

# Raccolta di esercizi di BD e delle relative risoluzioni

Antonio Albano e Giorgio Ghelli

## 1. DBMS

- 1) Si commentino brevemente le funzionalità offerte da un sistema relazionale che presenti il seguente menu iniziale:

- F1 - Create/Alter db-table-index
- F2 - Data Entry
- F3 - Interactive SQL
- F4 - Reports
- F5 - Forms
- F6 - Procedures
- F7 - Select Database
- F8 - Exit

- 2) Uno studente vuole farsi una BD ad uso personale per i dischi di musica che possiede usando un DBMS sul proprio PC. Quali delle seguenti funzionalità non sono necessarie:

- Schema logico
- Schema fisico
- Tabelle calcolate
- Schemi esterni
- Indipendenza logica
- Indipendenza fisica
- Gestore sicurezza
- Gestore integrità
- Gestore della concorrenza
- Gestore dell'affidabilità
- Linguaggio per interrogazioni
- Linguaggio per lo sviluppo di applicazioni

- 3) Elencare le differenze tra la nozione di tupla nei sistemi relazionali e la nozione di oggetto nel modello ad oggetti.

- 4) Elencare alcune domande (fra cinque e dieci) da fare ad un produttore di sistemi per la gestione di dati per stabilire se il sistema che propone può essere classificato come un sistema per la gestione di basi di dati centralizzate.

- 5) Si supponga che sia stato definito uno schema relazionale con le istruzioni:

```
CREATE TABLE R (K integer(8) not null, A integer(8), B integer(8) )  
PRIMARY KEY (K)
```

```
GRANT SELECT ON R TO caio
```

e siano stati immessi dei dati in R.

Usando i comandi:

```
DROP TABLE Nome
CREATE TABLE Nome (Attributo Tipo, ...) AS ExprSQL
CREATE VIEW Nome (Attributo, ...) AS ExprSQL
```

mostrare come si possa modificare lo schema in modo che i dati di R vengano memorizzati esclusivamente nelle tabelle:

```
R1( K integer(8) not null, A integer(8) )
R2( K integer(8) not null, B integer(8) )
```

e l'utente "caio" possa continuare a lavorare sulla base di dati come se esistesse la tabella:

```
R (K integer(8) not null, A integer(8), B integer(8) ).
```

## 2. PROGETTAZIONE CONCETTUALE

- 1) Si supponga che per archiviare dati sull'inventario delle apparecchiature di un'azienda sia stato usato un foglio elettronico con la seguente struttura:

Inventario(NInventario, Modello, Descrizione, NumeroDiSerie, Costo, Responsabile, TelefonoResponsabile)

---

NInv.	Modello	Descrizione	NumeroDiSerie	Costo	Responsabile	Tel.Resp.
...						
111	SUN3	Stazione di lavoro Sun ...	ajk078546	25000	Caio	576
112	PB180	Notebook Macintosh...	a908mnb	6000	Tizio	587
113	SUN3	Stazione di lavoro Sun ...	ajp890796	27000	Tizio	587
...						

---

Il numero di inventario identifica un'apparecchiatura. Un'apparecchiatura ha un costo, un modello e un numero di serie. Apparecchiature dello stesso modello possono avere costi differenti, perché acquistati in momenti diversi, ma hanno la stessa descrizione. Il numero di serie è una caratteristica dell'apparecchiatura e due diverse apparecchiature dello stesso modello hanno numero di serie diverso. Ogni apparecchiatura ha un responsabile, che può avere più apparecchiature, ma un unico numero di telefono. I responsabili sono identificati dal cognome.

Si definisca uno schema concettuale grafico della base di dati.

- 2) Si vogliono gestire le prenotazioni per spettacoli teatrali in Italia. Di uno spettacolo interessa il regista, il titolo, l'autore, la compagnia e gli attori principali. Uno spettacolo viene rappresentato più volte. Di ogni rappresentazione interessa la città, la data, l'ora e il teatro. Le prenotazioni riguardano una rappresentazione e di esse interessa il nome della persona che ha effettuato la prenotazione, il recapito telefonico, e la lista dei posti prenotati. Un posto è codificato da una coppia (numero fila, numero posto). Ad ogni spettacolo, ad ogni

rappresentazione e ad ogni prenotazione è assegnato un codice che lo identifica. Si definisca uno schema concettuale grafico della base di dati.

- 3) Si vogliono rappresentare le seguenti informazioni relative alle tesi di un corso di laurea. Di una tesi interessa il titolo, gli studenti che la svolgono (anche più di uno), i relatori (anche più di uno), il controrelatore, il tipo (ricerca, rasegna, progetto) e l'area (basi di dati, programmazione logica, architettura, ecc.). Di uno studente interessa la matricola, il cognome, la data prevista di laurea. I relatori possono essere docenti universitari o dipendenti di altri enti, mentre i controrelatori sono sempre docenti. Di un relatore interessa il cognome e l'ente di appartenenza, di un docente interessa anche il dipartimento di appartenenza. Si dia uno schema concettuale grafico della base di dati. Si definisca uno schema concettuale grafico della base di dati.
- 4) Un circolo del tennis vuole memorizzare le prenotazioni dei propri campi da tennis da parte dei propri soci. Ogni prenotazione riguarda un singolo socio ed un singolo campo da tennis, prenotato per una certa ora ed una certa data (per semplicità si rappresentino tanto la data che l'ora tramite una stringa). Un singolo campo può essere prenotato, ad una certa ora di una certa data, da un solo socio; un singolo socio può effettuare più prenotazioni. Di ogni socio interessano nome, cognome e recapito. Di un campo interessa ricordare se sia coperto o scoperto e se sia in cemento o in terra battuta. Ogni socio ha un codice numerico che lo identifica; similmente per ogni campo. Si definisca uno schema concettuale grafico della base di dati.
- 5) Nel selezionare gli articoli da accettare per una conferenza, ciascun articolo viene spedito ad un certo numero di revisori, ciascuno dei quali riceve normalmente più articoli da valutare. Ogni revisore assegna un punteggio ad ogni articolo. Un revisore è identificato da un codice e un articolo è identificato da un numero. Si supponga per semplicità che un autore sia anche identificato da nome e cognome. Si definisca uno schema concettuale grafico della base di dati.
- 6) Si vuole gestire l'allocazione di aule dedicate a seminari. Un seminario può essere tenuto in più parti, che possono anche svolgersi in aule diverse. Di un seminario interessano Responsabile, Titolo e Codice (che lo identifica). Di un'aula interessano Nome (che la identifica) e Capacità. Interessa inoltre sapere per ogni parte di un seminario la data, l'ora di inizio e fine, e l'aula in cui si svolge. Si definisca uno schema concettuale grafico della base di dati.
- 7) Si vogliono gestire i dati di interesse di una compagnia di assicurazione ramo RCA. Interessano i dati sui clienti, auto e incidenti. Di un cliente interessano codice fiscale (che lo identifica), nome e indirizzo. Di un'auto interessano targa e modello. Di un incidente interessa l'auto assicurata coinvolta (si suppone che sia unica) il danno (in lire) e la percentuale di colpa. Un cliente può avere più automobili e un'automobile ha un solo proprietario. Un'automobile può essere stata coinvolta in più incidenti. Si definisca uno schema concettuale grafico della base di dati.
- 8) Una società di rilevazioni ha effettuato dei sondaggi relativi ai voti espressi dagli elettori in una consultazione nazionale. Il territorio è diviso in circoscrizioni, in ogni circoscrizione si presentano più partiti (ma non tutti i partiti si presentano in tutte le circoscrizioni). Ogni elettore vota in una specifica circoscrizione. Di ogni elettore interrogato interessano l'età, il sesso, il voto che ha dichiarato di avere espresso, e la circoscrizione in cui vota; per

semplicità si supponga che tutti gli interrogati o non rispondono o dichiarano di avere votato uno specifico partito. Di ogni partito interessa il nome che lo contraddistingue e l'anno di fondazione. Di ogni circoscrizione interessano il nome, che la distingue, e la popolazione. Si definisca uno schema concettuale grafico della base di dati.

