

**Laurea in Informatica
A.A. 2021-2022**

Corso "Base di Dati"

Esercitazione Algebra Relazionale

Prof. Massimiliano de Leoni



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

Considerare una relazione

$R(A, \underline{B}, \underline{C}, D, E).$

Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di R:

1. $\pi_{ABCD}(R)$

2. $\pi_{AC}(R)$

3. $\pi_{BC}(R)$

4. $\pi_C(R)$

5. $\pi_{CD}(R)$

Considerare una relazione

$R(A, \underline{B}, \underline{C}, D, E).$

Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di R:

1. $\pi_{ABCD}(R)$ **SI**

2. $\pi_{AC}(R)$

3. $\pi_{BC}(R)$

4. $\pi_C(R)$

5. $\pi_{CD}(R)$

Considerare una relazione

$R(A, \underline{B}, \underline{C}, D, E).$

Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di R:

1. $\pi_{ABCD}(R)$ **SI**
2. $\pi_{AC}(R)$ **NO**
3. $\pi_{BC}(R)$
4. $\pi_C(R)$
5. $\pi_{CD}(R)$

Considerare una relazione

$R(A, \underline{B}, \underline{C}, D, E).$

Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di R:

- | | |
|--------------------|-----------|
| 1. $\pi_{ABCD}(R)$ | SI |
| 2. $\pi_{AC}(R)$ | NO |
| 3. $\pi_{BC}(R)$ | SI |
| 4. $\pi_C(R)$ | |
| 5. $\pi_{CD}(R)$ | |

Considerare una relazione

$R(A, \underline{B}, \underline{C}, D, E).$

Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di R:

- | | |
|--------------------|-----------|
| 1. $\pi_{ABCD}(R)$ | SI |
| 2. $\pi_{AC}(R)$ | NO |
| 3. $\pi_{BC}(R)$ | SI |
| 4. $\pi_C(R)$ | NO |
| 5. $\pi_{CD}(R)$ | |

Considerare una relazione

$R(A, \underline{B}, \underline{C}, D, E).$

Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di R:

- | | |
|--------------------|-----------|
| 1. $\pi_{ABCD}(R)$ | SI |
| 2. $\pi_{AC}(R)$ | NO |
| 3. $\pi_{BC}(R)$ | SI |
| 4. $\pi_C(R)$ | NO |
| 5. $\pi_{CD}(R)$ | NO |

Considerare le relazioni

$R_1(\underline{A}, B, C)$ con cardinalità N_1

$R_2(\underline{D}, E, F)$ con cardinalità N_2

Assumere che sia definito un vincolo di integrità referenziale fra:
l'attributo C di R_1 e la chiave D di R_2

Indicare la cardinalità (K) di ciascuno dei seguenti join (specificare l'intervallo nel quale essa può variare):

1. $R_1 \bowtie_{A=D} R_2$

2. $R_1 \bowtie_{C=D} R_2$

3. $R_1 \bowtie_{A=F} R_2$

4. $R_1 \bowtie_{B=E} R_2$

Considerare le relazioni

$R_1(\underline{A}, B, C)$ con cardinalità N_1

$R_2(\underline{D}, E, F)$ con cardinalità N_2

Assumere che sia definito un vincolo di integrità referenziale fra:
l'attributo C di R_1 e la chiave D di R_2

Indicare la cardinalità (K) di ciascuno dei seguenti join (specificare l'intervallo nel quale essa può variare):

$$1. K_1 = R_1 \bowtie_{A=D} R_2 \qquad 0 \leq |K_1| \leq \min(N_1, N_2)$$

$$2. K_2 = R_1 \bowtie_{C=D} R_2$$

$$3. K_3 = R_1 \bowtie_{A=F} R_2$$

$$4. K_4 = R_1 \bowtie_{B=E} R_2$$

Considerare le relazioni

$R_1(\underline{A}, B, C)$ con cardinalità N_1

$R_2(\underline{D}, E, F)$ con cardinalità N_2

Assumere che sia definito un vincolo di integrità referenziale fra:
l'attributo C di R_1 e la chiave D di R_2

Indicare la cardinalità (K) di ciascuno dei seguenti join (specificare l'intervallo nel quale essa può variare):

1. $K_1 = R_1 \bowtie_{A=D} R_2$

$$0 \leq |K_1| \leq \min(N_1, N_2)$$

2. $K_2 = R_1 \bowtie_{C=D} R_2$

$$|K_2| = N_1$$

3. $K_3 = R_1 \bowtie_{A=F} R_2$

4. $K_4 = R_1 \bowtie_{B=E} R_2$

Considerare le relazioni

$R_1(\underline{A}, B, C)$ con cardinalità N_1

$R_2(\underline{D}, E, F)$ con cardinalità N_2

Assumere che sia definito un vincolo di integrità referenziale fra:
l'attributo C di R_1 e la chiave D di R_2

Indicare la cardinalità (K) di ciascuno dei seguenti join (specificare l'intervallo nel quale essa può variare):

$$1. K_1 = R_1 \bowtie_{A=D} R_2 \qquad 0 \leq |K_1| \leq \min(N_1, N_2)$$

$$2. K_2 = R_1 \bowtie_{C=D} R_2 \qquad |K_2| = N_1$$

$$3. K_3 = R_1 \bowtie_{A=F} R_2 \qquad 0 \leq |K_3| \leq N_2$$

$$4. K_4 = R_1 \bowtie_{B=E} R_2$$

Considerare le relazioni

$R_1(\underline{A}, B, C)$ con cardinalità N_1

$R_2(\underline{D}, E, F)$ con cardinalità N_2

Assumere che sia definito un vincolo di integrità referenziale fra:
l'attributo C di R_1 e la chiave D di R_2

Indicare la cardinalità (K) di ciascuno dei seguenti join (specificare l'intervallo nel quale essa può variare):

$$1. K_1 = R_1 \bowtie_{A=D} R_2 \quad 0 \leq |K_1| \leq \min(N_1, N_2)$$

$$2. K_2 = R_1 \bowtie_{C=D} R_2 \quad |K_2| = N_1$$

$$3. K_3 = R_1 \bowtie_{A=F} R_2 \quad 0 \leq |K_3| \leq N_2$$

$$4. K_4 = R_1 \bowtie_{B=E} R_2 \quad 0 \leq |K_4| \leq N_1 \cdot N_2$$

In questa esercitazione, si assuma la seguente notazione:

1. Il simbolo \bowtie_C denota il join di relazioni che “mette insieme” le tuple delle due relazioni che soddisfano la condizione **C**
2. Il simbolo \bowtie senza condizione denota il join naturale, cioè join su valori uguali di attributi con lo stesso nome.
3. Si può usare il punto (“.”) per indicare l’attributo di una relazione (per esempio **R1.a** indica l’attributo **a** della relazione **R1**)

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

1. Trovare i nomi dei fornitori che forniscono pezzi rossi
2. Trovare i fid dei fornitori che forniscono pezzi rossi o pezzi verdi
3. Trovare i fid dei fornitori che forniscono pezzi rossi o si trovano a via Cavour
4. Trovare i fid dei fornitori che forniscono sia pezzi rossi che pezzi verdi
5. Trovare coppie di fid tali che il fornitore con il primo fid applica per alcuni tipi di pezzo un prezzo maggiore di quello del fornitore con il secondo fid.
6. Trovare i pid dei tipi di pezzi forniti da almeno due diversi fornitori
7. Trovare i pid del tipo di pezzo più costoso fornito da “UniPd” (assumendo che sia uno)

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

1. Trovare i nomi dei fornitori che forniscono pezzi rossi

$$\pi_{\text{fnome}}(\text{FORNITORI} \bowtie (\sigma_{\text{colore}=\text{"Rosso"}}(\text{PEZZI}) \bowtie \text{CATALOGO}))$$

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

2. Trovare i fid dei fornitori che forniscono pezzi rossi o pezzi verdi

$$\begin{aligned} & \pi_{\text{fid}}((\sigma_{\text{colore}=\text{"Rosso"}}(\text{PEZZI})) \bowtie \text{CATALOGO}) \\ & \quad \cup \\ & \pi_{\text{fid}}((\sigma_{\text{colore}=\text{"Verdi"}}(\text{PEZZI})) \bowtie \text{CATALOGO}) \end{aligned}$$

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

3. Trovare i fid dei fornitori che forniscono pezzi rossi o si trovano a via Cavour

$$\begin{aligned} & \pi_{\text{fid}} \left((\sigma_{\text{colore}=\text{"Rosso"}}(\text{PEZZI}) \bowtie \text{CATALOGO}) \right) \\ & \cup \\ & \pi_{\text{fid}} \left(\sigma_{\text{indirizzo}=\text{"Via Cavour"}}(\text{FORNITORI}) \right) \end{aligned}$$

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

4. Trovare i fid dei fornitori che forniscono sia pezzi rossi che pezzi verdi

$$\pi_{\text{fid}}(\text{FORNITORI} \bowtie (\sigma_{\text{colore}=\text{"Rosso"}}(\text{PEZZI}) \bowtie \text{CATALOGO}))$$
$$\cap$$
$$\pi_{\text{fid}}(\text{FORNITORI} \bowtie (\sigma_{\text{colore}=\text{"Verdi"}}(\text{PEZZI}) \bowtie \text{CATALOGO}))$$

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

5. Trovare coppie di fid tali che il fornitore con il primo fid applica per alcuni tipi di pezzi un prezzo maggiore di quello del fornitore con il secondo fid.

R1 = Catalogo

R2 = Catalogo

$\pi_{R1.fid, R2.fid} (R1 \bowtie_{R1.pid=R2.pid \wedge R1.costo > R2.costo} R2)$

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

6. Trovare i pid dei tipi di pezzi forniti da almeno due diversi fornitori

R1 = Catalogo

R2 = Catalogo

$\pi_{R1.pid} (R1 \bowtie_{R1.pid=R2.pid \wedge R1.fid \neq R2.fid} R2)$

Si consideri lo schema di base di dati che contiene le seguenti relazioni:

FORNITORI (fid: integer, fnome: String, indirizzo: String)

PEZZI(pid: integer, pnome: String, colore: String)

CATALOGO (fid: integer, pid: integer, costo: real)

7. Trovare i pid del tipo di pezzo più costoso fornito da “UniPd”
(assumendo che sia uno)

$R1 = \text{Catalogo} \bowtie (\sigma_{\text{fnome}=\text{"UniPd"}} \text{Fornitori})$

$R2 = \text{Catalogo} \bowtie (\sigma_{\text{fnome}=\text{"UniPd"}} \text{Fornitori})$

$\pi_{R1.pid} (R1 \bowtie_{R1.fid=R2.fid \wedge R1.costo < R2.costo} R2)$

AR) Si consideri lo schema relazionale composto dalle seguenti relazioni:

IMPIEGATO (Matricola, Cognome, Stipendio, Dipartimento)

DIPARTIMENTO (Codice, Nome, Sede, Direttore)

Con i seguenti vincoli di integrità referenziale:

- tra Dipartimento della relazione IMPIEGATO e Codice della relazione DIPARTIMENTO
- tra Direttore della relazione DIPARTIMENTO e Matricola della relazione IMPIEGATO.

Scrivere nel foglio delle risposte le espressioni in algebra relazionale per le seguenti interrogazioni:

- Trovare i cognomi degli impiegati che NON sono direttori di dipartimento (se un impiegato non direttore di dipartimento ha lo stesso cognome di un direttore di dipartimento qualsiasi, allora tale cognome non deve comparire)
- Trovare i nomi dei dipartimenti in cui lavorano impiegati che guadagnano più di 60.000 euro

AR) Si consideri lo schema relazionale composto dalle seguenti relazioni:

IMPIEGATO (Matricola, Cognome, Stipendio, Dipartimento)

DIPARTIMENTO (Codice, Nome, Sede, Direttore)

Con i seguenti vincoli di integrità referenziale:

- tra Dipartimento della relazione IMPIEGATO e Codice della relazione DIPARTIMENTO
- tra Direttore della relazione DIPARTIMENTO e Matricola della relazione IMPIEGATO.

Scrivere nel foglio delle risposte le espressioni in algebra relazionale per le seguenti interrogazioni:

- a) Trovare i cognomi degli impiegati che NON sono direttori di dipartimento (se un impiegato non direttore di dipartimento ha lo stesso cognome di un direttore di dipartimento qualsiasi, allora tale cognome non deve comparire)

Possibile soluzione:

$\pi_{\text{Cognome}}(\text{IMPIEGATO}) - \pi_{\text{Cognome}}(\text{DIPARTIMENTO} \bowtie_{\text{Direttore=Matricola}} \text{IMPIEGATO})$

- b) Trovare i nomi dei dipartimenti in cui lavorano impiegati che guadagnano più di 60.000 euro

Possibile soluzione:

$\pi_{\text{Nome}}(\sigma_{\text{Stipendio}>60000}(\text{IMPIEGATO} \bowtie_{\text{Dipartimento=Codice}} \text{DIPARTIMENTO}))$

**Laurea in Informatica
A.A. 2020-2021**

Corso "Base di Dati"

**Concetti di base di SQL:
Esercitazione**

Dott. Massimiliano de Leoni



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

Dato il seguente schema:

ESECUZIONE(CodiceReg, TitoloCanz, Anno)

AUTORE(Nome, TitoloCanzone)

CANTANTE(NomeCantante, CodiceReg)

Formulare le interrogazioni SQL che permettono di determinare:

Gli autori e i cantanti **puri**, ovvero autori che non hanno mai registrato una canzone e cantanti che non hanno mai scritto una canzone.

Dato il seguente schema:

ESECUZIONE(CodiceReg,TitoloCanz,Anno)

AUTORE(Nome,TitoloCanzone)

CANTANTE(NomeCantante,CodiceReg)

Formulare le interrogazioni SQL che permettono di determinare:

Gli autori e i cantanti **puri**, ovvero autori che non hanno mai registrato una canzone e cantanti che non hanno mai scritto una canzone.

Soluzione:

```
SELECT Nome FROM AUTORE
WHERE Nome NOT IN
      ( SELECT NomeCantante FROM CANTANTE )
UNION
SELECT NomeCantante FROM CANTANTE
WHERE NomeCantante NOT IN
      ( SELECT Nome FROM AUTORE )
```

Dato il seguente schema:

ESECUZIONE(CodiceReg, TitoloCanz, Anno)

AUTORE(Nome, TitoloCanzone)

CANTANTE(NomeCantante, CodiceReg)

Formulare le interrogazioni SQL che permettono di determinare:

I cantanti che hanno cantato canzoni che sono anche cantate da altri cantanti.

Soluzione:

```
SELECT NomeCantante
FROM CANTANTE JOIN ESECUZIONE ON ESECUZIONE.CodiceReg=CANTANTE.CodiceReg
WHERE TitoloCanz
IN
(SELECT E1.TitoloCanz
FROM ESECUZIONE E1, CANTANTE C1, ESECUZIONE E2, CANTANTE C2
WHERE E1.CodiceReg=C1.CodiceReg AND E2.CodiceReg=C2.CodiceReg AND
      E1.TitoloCanz=E2.TitoloCanz AND C1.NomeCantante<>C2.NomeCantante)
```

Dato il seguente schema:

ESECUZIONE(CodiceReg,TitoloCanz,Anno)

AUTORE(Nome,TitoloCanzone)

CANTANTE(NomeCantante,CodiceReg)

Formulare le interrogazioni SQL che permettono di determinare:

Restituisce il numero di esecuzioni registrate per ogni anno di canzoni di «Mogol»

Soluzione:

```
SELECT Anno,COUNT(*)  
FROM CANTANTE JOIN AUTORE ON ESECUZIONE.TitoloCanz=AUTORE.TitoloCanzone  
WHERE Nome="Mogol"  
GROUP BY Anno
```

Dato il seguente schema:

ESECUZIONE(CodiceReg,TitoloCanz,Anno)

AUTORE(Nome,TitoloCanzone)

CANTANTE(NomeCantante,CodiceReg)

Formulare le interrogazioni SQL che permettono di determinare:

Restituisce il numero di esecuzioni registrate per ogni anno per ogni canzone di «Mogol»

Soluzione:

```
SELECT TitoloCanz, Anno,COUNT(*)  
FROM CANTANTE JOIN AUTORE ON ESECUZIONE.TitoloCanz=AUTORE.TitoloCanzone  
WHERE Nome="Mogol"  
GROUP BY TitoloCanz, Anno
```

Dato il seguente schema:

ESECUZIONE(CodiceReg, TitoloCanz, Anno)

AUTORE(Nome, TitoloCanzone)

CANTANTE(NomeCantante, CodiceReg)

Formulare le interrogazioni SQL che permettono di determinare:

Restituire il/i nome/i dello/degli autore/i che ha scritto la canzone con più esecuzioni, insieme con il nome della canzone stessa

Soluzione:

```
SELECT *
FROM Autore
WHERE TitoloCanz
IN
(
    SELECT TitoloCanz
    FROM ESECUZIONE
    GROUP BY TitoloCanz
    HAVING COUNT(*) >
        SELECT MAX(CONTA) FROM (SELECT COUNT(*) AS CONTA FROM ESECUZIONE GROUP BY
                                TitoloCanzone) AS CONTEGGIO)
```

Esercizi

Si effettui la progettazione concettuale delle basi di dati nei due esercizi che seguono, in accordo con le corrispondenti specifiche e fornendo un diagramma ER per ogni esercizio. Specificare gli eventuali vincoli che non possono essere espressi direttamente nei diagrammi ER;

Esercizio 1

La serra **PianteBellissime** vuole dotarsi di un sistema informativo per gestire la manutenzione e la vendita delle piante. La serra è specializzato in piante di alto fusto, ma vende anche piante fiorite.

Per la piante al alto fusto, si vogliono memorizzare i singoli esemplari in magazzino. Ogni esemplare è contraddistinto da un codice individuale, la specie, il nome botanico, l'età in anni, il clima ed il tipo di terriccio ideali.

