione che gli schemi delle relazioni siano identici. Il risultato dell'operatore insiemistico su $r_1(A)$ e $r_2(A)$ è una relazione che ha:

- Schema identico
- Unione: $r_1 \cup r_2$ (unione delle tuple)
- Differenza: $r_1 - r_2$ (tuple di r_1 non contenute in r_2)

7.4.1 Cardinalità dell'unione

$$0 \leq |r_1(A) \cup r_2(A)| \leq |r_1(A)| + |r_2(A)|$$

7.5 Composizione di Operazioni

Si possono comporre le espressioni algebriche unendo insieme gli operatori.

8 Notazione albero sintattico

[image missing]

8.1 Correttezza sintattica

Nelle espressioni algebriche è necessario controllare la correttezza sintattica, ovvero la coerenza tra operatori e a argomenti.

8.2 Operatore di Ridenominazione

L'operatore ha come argomento $r(A)$ ed il suo compito è quello di rinominare gli attributi della relazione. Data una relazione il risultato dell'operazione

$$\rho_{B_i,B_j,..,B_k \leftarrow A_i,A_j,..,A_k}(r)$$

è una relazione **virtuale** (ovvero non modifica lo schema) con schema identico, tranne per gli attributi che sono stati ridenominati. L'istanza rimane intoccata. Ricordarsi di controllare i domini. In caso si consideri necessario è possibile rinominare anche il nome dello schema.

$\rho_{UTENTI_{(CF,Provincia)} \leftarrow PAZIENTI_{(COD,Residenza)}}$

9 Prodotto Cartesiano

Date due relazioni $r_1(A)$ e $r_2(B)$ con intersezione nulla, l'operazione

$$r_1(A) \times r_2(B)$$

produce come risultato la relazione r' con schema R' composto dall'unione degli schemi $A \cup B$ e istanza combinazione di tutte le tuple di r_1 con r_2. [Immagine esempio]. Il prodotto cartesiano si avvale della proprietà commutativa. Questo operatore di base viene utilizzato nel calcolo di altri operatori derivati, il più importante: **join**.

9.1 Cardinalità del Prod. Cartesiano

$$0 \leq |r_1(A) \times r_2(B)| = |r_1(A)| \cdot |r_2(B)|$$

10 Operatore di Join

Questo operatore serve a costruire informazioni estratte da più relazioni mettendo in correlazione con diverse informazioni di diverse relazioni. Il theta-join è definito come:

$$r_1(A) \bowtie_\theta r_2(B) = \sigma_\theta(r_1(A) \times r_2(B))$$

Posso stabilire delle condizioni di calcolo:

$$\bowtie_{PAZ=COD}$$

accoppierà solo le tuple che soddisfano il vincolo imposto. Gli schemi su cui si fa join sono **disgiunti**.

10.1 Cardinalità del join

Essendo il join una forma di selezione del prodotto cartesiano più efficiente, la cardinalità si calcola nel seguente modo:

$$0 \leq |\sigma_\theta(r_1(A) \times r_2(B))| \leq |r_1(A) \times r_2(B)|$$

sapendo che $|r_1(A) \times r_2(B)| = |r_1(A)| \cdot |r_2(B)|$, allora

$$0 \leq |r_1(A) \bowtie_\theta r_2(B)| \leq |r_1(A)| \cdot |r_2(B)|$$

10.2 Join e Selezione

10.3 Esempio

Tutti i medici che hanno operato Missoni Luigi:

$\Pi_{MATR,CM,NM}((\sigma_{Cognome="Missoni" \wedge Nome="Luigi"}(pazienti) \bowtie_C$

[missing]

10.4 Dot Notation

Per evitare situazioni di ambiguità con la stessa nominazione, possiamo utilizzare la **Dot Notation** per evitare di rinominare gli attributi segnandogli assieme alla relazione a cui appartiene. Es:$\rho_{medici.Reparto=ricoveri.Reparto}$

10.5 Interrogazioni con conteggio

In algebra relazionale non è possibile effettuale interrogazioni con conteggi, per eseguire questo tipo di calcolo si utilizzano specifici operatori in SQL. In algebra relazionale si possono però fare dei calcoli del tipo: "Elencare i pazienti che hanno subito due o più ricoveri"; una situazione tale comporterà l'eventuale presenza di molteplici tuple con nome uguale, ma con date di ricovero diverse. Ci serve quindi poter relazionare a se stessa la relazione.
Bisognerà rinominare le tutti gli attributi e fare un theta join con criteri del tipo: "$Inizio1 \neq Inizio2$". Ci ritroviamo adesso con tuple uguali, per risolverlo inseriamo $Inizio1 < Inizio2$. Effettivamente ci ritroveremo a rinominare l'intera relazione e ad usare la Dot Notation.