- 9) Si vogliono gestire informazioni sui dipendenti di una ditta. Di un dipendente interessa il nome, il codice, che lo identifica, e lo stipendio. Un dipendente può essere un tecnico, un responsabile di progetto, oppure un amministrativo. Di un tecnico interessa anche la qualifica e di un responsabile di progetto interessa anche il budget. Ogni tecnico è diretto da un responsabile di progetto, il quale dirige più tecnici. Si definisca uno schema concettuale grafico della base di dati.
- 10) Si vogliono gestire i dati relativi a dirigenti e progetti di una ditta divisa in più dipartimenti. Ogni dirigente appartiene ad un unico dipartimento e più dirigenti possono appartenere allo stesso dipartimento. Un dipartimento lavora a più progetti e un progetto è seguito da uno o più dipartimenti. Ogni progetto ha un unico direttore che è uno dei dirigenti dei dipartimenti coinvolti nel progetto. Un dirigente può dirigere al più un progetto. Di un dipartimento interessa il nome (che lo identifica) e l'indirizzo. Di un progetto interessa il nome (che lo identifica) e il finanziamento. Di un dirigente interessa la matricola (che lo identifica) e il nome. Si definisca uno schema concettuale grafico della base di dati.
- 11) Si vogliono trattare informazioni relative a stati e fiumi. Di uno stato interessa il nome, che lo identifica, la popolazione e gli stati confinanti. Di un fiume interessa il nome, la lunghezza e gli stati attraversati. In uno stato i nomi dei fiumi sono tutti diversi. Si ignori il fatto che il nome di un fiume, o stato, può cambiare a seconda della lingua considerata. Si definisca uno schema concettuale grafico della base di dati.
- 12) Una banca gestisce informazioni sui mutui, clienti e rate per produrre tabulati del tipo mostrato in figura. Un cliente può avere più di un mutuo.

RESOCONTO MUTUO			
COD-MUTUO: 250		DATA: 7/2/94	
SCADENZA: 1/1/2000			
AMMONTARE: 70 M			
COD-CLIENTE: 2000			
NOME CLIENTE: Rossi Mario			
INDIRIZZO CLIENTE: Via Roma, 13 -Pisa			
NUM-RATA	DATA SCADENZA	AMMONTARE	DATA VERSAMENTO
1	1/7/93	6 M	29/6/93
2	1/1/94	6 M	30/12/93
3	1/7/94	6.1 M	--
.....	.....	.....	.....

Si definisca uno schema concettuale grafico della base di dati.

- 13) Si vuole realizzare un gioco di esplorazione su calcolatore. Il gioco si compone di più "stanze"; in ogni momento il giocatore si trova all'interno di una "stanza" dove può compiere alcune azioni, oppure chiedere di spostarsi in una direzione (su, giù, Nord, Ovest,...) per entrare in un'altra stanza. Si desidera memorizzare la mappa del gioco, ovvero per ogni stanza a quali altre stanza essa sia collegata, e per ogni collegamento tra due

stanze la direzione del collegamento. Inoltre, di ogni stanza interessano il nome (che la identifica), l'aspetto (un valore di tipo Disegno), e le azioni che si possono eseguire (una sequenza di valori di tipo Azione).

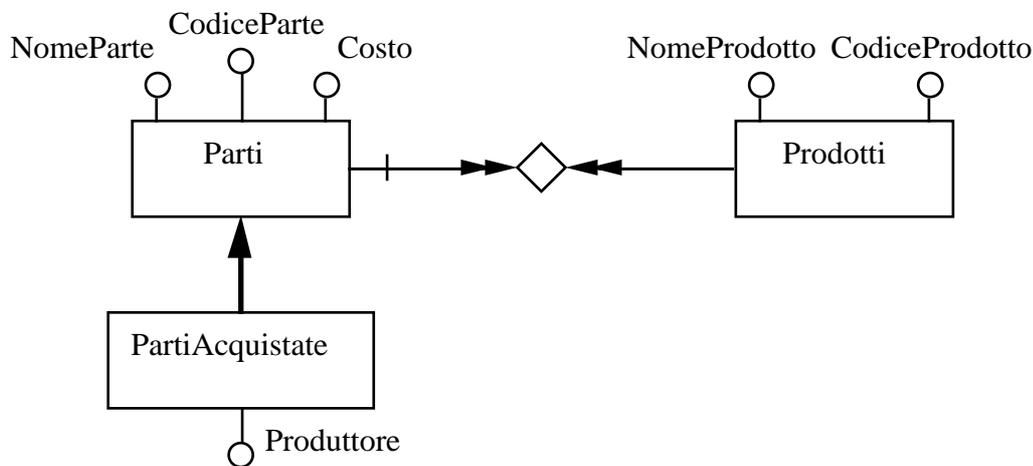
Si definisca uno schema concettuale grafico della base di dati.

- 14) Si vogliono trattare informazioni relative a attori e registi di film. Di attori e registi interessa il nome, che lo identifica, l'anno di nascita e la nazionalità. Un attore può essere anche un regista. Di un film interessano il titolo, l'anno di produzione, gli attori e il regista. Due film prodotti lo stesso anno hanno titolo diverso. Si definisca uno schema concettuale grafico della base di dati.
- 15) Si vogliono memorizzare informazioni relative allo schema di una base di dati relazionale, ottenendo così una meta-base di dati. Si vogliono memorizzare informazioni relative a relazioni, attributi ed utenti. Un attributo ha un nome, una relazione di appartenenza ed un dominio; in una singola relazione attributi diversi hanno nome diverso, ma attributi in relazioni diverse possono avere lo stesso nome (ad esempio un attributo di nome *Matricola* nella relazione *Studenti* ed un attributo di nome *Matricola* nella relazione *Esami* sono considerati due attributi diversi con lo stesso nome). Una relazione ha un nome, che la identifica, un tipo (relazione utente o relazione di sistema), e ad essa appartiene un insieme di attributi. Un utente ha un nome, che lo identifica, e ci interessa memorizzare a quali relazioni ha diritto di accedere.  
Si definisca uno schema concettuale grafico della base di dati.
- 16) Si vuole automatizzare il sistema di gestione degli animali in uno zoo. Ogni esemplare di animale ospitato è identificato dal suo genere (es. zebra) e da un codice unico all'interno del genere di appartenenza. Per ogni esemplare si memorizzano la data di arrivo nello zoo, il nome proprio, il sesso, il paese di provenienza e la data di nascita.  
Lo zoo è diviso in aree; in ogni area c'è un insieme di case, ognuna destinata ad un determinato genere di animali. Ogni casa contiene un insieme di gabbie, ognuna contenente un solo esemplare.  
Ogni casa ha un addetto che pulisce ciascuna gabbia in un determinato giorno della settimana.  
Gli animali sono sottoposti periodicamente a controllo veterinario; in un controllo, un veterinario rileva il peso degli esemplari, diagnostica un'eventuale malattia e prescrive il tipo di dieta da seguire.  
Si definisca uno schema concettuale grafico della base di dati.
- 17) Un'azienda vuole trattare le informazioni sugli impiegati, i dipartimenti e i progetti in corso. Di un impiegato interessano il codice, assegnato dalla azienda, che lo identifica, nome e cognome, anno di nascita, il sesso e i familiari a carico, dei quali interessano il nome, il sesso, la relazione di parentela e l'anno di nascita. Di un dipartimento interessano il numero, che lo identifica, il nome e la città dove si trova. Di un progetto interessano il numero, che lo identifica, e il codice. Gli impiegati afferiscono a un dipartimento, che gestisce più progetti ed è diretto da un impiegato. Gli impiegati partecipano a più progetti, che si svolgono presso dipartimenti di città diverse, a ognuno dei quali dedicano una percentuale del proprio tempo. Gli impiegati sono coordinati da un responsabile.  
Si definisca uno schema concettuale grafico della base di dati.
- 18) Un'azienda ha più linee di prodotti ognuna delle quali è controllata da un responsabile ed è stata progettata da un dirigente tecnico. I responsabili sono dipendenti che svolgono