Per le piante fiorite, non è di intesse memorizzare i singoli esemplari, ma solo le quantità a disposizione in forma aggregata. Di conseguenza, i tipi di piante fiorite non hanno un codice individuale, sono raggruppate in insiemi identificati tramite il nome botanico; di ogni insieme sono noti il periodo e la durata della fioritura, il numero di esemplari in magazzino, nonché il costo del singolo esemplare (uguale per tutti gli esemplari).¹

Le piante di alto fusto, se non vendute, devono essere sottoposte ad un trattamento annuale di svasamento con terricci diversi a seconda del tipo. A tale scopo, occorre associare ogni specie di pianta ad uno o più terricci.

La serra ha una clientela di due tipi: singoli individui che acquistano al dettaglio e negozianti che acquistano all'ingrosso. Soltanto per questi ultimi è necessario conoscere nome, cognome, codice fiscale e partita iva.

Di ogni vendita interessa memorizzare la data, l'identificativo, le piante acquistate. Ogni vendita contiene uno o più tipi di piante fiorite e piante ad alto fusto. Per ogni vendita, ogni tipo di pianta fiorita è venduto in una certa quantità; viceversa, ogni vendita contiene solo un esemplare di un tipo di pianta ad alto fusto, di cui si vuole conoscere il prezzo di vendita, che può cambiare ad ogni vendita. Solo per le vendite all'ingrosso, si interessa conoscere l'acquirente.

Soluzione Progettazione Concettuale

Si vuole realizzare una base di dati per l'organizzazione di una serra. Si noti che le istanze dell'entità "Pianta Alto Fusto" sono gli esemplari specifici, mentre le istanze di "Pianta Fiorita" si riferiscono ai tipi di piante (per es. *Palma Nana*) e non agli specifici esemplari di quel tipo

Ogni **pianta Alto Fusto** è caratterizzata da:

- Età
- Nome Botanico
- Codice Individuale

¹ Si noti quindi la differenza tra "piante fiorite" e "piante ad alto fusto", rappresentate come due entità separate nel diagramma ER. Un'istanza dell'entità "piante fiorite" rappresenta un tipo di pianta fiorita, che può essere disponibile in magazzino in un certo numero di entità. Viceversa, le istanze di "piante ad alto fusto" rappresentano i singolo esemplari: due istanze si possono riferire a due esemplari dello stesso tipo di pianta ad alto fusto.

- Clima

Ogni **Pianta Alto Fusto** appartiene ad una **Specie**, ma una **Specie** ha una o più Piante in vendita. La **Specie** è identificata da:

- Nome Specie

Ogni **Specie** di pianta è associata ad uno o più **Terricci**, ma un tipo di **Terriccio** può avere più **Specie** di piante associate. Il **Terriccio** è identificato da:

- Tipo Terreno

Ogni **Pianta Alto Fusto** può essere stata **Venduta** oppure no ad un determinato prezzo, ma una **Vendita** può contenere più di una **Pianta Alto Fusto**. Ogni **Vendita** è caratterizzata da:

- ID
- Data di vendita

Ogni **Pianta Fiorita** può essere stata **Venduta** oppure no in un determinato numero di esemplari, ma una **Vendita** può contenere più di una **Pianta Fiorita**. Ogni **Pianta Fiorita** è identificata da:

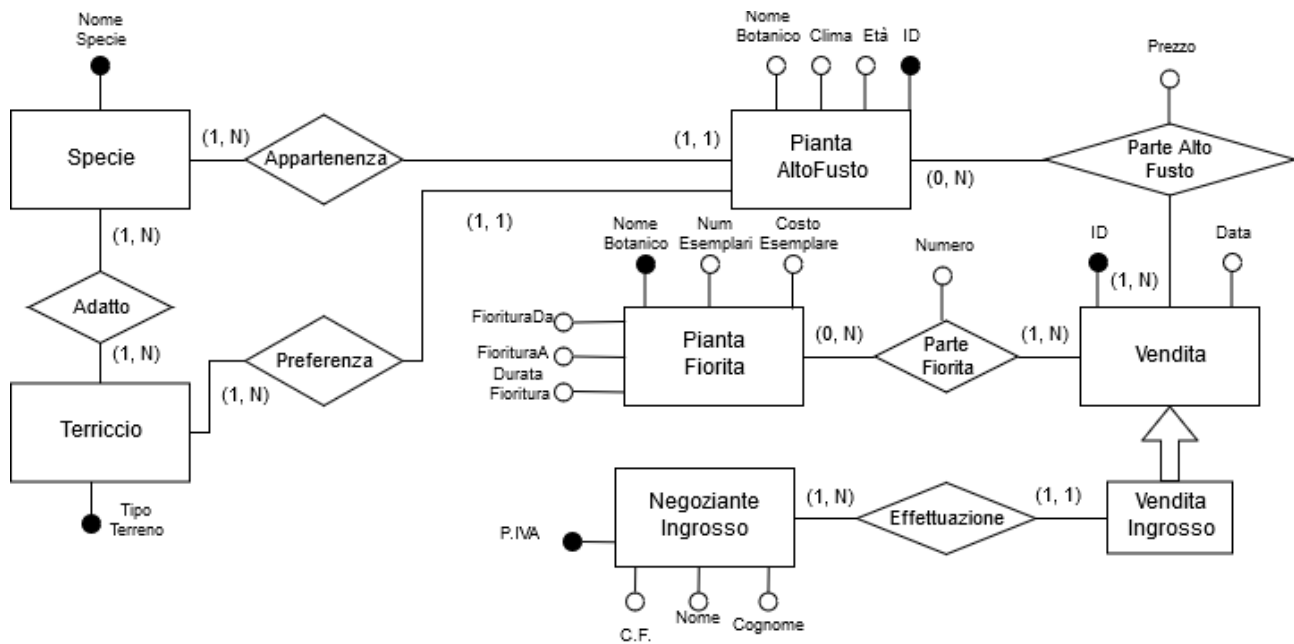
- Data inizio Fioritura
- Data fine Fioritura
- Durata della fioritura
- Numero di Esemplari
- Costo del singolo esemplare

Si noti l'assenza della Generalizzazione Pianta, in cui Pianta AltoFusto e Pianta Fiorita sarebbero state specializzazioni. Si ricordi che le istanze di Pianta AltoFusto sono i singoli esemplari in magazzino, mentre le istanze di Pianta Fiorita sono i tipi botanici. Non ha senso qui fare questa generalizzazione perché sono due concetti parecchio diversi.

Venditore Ingrosso è una specializzazione di **Venditore**. (**Venditore** può o non essere all'ingrosso, mentre ogni **Venditore Ingrosso** è anche un **Venditore**). Nel caso in cui una **Vendita** sia **all'Ingrosso**, essa è effettuata da un **Negoziante all'Ingrosso**, ma un **Negoziante all'Ingrosso** può prendere parte a più **Vendite all'ingrosso**. Ogni negoziante è caratterizzato da :

- Partita IVA (chiave)
- Codice Fiscale
- Nome
- Cognome

Diagramma ER:



Vincoli di Integrità Non Rappresentabili in ER:

- Una **Vendita** riguarda almeno una **Piante Fiorita** o una **Piante Alto Fusto**.

Esercizio 2

Un'azienda che gestisce gli eventi di uno spazio di fiera vuole progettare una base di dati per la memorizzazione delle informazioni di suo interesse.

L'azienda ha il compito di gestire tutti gli eventi che sono organizzati nello spazio di fiera. Questi eventi sono caratterizzati da un nome, dalla durata dell'evento e, se disponibile, dal sito web dell'evento.

L'evento si svolge nei padiglioni dello spazio di fiera. È di interesse memorizzare quali padiglioni sono occupati in quali giorni. Inoltre i padiglioni, oltre ad essere identificati mediante un codice alfanumerico, sono classificabili in padiglioni grandi, medi e piccoli. Ogni padiglione ha associato un espositore, di cui interessa il nome, la partita iva, ed il riferimento ad una persona di contatto (email).

Il padiglione ha un programma delle giornate in cui si svolge l'evento che risulta organizzato in sessioni. A ciascuna sessione è associato un titolo, la giornata in cui si svolge e un insieme di speaker, di cui interessa nome, cognome, email, oltre che titolo e durata della relazione che dovranno esporre.

Si noti che uno speaker può effettuare più di una relazione all'interno dello stesso evento. I partecipanti all'evento devono iscriversi fornendo i loro dati anagrafici (nome, cognome, email). Anche gli speaker devono iscriversi all'evento. Tutti i partecipanti tranne gli speaker devono poi pagare un biglietto di iscrizione caratterizzato da un codice identificativo ed un costo.

Si vuole realizzare una base di dati per l'organizzazione di una fiera.

Un **Evento** è identificato da:

- Sito Web
- Nome
- Data Inizio
- Data Fine

I **Padiglioni** e gli **Espositori** sono **Assegnati** ad un **Evento**, ma un **Evento** ha **Assegnati** più **Padiglioni** ed **Espositori**. Ogni **Assegnazione** è caratterizzata da:

- Data Inizio
- Data Fine

Ogni **Espositore** è associato a più **Assegnazioni**, ma ogni **Assegnazione** prevede un **Espositore**. Ogni **Espositore** è caratterizzato da:

- Nome
- Partita IVA
- Email

Ogni **Padiglione** è associato in più **Assegnazioni**, ma ogni **Assegnazione** prevede un **Padiglione**. Ogni **Padiglione** è caratterizzato da:

- Codice

Padiglione è generalizzazione completa delle entità **Piccolo**, **Medio**, **Grande**, i quali ne rappresentano la dimensione.

Ad ogni **Padiglione** sono associate una o più **Sessioni**, mentre una **Sessione** è univocamente assegnata ad un **Padiglione**. Inoltre ogni **Sessione** si riferisce ad un evento. Ogni **Sessione** è caratterizzata da:

- Data
- Titolo

Ogni **Evento** prevede più Biglietti, ma ogni **Biglietto** è assegnato univocamente ad un **Evento**. Ogni **Biglietto** è identificato da:

- Codice
- Costo

Partecipante è generalizzazione completa di **Speaker** e **Non Speaker**. Ciò significa che ogni **Partecipante** può essere uno **Speaker** o un **Non Speaker**. Inoltre, ogni **Partecipante** può partecipare a più **Eventi**, ed anche ogni **Evento** ha più **Partecipanti**. Ogni **Partecipante** è caratterizzato da:

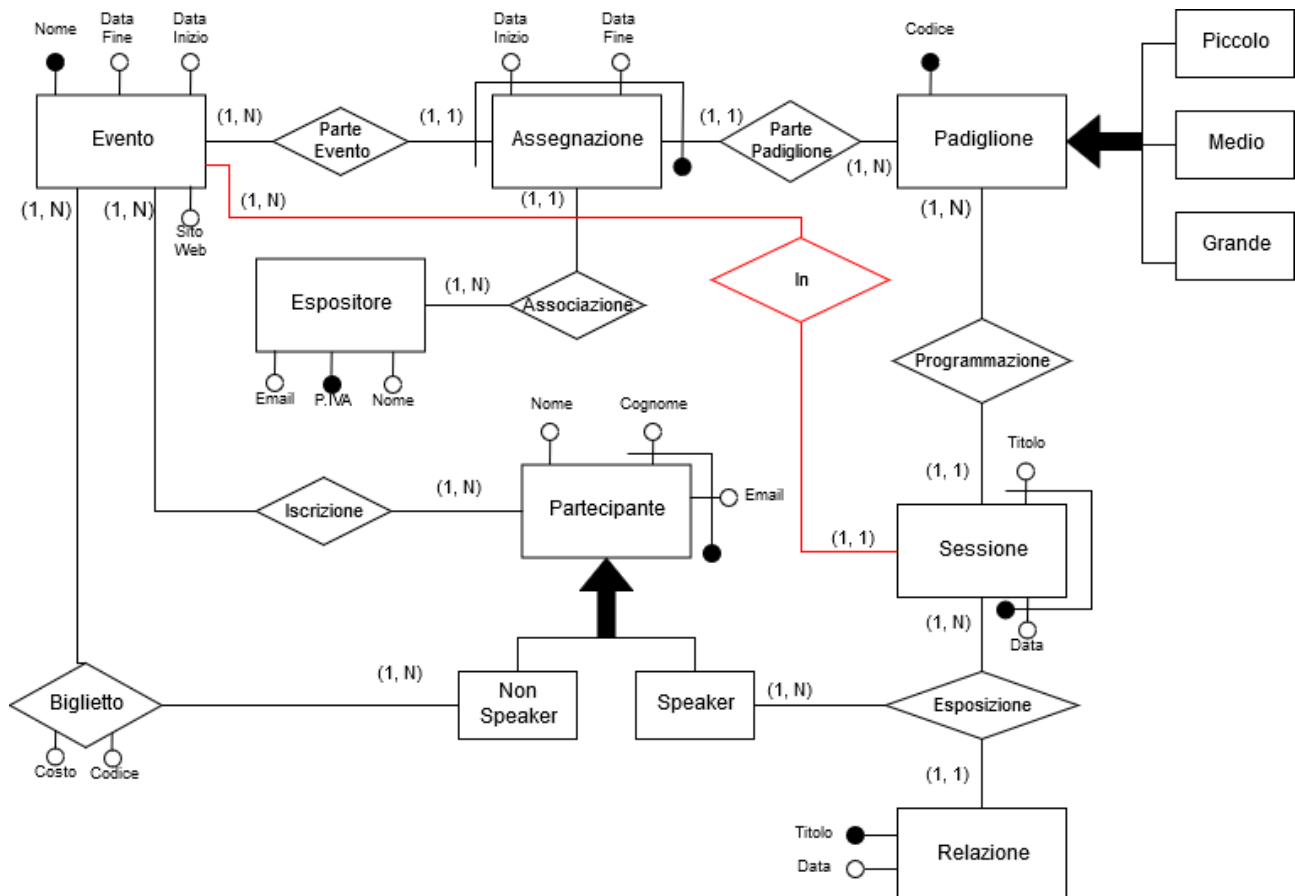
- Email
- Nome
- Cognome

Se un **Partecipante** è uno **Non-Speaker**, esso potrà pagare uno o più **Biglietti**, mentre ad ogni **Biglietto** è assegnato ad ogni **Non-Speaker**.

Ogni **Speaker** può esporre più **Relazioni** in più **Sessioni**, mentre ogni **Relazione** è unicamente assegnata ad uno **Speaker** e ad una **Sessione** ed ogni **Sessione** può avere più **Speaker** con più **Relazioni**. Ogni **Relazione** è identificata da:

- Titolo
- Data

Diagramma ER:



L'uso del colore rosso è per rendere la relazione "In" più visibile a causa delle sovrapposizioni.

Vincoli di Integrità Non Rappresentabili in ER:

- Se uno **Non-Speaker** partecipa ad un **Evento**, allora deve esistere anche un **Biglietto** ad egli associato.
- Il periodo in cui un **Padiglione** viene assegnato ad un evento deve essere compatibile con il periodo in cui l'**Evento** stesso si svolge (in termini di Data Inizio e Data Fine dell'evento).

Esercizio 3

Si vuole realizzare una base di dati che gestisca procedimenti sanzionatori nel contesto di rilevazioni statistiche ufficiali di carattere nazionale. Per alcune rilevazioni statistiche ufficiali esiste, infatti, l'obbligo di risposta da parte dei soggetti contattati per la conduzione delle rilevazioni. Qualora il soggetto contattato non risponda al questionario inviatogli, dopo un prefissato intervallo di tempo, ha inizio un procedimento sanzionatorio che consta di due fasi principali: invio della diffida al soggetto non rispondente e, qualora tale soggetto continui ad essere inadempiente (cioè non risponda al questionario), invio della sanzione che il soggetto stesso dovrà pagare. I soggetti possono essere persone fisiche o imprese. Delle persone fisiche interessa memorizzare il codice fiscale, delle imprese il codice fiscale o la partita iva in maniera alternativa. Inoltre è di interesse l'indirizzo cui il soggetto è contattabile. Si noti che le imprese possono prevedere delle unità locali, ovvero l'impresa si articola secondo una struttura che consiste di un'impresa centrale ed eventualmente di un insieme di imprese "periferiche". Un procedimento viene avviato in relazione alla non-risposta ad una specifica edizione di un'indagine. Ogni indagine è caratterizzata da un nome (es. Forze di lavoro), da una frequenza con cui le sue edizioni occorrono (es. trimestrale) e dalle specifiche edizioni che sono occorse (es. primo trimestre 2011).

Le edizioni, che hanno un codice univoco nell'ambito dell'indagine in cui sono svolte, hanno una data di inizio ed una data di fine che caratterizzano l'inizio e la fine della rilevazione sul campo dei dati oggetto dell'indagine. Nell'ambito di un procedimento è prodotto un insieme di documenti che costituisce il fascicolo del procedimento. Un fascicolo ha un codice che lo identifica nell'ambito del procedimento a cui è legato. I documenti, che dispongono di un ID univoco nell'ambito del fascicolo in cui sono redatti, sono rappresentati da un nome, un tipo, una data di produzione e dal path relativo al file cui sono associati. Del procedimento, oltre alle informazioni necessarie a desumere il suo avanzamento, interessa memorizzare la data di inizio e, qualora sia stato archiviato, l'esito della archiviazione (ad esempio archiviato perché il soggetto ha risposto) e la data di archiviazione.

Soluzione Progettazione Concettuale

Sarà oggetto della prossima esercitazione.

Esercizi

Si effettuano:

1. la progettazione concettuale delle basi di dati nei tre esercizi che seguono, in accordo con le corrispondenti specifiche e fornendo un diagramma ER per ogni esercizio. Specificare gli eventuali vincoli che non possono essere espressi direttamente nei diagrammi ER;
2. la progettazione logica sulla base dei diagrammi ER ottenuti come risultato della progettazione concettuale. Si mostrino anche i risultati intermedi della progettazione logica, con particolare riferimento ai diagrammi ER ristrutturati.