10.6 Casi Particolari di Join

10.6.1 Equi-Join

Caso particolare del θ join dove $r_1(A)$ e $r_2(B)$ sono da intendersi con una congiunzione di uguaglianze. Vengono visti diversi casi:

1. r_1 con partecipazione completa al join, ovvero ogni tupla di r_1 trova una corrispondenza in r_2. La cardinalità massima sarà $\leq |r_1(A)| \cdot |r_2(A)|$

2. r_1 in equi-join con r_2 in corrispondenza con la chiave principale di r_2, ovvero $B_i = PK_j$ dove $B_i \in B$ e $PK_j \in PK$. Essendo PK la chiave principale univoca: $0 \leq |r_1(A) \bowtie_\theta r_2(B)| \leq |r_1(A)|$.

3. r_1 in equi-join con r_2 come nel caso precedente, ma con vincoli di integrità referenziale. Il vincolo in questione è: [missing]. Quindi la cardinalità sarà $= |r_1(A)|$

11 Natural Join

11.1 Casi limite del natural join

Se A e B sono due schemi disgiunti, il natural join diventa equivalente al prodotto cartesiano in quando si comporta come un equi-join senza alcun attributo su cui verificare l'uguaglianza.

12 Semi-Join $\ltimes$

Date due relazioni $r_1(A)$ $r_2(A)$ definiamo il semi-join come:

$$r_1(A) \ltimes_\theta r_2(B) = \Pi_A(r_1(A) \bowtie_\theta r_2(B))$$

Esempio: Elencare tutti i dati dei primari:

$$medici \ltimes_{\text{MATR=Primario}} reparti$$

Si può vedere semplicemente come un join dove omettiamo tutti gli attributi del parte aggiunta, tenendo solo quelli della relazione base.

Parte III

Algebra Relazionale pt.2

Osserviamo i seguenti esempi:

- Elencare i pazienti non residenti a Torino (si può vedere come pazienti con residenza diversa da Torino) **Negazione Inessenziale**
- Elencare i medici non primari (Si può solo vedere come i medici che non sono primari) **Negazione Essenziale**

12.1 Negazione Essenziale

Per Elencare i medici non primari:

1. Si definisce l'universo U
2. Si risponde alla domanda in forma positiva p (Elencare i medici primari)
3. Si trova la risposta all'interrogazione $R = U - P$

12.2 Negazione Inessenziale

Per Elencare i pazienti non residenti in città in cui risiede un medico:

1. U = pazienti
2. P = pazienti $\ltimes_{Paz.res=Med.res}$ Medici
3. $R = U - P$

13 Quantificatori Universali

Quando si trova "tutti", "ogni", "sempre" anche nelle loro forme implicite, siamo davanti ad un caso di quantificatore universale. Indica un rapporto di relazione verso **tutti** gli elementi della relazione (o di quella a cui relazionata).

13.1 Operatore Derivato Quoziente

Dati due insiemi disgiunti di attributi A e B, L'operatore derivato di quoziente si definisce come:

$$r(A,B) \div s(B) = \Pi_A(r) - \Pi_A((\Pi_A(r) \times s) - r)$$

la cui cardinalità sarà:

$$|r(A,B) \div s(B)| \leq |\Pi_A(r)| \leq |r|$$

Estrae tutti gli elementi di **che sono in relazione** con tutti gli elementi di B

14 Semantica di Codd

15 Proprietà degli operatori Algebrici

- Proprietà distributive della selezione
  - Rispetto alla Proiezione
  -

15.1 Proprietà commutativa del prodotto Cartesiano

Siccome lo schema risultato del prodotto cartesiano è l'unione dello schema e l'ordine degli attributi in un relazione è irrilevante, allora

$$r(A) \times s(B) = s(B) \times r(A)$$

15.2 Proprietà commutativa del θ join

Ricordando che $r(A) \bowtie_\theta = \sigma_\theta(r(A) \times s(B))$ possiamo scrivere che

$$r(A) \bowtie$$

[missing]

15.3 Proprietà associativa del prodotto cartesiano

Siccome:

- $r(A) \times s(B)$ è $A \cup B$
-
-

[missing]

$$(r(A) \times s(B)) \times u(C) = r(A) \times (s(B) \times u(C))$$

15.4 Proprietà associativa del θ join

[missing]

15.5 Proprietà della selezione multiple

La selezione $p \wedge q$ sceglie le tuple che soddisfano sia p che q. La selezione di p di quelle selezionate da q possiede lo stesso significato.

15.6 Proprietà della sostituzione degli Operatori

$$\sigma_{p \wedge q}(r(A)) = (\sigma_p(r(A)) \cap \sigma_q(r(A)))$$

(Dimostrazione simile a quella precedente)

15.7 Proprietà distributiva della selezione rispetto alla proiezione

$$\sigma_p \Pi_X(r(A)) = \Pi_x \sigma_p(r(A))$$

Vale la condizione ove il predicato p è definito solo su attributi X.