mansioni di controllo. I dirigenti sono dipendenti che coordinano l'attività di un gruppo di studio al quale partecipano più dipendenti. Un dipendente può partecipare a più gruppi di studio. Di un dipendente interessano il nome, il codice fiscale, che lo identifica, e lo stipendio. Di un dirigente interessano il nome del gruppo di studio che coordina, che lo identifica. Di un responsabile interessa l'anzianità di servizio. Di una linea di prodotto interessano il nome del prodotto, che la identifica, e il costo di produzione. Un dirigente può aver progettato più linee di prodotti e un responsabile può controllare più linee di prodotti. Si definisca uno schema concettuale grafico della base di dati.

### 3. PROGETTAZIONE RELAZIONALE E SQL

1) Per la base di dati in figura:



- a) Definire uno schema relazionale e dire in quale forma normale è lo schema.
- b) Scrivere in SQL le seguenti interrogazioni:
  - “trovare i nomi di tutti i prodotti”;
  - “trovare i nomi di tutte le parti usate da un prodotto con CodiceProdotto “XY” “;
  - “trovare i nomi di tutti i prodotti che contengono almeno una parte acquistata”.

2) All'Ufficio della Motorizzazione interessano i seguenti fatti sulle persone che posseggono auto:

- di ogni persona il codice fiscale, che l'identifica, il cognome e l'indirizzo;
- di ogni auto la targa, che l'identifica, l'anno di immatricolazione e i cavalli fiscali;
- l'associazione fra auto e proprietari, nell'ipotesi che un auto abbia un solo proprietario e un proprietario possa avere più auto.

Si supponga che per la base di dati sia stata definito il seguente schema relazionale:

Proprietari(CodiceFiscale,Cognome,Indirizzo, Targa)  
 Auto(Targa,AnnoImmatricolazione,CavalliFiscali)

Dire se lo schema presenta anomalie, giustificando la risposta. Dare, rispetto a tale schema, le espressioni SQL che ritornano i cognomi dei proprietari di auto con 20 cavalli fiscali, e

l'anno di immatricolazione delle auto possedute dalle persone con cognome Rossi.

- 3) Si considerino due diverse basi di dati, contenenti le seguenti informazioni riguardanti i docenti:
- di un docente interessa il codice fiscale di tipo stringa, che lo identifica, i recapiti telefonici (in generale più di uno e di tipo stringa) e i nomi dei corsi che tiene (in generale più di uno e di tipo stringa);
  - di un docente interessa il codice fiscale di tipo stringa, che lo identifica, ed i recapiti (in generale più di uno) costituiti da via e città entrambi di tipo stringa.

Dare i due schemi relazionali e dire in quale forma normale si trovano.

- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.4 e si dica quale forma normale soddisfa. Definire in SQL l'interrogazione per trovare il codice di tutti i campi per i quali esiste almeno una prenotazione effettuata da un socio con cognome "Rossi".
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.5 e si dica in quale forma normale si trova. Definire in SQL l'interrogazione per trovare il titolo e il punteggio dei lavori valutati da un revisore con nome X.
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.6 e si dica in quale forma normale si trova.
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.7 e si dica in quale forma normale si trova. Definire due interrogazioni in SQL per trovare il nome e l'indirizzo dei clienti che: a) hanno avuto almeno un incidente e b) non hanno avuto incidenti.
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.8 e si dica in quale forma normale si trova. Definire un'interrogazione in SQL per trovare il numero dei voti espressi per ogni partito nella circoscrizione C.
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.9 e si dica in quale forma normale si trova. Definire un'interrogazione in SQL per trovare il nome di tutti i tecnici che sono diretti dal responsabile del tecnico con codice X.
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.10 e si dica in quale forma normale si trova. Definire un'interrogazione in SQL per trovare tutti i finanziamenti dei progetti in cui è coinvolto il dipartimento del dirigente con matricola specificata M.
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.11 e si dica in quale forma normale si trova. Definire un'interrogazione in SQL per trovare i nomi dei fiumi che attraversano uno stato confinante con lo stato di nome "X".
- Si dia lo schema relazionale per la base di dati dell'Esercizio 2.15 e si dica in quale forma normale si trova. Definire in SQL le interrogazioni:
  - Per un utente di nome "Luigi", trovare i nomi di tutti gli attributi di tutte le relazioni a cui ha diritto di accedere.
  - Trovare i nomi di tutti gli utenti che hanno diritto di accedere ad almeno un attributo di nome "XXX".

Supponiamo di aggiungere il vincolo che due attributi che hanno lo stesso nome devono avere anche lo stesso dominio. Lo schema relazionale disegnato in precedenza quali forme normali rispetta in questo caso?

- 13) Sono dati i seguenti schemi relazionali, utilizzati per la gestione di un laboratorio:

COMPUTER (codComputer, marca, modello, indirizzoFornitore)  
 INSTALLAZIONE (codComputer, codSoftware, descrizSoftware, dataInstallaz)

(computer con uguale modello sono identici e differenziati solo dal codice; i computer della stessa marca - ad esempio IBM - hanno tutti lo stesso fornitore; uno stesso software è in genere installato su più computer).

Si chiede di:

- Indicare tutte le dipendenze funzionali presenti nei due schemi. Se uno o entrambi gli schemi risultano non normalizzati, si determini un insieme di schemi che siano in forma normale di Boyce-Codd e risultino equivalenti, dal punto di vista informativo, agli schemi dati.
- Sullo schema normalizzato, scrivere un'interrogazione SQL che restituisca l'indirizzo di tutti i fornitori di calcolatori sui quali è stato installato del software dopo del 1-1-1995.

- 14) Una palestra ospita diversi corsi appartenenti a diverse tipologie (aerobica, danza classica, danza jazz...). Ogni corso ha una sigla, che lo identifica, un insegnante, e alcuni allievi. Un insegnante offre in generale più corsi, anche con diverse tipologie, e anche un allievo può essere iscritto a più corsi. Di ogni insegnante interessano il nome (che lo identifica) e l'indirizzo. Di ogni allievo interessano il nome (che lo identifica) e il numero di telefono. Per ogni allievo interessa sapere, per ogni corso che frequenta, quanto ha già versato finora. La palestra gestisce attualmente i dati con un'unica relazione così strutturata:

Allievo	TelAll	CodCorso	TipoCorso	Saldo	Insegn.	IndInsegn.
Luigi	303130	C1	Scherma	300.000	Mario	Via Roma
Pietro	304140	C1	Scherma	200.000	Mario	Via Roma
Luigi	303130	C4	Scherma	300.000	Antonio	Via Udine
Andrea	303130	C3	Danza	300.000	Filippo	Via Roma
Pietro	304140	C7	Judo	250.000	Mario	Via Roma
.....	.....					