Esercizio 1

La serra **PianteBellissime** vuole dotarsi di un sistema informativo per gestire la manutenzione e la vendita delle piante. La serra è specializzata in piante di alto fusto, ma vende anche piante fiorite.

Per la piante al alto fusto, si vogliono memorizzare i singoli esemplari in magazzino. Ogni esemplare è contraddistinto da un codice individuale, la specie, il nome botanico, l'età in anni, il clima ed il tipo di terriccio ideali.

Per le piante fiorite, non è di interesse memorizzare i singoli esemplari, ma solo le quantità a disposizione in forma aggregata. Di conseguenza, i tipi di piante fiorite non hanno un codice individuale, sono raggruppate in insiemi identificati tramite il nome botanico; di ogni insieme sono noti il periodo e la durata della fioritura, il numero di esemplari in magazzino, nonché il costo del singolo esemplare (uguale per tutti gli esemplari).¹

Le piante di alto fusto, se non vendute, devono essere sottoposte ad un trattamento annuale di svasamento con terricci diversi a seconda del tipo. A tale scopo, occorre associare ogni specie di pianta ad uno o più terricci.

La serra ha una clientela di due tipi: singoli individui che acquistano al dettaglio e negozianti che acquistano all'ingrosso. Soltanto per questi ultimi è necessario conoscere nome, cognome, codice fiscale e partita iva.

Di ogni vendita interessa memorizzare la data, l'identificativo, le piante acquistate. Ogni vendita contiene uno o più tipi di piante fiorite e piante ad alto fusto. Per ogni vendita, ogni tipo di pianta fiorita è venduto in una certa quantità; viceversa, ogni vendita contiene solo un esemplare di un tipo di pianta ad alto fusto, di cui si vuole conoscere il prezzo di vendita, che può cambiare ad ogni vendita. Solo per le vendite all'ingrosso, si interessa conoscere l'acquirente.

Soluzione Progettazione Concettuale

Si vuole realizzare una base di dati per l'organizzazione di una serra. Si noti che le istanze dell'entità "Pianta Alto Fusto" sono gli esemplari specifici, mentre le istanze di "Pianta Fiorita" si riferiscono ai tipi di piante (per es. *Palma Nana*) e non agli specifici esemplari di quel tipo

¹ Si noti quindi la differenza tra "piante fiorite" e "piante ad alto fusto", rappresentate come due entità separate nel diagramma ER. Un'istanza dell'entità "piante fiorite" rappresenta un tipo di pianta fiorita, che può essere disponibile in magazzino in un certo numero di entità. Viceversa, le istanze di "piante ad alto fusto" rappresentano i singoli esemplari: due istanze si possono riferire a due esemplari dello stesso tipo di pianta ad alto fusto.

Ogni **pianta Alto Fusto** è caratterizzata da:

- Età
- Nome Botanico
- Codice Individuale
- Clima

Ogni **Pianta Alto Fusto** appartiene ad una **Specie**, ma una **Specie** ha una o più Piante in vendita. La **Specie** è identificata da:

- Nome Specie

Ogni **Specie** di pianta è associata ad uno o più **Terricci**, ma un tipo di **Terriccio** può avere più **Specie** di piante associate. Il **Terriccio** è identificato da:

- Tipo Terreno

Ogni **Pianta Alto Fusto** può essere stata **Venduta** oppure no ad un determinato prezzo, ma una **Vendita** può contenere più di una **Pianta Alto Fusto**. Ogni **Vendita** è caratterizzata da:

- ID
- Data di vendita

Ogni **Pianta Fiorita** può essere stata **Venduta** oppure no in un determinato numero di esemplari, ma una **Vendita** può contenere più di una **Pianta Fiorita**. Ogni **Pianta Fiorita** è identificata da:

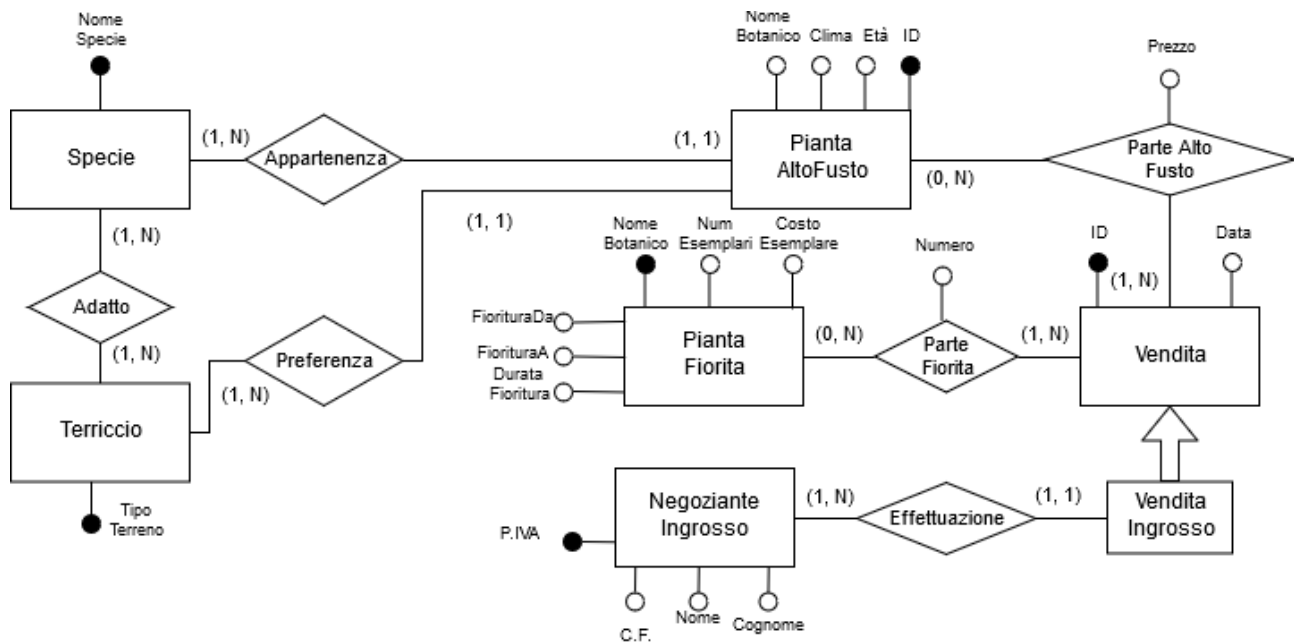
- Data inizio Fioritura
- Data fine Fioritura
- Durata della fioritura
- Numero di Esemplari
- Costo del singolo esemplare

Si noti l'assenza della Generalizzazione Pianta, in cui Pianta AltoFusto e Pianta Fiorita sarebbero state specializzazioni. Si ricordi che le istanze di Pianta AltoFusto sono i singoli esemplari in magazzino, mentre le istanze di Pianta Fiorita sono i tipi botanici. Non ha senso qui fare questa generalizzazione perché sono due concetti parecchio diversi.

Venditore Ingrosso è una specializzazione di **Venditore**. (**Venditore** può o non essere all'ingrosso, mentre ogni **Venditore Ingrosso** è anche un **Venditore**). Nel caso in cui una **Vendita** sia **all'Ingrosso**, essa è effettuata da un **Negoziante all'Ingrosso**, ma un **Negoziante all'Ingrosso** può prendere parte a più **Vendite all'ingrosso**. Ogni negoziante è caratterizzato da :

- Partita IVA (chiave)
- Codice Fiscale
- Nome
- Cognome

Diagramma ER:



Vincoli di Integrità Non Rappresentabili in ER:

- Una **Vendita** riguarda almeno una **Piante Fiorita** o una **Piante Alto Fusto**.

Soluzione Progettazione Logica

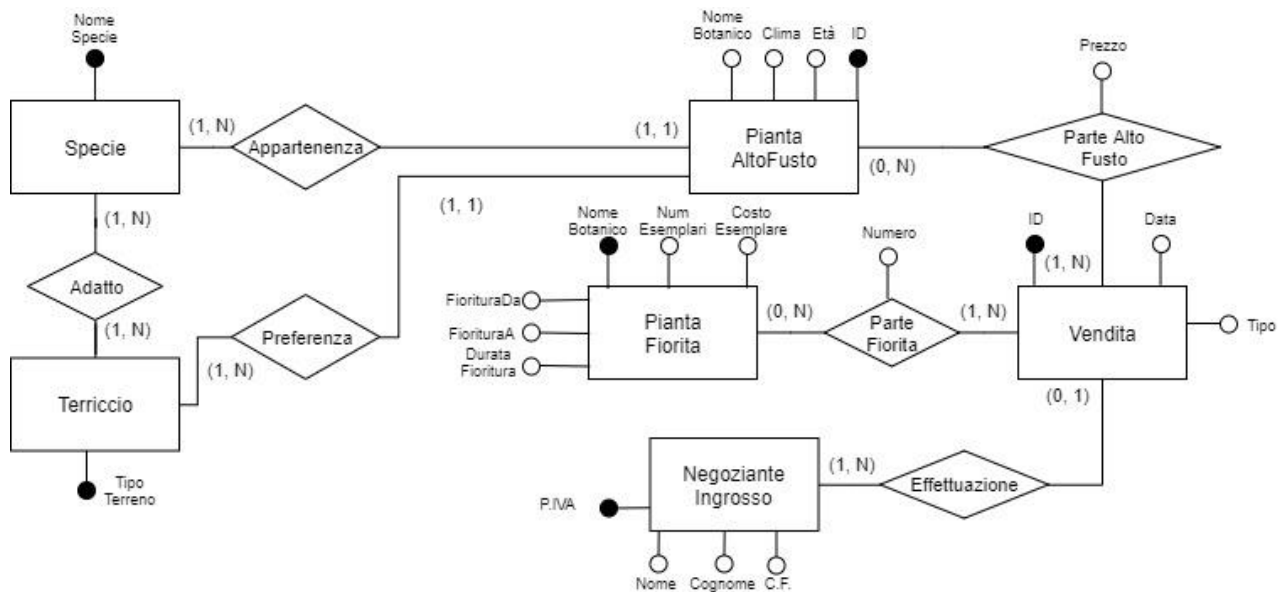
Per la progettazione logica, si partirà dalla prima soluzione, cioè senza la modellazione dell'entità "Trattamento" e le altre relazioni ed entità collegate

Ristrutturazione schema Entità-Relazione:

Dato che i sistemi tradizionali per la gestione delle basi di dati non consentono di rappresentare costrutti quali generalizzazioni, gerarchie, attributi composti o identificatori esterni, risulta necessario ristrutturare lo schema precedente trasformandoli in elementi rappresentabili. In questo caso, nella trasformazione, bisogna solamente rimuovere la generalizzazione Vendita-Vendita Ingrosso. Per far ciò aggiungiamo un attributo "Tipo" alla classe Vendita e aggiungiamo il seguente vincolo di integrità:

- L'attributo *Tipo* della classe **Vendita** può assumere solamente i valori {All'ingrosso, Non all'ingrosso}.

Diagramma E-R ristrutturato:



Schema tradotto in modello relazionale:

Terriccio (TipoTerreno)

Specie (NomeSpecie)

Adatto (Specie, Terriccio)

- Adatto.Terriccio -> Terriccio.TipoTerreno
- Adatto.Specie -> Specie.NomeSpecie

PiantaAltoFusto (ID, NomeBotanico, Età, Clima, TipoTerreno, NomeSpecie)

ParteAltoFusto (Vendita, PiantaAltoFusto, Prezzo)

- ParteAltoFusto.Vendita -> Vendita.ID
- ParteAltoFusto.PiantaAltoFusto -> PiantaAltoFusto.ID

Vendita (ID, Data, Tipo, PIVANegoziante) (PIVANegoziante è chiave esterna a Negoziante all'ingrosso partita IVA)

- PIVA può essere NULL

ParteFiorita (Vendita, PiantaFiorita, Numero)

- ParteFiorita.Vendita -> Vendita.ID
- ParteFiorita. PiantaFiorita -> PiantaFiorita.NomeBotanico

PiantaFiorita (NomeBotanico, NumEsemplari, CostoEsemplare, FiorituraDa, FiorituraA, DurataFioritura)

Negoziante Ingrosso (PIVA, CF, Nome, Cognome)

Esercizio 2

Un'azienda che gestisce gli eventi di uno spazio di fiera vuole progettare una base di dati per la memorizzazione delle informazioni di suo interesse.

L'azienda ha il compito di gestire tutti gli eventi che sono organizzati nello spazio di fiera. Questi eventi sono caratterizzati da un nome, dalla durata dell'evento e, se disponibile, dal sito web dell'evento.

L'evento si svolge nei padiglioni dello spazio di fiera. È di interesse memorizzare quali padiglioni sono occupati in quali giorni. Inoltre i padiglioni, oltre ad essere identificati mediante un codice alfanumerico, sono classificabili in padiglioni grandi, medi e piccoli. Ogni padiglione ha associato un espositore, di cui interessa il nome, la partita iva, ed il riferimento ad una persona di contatto (email).

Il padiglione ha un programma delle giornate in cui si svolge l'evento che risulta organizzato in sessioni. A ciascuna sessione è associato un titolo, la giornata in cui si svolge e un insieme di speaker, di cui interessa nome, cognome, email, oltre che titolo e durata della relazione che dovranno esporre.

Si noti che uno speaker può effettuare più di una relazione all'interno dello stesso evento. I partecipanti all'evento devono iscriversi fornendo i loro dati anagrafici (nome, cognome, email). Anche gli speaker devono iscriversi all'evento. Tutti i partecipanti tranne gli speaker devono poi pagare un biglietto di iscrizione caratterizzato da un codice identificativo ed un costo.

Si vuole realizzare una base di dati per l'organizzazione di una fiera.

Un **Evento** è identificato da:

- Sito Web
- Nome
- Data Inizio
- Data Fine

I **Padiglioni** e gli **Espositori** sono **Assegnati** ad un **Evento**, ma un **Evento** ha **Assegnati** più **Padiglioni** ed **Espositori**. Ogni **Assegnazione** è caratterizzata da:

- Data Inizio
- Data Fine

Ogni **Espositore** è associato a più **Assegnazioni**, ma ogni **Assegnazione** prevede un **Espositore**. Ogni **Espositore** è caratterizzato da:

- Nome
- Partita IVA
- Email

Ogni **Padiglione** è associato in più **Assegnazioni**, ma ogni **Assegnazione** prevede un **Padiglione**. Ogni **Padiglione** è caratterizzato da:

- Codice

Padiglione è generalizzazione completa delle entità **Piccolo**, **Medio**, **Grande**, i quali ne rappresentano la dimensione.

Ad ogni **Padiglione** sono associate una o più **Sessioni**, mentre una **Sessione** è univocamente assegnata ad un **Padiglione**. Inoltre ogni **Sessione** si riferisce ad un evento. Ogni **Sessione** è caratterizzata da:

- Data
- Titolo

Ogni **Evento** prevede più Biglietti, ma ogni **Biglietto** è assegnato univocamente ad un **Evento**. Ogni **Biglietto** è identificato da:

- Codice
- Costo

Partecipante è generalizzazione completa di **Speaker** e **Non Speaker**. Ciò significa che ogni **Partecipante** può essere uno **Speaker** o un **Non Speaker**. Inoltre, ogni **Partecipante** può partecipare a più **Eventi**, ed anche ogni **Evento** ha più **Partecipanti**. Ogni **Partecipante** è caratterizzato da:

- Email
- Nome
- Cognome

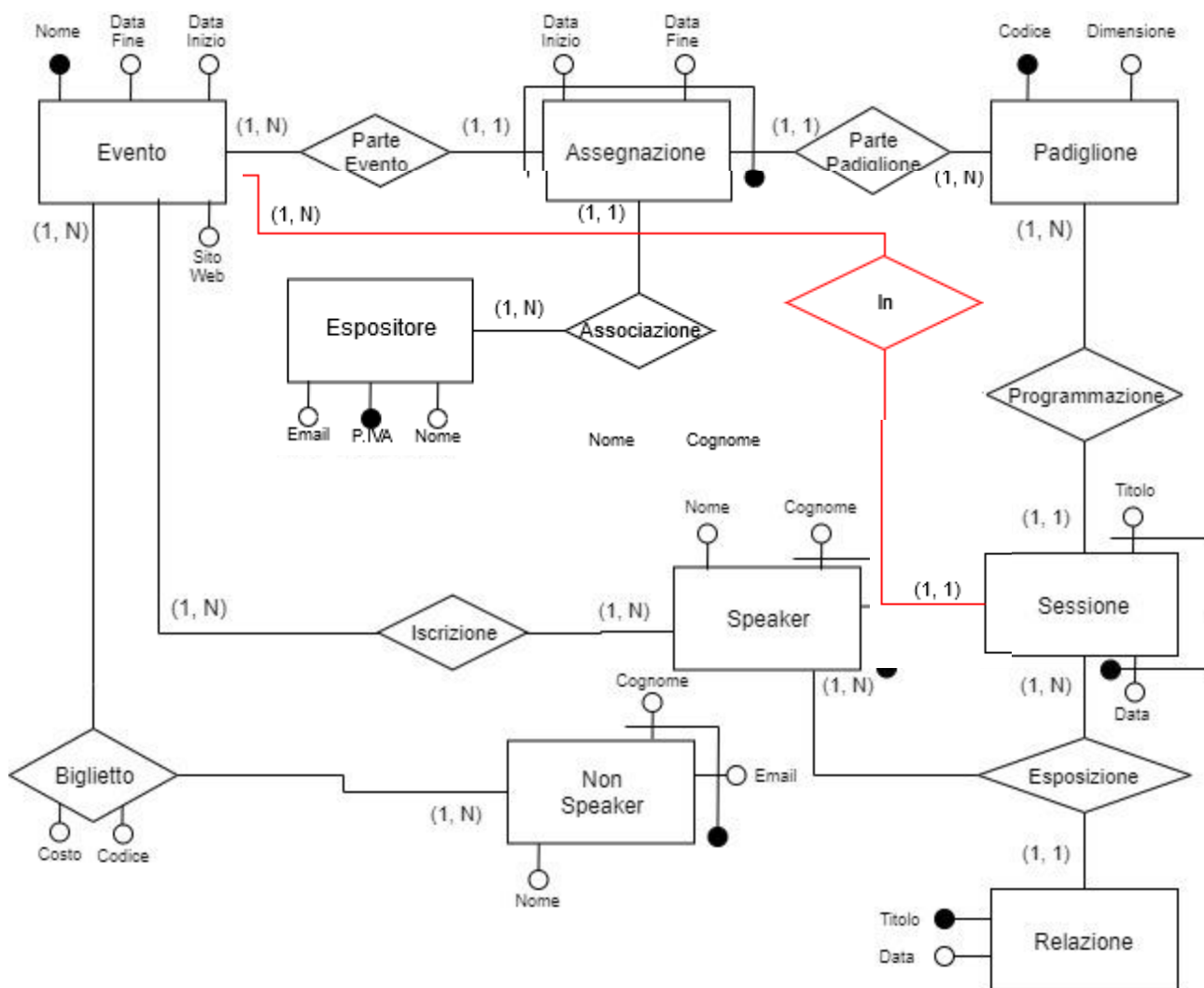
- Se uno **Non-Speaker** partecipa ad un **Evento**, allora deve esistere anche un **Biglietto** ad egli associato.
- Il periodo in cui un **Padiglione** viene assegnato ad un evento deve essere compatibile con il periodo in cui l'**Evento** stesso si svolge (in termini di Data Inizio e Data Fine dell'evento).