15.8 Proprietà distributiva della selezione rispetto al prodotto cartesiano

15.9 Proprietà distributiva della selezione rispetto al Join

$$\sigma_p(r(A) \bowtie_\theta s(B))$$

Per essere applicata **necessita** che p non coinvolga sia attributi di A che di B.

15.10 Proprietà distributiva della selezione rispetto a unione e differenza

15.11 Proprietà della selezione multipla

15.12 Proprietà della proiezione multipla

15.13 Proprietà distributiva della proiezione rispetto al prodotto cartesiano

15.14 Proprietà distributiva della proiezione rispetto al join

Parte IV
Laboratorio

Software online utilizzato per gli esercizi: Diagrams

16 Generalizzazione

Una generalizzazione mette in relazione una o più entità comprendendole come casi particolari:

- E è generazione di $E_1, E_2, ...$;
- $E_1, E_2, ...$ sono specializzazioni di E.

Esistono più tipi di generalizzazione in base alla situazione; se ogni occorrenza del genitore sono occorrenze di almeno una delle entità figlie, allora è **parziale**; se ogni occorrenza del genitore è occorrenza di **al più** una figlia **Esclusiva**.

17 Esercizi

17.1 Esercizio 1

17.2 Esercizio 2

Generalizzazioni sbagliate, mancano identificatori, mancano alcuni attributi

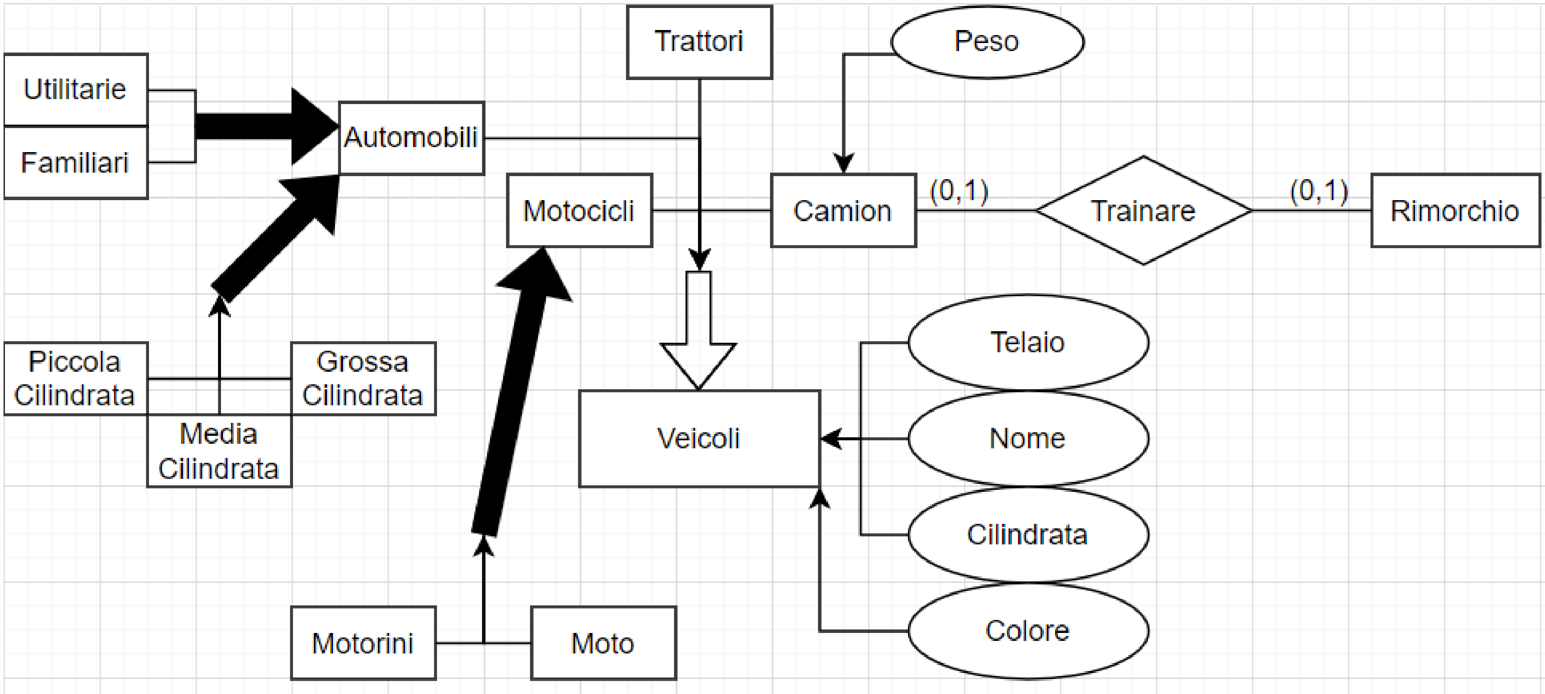

Figura 1: Testo dell'esercizio 2