Si chiede di:

- Dare una copertura canonica delle dipendenze in tale schema ed elencare le chiavi candidate;
- Applicare allo schema l'algoritmo di sintesi per portarlo in 3FN, e dire se lo schema così ottenuto è anche in BCNF; (facoltativo: applicare allo schema l'algoritmo di decomposizione per portarlo in BCFN, e dire se tale decomposizione preserva le dipendenze);

- 15) Si dia lo schema relazionale per la base di dati dell'Esercizio 2.17 e si dica quale forma normale soddisfa. Definire in SQL le seguenti interrogazioni:

1. Trovare il nome delle impiegate

2. Trovare i nomi dei dipartimenti di Pisa che gestiscono almeno due progetti e il numero dei progetti gestiti.
3. Di ogni impiegato trovare il nome e il nome del dipartimento dove lavora.
4. Trovare i nomi e l'età (all'anno attuale) delle impiegate più anziane del loro supervisore.
5. Trovare il cognome di ogni impiegato senza familiari a carico
6. Di ogni impiegato trovare il nome, il totale del tempo dedicato ai progetti a cui partecipa.
7. Trovare il nome degli impiegati che afferiscono a dipartimenti diretti da un impiegato di sesso diverso.
8. Di ogni impiegato, trovare il nome, il codice del progetto a cui partecipa e il tempo ad esso dedicato.
9. Trovare i nomi dei dipartimenti, il codice dei progetti che gestiscono e i numero dei partecipanti ad ogni progetto
10. Trovare i nomi degli impiegati che partecipano a progetti dei dipartimenti di Pisa
11. Degli impiegati che partecipano ai progetti dei dipartimenti di Pisa, trovare i nomi degli impiegati e del dipartimento di appartenenza.
12. Nome e anno nascita dei figli dell'impiegato con codice 010:
13. Di ogni impiegato trovare il cognome e il nome, anno nascita e sesso dei figli a carico.

#### *Restrizioni con quantificatori*

14. Trovare il cognome degli impiegati che hanno tutti i familiari dello stesso sesso dell'impiegato.
15. Trovare il cognome degli impiegati che hanno tutti i familiari dello stesso sesso.
16. Trovare il cognome degli impiegati che lavorano solo a progetti che li impegnano per il 20 per cento del loro tempo.
17. Trovare il cognome degli impiegati che lavorano (solo) a progetti gestiti tutti dal dipartimento 2.
18. Trovare il cognome degli impiegati che lavorano a progetti gestiti tutti dallo stesso dipartimento.
19. Trovare il cognome degli impiegati che lavorano a tutti i progetti a cui partecipa l'impiegato con codice 100 (o 010).

#### *Uso di funzioni di aggregazione*

20. Per i dipartimenti della stessa città, trovare il nome della città e il numero dei loro dipendenti.

21. Raggruppa i progetti per dipartimento, e trovare il nome del dipartimento e il numero dei progetti per dipartimento.

22. Raggruppa i progetti per dipartimento, e trovare il nome del dipartimento e il numero dei progetti per dipartimento e numero totale dei partecipanti e il massimo tempo dedicato da un partecipante:

23. Raggruppare gli impiegati per sesso e trovare per ogni gruppo il loro numero e il numero totale dei familiari

24. Raggruppare gli impiegati per fasce di età e trovare per ogni gruppo il loro numero e il numero totale dei familiari

25. Raggruppare gli impiegati per dipartimento, e per ogni gruppo trovare il nome e città del dipartimento, il numero degli impiegati, l'anno di nascita dell'impiegato più anziano.

#### 4. NORMALIZZAZIONE DI SCHEMI RELAZIONALI

1) Dimostrare che per lo schema  $R\langle T, F \rangle$  se un attributo  $A$  non appare a destra di una DF, allora  $A$  fa parte di ogni chiave di  $R$ .

2) Sia  $R\langle T, F \rangle$  uno schema relazionale, con  $F$  una copertura canonica. Si dimostri che se lo schema ha una sola chiave candidata ed è in 3FN allora è anche in FNBC.  
Suggerimento: si inizi con lo scrivere la definizione di 3FN e di FNBC (obbligatorio), e si dia un nome (ad es.  $Y$ ) all'insieme di attributi che forma la sola chiave di  $R\langle T, F \rangle$ . Si ragioni per assurdo, supponendo che  $R\langle T, F \rangle$  sia in 3FN ma non in FNBC.

3) Discutere la complessità dei seguenti problemi di decisione:

PRIMALITA': Dato lo schema di relazione  $R\langle T, F \rangle$ ,  $A \in T$ ,  $A$  è primo?

TEST 3FN: Dato lo schema di relazione  $R\langle T, F \rangle$ ,  $R$  è in 3FN rispetto ad  $F$ ?

TEST FNBC: Dato lo schema di relazione  $R\langle T, F \rangle$ ,  $R$  è in FNBC rispetto ad  $F$ ?

TEST FNBC DI SOTTOSHEMA: Dato lo schema di relazione  $R\langle T, F \rangle$ , dato  $X \subseteq T$ ,  $X$  è in FNBC rispetto alla proiezione di  $F$  su  $X$ ?

4) Discutere la complessità del seguente problema di decisione: dato uno schema  $R\langle T, F \rangle$ ,  $F$  è in forma canonica?

5) Supponiamo che una dipendenza funzionale  $X \rightarrow Y$  sia soddisfatta da un'istanza di relazione  $r$ . Sia  $r' \subseteq r$  (quindi  $r'$  è una relazione con gli stessi attributi di  $r$ ).  $r'$  soddisfa  $X \rightarrow Y$ ? Se sì, dire perché, altrimenti dare un controesempio.

6) Supponiamo che una dipendenza funzionale  $X \rightarrow Y$  sia soddisfatta da due istanze di

relazione con gli stessi attributi r ed s.  $r \cap s$  soddisfa  $X \rightarrow Y$ ?  $r \cup s$  soddisfa  $X \rightarrow Y$ ? Se sì, dire perché, altrimenti dare un controesempio.

7) Descrivere un algoritmo per trovare una chiave di uno schema relazionale  $R(T,F)$  e discuterne il costo.

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

$$F = \{AB \rightarrow CDE, AC \rightarrow BDE, B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}.$$

Si richiede di:

- portare F in forma canonica;
- determinare le possibili chiavi;
- mostrare che lo schema non è in terza forma normale;
- portare lo schema in terza forma normale.

9) Dare la definizione di Forma Normale di Boyce Codd ed indicare la complessità dell'operazione di verificare se uno schema  $R\langle T, F \rangle$  soddisfa tale forma normale. Definire l'algoritmo di normalizzazione per decomposizione e valutarne la complessità.

10) Si dia una definizione formale dell'operazione di giunzione naturale e del concetto di decomposizione che preserva i dati per uno schema relazionale.

11) Si consideri la relazione R con attributi ABCD e le dipendenze:  $AB \rightarrow C$ ,  $C \rightarrow D$  e  $D \rightarrow A$ .

- Trovare tutte le chiavi di R.
- Dire se R è in 3NF.
- Dire se R è in BCNF.
- Applicare l'algoritmo di sintesi.
- Applicare l'algoritmo di analisi (opzionale).

12) Si consideri lo schema  $R(A,B,C,D,E)$  con le DF,  $A \rightarrow BC$ ,  $CD \rightarrow E$ ,  $B \rightarrow D$ ,  $E \rightarrow A$ , e la decomposizione di R in  $R_1(A,B,C)$  and  $R_2(A,D,E)$ .