Eliminazione delle generalizzazioni

Il diagramma ER possiede 2 generalizzazioni:

- Rimuoviamo le classi Piccolo, Medio, Grande e aggiungiamo un nuovo attributo *Dimensione* alla classe Padiglione. Aggiungiamo quindi il vincolo che tale attributo può assumere solamente i valori {Piccolo, Medio, Grande}.
- Le classi **Speaker** e **Non-Speaker** ereditano gli attributi e le relazioni della classe **Partecipante**. Tuttavia **Non-Speaker** era già in relazione con **Evento**, quindi nessuna relazione viene aggiunta.

Diagramma E-R ristrutturato:



Schema tradotto in modello relazionale:

Evento (Nome, DataFine, DataInizio, SitoWeb)

Assegnazione (DataInizio, DataFine, Padiglione, Nome-Evento, PIVA_Espositore,)

- Assegnazione.Padiglione -> Padiglione.Codice
- Assegnazione.PIVA_Espositore -> Espositore.PIVA
- Assegnazione.Nome-Evento -> Evento.Nome

Espositore (PIVA, Nome, E-mail)

Padiglione (Codice, Dimensione)

Sessione (Titolo, Data, Padiglione, Evento)

- Sessione.Padiglione -> Padiglione.Codice
- Sessione.Evento -> Evento.Nome

Note:

- SitoWeb avrebbe cardinalità (0,1) (andrebbe l'asterisco)

- Da parte di Relazione si ha che

la chiave verso Sessione dovrebbe comunque

essere composta da Titolo e Data; forse viene

lasciato l'attributo singolo in virtù del caso relazionale

Esposizione che copre già il collegamento.

Io, in ogni caso, lo farei nel modo classico descritto

Speaker (Cognome, Email, Nome)

Iscrizione (Evento, SpeakerCognome, SpeakerEmail)

- Iscrizione.Evento -> Evento.Nome
- Iscrizione.SpeakerCognome -> Speaker.Cognome
- Iscrizione.SpeakerEmail -> Speaker.Email

Biglietto (Evento, Non-SpeakerCognome, Non-SpeakerEmail, Costo, Codice)

- Biglietto.Evento -> Evento.Nome
- Biglietto.(Non-SpeakerCognome, Non-SpeakerEmail) -> Non-Speaker.(Cognome,Email)²

Non-Speaker (Cognome, Email, Nome)

Relazione (Titolo, Data, Speaker, Sessione)

Esposizione (SpeakerCognome, SpeakerEmail, SessioneTitolo, SessioneData, Relazione)

- Esposizione.SpeakerCognome -> Speaker.Cognome
- Esposizione.SpeakerEmail -> Speaker.Email
- Esposizione.SessioneTitolo -> Sessione.Titolo
- Esposizione.SessioneData -> Sessione.Data
- Esposizione.Relazione -> Relazione.Titolo

² Notare che questa soluzione implica che ogni coppia (non-speaker-cognome,con-speaker-email) di ogni tupla di Biglietto deve essere presente come coppia (cognome,email) di 1+ tuple della tabella Non-Speaker.

Esercizio 3

Si vuole realizzare una base di dati che gestisca procedimenti sanzionatori nel contesto di rilevazioni statistiche ufficiali di carattere nazionale. Per alcune rilevazioni statistiche ufficiali esiste, infatti, l'obbligo di risposta da parte dei soggetti contattati per la conduzione delle rilevazioni. Qualora il soggetto contattato non risponda al questionario inviatogli, dopo un prefissato intervallo di tempo, ha inizio un procedimento sanzionatorio che consta di due fasi principali: invio della diffida al soggetto non rispondente e, qualora tale soggetto continui ad essere inadempiente (cioè non risponda al questionario), invio della sanzione che il soggetto stesso dovrà pagare. I soggetti possono essere persone fisiche o imprese. Delle persone fisiche interessa memorizzare il codice fiscale, delle imprese il codice fiscale o la partita iva in maniera alternativa. Inoltre è di interesse l'indirizzo cui il soggetto è contattabile. Si noti che le imprese possono prevedere delle unità locali, ovvero l'impresa si articola secondo una struttura che consiste di un'impresa centrale ed eventualmente di un insieme di imprese "periferiche". Un procedimento viene avviato in relazione alla non-risposta ad una specifica edizione di un'indagine. Ogni indagine è caratterizzata da un nome (es. Forze di lavoro), da una frequenza con cui le sue edizioni occorrono (es. trimestrale) e dalle specifiche edizioni che sono occorse (es. primo trimestre 2011).

Le edizioni, che hanno un codice univoco nell'ambito dell'indagine in cui sono svolte, hanno una data di inizio ed una data di fine che caratterizzano l'inizio e la fine della rilevazione sul campo dei dati oggetto dell'indagine. Nell'ambito di un procedimento è prodotto un insieme di documenti che costituisce il fascicolo del procedimento. Un fascicolo ha un codice che lo identifica nell'ambito del procedimento a cui è legato. I documenti, che dispongono di un ID univoco nell'ambito del fascicolo in cui sono redatti, sono rappresentati da un nome, un tipo, una data di produzione e dal path relativo al file cui sono associati. Del procedimento, oltre alle informazioni necessarie a desumere il suo avanzamento, interessa memorizzare la data di inizio e, qualora sia stato archiviato, l'esito della archiviazione (ad esempio archiviato perché il soggetto ha risposto) e la data di archiviazione.

Soluzione Progettazione Concettuale

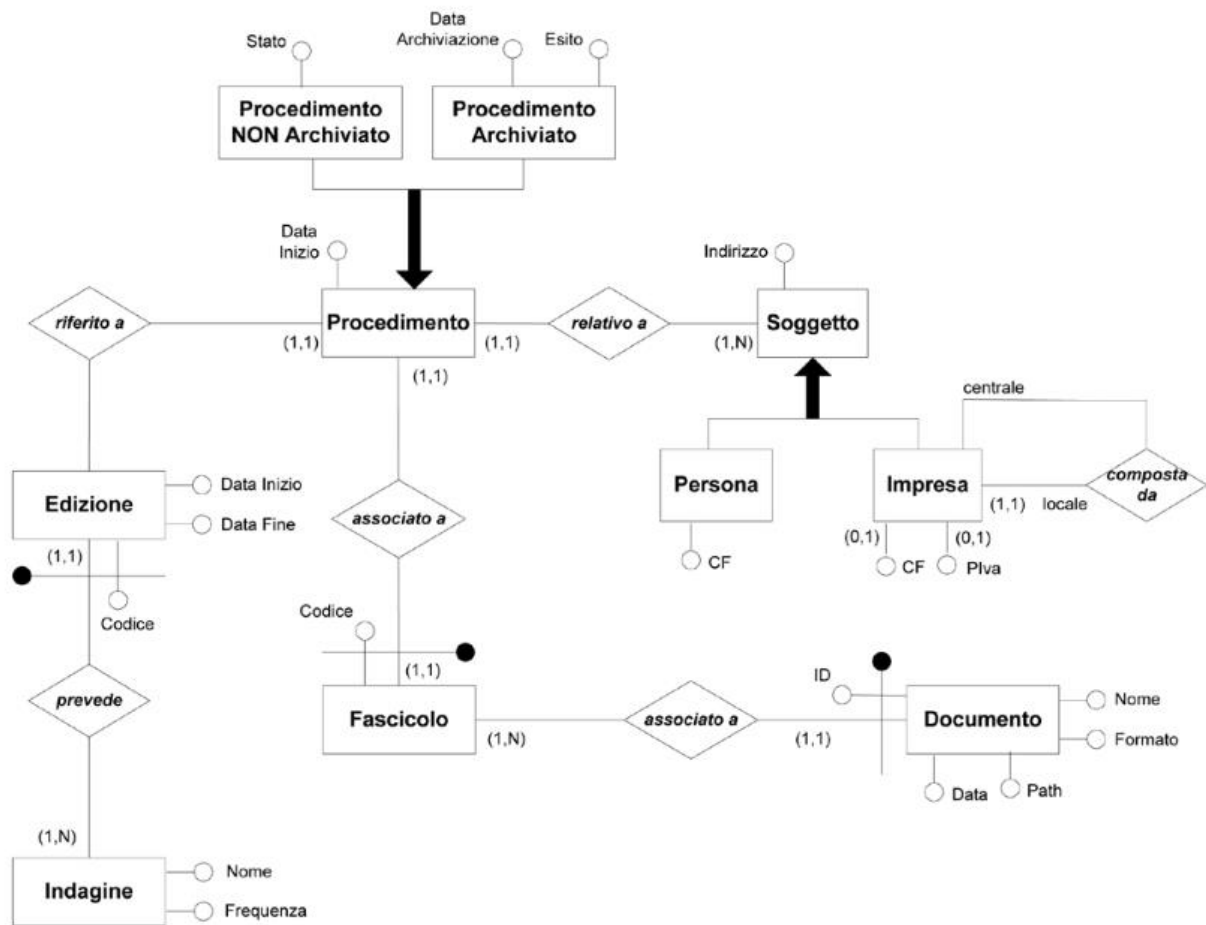
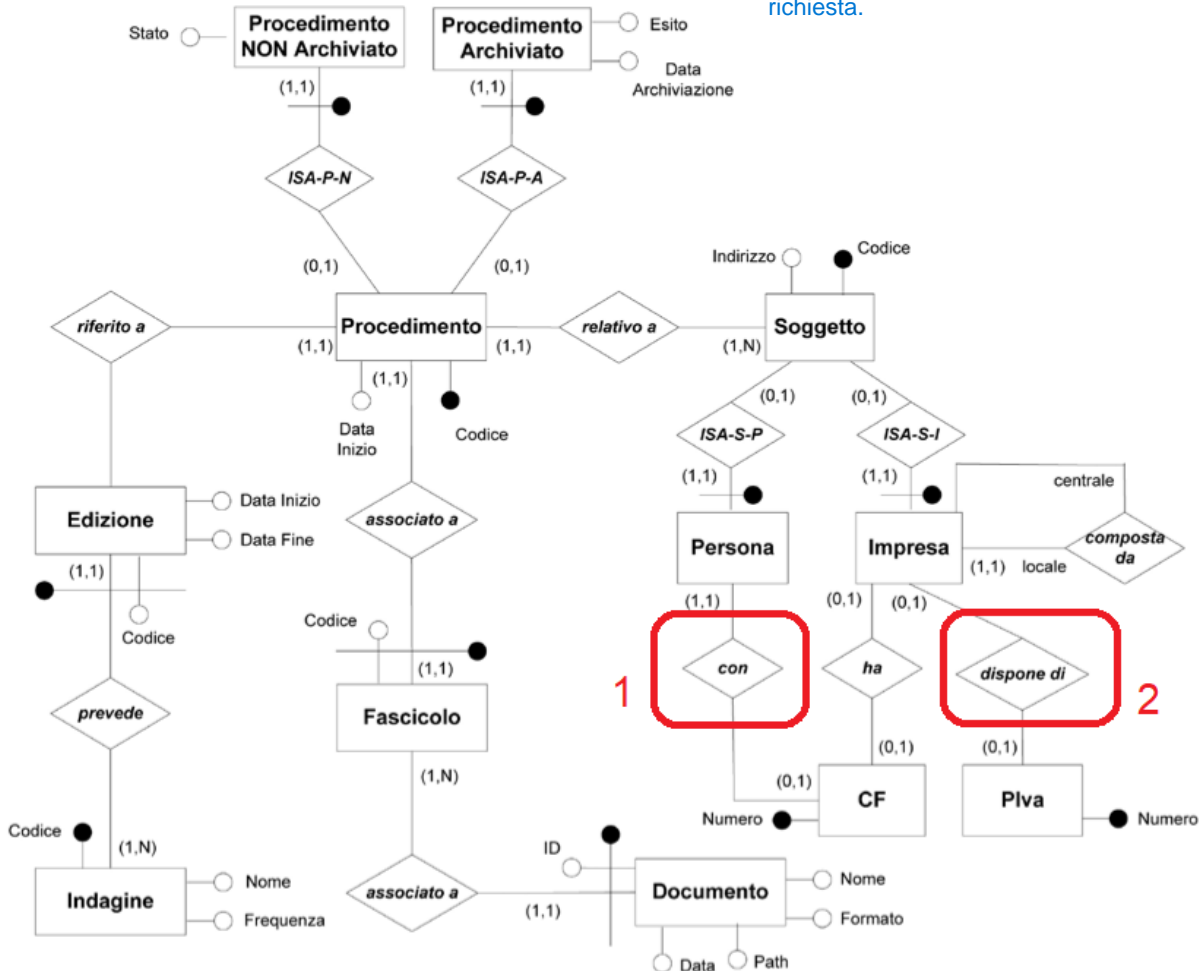


Diagramma E-R ristrutturato

"Informazioni necessarie a desumere l'avanzamento del procedimento" sembra sia l'unico motivo razionale per il quale è stato inserito l'attributo "Stato"; non sembra, infatti, comparire nel testo come esplicita richiesta.



Le scelte effettuate nella fase di ristrutturazione hanno avuto come obiettivo quello di evitare la presenza di valori NULL nella base di dati. Questo ha comportato una traduzione "ridondante" delle generalizzazioni, in cui sia l'entità padre che quelle figlie sono state mantenute nello schema, e una traduzione degli attributi con cardinalità (0,1) in nuove entità specifiche. Note che quest'ultimo aspetto ha comportato la modifica di uno dei vincoli esterni (quello identificato nel rettangolo num. 2) e l'aggiunta di un nuovo vincolo eterno (quello identificato nel rettangolo num. 1), rispetto a quelli definiti in fase di Progettazione Concettuale. Inoltre, dato che, per ogni entità, è necessario individuare principale, è stato introdotto un codice, i cui valori sono speciali ed hanno l'unico scopo di identificare le istanze dell'entità, per tutte quelle entità che naturalmente non ne dispongono.

Schema Logico:

Procedimento(Codice, Soggetto, Edizione, Indagine, DataInizio)

- Procedimento.Soggetto --> Soggetto.Codice)
- Procedimento.(Edizione,Indagine) --> Edizione.(Codice, Indagine)

ProcedimentoNonArchiviato(Codice,Stato)

- ProcedimentoNonArchiviato.Codice -->Procedimento.Codice)

ProcedimentoArchiviato(Codice,Esito,DataArchiviazione)

- ProcedimentoArchiviato.Codice -->Procedimento.Codice

Indagine(Codice,Nome,Frequenza)

Edizione(Codice,Indagine,DataInizio,DataFine)

- Edizione.Indagine --> Indagine.Codice

Fascicolo(Codice,Procedimento)

- Fascicolo.Procedimento --> Procedimento.Codice

Documento(ID,Fascicolo,Data,Path,Nome,Formato)

- Documento.Fascicolo -->Fascicolo.Codice

Soggetto(Codice,Indirizzo)

Persona(Codice,CF)

- Persona.Codice -->Soggetto(Codice)
- Persona.CF -->CF.Numero

Impresa (Codice,CodiceCentrale)

Impresa.Codice -->Soggetto.Codice

Impresa.CodiceCentrale --> Impresa.Codice

CF(Numero)

Plva(Numero)

ha(CF,Impresa)

- ha.CF --> CF.Numero
- ha.Impresa --> Impresa.Codice

ImpresadisponeDi(Plva, Impresa)

- disponeDi.Plva --> Plva(Numero)
- disponeDi.Impresa --> Impresa.Codice

**Laurea in Informatica
A.A. 2021-2022**

Corso "Base di Dati"

Esercitazione Normalizzazione

Prof. Massimiliano de Leoni



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

Data la relazione $R(A,B,C,D)$ con dipendenze funzionali $\{ C \rightarrow D, C \rightarrow A, B \rightarrow C \}$.

1. Mostrare tutte le chiavi di R e motivare perché ognuna è chiave.
2. Dire quale dipendenze violano la forma normale di Boyce Codd (BCNF), spiegandone la ragione.
3. Decomporre in BCNF

Data la relazione $R(A,B,C,D)$ con dipendenze funzionali $\{ C \rightarrow D, C \rightarrow A, B \rightarrow C \}$.

1. Mostrare tutte le chiavi di R e motivare perché ognuna è chiave.

$B^+ = \{B, C, A, D\}$ e $C^+ = \{C, A, D\}$.