1. Trovare le chiavi di R.
2. La decomposizione conserva i dati?
3.  $R_2$  è in FNBC?
4. La decomposizione conserva le dipendenze?

13) Si consideri il seguente schema relazionale:

Vendite(Commesso, Negozio, Città, Data, CodiceProdotto, Taglia, Colore)  
Prodotti(CodiceProdotto, Taglia, Colore, Prezzo)

Supponiamo che:

- Ogni commesso lavora in un solo negozio
- Ogni negozio si trovi in una sola città
- Un dato prodotto abbia sempre lo stesso prezzo
- Ogni prodotto sia disponibile in più taglie e colori.

Basandosi solo su queste assunzioni:

- a) si diano le dipendenze funzionali,
- b) si specifichino le chiavi delle due relazioni,
- c) si dica se lo schema è in FNBC, se no lo si decomponga.

- 14) Un'agenzia fornisce personale ad alberghi e raccoglie le informazioni sui contratti in corso nella tabella R(CF, NContratto, Nome, NAlbergo, IndirizzoA). Si mostra un esempio di istanza di R:

10	C1	N1	H1	Via1
20	C1	N2	H1	Via1
30	C2	N3	H2	Via2
10	C2	N1	H2	Via2

- a) Trovare le dipendenze funzionali.
- b) Trovare le chiavi di R.
- c) Dire se R è in 3FN. Se non lo è, trovare una decomposizione in 3FN che preservi dati e dipendenze.

- 15) Si consideri lo schema relazionale R(A, B, C, D). Per ognuno dei seguenti insiemi di DF trovare: a) le chiavi, b) la forma normale più forte che soddisfa R, c) se R non è in FNBC, dire se esiste una decomposizione in FNBC che preservi dati e dipendenze.

- a)  $B \rightarrow C, D \rightarrow A$
- b)  $A \rightarrow B, BC \rightarrow D, A \rightarrow C$

- 16) Si consideri la seguente relazione, che descrive i finanziamenti ricevuti dai docenti che afferiscono a dei dipartimenti. Ogni docente afferisce ad un unico dipartimento, ed ha un unico telefono. Un finanziamento può riguardare più docenti, e un docente può avere più finanziamenti. Ogni dipartimento ha un unico indirizzo. Ogni finanziamento ha un unico ammontare ed un'unica scadenza.

Finanziamenti(CodDipartimento, IndirizzoDipartimento, CodDocente, TelDocente, CodFinanziamento, Ammontare, Scadenza)

Si chiede di:

- 1) Definire una copertura canonica per le dipendenze funzionali valide su tale relazione
- 2) Elencare tutte le chiavi della relazione
- 3) Dire se la relazione è in terza forma normale e se è in forma normale di Boyce Codd
- 4) Applicare l'algoritmo di sintesi alla relazione
- 5) Applicare l'algoritmo di analisi alla relazione
- 6) Dire come cambiano le dipendenze funzionali se si aggiunge il vincolo che ogni finanziamento riguarda un unico docente

# SOLUZIONI

## 1. DBMS

- 1) F1 permette di definire nuove basi di dati, di definire e modificare lo schema logico di una base di dati, e infine di definire e modificare un aspetto dello schema fisico, la presenza di indici.  
F2 fornisce un ambiente che facilita l'immissione di dati nella base di dati.  
F3 permette di formulare ed eseguire interrogazioni SQL.  
F4 dà l'accesso ad un generatore di rapporti, cioè un ambiente per definire il formato ed il contenuto di stampe da prodursi a partire dal risultato di interrogazioni.  
F5 dà l'accesso ad un generatore di forme (moduli elettronici). Un modulo elettronico ha parti di testo e campi usati per visualizzare o immettere dati, come accade in un foglio elettronico. Il contenuto di un campo può essere calcolato con un'espressione SQL.  
F6 fornisce un ambiente per definire procedure che operano sulla base di dati.  
F7 permette di selezionare una delle basi di dati definite in precedenza.  
F8 termina la sessione di lavoro.
- 2) Da fare.
- 3) Le tuple nel modello relazionale hanno campi di tipo elementare, non hanno una nozione di identità, non hanno componenti procedurali come i metodi degli oggetti, non hanno nozioni di inclusione, eredità, incapsulazione. Dal punto di vista della modellazione, ne consegue, in particolare, che:
  - nel modello relazionale non è possibile rappresentare le entità della realtà con un'unica tupla se queste hanno una struttura complessa (ad esempio, se hanno attributi multivalore);
  - nel modello relazionale le associazioni si rappresentano usando il meccanismo delle chiavi esterne;
  - nel modello relazionale è necessario aggiungere un campo chiave a tutte quelle classi in cui è possibile che si abbiano due entità con gli stessi valori per tutti gli attributi.
- 4)
  - 1) Schema logico: il sistema gestisce insiemi di dati e associazioni tra di loro? Quale modello dei dati adotta? Il sistema gestisce e rende accessibile un catalogo delle definizioni dei dati (meta-dati)? Cosa si può definire in uno schema? Si può modificare uno schema con la base di dati già creata?
  - 2) Schema fisico: il sistema permette di definire l'organizzazione fisica dei dati e le strutture di accesso in maniera dichiarativa? Quali tipi di indici si possono definire? In che misura le applicazioni sono indipendenti da queste scelte?
  - 3) Schemi esterni: il sistema consente di definire "viste" dei dati, calcolate o memorizzate? Quali possibilità sono previste per modificare la struttura logica dei dati accessibili dalle viste?
  - 4) Controllo dei dati: il sistema permette di associare dei vincoli ai dati, e cura il mantenimento di tali vincoli?

- 5) Accesso ai dati: il sistema prevede un linguaggio di interrogazione e di generazione di rapporti? Il sistema permette di accedere ai dati da un programma? In quali linguaggi può essere scritto un tale programma?
- 6) Sicurezza: il sistema permette di definire utenti e classi di utenti e di stabilire i diritti di accesso ai diversi dati delle diverse classi di utenti, e controlla poi in maniera inviolabile che gli utenti non eseguano operazioni non ammesse?
- 7) Affidabilità: il sistema permette di definire transazioni, cioè sequenze di azioni da eseguire atomicamente? Il sistema garantisce l'atomicità delle transazioni indipendentemente dal tipo di guasto? Il sistema garantisce la persistenza degli effetti delle transazioni terminate normalmente?
- 8) Controllo della concorrenza: il sistema permette l'esecuzione concorrente di transazioni? Il sistema garantisce la serializzabilità dell'esecuzione di transazioni concorrenti?
- 9) Prestazioni: qual'è il numero massimo di classi, di attributi per ciascuna classe, di indici per ciascuna classe, di utenti, di gruppi di utenti, di connessioni contemporanee che il sistema può gestire? Quali limiti esistono sulla quantità di dati memorizzabili nella base di dati? Che tipo di ottimizzazione effettua il sistema sulle interrogazioni che gli vengono proposte?
- 10) Strumenti: il sistema mette a disposizione strumenti che permettano all'amministratore di valutare la bontà dell'organizzazione, quali, ad esempio, strumenti per verificare la frequenza di accesso ai dati e l'efficienza di esecuzione delle transazioni? Il sistema mette a disposizione strumenti per i programmatori, ad esempio per costruire agevolmente applicazioni con un'interfaccia grafica?