Quindi, B è chiave perché la sua chiusura contiene tutti gli attributi della relazione

2. Dire quale dipendenze violano la forma normale di Boyce Codd (BCNF), spiegandone la ragione.

3. Decomporre in BCNF

Data la relazione $R(A,B,C,D)$ con dipendenze funzionali $\{ C \rightarrow D, C \rightarrow A, B \rightarrow C \}$.

B è chiave.

2. Dire quale dipendenze violano la forma normale di Boyce Codd (BCNF), spiegandone la ragione.

$C \rightarrow D$ e $C \rightarrow A$ violano BCNF perché C non è una superchiave della relazione

3. Decomporre in BCNF

Data la relazione $R(A,B,C,D)$ con dipendenze funzionali $\{ C \rightarrow D, C \rightarrow A, B \rightarrow C \}$.

B è chiave.

2. Dire quale dipendenze violano la forma normale di Boyce Codd (BCNF), spiegandone la ragione.

3. Decomporre in BCNF

Usando $C \rightarrow D$, si ottiene $R_1(\underline{C}, D)$ e, togliendo D da R , si ottiene $\underline{R}(A, B, C)$

Usando $C \rightarrow A$, si ottiene $R_2(\underline{C}, A)$ e, togliendo A da \underline{R} , si ottiene $R_3(\underline{B}, C)$

Quindi si ottengono tre relazioni: $R_1(\underline{C}, D)$, $R_2(\underline{C}, A)$ e $R_3(\underline{B}, C)$.

È anche possibile ricomporre R_1 e R_2 ottenendo $R_4(\underline{C}, A, D)$ e $R_3(\underline{B}, C)$.

Considerare uno schema di relazione R ($E, N, L, C, S, D, M, P, A$) con le seguenti dipendenze funzionali:

$E \rightarrow NS$,

$NL \rightarrow EMD$,

$EN \rightarrow LCD$,

$C \rightarrow S$,

$D \rightarrow M$,

$M \rightarrow D$,

$EPD \rightarrow A$,

$NLCP \rightarrow A$.

Calcolare una **copertura ridotta** per tale insieme e decomporre la relazione in **terza forma normale**.

Una relazione R con chiavi K_1, \dots, K_n è in Terza Forma Normale se:

Per ogni dipendenza funzionale non banale $X \rightarrow Y$, almeno una delle seguenti condizioni sono valide:

- X è superchiave (BCNF)
- ogni attributo in Y è contenuto in almeno una tra le chiavi K_1, \dots, K_n .

- Un insieme di dipendenze F è una copertura ridotta:
 - **non ridondante** se non esiste dipendenza $f \in F$ tale che $F - \{f\}$ implica f ;
 - **ridotto** se
 - **non ridondante** se non esiste dipendenza $f \in F$ tale che $F - \{f\}$ implica f ;
 - non esiste un insieme F' equivalente a F ottenuto eliminando attributi dai primi membri di una o più dipendenze di F .
- Esempio (parte in rosso rimovibile):
 - $\{A \rightarrow B; AB \rightarrow C; A \rightarrow C\}$ è ridondante;
 - $\{A \rightarrow B; AB \rightarrow C\}$ non è ridondante né ridotto;
 - $\{A \rightarrow B; A \rightarrow C\}$ è ridotto

I passi per calcolare la copertura ridotta di una relazione sono i seguenti:

1. Sostituzione dell'insieme dato con quello equivalente che ha tutti i secondi membri costituiti da singoli attributi;
2. Per ogni dipendenza verifica dell'esistenza di attributi eliminabili dal primo membro;
3. Eliminazione delle dipendenze ridondanti.

Esercizio 2 (passo 1)



1. Sostituzione dell'insieme dato con quello equivalente che ha tutti i secondi membri costituiti da singoli attributi

$E \rightarrow NS$

$NL \rightarrow EMD$

$EN \rightarrow LCD$

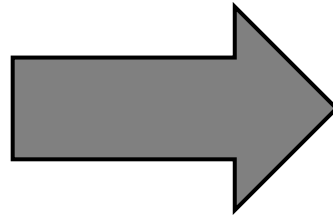
$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$EPD \rightarrow A$

$NLCP \rightarrow A$



Esercizio 2 (passo 1)



1. Sostituzione dell'insieme dato con quello equivalente che ha tutti i secondi membri costituiti da singoli attributi

$E \rightarrow NS$

$NL \rightarrow EMD$

$EN \rightarrow LCD$

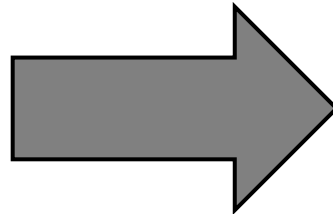
$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$EPD \rightarrow A$

$NLCP \rightarrow A$



$E \rightarrow S$

$E \rightarrow N$

$NL \rightarrow E$

$NL \rightarrow M$

$NL \rightarrow D$

$EN \rightarrow L$

$EN \rightarrow C$

$EN \rightarrow D$

$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$EPD \rightarrow A$

$NLCP \rightarrow A$

Esercizio 2 (passo 2)



2. Per ogni dipendenza verifica dell'esistenza di attributi eliminabili dal primo membro

$E \rightarrow S$

$E \rightarrow N$

$NL \rightarrow E$

$NL \rightarrow M$

$NL \rightarrow D$

$EN \rightarrow L$

$EN \rightarrow C$

$EN \rightarrow D$

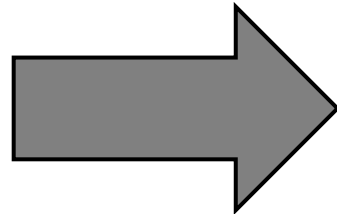
$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$EPD \rightarrow A$

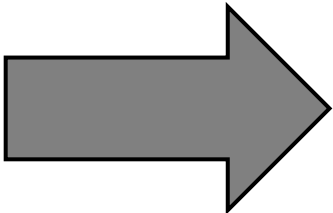
$NLCP \rightarrow A$



Esercizio 2 (passo 2)



2. Per ogni dipendenza verifica dell'esistenza di attributi eliminabili dal primo membro

$E \rightarrow S$		$E \rightarrow S$	
$E \rightarrow N$		$E \rightarrow N$	
$NL \rightarrow E$		$NL \rightarrow E$	
$NL \rightarrow M$		$NL \rightarrow M$	
$NL \rightarrow D$		$NL \rightarrow D$	
$EN \rightarrow L$		$E \rightarrow L$	$(EN \rightarrow L, E \rightarrow N)$
$EN \rightarrow C$		$E \rightarrow C$	$(EN \rightarrow C, E \rightarrow N)$
$EN \rightarrow D$		$E \rightarrow D$	$(EN \rightarrow D, E \rightarrow N)$
$C \rightarrow S$		$C \rightarrow S$	
$D \rightarrow M$		$D \rightarrow M$	
$M \rightarrow D$		$M \rightarrow D$	
$EPD \rightarrow A$		$EP \rightarrow A$	$(EPD \rightarrow A, E \rightarrow D)$
$NLCP \rightarrow A$		$NLP \rightarrow A$	$(NLCP \rightarrow A, NL \rightarrow E, E \rightarrow C)$

3. Eliminazione delle dipendenze ridondanti

$E \rightarrow S$

$E \rightarrow N$

$NL \rightarrow E$

$NL \rightarrow M$

$NL \rightarrow D$

$E \rightarrow L$

$E \rightarrow C$

$E \rightarrow D$

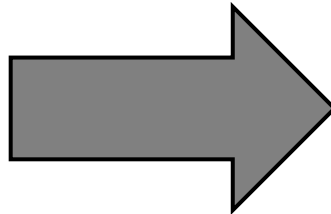
$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$EP \rightarrow A$

$NLP \rightarrow A$



Esercizio 2 (passo 3)



3. Eliminazione delle dipendenze ridondanti

~~E~~ → S

E → N

NL → E

~~NL~~ → M

~~NL~~ → D

E → L

E → C

E → D

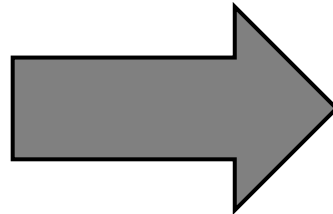
C → S

D → M

M → D

~~EP~~ → A

NLP → A



E → N

NL → E

E → L

E → C

E → D

C → S

D → M

M → D

NLP → A

Ho ottenuto una copertura ridotta

$E \rightarrow N$

$NL \rightarrow E$

$E \rightarrow L$

$E \rightarrow C$

$E \rightarrow D$

$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$NLP \rightarrow A$

Non abbiamo le chiavi → Occorre Individuare le chiavi partendo dalla copertura ridotta

$E \rightarrow N$

$NL \rightarrow E$

$E \rightarrow L$

$E \rightarrow C$

$E \rightarrow D$

$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$NLP \rightarrow A$

Eseguo le chiusure dei primi membri

E^+

NL^+

C^+

D^+

M^+

NLP^+

Non abbiamo le chiavi → Occorre Individuare le chiavi partendo dalla copertura ridotta

E → N

NL → E

E → L

E → C

E → D

C → S

D → M

M → D

NLP → A

Eseguo le chiusure dei primi membri

$E^+ = \{E, N, L, D, C, S, M\}$ Non chiave
perchè mancano A, P

$NL^+ = \{E, N, L, D, C, S, M\}$ Non chiave
perchè mancano A, P

$C^+ = \{C, S\}$ Non chiave

$D^+ = \{D, M\}$ Non chiave

$M^+ = \{M, D\}$ Non chiave

$NLP^+ = \{N, L, P, A, E, D, C, S, M\}$ è chiave

Non abbiamo le chiavi → Occorre Individuare le chiavi partendo dalla copertura ridotta

E → N

NL → E

E → L

E → C

E → D

C → S

D → M

M → D

NLP → A

Eseguo le chiusure dei primi membri

$E^+ = \{E, N, L, D, C, S, M\}$ Non chiave
perchè mancano A, P

$NL^+ = \{E, N, L, D, C, S, M\}$ Non chiave
perchè mancano A, P

$C^+ = \{C, S\}$ Non chiave

$D^+ = \{D, M\}$ Non chiave

$M^+ = \{M, D\}$ Non chiave

$NLP^+ = \{N, L, P, A, E, D, C, S, M\}$ è chiave

Individuare le chiavi partendo dalla copertura ridotta

Eseguo le chiusure dei primi membri

$E \rightarrow N$

$NL \rightarrow E$

$E \rightarrow L$

$E \rightarrow C$

$E \rightarrow D$

$C \rightarrow S$

$D \rightarrow M$

$M \rightarrow D$

$NLP \rightarrow A$

$NLP^+ = \{N, L, P, A, E, D, C, S, M\}$ è **chiave**

$EP^+ = \{E, N, L, D, C, S, M, P, A\}$ anche questa è chiave!

Ho ottenuto una copertura ridotta

$E \rightarrow N$
 $NL \rightarrow E$
 $E \rightarrow L$
 $E \rightarrow C$
 $E \rightarrow D$
 $C \rightarrow S$
 $D \rightarrow M$
 $M \rightarrow D$
 $NLP \rightarrow A$

Chiavi **NLP, EP**

Sintesi di schema in
terza forma normale

Dati uno schema $R(U)$ e un insieme di dipendenze F su U , con chiavi K_1, \dots, K_n

1. Viene calcolata una copertura ridotta G di F
2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$
3. Viene costruita una relazione per ogni sotto-insieme
4. Se esistono due relazioni $S(X)$ e $T(Y)$ con $X \subseteq Y$, S viene eliminata
5. Se, per qualche i , non esiste una relazione $S(X)$ con $K_i \subseteq X$, viene aggiunta una relazione $T(K_i)$

2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$

$E \rightarrow N$

$E^+ = \{E, N, L, D, C, S, M\}$ Non chiave
perchè mancano A, P

$NL \rightarrow E$

$E \rightarrow L$

$NL^+ = \{E, N, L, D, C, S, M\}$ Non chiave
perchè mancano A, P

$E \rightarrow C$

$E \rightarrow D$

$C \rightarrow S$

$C^+ = \{C, S\}$ Non chiave

$D \rightarrow M$

$D^+ = \{D, M\}$ Non chiave

$M \rightarrow D$

$NLP \rightarrow A$

$M^+ = \{M, D\}$ Non chiave

$NLP^+ = \{N, L, P, A, E, D, C, S, M\}$ è chiave

2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$

$E \rightarrow N$	}	$E^+ = \{ E, N, L, C, D, S, M \}$
$E \rightarrow L$		
$E \rightarrow C$		
$E \rightarrow D$		
$NL \rightarrow E$	}	$NL^+ = \{ E, N, L, C, D, S, M \}$
$C \rightarrow S$	}	$CS^+ = \{ C, S \}$
$D \rightarrow M$	}	$D^+ = \{ D, M \} ; M^+ = \{ D, M \}$
$M \rightarrow D$		
$NLP \rightarrow A$	}	$NLP^+ = \{ E, N, L, C, D, C, S, M, A \}$

2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$

$E \rightarrow N$
 $E \rightarrow L$
 $E \rightarrow C$
 $E \rightarrow D$

$$E^+ = \{ E, N, L, C, D, S, M \}$$

CHIUSURE COINCIDONO

$NL \rightarrow E$

$$NL^+ = \{ E, N, L, C, D, S, M \}$$

$C \rightarrow S$

$$C^+ = \{ C, S \}$$

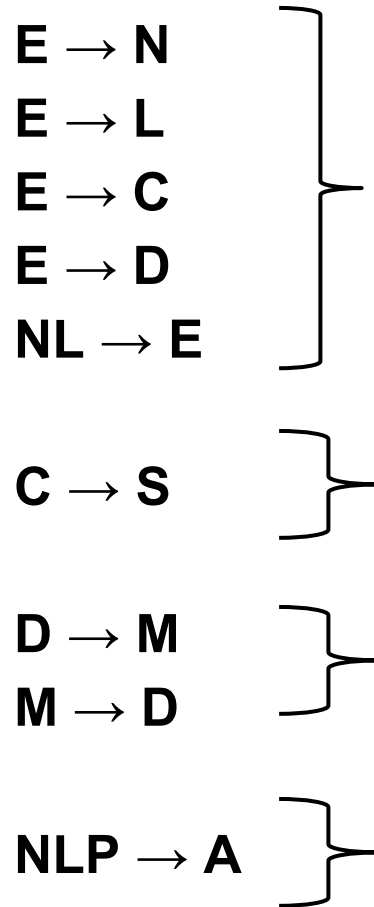
$D \rightarrow M$
 $M \rightarrow D$

$$D^+ = \{ D, M \} ; M^+ = \{ D, M \}$$

$NLP \rightarrow A$

$$NLP^+ = \{ E, N, L, C, D, C, S, M, A \}$$

3. Viene costruita una relazione per ogni sotto-insieme



3. Viene costruita una relazione per ogni sotto-insieme

$$\left. \begin{array}{l} E \rightarrow N \\ E \rightarrow L \\ E \rightarrow C \\ E \rightarrow D \\ NL \rightarrow E \end{array} \right\} R_1 (E, N, L, C, D)$$

$$C \rightarrow S \left\} R_2 (C, S)$$

$$\left. \begin{array}{l} D \rightarrow M \\ M \rightarrow D \end{array} \right\} R_3 (D, M)$$

$$NLP \rightarrow A \left\} R_4 (N, L, P, A)$$

4. Se esistono due relazioni $S(X)$ e $T(Y)$ con $X \subseteq Y$, S viene eliminata

$E \rightarrow N$	{	$R_1 (E, N, L, C, D)$
$E \rightarrow L$		
$E \rightarrow C$		
$E \rightarrow D$		
$NL \rightarrow E$		

$C \rightarrow S$	}	$R_2 (C, S)$
-------------------	---	--------------

$D \rightarrow M$	{	$R_3 (D, M)$
$M \rightarrow D$		

$NLP \rightarrow A$	}	$R_4 (N, L, P, A)$
---------------------	---	--------------------

Non accade

5. Se, per qualche i , non esiste una relazione $S(X)$ con $K_i \subseteq X$, viene aggiunta una relazione $T(K_i)$ **Chiavi NLP, EP**

$E \rightarrow N$
 $E \rightarrow L$
 $E \rightarrow C$
 $E \rightarrow D$
 $NL \rightarrow E$ } $R_1 (E, N, L, C, D)$

$C \rightarrow S$ } $R_2 (C, S)$

$D \rightarrow M$
 $M \rightarrow D$ } $R_3 (D, M)$

$NLP \rightarrow A$ } $R_4 (N, L, P, A)$

**Si aggiunge
 $R_5 (E, P)$**

Esercizio 2



Le chiavi delle relazioni sono sottolineate:

$E \rightarrow N$
 $E \rightarrow L$
 $E \rightarrow C$
 $E \rightarrow D$
 $NL \rightarrow E$

} $R_1 (\underline{E}, \underline{N}, L, C, D)$

$C \rightarrow S$

} $R_2 (\underline{C}, S)$

$D \rightarrow M$
 $M \rightarrow D$

} $R_3 (\underline{D}, \underline{M})$

$NLP \rightarrow A$

} $R_4 (\underline{N}, \underline{L}, \underline{P}, A)$

$R_5 (\underline{E}, \underline{P})$

Dato schema $R(A, B, C, D, E, F)$ con dipendenze:
 $CE \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow BE$, $B \rightarrow F$, $AD \rightarrow CF$

1. Trovare copertura ridotta G
2. Trovare tutte le chiavi
3. Dire se ci sono e quali dipendenze violano 3NF
4. Normalizzare lo schema in 3NF
5. Lo schema normalizzato al punto 4 è anche in BCNF?