5) Si può agire come segue:

```
CREATE TABLE R1( K integer(8) not null, A integer(8) )
AS SELECT K, A FROM R PRIMARY KEY (K)

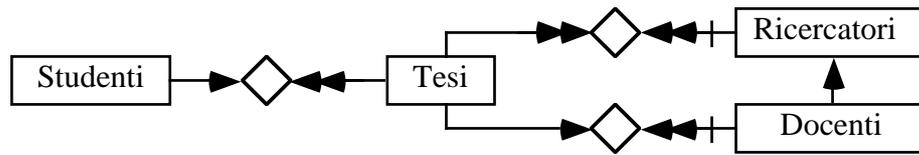
CREATE TABLE R2( K integer(8) not null, B integer(8) )
AS SELECT K, B FROM R PRIMARY KEY (K)
DROP TABLE R
CREATE VIEW R (K,A,B)
AS SELECT K, A, B FROM R1, R2 WHERE R1.K = R2.K
GRANT SELECT ON R TO caio
```

## 2. PROGETTAZIONE CONCETTUALE

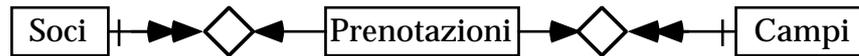
- 1) Da fare.
- 2) Da fare.
- 3) Schema grafico:

Per semplicità immaginiamo di gestire solo studenti a cui è stata assegnata la tesi e solo tesi

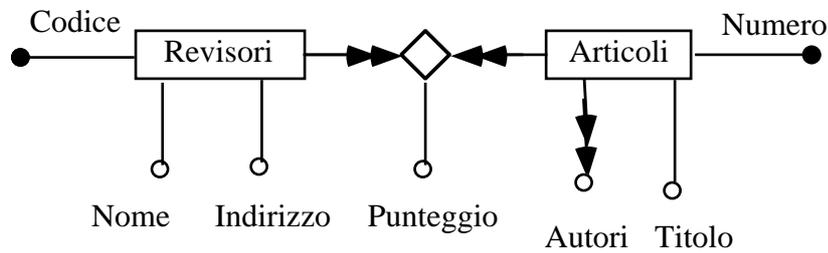
a cui è stato assegnato un controrelatore. La classe Ricercatori contiene dipendenti di un qualunque ente di ricerca, Università compresa.



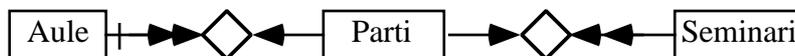
4) Schema grafico:



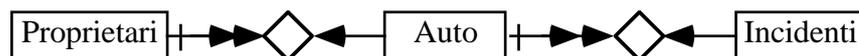
5) Schema grafico:



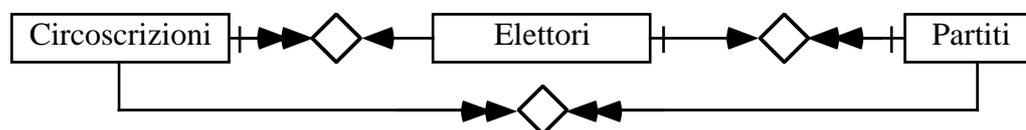
6) Schema grafico:



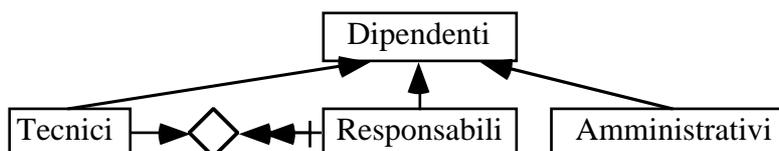
7) Schema grafico:



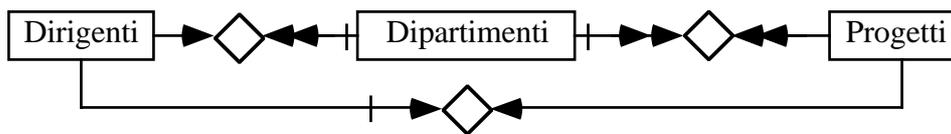
8) Schema grafico:



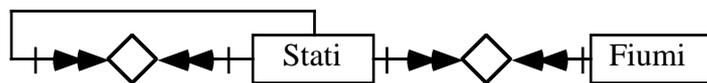
9) Schema grafico:



10) Schema grafico:

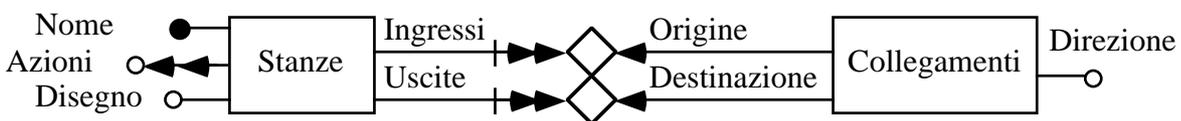


11) Schema grafico:



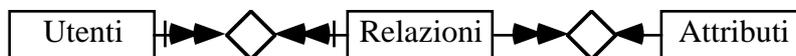
12) Da fare.

13) Schema grafico:

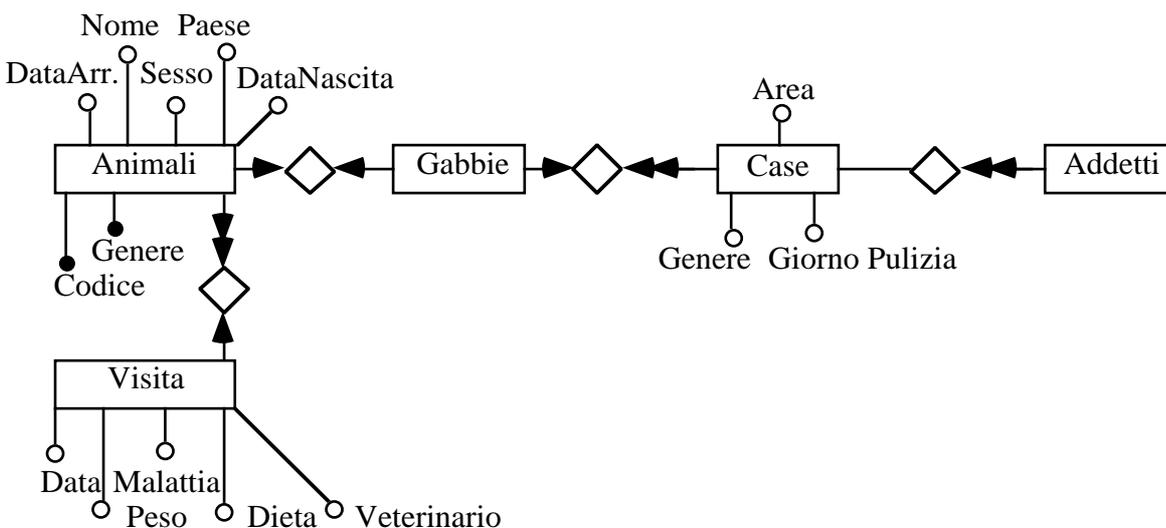


14) Da fare.