Dato schema $R(A, B, C, D, E, F)$ con dipendenze:
 $CE \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow BE$, $B \rightarrow F$, $AD \rightarrow CF$

1. Trovare copertura ridotta G
 $C \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow B$, $D \rightarrow E$, $B \rightarrow F$, $AD \rightarrow C$
2. Trovare tutte le chiavi
3. Dire se ci sono e quali dipendenze violano 3NF
4. Normalizzare lo schema in 3NF

Dato schema $R(A, B, C, D, E, F)$ con dipendenze:
 $CE \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow BE$, $B \rightarrow F$, $AD \rightarrow CF$

1. Trovare copertura ridotta G
 $C \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow B$, $D \rightarrow E$, $B \rightarrow F$, $AD \rightarrow C$
2. Trovare tutte le chiavi
 AD , C
3. Dire se ci sono e quali dipendenze violano 3NF
4. Normalizzare lo schema in 3NF
5. Indicare e spiegare se o meno la normalizzazione produce uno schema anche in BCNF

Dato schema $R(A, B, C, D, E, F)$ con dipendenze:
 $CE \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow BE$, $B \rightarrow F$, $AD \rightarrow CF$

1. Trovare copertura ridotta G
 $C \rightarrow A$, $C \rightarrow D$, $A \rightarrow B$, $D \rightarrow B$, $D \rightarrow E$, $B \rightarrow F$, $AD \rightarrow C$
2. Trovare tutte le chiavi
 AD , C
3. Dire se ci sono (ed eventualmente quali) dipendenze che violano 3NF. **Ricordiamo** che 3NF richiede che per ogni FD $X \rightarrow Y$ sia soddisfatto:
 1. X contiene chiave K di r
oppure
 2. ogni attributo in Y è contenuto in almeno una chiave di r
4. Normalizzare lo schema in 3NF
5. Indicare e spiegare se o meno la normalizzazione produce uno schema anche in BCNF

Ho:

$C \rightarrow A,$
 $C \rightarrow D,$
 $A \rightarrow B,$
 $D \rightarrow B,$
 $D \rightarrow E,$
 $B \rightarrow F,$
 $AD \rightarrow C$

Con chiavi: AD, C

Dire se ci sono e quali dipendenze violano 3NF, per ogni FD $X \rightarrow Y$:

- X contiene chiave K di r
- ogni attributo in Y è contenuto in almeno una chiave di r

Ho:

$C \rightarrow A$, non viola 3NF

$C \rightarrow D$, non viola 3NF

$A \rightarrow B$, viola (A non super chiave e B non presente in chiave)

$D \rightarrow B$, viola (D non super chiave e B non presente in chiave)

$D \rightarrow E$, viola (D non super chiave e E non presente in chiave)

$B \rightarrow F$, viola (B non super chiave e F non presente in chiave)

$AD \rightarrow C$ non viola 3NF

Con chiavi: AD, C

Non è in 3NF

4. Normalizzare lo schema in 3NF

$C \rightarrow A,$
 $C \rightarrow D,$
 $A \rightarrow B,$
 $D \rightarrow B,$
 $D \rightarrow E,$
 $B \rightarrow F,$
 $AD \rightarrow C$

Con chiavi: AD, C

4. Normalizzare lo schema in 3NF

Calcolo la chiusura dei primi membri:

$C \rightarrow A,$ $C^+ = \{ A, B, C, D, E, F \}$

$C \rightarrow D,$

$A \rightarrow B,$ $A^+ = \{ A, B, F \}$

$D \rightarrow B,$ $D^+ = \{ D, B, F, E \}$

$D \rightarrow E,$

$B \rightarrow F,$ $B^+ = \{ B, F \}$

$AD \rightarrow C$ $AD^+ = \{ A, B, C, D, E, F \}$

4. Normalizzare lo schema in 3NF

$C \rightarrow A,$
 $C \rightarrow D,$

$C^+ = \{ \mathbf{A, B, C, D, E, F} \}$

$A \rightarrow B,$

$A^+ = \{ A, B, F \}$

$D \rightarrow B,$
 $D \rightarrow E,$

$D^+ = \{ D, B, F, E \}$

$B \rightarrow F,$

$B^+ = \{ B, F \}$

$AD \rightarrow C$

$AD^+ = \{ \mathbf{A, B, C, D, E, F} \}$

Le chiusure di C e AD coincidono, quindi vanno considerati nella stessa partizione

4. Normalizzare lo schema in 3NF

$C \rightarrow A,$
 $C \rightarrow D,$
 $AD \rightarrow C$

R1 (C, A, D)

key C, AD

$A \rightarrow B,$

R2 (A, B)

key A

$D \rightarrow B,$
 $D \rightarrow E,$

R3 (B, E, D)

key D

$B \rightarrow F,$

R4 (B, F)

key B

4. Normalizzare lo schema in 3NF

$C \rightarrow A,$
 $C \rightarrow D,$
 $AD \rightarrow C$

R1 (C, A, D)

chiavi C, AD

$A \rightarrow B,$

R2 (A, B)

chiave A

$D \rightarrow B,$
 $D \rightarrow E,$

R3 (B, E, D)

chiave D

$B \rightarrow F,$

R4 (B, F)

chiave B

5. Indicare e spiegare se o meno la normalizzazione produce uno schema anche in BCNF

$C \rightarrow A,$ $C \rightarrow D,$ $AD \rightarrow C$	R1 (C, A, D)	chiavi C, AD
--	--------------	--------------

$A \rightarrow B,$	R2 (A, B)	chiave A
--------------------	-----------	----------

$D \rightarrow B,$ $D \rightarrow E,$	R3 (B, E, D)	chiave D
--	--------------	----------

$B \rightarrow F,$	R4 (B, F)	chiave B
--------------------	-----------	----------

Tutte le dipendenze funzionali non violano BCNF:

- $C \rightarrow A$ e $C \rightarrow D$ si applicano su R1 dove C è chiave
- $AD \rightarrow C$ si applica su R1 dove AD è chiave.
- $A \rightarrow B$ si applica su R2 dove A è chiave
- ...

Dato lo schema R (A, B, C, D, E, F, G)

Con dipendenze:

$AF \rightarrow BE$, $EF \rightarrow BCD$, $A \rightarrow F$, $B \rightarrow C$

1. Trovare copertura ridotta
2. Trovare tutte le altre chiavi, in aggiunta alla chiave primaria data
3. Dire se ci sono e quali dipendenze violano 3NF
4. Normalizzare in 3NF

R (A, B, C, D, E, F, G)

con:

$AF \rightarrow BE$, $EF \rightarrow BCD$, $A \rightarrow F$, $B \rightarrow C$

1. Trovare copertura ridotta

Calcolo chiusure transitive:

$A^+ = \{A, B, E, F, D, C\}$, $B^+ = \{B, C\}$, $EF^+ = \{B, D, C, E, F\}$,

Otteniamo quindi:

$A \rightarrow B$, $A \rightarrow E$, $EF \rightarrow B$, $EF \rightarrow D$, $A \rightarrow F$, $B \rightarrow C$

2. Trovare tutte le chiavi

3. Dire se ci sono e quali dipendenze violano 3NF

4. Normalizzare in 3NF

R (A, B, C, D, E, F, G)

con:

$AF \rightarrow BE$, $EF \rightarrow BCD$, $A \rightarrow F$, $B \rightarrow C$

1. Trovare copertura ridotta

$A \rightarrow B$, $A \rightarrow E$, $EF \rightarrow B$, $EF \rightarrow D$, $A \rightarrow F$, $B \rightarrow C$

2. Trovare tutte le chiavi

A^+ contiene tutti gli attributi, inoltre G è già una chiave

Quindi le chiavi sono: A, G

3. Dire se ci sono e quali dipendenze violano 3NF

4. Normalizzare in 3NF

R (A, B, C, D, E, F, G)

con:

$AF \rightarrow BE$, $EF \rightarrow BCD$, $A \rightarrow F$, $B \rightarrow C$

1. Trovare copertura ridotta

$A \rightarrow B$, $A \rightarrow E$, $EF \rightarrow B$, $EF \rightarrow D$, $A \rightarrow F$, $B \rightarrow C$

2. Trovare tutte le chiavi

A, G

3. Dire se ci sono e quali dipendenze violano 3NF. **Ricordiamo** che 3NF richiede che per ogni FD $X \rightarrow Y$ sia soddisfatto:
 1. X contiene chiave K di r
oppure
 2. ogni attributo in Y è contenuto in almeno una chiave di r
4. Normalizzare in 3NF

R (A, B, C, D, E, F, G)

con:

$A \rightarrow B$,

$A \rightarrow E$,

$EF \rightarrow B$,

$EF \rightarrow D$,

$A \rightarrow F$,

$B \rightarrow C$

Con chiavi: A, G

Dire se ci sono e quali dipendenze violano 3NF, per ogni FD $X \rightarrow Y$:

- X contiene chiave K di r
- ogni attributo in Y è contenuto in almeno una chiave di r

Esercizio 4



R (A, B, C, D, E, F, G)

con:

$A \rightarrow B$, non viola 3NF

$A \rightarrow E$, non viola 3NF

$EF \rightarrow B$, viola (EF non super chiave e B non presente in chiave)

$EF \rightarrow D$, viola (EF non super chiave e D non presente in chiave)

$A \rightarrow F$, non viola 3NF

$B \rightarrow C$ viola (B non super chiave e C non presente in chiave)

Con chiavi: A, G

Non è in 3NF

$R(A, B, C, D, E, F, G)$

4. Normalizzare in 3NF

1. Copertura Ridotta D di F:

$A \rightarrow B, A \rightarrow E, EF \rightarrow B, EF \rightarrow D, A \rightarrow F, B \rightarrow C$

2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$

$\{A \rightarrow B, A \rightarrow E, A \rightarrow F\}, \{EF \rightarrow B, EF \rightarrow D\}, \{B \rightarrow C\}$

3. Viene costruita una relazione per ogni sotto insieme:

Abbiamo una chiave per relazione:

$R_1(\underline{A}, B, E, F), R_2(\underline{E}, \underline{F}, B, D), R_3(\underline{B}, C)$

$R(A, B, C, D, E, F, G)$

4. Normalizzare in 3NF

3. Viene costruita una relazione per ogni sotto insieme:

$R_1(\underline{A}, B, E, F), R_2(\underline{E}, \underline{F}, B, D), R_3(\underline{B}, C)$

4. Se esistono due relazioni $S(X)$ e $T(Y)$ con $X \subseteq Y$, S viene eliminata

Non accade, quindi stesse relazioni.

$R_1(A, B, E, F), R_2(E, F, B, D), R_3(B, C)$

$R(A, B, C, D, E, F, G)$

4. Normalizzare in 3NF

4. Se esistono due relazioni $S(X)$ e $T(Y)$ con $X \subseteq Y$, S viene eliminata

Non accade, quindi stesse relazioni.

$R_1(A, B, E, F)$, $R_2(E, F, B, D)$, $R_3(B, C)$

5. Se, per qualche i , non esiste una relazione $S(X)$ con $K_i \subseteq X$, viene aggiunta una relazione $T(K_i)$

Nessuna relazione contiene G , aggiungiamo una relazione

$R_1(A, B, E, F)$, $R_2(E, F, B, D)$, $R_3(B, C)$, **$R_4(G)$**

**Laurea in Informatica
A.A. 2021-2022**

Corso "Base di Dati"