15) Schema grafico:



16) Schema grafico:



Gli attributi *Genere* di *Animali* e *Case* potrebbero essere sostituiti con un'associazione con un'apposita classe *Generi*. Il vincolo potrebbe essere mantenuto derivando l'attributo

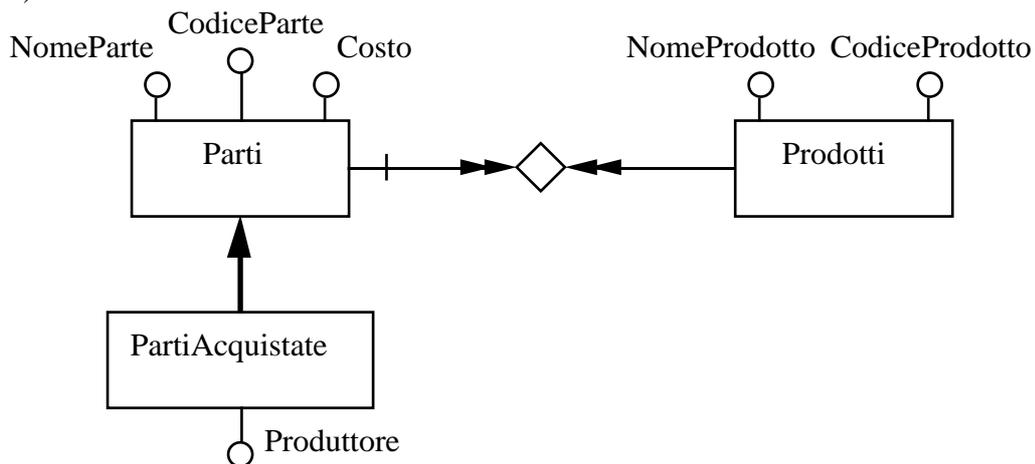
*Genere di Animali da Gabbia.Casa.Genere.* L'attributo *Area* di *Case* potrebbe essere sostituito con un'associazione con un'apposita classe *Aree*.

17) Da fare.

18) Da fare.

### 3. PROGETTAZIONE RELAZIONALE E SQL

1)



*Schema relazionale:*

Prodotti(CodiceProdotto,NomeProdotto)  
 Parti(CodiceParte,NomeParte,Costo)  
 PartiProdotti(CodiceProdotto,CodiceParte)  
 PartiAcquistate(CodiceParte,Produttore)

*oppure:*

Prodotti(CodiceProdotto,NomeProdotto)  
 PartiNonAcquistate(CodiceParte,NomeParte,Costo)  
 PartiProdotti(CodiceProdotto,CodiceParte)  
 PartiAcquistate(CodiceParte,NomeParte,Costo,Produttore)

*oppure:*

Prodotti(CodiceProdotto,NomeProdotto)  
 Parti(CodiceParte,NomeParte,Costo)  
 PartiProdotti(CodiceProdotto,CodiceParte)

Tutti gli schemi sono in BCNF poiché tutte le dipendenze funzionali sono conseguenza di vincoli di chiave.

Interrogazioni (con riferimento al primo schema):

```
select NomeProdotto
from Prodotti
```

```
select distinct NomeParte
from Parti, PartiProdotti
where Parti.CodiceParte = PartiProdotti.CodiceParte
and PartiProdotti.CodiceProdotto = "XY"
```

```
select distinct NomeProdotto
from Prodotti, PartiProdotti, PartiAcquistate
where Prodotti.CodiceProdotto = PartiProdotti.CodiceProdotto
and PartiAcquistate.CodiceParte = PartiProdotti.CodiceParte
```

### Osservazione:

La giunzione:

```
select distinct NomeParte
from Parti, PartiProdotti
where Parti.CodiceParte = PartiProdotti.CodiceParte
and PartiProdotti.CodiceProdotto = "XY"
```

Potrebbe essere espressa anche facendo uso di select annidati:

```
select NomeParte
from Parti
where CodiceParte in
(select CodiceParte
from PartiProdotti
where PartiProdotti.CodiceProdotto = "XY")
```

```
select NomeParte
from Parti P
where exists
(select *
from PartiProdotti
where PartiProdotti.CodiceProdotto = "XY"
and PartiProdotti = P.PartiProdotti)
```

Le tre interrogazioni hanno la stessa semantica, ma la prima è da preferirsi, poiché la generazione del piano di accesso utilizzato per eseguire la giunzione è lasciata al sistema, mentre le altre due specificano parzialmente l'algoritmo da utilizzarsi per eseguire la giunzione. La seconda formulazione forza, in pratica, il sistema ad eseguire per prima cosa la restrizione su *PartiProdotti*, mentre la terza chiede al sistema di valutare nuovamente la **select** interna per ogni elemento della relazione esterna *Parti*. Si richiede quindi di utilizzare sempre la prima formulazione.

2) Schema grafico:



Valgono le seguenti dipendenze funzionali:

Proprietari: CodiceFiscale → Cognome, Indirizzo;

Targa → CodiceFiscale; (Chiave candidata: Targa)

Auto: Targa → AnnoImmatricolazione, CavalliFiscali; (Chiave candidata: Targa)

Lo schema non rispetta quindi la terza FN (e quindi neppure la BCNF) poiché la relazione Proprietari contiene una dipendenza transitiva di due attributi non primi da una chiave candidata.

Interrogazione SQL:

```
select distinct Cognome
from Auto, Proprietari
where Proprietari.Targa = Auto.Targa AND CavalliFiscali = 20
```

```
select AnnoImmatricolazione
from Auto, Proprietari
where Proprietari.Targa = Auto.Targa AND Cognome = "Rossi"
```

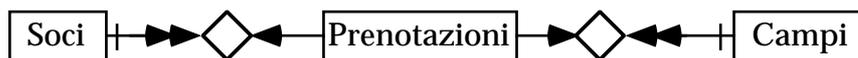
3) Schemi relazionali:

(a) Docenti(CF); Telefoni(CF, Telefono); Corsi(CF, Corso)

(b) Docenti(CF); Indirizzi(CF, Via, Citta)

Se si suppone che ogni docente abbia un numero di telefono o tenga un corso, la relazione Docenti non è necessaria nel primo schema. Similmente, se ogni docente ha un indirizzo, la relazione Docenti non è necessaria nel secondo. Lo schema è in FNBC poiché tutte le dipendenze funzionali derivano da vincoli di chiavi.

4) Schema grafico:



Dip. Funzionali:

Prenotazioni: CodCampo, Ora, Giorno → CodSocio;

CodCampo → CampoCoperto, TerraBattuta

Unica chiave candidata: CodCampo, Ora, Giorno

Soci: Codice → Nome, Cognome, Recapito

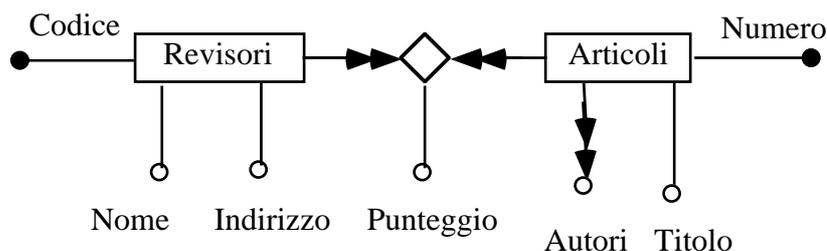
Unica chiave candidata: Codice

Lo schema non è in 2FN a causa delle dipendenze parziali che esistono nella classe Prenotazioni; lo schema non soddisfa quindi neppure le forme normali più restrittive (3FN, FNBC, 4FN).

```

select    CodCampo
from      Prenotazioni, Soci
where     Soci.Codice = Prenotazioni.CodSocio and Soci.Cognome="Rossi"
  
```

5) Schema grafico:



```

Revisori(Codice, Nome, Indirizzo)
Articoli(Titolo, Numero)
Punteggi(CodRevisore*, NumArticolo*, Punti)
AutoriArticoli(Nome, Ente, NumArticolo*)
  
```

Supponiamo che esista un unico revisore con nome X.

```

select    Titolo, Punti
from      Revisori, Punteggi, Articoli
where     Nome = X
           and Revisori.Codice = Punteggi.CodRevisore
           and Articoli.Numero = Punteggi. NumArticolo
  
```

6) Schema grafico



Schema relazionale

```

Aule(Nome, Capienza)
Parti(Data,OraInizio,OraFine,CodiceSeminario,NomeAula)
Seminari(Responsabile,Titolo,Codice)
  
```

Dipendenze funzionali:

Aule: Nome → Capienza

Parti: NomeAula, Data, OraInizio → OraFine, CodiceSeminario

NomeAula, Data, OraFine → OraInizio, CodiceSeminario

CodiceSeminario, Data, OraInizio → OraFine, NomeAula

CodiceSeminario, Data, OraFine → OraInizio, NomeAula

Seminari: Codice → Responsabile, Titolo

Il sistema è in BCNF poiché, riportando la copertura sopra data in forma canonica, tutti i membri sinistri di tale copertura sono chiavi. Osserviamo che le prime dipendenze funzionali definite sulla relazione Parti derivano dal vincolo che due seminari non possono avvenire contemporaneamente nella stessa aula, ma non riescono ad esprimere completamente tale vincolo. Similmente, le altre due dipendenze esprimono parzialmente il vincolo che due parti diverse di uno stesso seminario non si possono sovrapporre.