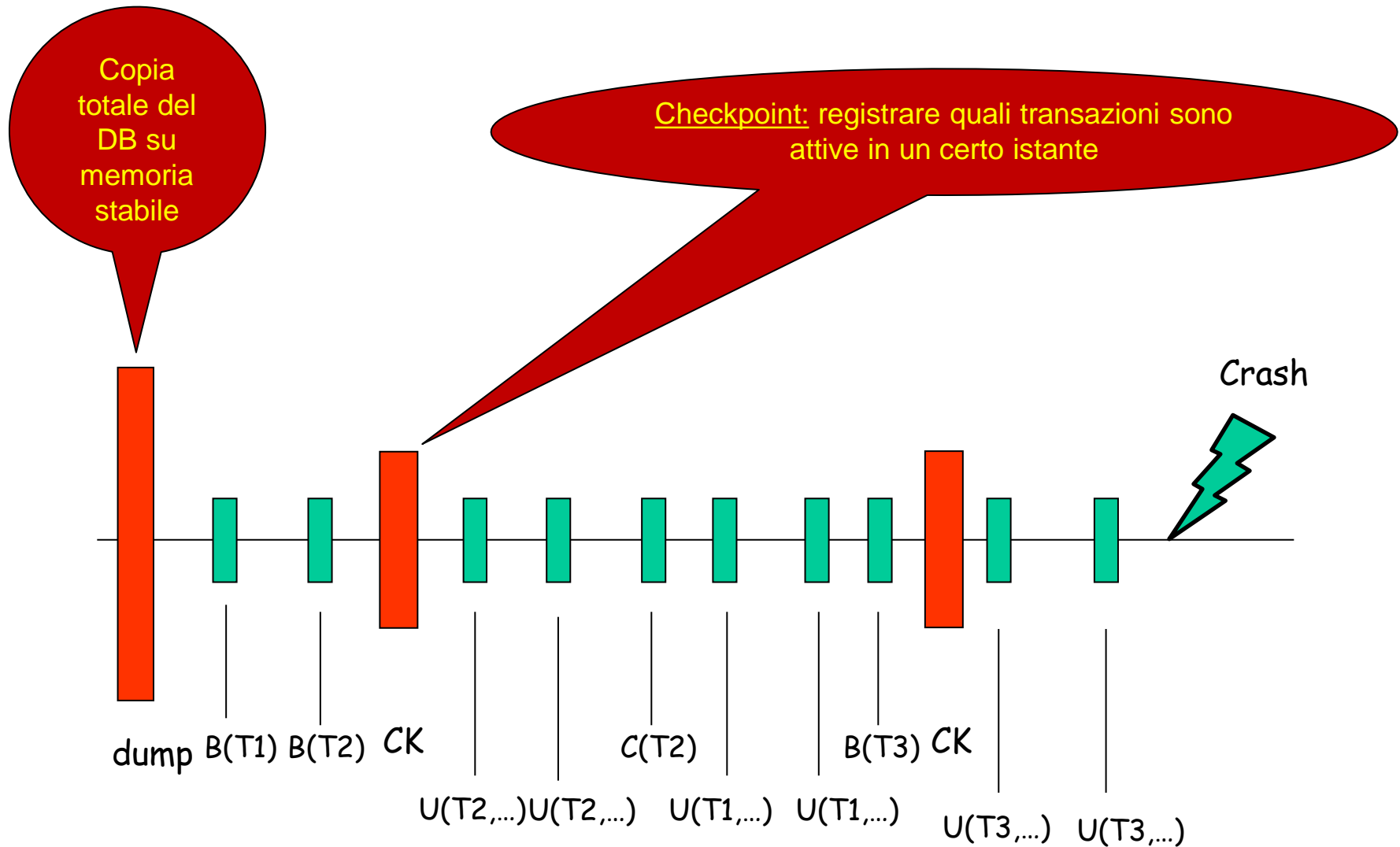
Esercitazione Transazioni

Prof. Massimiliano de Leoni



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

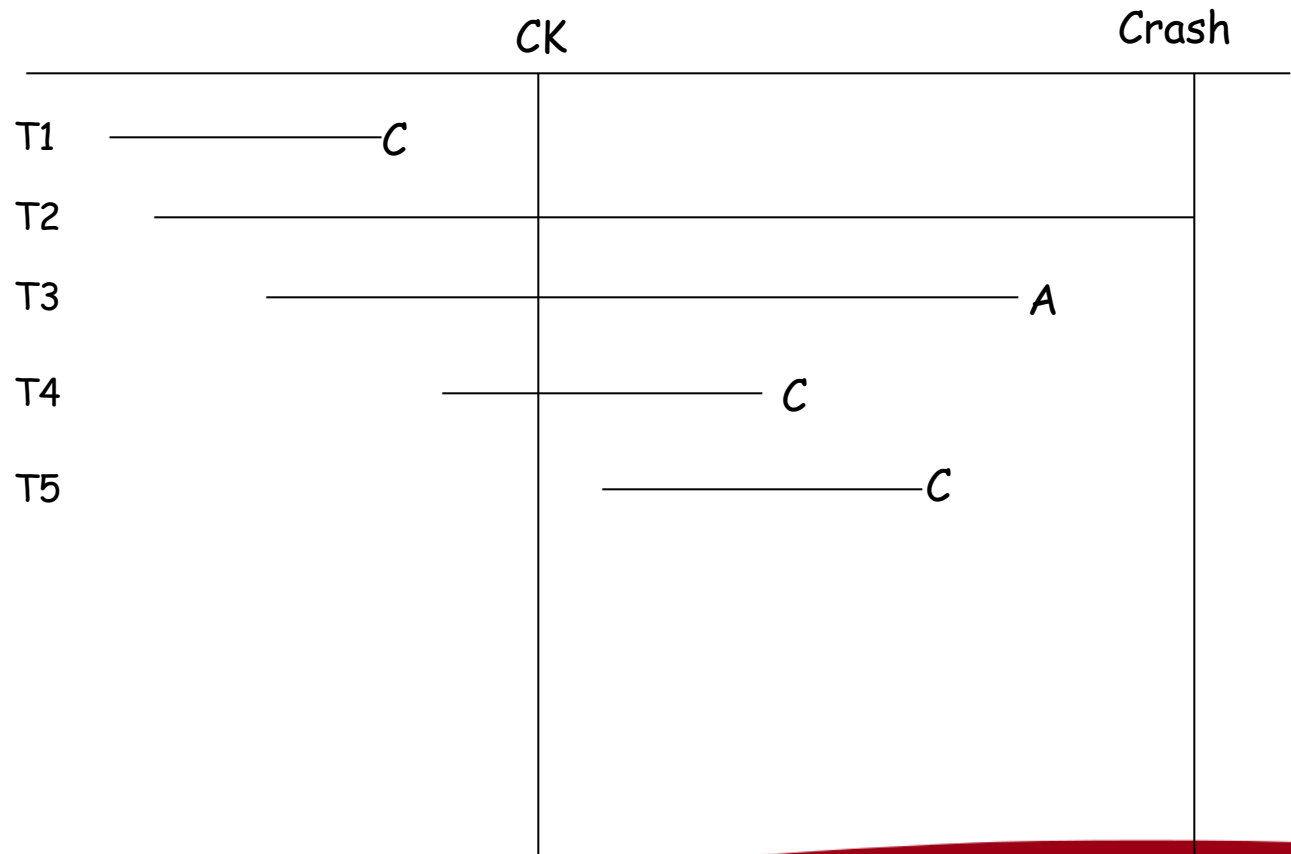
Log per guasti dispositivo



Esempio di warm restart



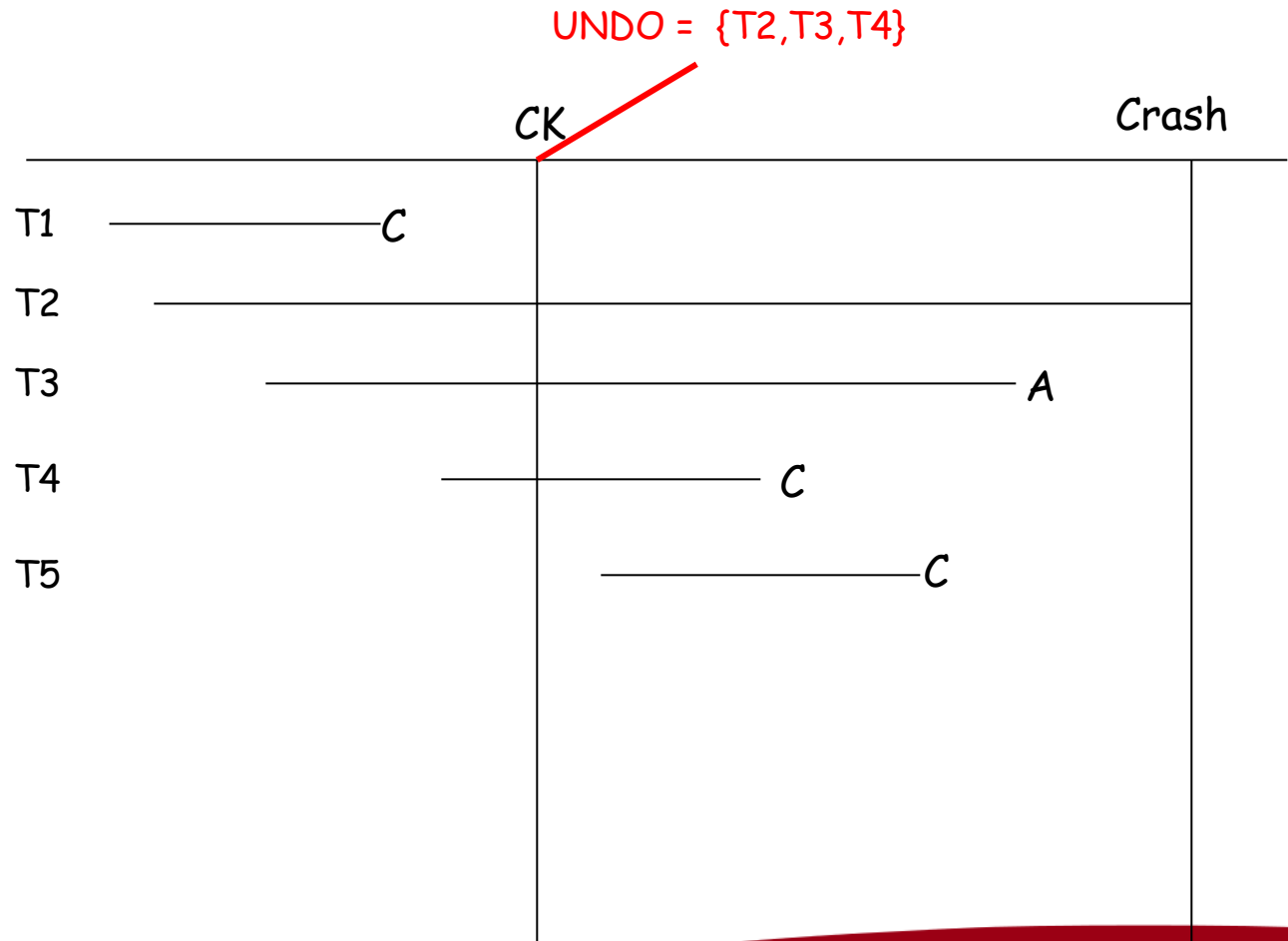
B(T1)
 B(T2)
 U(T2, O1, B1, A1)
 I(T1, O2, A2)
 B(T3)
 C(T1)
 B(T4)
 U(T3, O2, B3, A3)
 U(T4, O3, B4, A4)
 CK(T2, T3, T4)
 C(T4)
 B(T5)
 U(T3, O3, B5, A5)
 U(T5, O4, B6, A6)
 D(T3, O5, B7)
 A(T3)
 C(T5)
 I(T2, O6, A8)



1. Ricerca dell'ultimo checkpoint



B(T1)
B(T2)
U(T2, O1, B1, A1)
I(T1, O2, A2)
B(T3)
C(T1)
B(T4)
U(T3, O2, B3, A3)
U(T4, O3, B4, A4)
CK(T2, T3, T4)
C(T4)
B(T5)
U(T3, O3, B5, A5)
U(T5, O4, B6, A6)
D(T3, O5, B7)
A(T3)
C(T5)
I(T2, O6, A8)



2. Costruzione degli insiemi UNDO e REDO



- | | |
|----------------------|--|
| B(T1) | 0. UNDO = {T2,T3,T4}. REDO = {} |
| B(T2) | |
| 8. U(T2, O1, B1, A1) | 1. C(T4) → UNDO = {T2, T3}. REDO = {T4} |
| I(T1, O2, A2) | |
| B(T3) | 2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4} |
| C(T1) | |
| B(T4) | 3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5} |
| 7. U(T3,O2,B3,A3) | |
| 9. U(T4,O3,B4,A4) | |
| CK(T2,T3,T4) | |
| 1. C(T4) | |
| 2. B(T5) | |
| 6. U(T3,O3,B5,A5) | |
| 10. U(T5,O4,B6,A6) | |
| 5. D(T3,O5,B7) | |
| A(T3) | |
| 3. C(T5) | |
| 4. I(T2,O6,A8) | |

3. Fase UNDO



B(T1)

B(T2)

8. U(T2, O1, B1, A1)

I(T1, O2, A2)

B(T3)

C(T1)

B(T4)

7. U(T3, O2, B3, A3)

9. U(T4, O3, B4, A4)

CK(T2, T3, T4)

1. C(T4)

2. B(T5)

6. U(T3, O3, B5, A5)

10. U(T5, O4, B6, A6)

5. D(T3, O5, B7)

A(T3)

3. C(T5)

4. I(T2, O6, A8)

0. UNDO = {T2, T3, T4}. REDO = {}

1. C(T4) → UNDO = {T2, T3}. REDO = {T4}

2. B(T5) → UNDO = {T2, T3, T5}. REDO = {T4} Setup

3. C(T5) → UNDO = {T2, T3}. REDO = {T4, T5}

4. D(O6)

5. O5 = B7

6. O3 = B5

7. O2 = B3

8. O1 = B1

Undo

4. Fase REDO



B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}	
B(T2)		
8. U(T2, O1, B1, A1)	1. C(T4) → UNDO = {T2, T3}. REDO = {T4}	
I(T1, O2, A2)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4}	Setup
B(T3)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}	
C(T1)		
B(T4)		
7. U(T3,O2,B3,A3)	4. D(O6)	
9. U(T4,O3,B4,A4)	5. O5 = B7	
CK(T2,T3,T4)		
1. C(T4)	6. O3 = B5	Undo
2. B(T5)		
6. U(T3,O3,B5,A5)	7. O2 = B3	
10. U(T5,O4,B6,A6)	8. O1 = B1	
5. D(T3,O5,B7)		
A(T3)	9. O3 = A4	
3. C(T5)		Redo
4. I(T2,O6,A8)	10. O4 = A6	

Descrivere la ripresa a caldo, indicando la costituzione progressiva degli insiemi di UNDO e REDO e le azioni di recovery, a fronte del seguente log:

DUMP, B(T1), B(T2), B(T3), I(T1, O1, A1), D(T2, O2, B2), B(T4),
U(T4, O3, B3, A3), U(T1, O4, B4, A4), C(T2), CK(T1, T3, T4), B(T5), B(T6),
U(T5, O5, B5, A5), A(T3), CK(T1, T4, T5, T6), B(T7), A(T4),
U(T7, O6, B6, A6), U(T6, O3, B7, A7), B(T8), A(T7), guasto

Soluzione:

1) Per prima cosa bisogna percorrere il log a ritroso fino al più recente record di check-point:

$CK(T1, T4, T5, T6)$

Si costruiscono gli insiemi di UNDO e di REDO:

$UNDO = \{ T1, T4, T5, T6 \}$ $REDO = \{ \}$

Soluzione:

2) Il log viene percorso in avanti, aggiornando i due insiemi:

$B(T7) \text{ UNDO} = \{ T1, T4, T5, T6, T7 \} \text{ REDO} = \{\}$

$A(T4) \text{ UNDO} = \{ T1, T4, T5, T6, T7 \} \text{ REDO} = \{\}$

$B(T8) \text{ UNDO} = \{ T1, T4, T5, T6, T7, T8 \} \text{ REDO} = \{\}$

$A(T7) \text{ UNDO} = \{ T1, T4, T5, T6, T7 \} \text{ REDO} = \{\}$

Soluzione:

3) Il log viene ripercorso ancora a ritroso, fino all'operazione I(T1,O1,A1) inclusa, eseguendo le seguenti operazioni:

O3=B7

O6=B6

O5=B5

O4=B4

O3=B3

Delete O1

4) Il log viene ripercorso in avanti per rieseguire le operazioni di REDO, ma essendo vuoto questo insieme, nessuna operazione verrà eseguita.

Considera il seguente schedule:

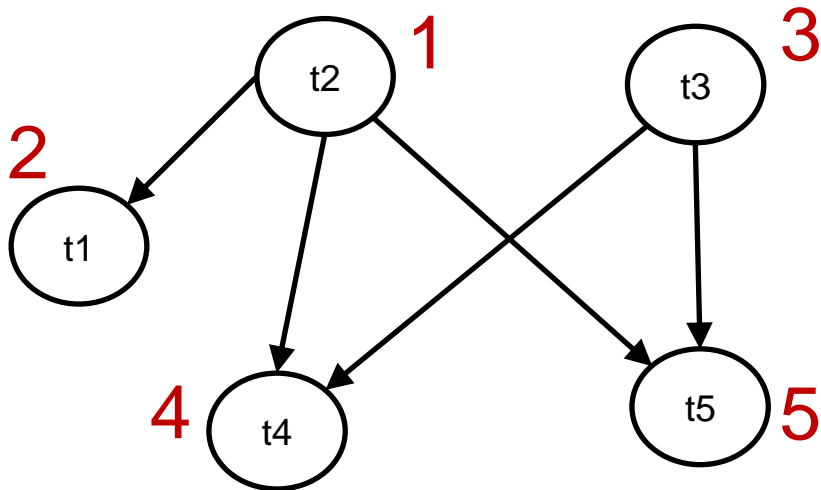
$S = r_2(x) \ r_1(x) \ w_3(t) \ w_1(x) \ r_3(y) \ r_4(t) \ r_2(y) \ w_2(z) \ w_5(y) \ w_4(z)$

S è conflict serializable? Se sì, mostrare uno schedule che è conflict-equivalente

Esercizio 2: Conflict Serializzabile?



$S = r_2(x) \ r_1(x) \ w_3(t) \ w_1(x) \ r_3(y) \ r_4(t) \ r_2(y) \ w_2(z) \ w_5(y) \ w_4(z)$



Schedule seriale conflict-equivalente:

$r_2(x) \ r_2(y) \ w_2(z) \ r_1(x) \ w_1(x) \ w_3(t) \ r_3(y) \ r_4(t) \ w_4(z) \ w_5(y)$

Dato lo schedule

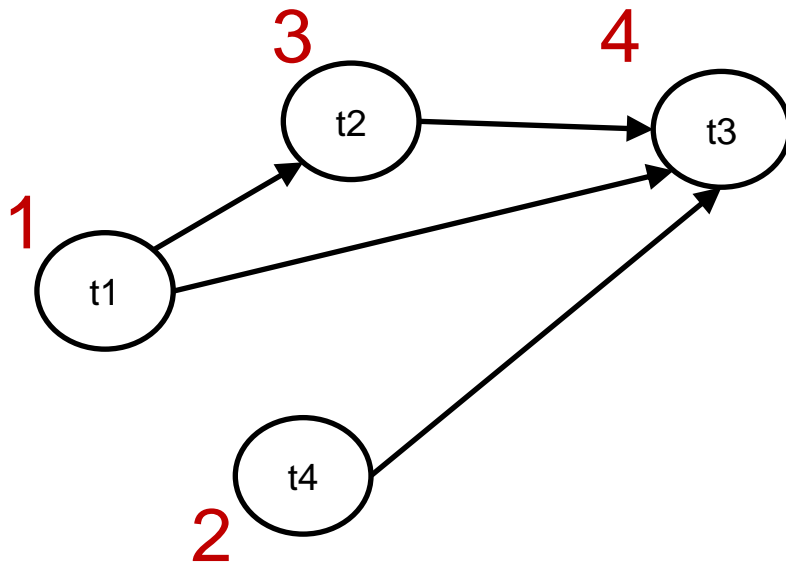
$S = r_1(x) \ w_2(x) \ r_3(x) \ w_1(u) \ w_3(v) \ r_3(y) \ r_2(y) \ w_3(u) \ w_4(t) \ w_3(t).$

Dire se è conflict-serializzabile e trovare uno schedule seriale conflict-equivalente

Esercizio 3



$S = r1(x) \ w2(x) \ r3(x) \ w1(u) \ w3(v) \ r3(y) \ r2(y) \ w3(u) \ w4(t) \ w3(t).$



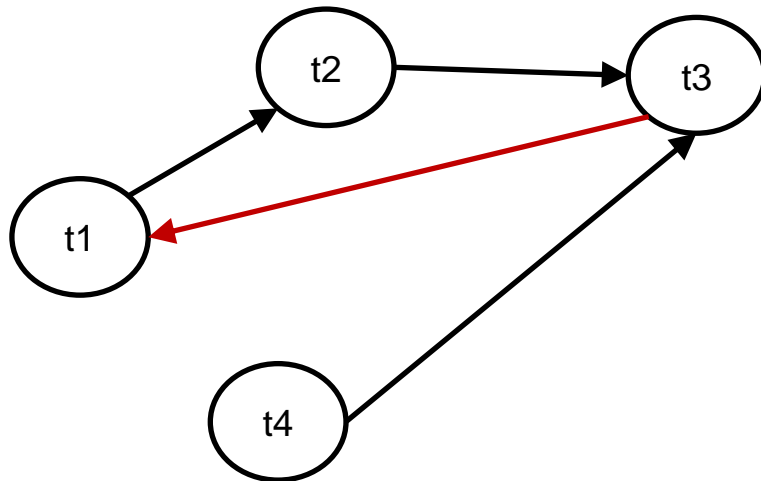
Schedule seriale conflict-equivalente:

$r1(x) \ w1(u) \ w4(t) \ w2(x) \ r2(y) \ r3(x) \ w3(v) \ r3(y) \ w3(u) \ w3(t)$

Esercizio 3 bis: spostiamo operaz. $w1(u)$ nel punto in rosso



$S = r1(x) \ w2(x) \ r3(x) \ w3(v) \ r3(y) \ r2(y) \ w3(u) \ w1(u) \ w4(t) \ w3(t).$



Ciclo $\langle t1, t2, t3, t1 \rangle \rightarrow$ non più conflict serializzabile

- L'operazione non può essere effettuata in quel punto
- Prima di effettuarla, occorre “rompere” il ciclo, cioè fare il commit o abort di $t2$ o $t3$ per togliere il nodo delle transazioni attive dal grafo

Indicare se i seguenti schedule sono VSR.

1. $r1(x), r2(y), w1(y), r2(x) w2,(x)$
2. $r1(x), r2(y), w1(x), w1(y), r2(x) w2,(x)$
3. $r1(x), r1(y), r2(y), w2(z), w1(z), w3(z), w3(x)$
4. $r1(y), r1(y), w2(z), w1(z), w3(z), w3(x), w1(x)$

Soluzione:

 = Elemento Relazione LEGGE - DA

1. $r1(x), r2(y), w1(y), r2(x) w2,(x)$

Questo schedule non è VSR, perché i due schedule seriali:

S1: $r1(x), w1(y), r2(y), r2(x), w2(x)$ e



S2: $r2(y), r2(x), w2(x), r1(x), w1(y)$



non sono view-equivalenti con lo schedule dato. Hanno entrambi una differente relazione LEGGE - DA

Soluzione:

2. $r1(x), r2(y), w1(x), w1(y), r2(x) w2,(x)$



Questo schedule non è VSR perché gli schedule

S1: $r1(x), w1(x), w1(y), r2(y), r2(x), w2(x)$



S2: $r2(y), r2(x), w2(x), r1(x), w1(x), w1(y)$



hanno entrambi una differente relazione LEGGE – DA

Soluzione:

3. $r1(x)$, $r1(y)$, $r2(y)$, $w2(z)$, $w1(z)$, $w3(z)$, $w3(x)$

Questo schedule è VSR e view-equivalente allo schedule seriale, in quanto sono caratterizzati dalle stesse scritture finali e dalle stesse relazioni LEGGI –DA.

S: $r2(y)$, $w2(z)$, $r1(x)$, $r1(y)$, $w1(z)$, $w3(z)$, $w3(x)$

Soluzione:

4. $r_1(y)$, $r_1(y)$, $w_2(z)$, $w_1(z)$, $w_3(z)$, $w_3(x)$, $w_1(x)$

Si noti che la transazione 1 ha due scritture: una su Z ed un'altra su X.

Ma anche la transazione 3 ha due scritture, una su Z ed un'altra su X.

Nello schedule di partenza, le scritture finali su X e Z sono originate rispettivamente dalle transazioni 1 e 3.

Nessuno schedule seriale potrà esibire le stesse scritture finali.

Questo schedule non è quindi VSR.

Classificare i seguenti schedule (come: NonSR, VSR, CSR).
Nel caso uno schedule sia VSR oppure CSR, indicare tutti gli schedule seriali e esso equivalenti.

1. $r1(x), w1(x), r2(z), r1(y), w1(y), r2(x), w2(x), w2(z)$
2. $r1(x), w1(x), w3(x), r2(y), r3(y), w3(y), w1(y), r2(x)$

Soluzione:

1. $r1(x), w1(x), r2(z), r1(y), w1(y), r2(x), w2(x), w2(z)$

Questo schedule è sia VSR che CSR, ed è conflict-equivalente (e view-equivalente) a:

S: $r1(x), w1(x), r1(y), w1(y), r2(z), r2(x), w2(x), w2(z)$

2. $r1(x), w1(x), w3(x), r2(y), r3(y), w3(y), w1(y), r2(x)$

Questo schedule è NonSR. In uno schedule seriale view-equivalente a questo schedule la transazione 1 dovrebbe seguire la transazione 3 a causa delle SCRITTURE FINALI su Y, ma dovrebbe anche precedere la transazione 3 a causa della relazione LEGGE – DA su X.

**Laurea in Informatica
A.A. 2021-2022**

Corso "Base di Dati"

Esercizi Vari

Prof. Massimiliano de Leoni



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

Verranno svolti esercizi riguardi i seguenti argomenti, nell'ordine:

1. Algebra Relazionale (73.3% dei rispondenti)
2. Terza Forma Normale (60% dei rispondenti)
3. Prima Forma Normale (60% dei rispondenti)
4. SQL con GROUP BY e HAVING (40% dei rispondenti)
5. Esercizi di modellazione con ER (33.3% dei rispondenti)

Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

Esprimere le seguenti query in Algebra Relazionale:

1. Le nazioni da cui parte ^{sarebbe "o" arriva, seguendo la logica della query} ~~e~~ arriva il volo con codice AZ274;
2. Le città da cui partono voli internazionali
3. Le città da cui partono solo voli nazionali
4. Le città con più piste;

Dato il seguente schema:

AEROPORTO (Citta, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittaPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

1. Le nazioni da cui parte e arriva il volo con codice AZ274:

$$\pi_{\text{Nazione}}(\sigma_{\text{IdVolo} = \text{'AZ274'}}(\text{VOLO})) \bowtie_{\text{CittaPart}=\text{Citta} \text{ OR } \text{CittaArr}=\text{Citta}} \text{AEROPORTO})$$

Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

2. Le città da cui partono voli internazionali:

$VP = (VOLO \bowtie_{CittàPart=Città} AEROPORTO);$

$VA = (VOLO \bowtie_{CittàArr=Città} AEROPORTO);$

$\pi_{VP.CittàPart} (VP \bowtie_{VP.IdVolo=VA.IdVolo \text{ AND } VP.Nazione \neq VA.Nazione} VA);$

Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

3. Le città da cui partono solo voli nazionali

$VP = (VOLO \bowtie_{CittàPart=Città} AEROPORTO);$

$VA = (VOLO \bowtie_{CittàArr=Città} AEROPORTO);$

$\pi_{Città} (AEROPORTO)$

-

$\pi_{Città} (\rho_{Città \leftarrow VP.CittàPart} ($
 $VP \bowtie_{VP.IdVolo=VA.IdVolo \text{ AND } VP.Nazione \neq VA.Nazione} VA$
 $))$

Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

4. Le città con più piste:

A1=AEROPORTO;

A2=AEROPORTO

$\pi_{\text{Città}}(\text{AEROPORTO}) - \pi_{\text{A1.Città}}(A1 \bowtie_{A1.\text{NumPiste} < A2.\text{NumPiste}} A2)$

Sia $R(A,B,C,D)$ uno schema di relazione su cui siano definite le dipendenze funzionali:

$$F = \{AB \rightarrow CD, C \rightarrow A, D \rightarrow B\}.$$

Si richiede di:

1. determinare le possibili chiavi

Se R non è in 3NF rispetto a F , allora:

2. mostrare le dipendenze che violano la 3NF
3. portare lo schema in terza forma normale.

Esercizio 1 - Terza Forma Normale



Sia $R(A,B,C,D)$ uno schema di relazione su cui siano definite le dipendenze funzionali:

$$F = \{AB \rightarrow CD, C \rightarrow A, D \rightarrow B\}.$$

Si richiede di:

1. determinare le possibili chiavi:

$$AB^+ = \{A, B, C, D\}$$

$$C^+ = \{C, A\}$$

$$D^+ = \{D, B\}$$

$$\text{Si noti che } C^+ \cup D^+ = \{A, B, C, D\} = CD^+$$

$$\text{Si noti che aggiungendo } B \text{ a } C^+, \text{ si ottiene } BC^+ = \{A, B, C, D\}$$

$$\text{Si noti che aggiungendo } A \text{ a } D^+, \text{ si ottiene } AD^+ = \{A, B, C, D\}$$

Quindi, AB, AD, BC e CD sono le chiavi

Esercizio 1 - Terza Forma Normale



AB, AD, BC e CD sono le chiavi

Sia $R(A,B,C,D,E)$ uno schema di relazione su cui siano definite le dipendenze funzionali:

$$F = \{AB \rightarrow CD, C \rightarrow A, D \rightarrow B\}.$$

Si richiede di:

Se R non è in 3NF rispetto a F , allora mostrare le dipendenze che violano la 3NF

Nulla da mostrare perché R è in 3NF rispetto alle dipendenze in F

(La parte 2 e 3 non è più rilevante)

Sia $R(A, B, C, D, E, G)$ uno schema di relazione su cui siano definite le dipendenze funzionali:

$$F = \{A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow G, CE \rightarrow D\}.$$

Si richiede di:

1. determinare le possibili chiavi

Se R non è in 3NF rispetto a F , allora:

2. mostrare le dipendenze che violano la 3NF
3. portare lo schema in terza forma normale.

Esercizio 2 - Esercizio Terza Forma Normale



Sia $R(A, B, C, D, E, G)$ uno schema di relazione su cui siano definite le dipendenze funzionali:

$$F = \{A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow G, CE \rightarrow D\}.$$

Si richiede di:

1. determinare le possibili chiavi:

$$A^+ = \{A, B\}$$

$$CB^+ = \{C, B, D, A\}$$

$$CE^+ = \{C, E, D, A, B, G\}$$

$$CD^+ = \{C, D, A, B\}$$

$$AE^+ = \{A, E, G, B\}$$

Aggiungendo E a CB^+ , si ottiene $ECB^+ = \{C, B, D, A, E\}$ ma B è superflua perchè $ECB^+ = CE^+$. Stesso dicasi per altre “estensioni”.

Quindi, CE è l'unica chiave.

Esercizio 2 - Terza Forma Normale



CE è l'unica chiave

Sia $R(A,B,C,D,E)$ uno schema di relazione su cui siano definite le dipendenze funzionali:

$$F = \{A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow G, CE \rightarrow D\}.$$

Si richiede di:

Se R non è in 3NF rispetto a F , allora mostrare le dipendenze che violano la 3NF

$A \rightarrow B, CD \rightarrow A, CB \rightarrow D, AE \rightarrow G$ violano la 3NF

Dati uno schema $R(U)$ e un insieme di dipendenze F su U , con chiavi K_1, \dots, K_n

1. Viene calcolata una copertura ridotta G di F
2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$
3. Viene costruita una relazione per ogni sotto-insieme
4. Se esistono due relazioni $S(X)$ e $T(Y)$ con $X \subseteq Y$, S viene eliminata
5. Se, per qualche i , non esiste una relazione $S(X)$ con $K_i \subseteq X$, viene aggiunta una relazione $T(K_i)$

1. Viene calcolata una copertura ridotta G di F

$A \rightarrow B,$

$CD \rightarrow A,$

$CB \rightarrow D,$

$AE \rightarrow G,$

$CE \rightarrow D$

Già copertura ridotta

2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$

$A \rightarrow B$

$CD \rightarrow A$

$CB \rightarrow D$

$AE \rightarrow G$

$CE \rightarrow D$

$A^+ = \{A, B\}$

$CD^+ = \{C, D, A, B\}$

$CB^+ = \{C, B, D, A\}$

$AE^+ = \{A, E, G, B\}$

$CE^+ = \{C, E, D, A, B, G\}$

Esercizio 2 - Terza Forma Normale



2. G viene partizionato in sottoinsiemi tali che due dipendenze funzionali $X \rightarrow A$ e $Y \rightarrow B$ sono insieme se $X_G^+ = Y_G^+$

$$\begin{array}{l} \mathbf{CD \rightarrow A} \\ \mathbf{CB \rightarrow D} \end{array} \left. \vphantom{\begin{array}{l} \mathbf{CD \rightarrow A} \\ \mathbf{CB \rightarrow D} \end{array}} \right\} \begin{array}{l} \mathbf{CD^+ = \{C, D, A, B\}} \\ \mathbf{CB^+ = \{C, D, A, B\}} \end{array}$$

$$\mathbf{A \rightarrow B} \left. \vphantom{\mathbf{A \rightarrow B}} \right\} \mathbf{A^+ = \{A, B\}}$$

$$\mathbf{AE \rightarrow G} \left. \vphantom{\mathbf{AE \rightarrow G}} \right\} \mathbf{AE^+ = \{A, E, G, B\}}$$

$$\mathbf{CE \rightarrow D} \left. \vphantom{\mathbf{CE \rightarrow D}} \right\} \mathbf{CE^+ = \{C, E, D, A, B, G\}}$$

Esercizio 2 - Terza Forma Normale



3. Viene costruita una relazione per ogni sotto-insieme

$$\left. \begin{array}{l} \mathbf{CD} \rightarrow \mathbf{A} \\ \mathbf{CB} \rightarrow \mathbf{D} \end{array} \right\} R_1 (A, B, C, D)$$

$$\mathbf{A} \rightarrow \mathbf{B} \left\} R_2 (A, B)$$

$$\mathbf{AE} \rightarrow \mathbf{G} \left\} R_3 (A, E, G)$$

$$\mathbf{CE} \rightarrow \mathbf{D} \left\} R_4 (C, D, E)$$

Esercizio 2 - Terza Forma Normale



4. Se esistono due relazioni $S(X)$ e $T(Y)$ con $X \subseteq Y$, S viene eliminata

$$\left. \begin{array}{l} \mathbf{CD} \rightarrow \mathbf{A} \\ \mathbf{CB} \rightarrow \mathbf{D} \end{array} \right\} R_1 (A, B, C, D)$$

$$\mathbf{A} \rightarrow \mathbf{B} \quad \left\} \quad \mathbf{R_2(A, B)} \quad \text{Si rimuove}$$

$$\mathbf{AE} \rightarrow \mathbf{G} \quad \left\} \quad R_3 (A, E, G)$$

$$\mathbf{CE} \rightarrow \mathbf{D} \quad \left\} \quad R_4 (C, D, E)$$

Esercizio 2 - Terza Forma Normale



5. Se, per qualche i , non esiste una relazione $S(X)$ con $K_i \subseteq X$, viene aggiunta una relazione $T(K_i)$

Chiave CE

$CD \rightarrow A$
 $CB \rightarrow D$ } $R_1 (A, B, C, D)$

$A \rightarrow B$ } $R_2 (A, B)$

$AE \rightarrow G$ } $R_3 (A, E, G)$

$CE \rightarrow D$ } $R_4 (C, D, E)$

**C'è già
 $R_4 (C, D, E)$
che contiene la
chiave CE**

Esercizio 2 - Terza Forma Normale



5. Se, per qualche i , non esiste una relazione $S(X)$ con $K_i \subseteq X$, viene aggiunta una relazione $T(K_i)$

Chiave CE

$CD \rightarrow A$
 $CB \rightarrow D$ } $R_1 (A, B, C, D)$ **Chiavi CD e CB**

$A \rightarrow B$ } $R_2 (A, B)$

$AE \rightarrow G$ } $R_3 (A, E, G)$ **Chiave AE**

$CE \rightarrow D$ } $R_4 (C, D, E)$ **Chiave CE**

Data una relazione $R(ABCD)$ con dipendenze funzionali
 $\{C \rightarrow D, C \rightarrow A, B \rightarrow C\}$

- (a) Identificare le possibili chiavi
- (b) Enumerare le dip.funz. che violano BCNF
- (c) Proporre una decomposizione in BCNF.
- (d) La decomposizione proposta è in BCNF e senza perdite nel join?

Relazione $R(ABCD)$ con $FD=\{C \rightarrow D, C \rightarrow A, B \rightarrow C\}$

(a) Identificare le possibili chiavi

$$C^+ = \{ C, D, A \}$$

$$B^+ = \{ B, C, D, A \}$$

Quindi, B è la sola possibile chiave

Relazione $R(\underline{A}BCD)$ con $FD=\{C \rightarrow D, C \rightarrow A, B \rightarrow C\}$

(b) enumerare le dip.funzi. che violano BCNF

Sono $\{C \rightarrow D, C \rightarrow A\}$: C non è super-chiave di R

Relazione $R(\underline{A}\underline{B}CD)$ con $FD=\{C \rightarrow D, C \rightarrow A, B \rightarrow C\}$

(c) Proporre una decomposizione in BCNF.

Usando $\{C \rightarrow D\}$, si ottiene $R_1(\underline{A}\underline{B}C)$ e $R_2(\underline{C}D)$

Usando $\{C \rightarrow A\}$, si ottiene $R_1(\underline{B}C)$, $R_2(\underline{C}D)$ e $R_3(\underline{C}A)$

(d) La decomposizione proposta è in BCNF e senza perdite nel join?

Sì, perchè gli attributi comuni di ogni coppia di relazioni è (super)chiave di una delle due relazioni. Per es. $R_1(\underline{B}C)$, $R_2(\underline{C}D)$ hanno in comune C che è chiave di R_2

Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

Le città francesi da cui partono più di 20 voli alla settimana diretti in Italia

```
SELECT CittàPart
FROM AEROPORTO as A1 join VOLO on A1.Città=CittàPart
      JOIN AEROPORTO as A2 on CittàArr=A2.Città
WHERE A1.Nazione='Francia' and A2.Nazione= 'Italia'
GROUP BY CittàPart
HAVING COUNT(*) >20
```

Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

Per ogni giorno della settimana, il tipo di aereo più utilizzato
(soluzione 1)

```
CREATE VIEW TIPOAEREO_X_GIORNO(GiornoSett, TipoAereo, Numero) AS
SELECT GiornoSett, TipoAereo, COUNT(*)
FROM VOLO
GROUP BY GiornoSett, TipoAereo;
```

```
SELECT *
FROM TIPOAEREO_X_GIORNO T1
WHERE Numero = (
    SELECT MAX(NUMERO)
    FROM TIPOAEREO_X_GIORNO T2
    WHERE T2.GiornoSett=T1.GiornoSett)
```


Dato il seguente schema:

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

Per ogni giorno della settimana, il tipo di aereo più utilizzato
(soluzione 2)

```
CREATE VIEW TIPOAEREO_X_GIORNO(GiornoSett, TipoAereo, Numero) AS
SELECT GiornoSett, TipoAereo, COUNT(*) AS Numero
FROM VOLO
GROUP BY GiornoSett, TipoAereo;
```

```
CREATE VIEW TIPOAEREO_MAX_X_GIORNO(GiornoSett,MaxNumero) AS
SELECT GiornoSett, MAX(Numero)
FROM TIPOAEREO_X_GIORNO
GROUP BY GiornoSett;
```

```
SELECT T.GiornoSett, T.TipoAereo
FROM TIPOAEREO_X_GIORNO T JOIN TIPOAEREO_MAX_X_GIORNO M
WHERE M.GiornoSett=T.GiornoSett AND T.Numero=M.MaxNumero
```

Si richiede di progettare lo schema ER concettuale di un'applicazione relativa alle edicole per la vendita di giornali. Di ogni edicola interessa il comune in cui essa è registrata, il codice, che è unico nell'ambito del comune in cui l'edicola stessa è registrata, la categoria, l'anno di inizio attività (non sempre disponibile), e i contratti che l'edicola ha con i distributori per l'approvvigionamento dei quotidiani.

Ogni contratto riguarda un'edicola, un quotidiano ed un distributore, ed è caratterizzato dal costo mensile a carico dell'edicola. Infine, per ogni edicola, interessa conoscere le varie persone che sono state proprietarie dell'edicola nei diversi anni, tenendo conto del fatto che in ogni anno un'edicola ha al massimo un proprietario. Di ogni persona interessa il codice fiscale (id), l'anno di nascita, il comune di nascita, ed il comune di residenza. Di ogni distributore di quotidiani interessa la partita IVA (id), il fatturato ed il comune in cui è situata la direzione. Di ogni quotidiano interessa il nome (identificativo), e l'anno di inizio pubblicazione. Di ogni comune interessa la provincia di appartenenza, il nome (unico nella provincia), ed il numero di abitanti. Dei comuni che sono capoluogo di provincia interessa l'attuale sindaco (con l'anno di elezione), ed il numero di assessori comunali. Di ogni provincia interessa il nome (identificativo) e la regione di appartenenza. Alcune province sono "autonome", e di esse interessa anche l'anno di istituzione.

Progettazione Concettuale

(cardinalità (1,n) omesse per leggibilità)