7) Schema grafico



Schema relazionale

Clienti(Codice, Nome, Indirizzo);

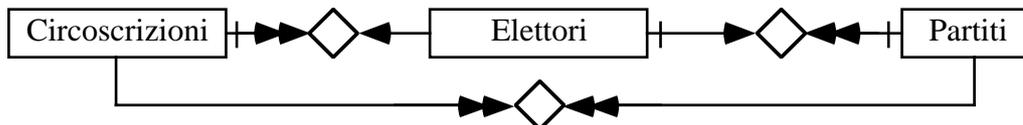
Auto(Targa, Modello, CodiceProprietario);

Incidenti (NumeroIncidente, Danno, PercentualeColpa, TargaAuto);

NumeroIncidente è un codice che identifica univocamente l'incidente, necessario per distinguere tra due incidenti che riguardano la stessa auto, con uguale danno e percentuale di colpa; si osservi come questo campo non fosse necessario nella modellazione ad oggetti.

Lo schema è in BCNF poiché tutte le dipendenze funzionali derivano da vincoli di chiave.

8) Schema grafico:



Schema relazionale.

Circoscrizioni(Nome, Popolazione)

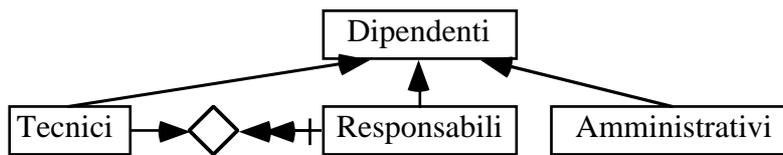
PartitiCircoscrizioni(Partito, Circoscrizione)

Elettori(Codice, Eta, Sesso, Voto, Circoscrizione)

Partiti(Nome, AnnoFondazione)

```
SELECT Voto, count(*)
FROM Elettori
WHERE Circoscrizione=C
GROUP BY Voto
```

9) Schema grafico



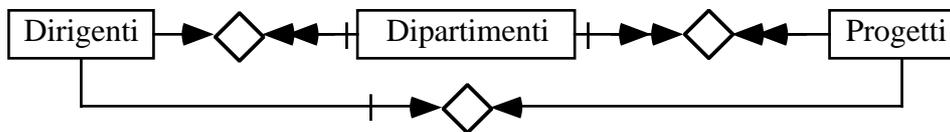
Schema relazionale

Dipendenti(Nome, Codice, Stipendio)  
 Tecnici(Codice\*, Qualifica, CodiceResponsabile\*)  
 Responsabili(Codice\*, Budget)  
 Amministrativi(Codice\*)

Lo schema si trova in BCNF, dato che tutte le dipendenze dipendono da vincoli di chiave.

```
c) SELECT t2.Nome
FROM Tecnici t1, Tecnici t2
WHERE t1.Codice = X AND t1.CodiceResponsabile = t2.CodiceResponsabile
```

10) Schema grafico



Schema relazionale

Dirigenti(Matricola, Nome, Dipartimento)  
 Progetti(Nome, Finanziamento, Direttore) due chiavi primarie: Nome e Direttore  
 DipProg(Dipartimento, Progetto)  
 Dipartimenti(Nome, Indirizzo)

Lo schema è in BCNF poiché tutti i vincoli derivano da vincoli di chiave

```
SELECT Progetti.Finanziamento
FROM Dirigenti, DipProg, Progetti
WHERE Dirigenti.Matricola = M
AND Dirigenti.Dipartimento = DipProg.Dipartimento;
AND DipProg.Progetto = Progetti.Nome;
```

11) Schema grafico;



Questo esercizio illustra bene il fatto che, in presenza di dipendenze funzionali particolari, diventano possibili diverse modellazioni relazionali della stessa situazione. In particolare, sono ragionevoli i seguenti schemi:

- (a) Stati(NomeStato,Popolazione) NS  $\rightarrow$  P  
 Confini(PrimoStato\*,SecondoStato\*)  
 Attraversamenti(Stato\*, CodiceFiume\*)  
 Fiumi(NomeFiume,Lunghezza, CodiceFiume) CF  $\rightarrow$  NF,L
- (b) Stati(NomeStato,Popolazione) NS  $\rightarrow$  P  
 Confini(PrimoStato\*,SecondoStato\*)  
 Attraversamenti (NomeFiume, CodiceFiume,Stato\*)NF,S  $\rightarrow$  CF; CF  $\rightarrow$  NF  
 Fiumi(CodiceFiume,Lunghezza) CF  $\rightarrow$  L
- (c) Stati(NomeStato,Popolazione) NS  $\rightarrow$  P  
 Confini(PrimoStato\*,SecondoStato\*)  
 Attraversamenti(Stato\*, NomeFiume, Lunghezza)NF,S  $\rightarrow$  L

Il primo schema è il più naturale, ed è quello che si ottiene operando una traduzione dello schema relazionale. Non presenta anomalie, è in BCNF, ma perde la dipendenza NF,S  $\rightarrow$  L. Il secondo schema è in 3FN ma non in BCNF, a causa della relazione Attraversamenti, ma preserva la dipendenza NF,S  $\rightarrow$  L. Il terzo schema è teoricamente perfetto, poiché è in BCNF e preserva sia dati che dipendenze, ma presenta ugualmente anomalie da ripetizione: si supponga di inserire un fiume che attraversa n stati, la sua lunghezza dovrà essere inserita n volte, e se una volta la lunghezza è inserita in modo errato il sistema non è in grado di rilevarlo.

#### Interrogazione

SQL (utilizzando il primo schema):

```
SELECT NomeFiume
FROM Confini, Attraversamenti x, Fiumi y
WHERE PrimoStato = "X"
AND SecondoStato = x.Stato AND x.CodiceFiume = y.CodiceFiume;
```

12) Schema grafico:



Utenti(NomeUtente)  
 Diritti(NomeUtente,NomeRelazione)  
 Relazioni(NomeRelazione,Tipo)

$\text{NomeRelazione} \rightarrow \text{Tipo}$   
 $\text{Attributi}(\underline{\text{NomeAttributo}}, \text{NomeRelazione}, \text{Dominio})$   
 $\text{NomeAttributo}, \text{NomeRelazione} \rightarrow \text{Dominio}$

Lo schema è in BCFN poiché tutte le dipendenze corrispondono a vincoli di chiavi. È in 4FN poiché è in 3FN e tutte le dipendenze multivalore non banali sono dipendenze funzionali.

Interrogazioni:

```

select   NomeAttributo
from     Diritti,Attributi
where    NomeUtente = "Luigi"
           and Diritti.NomeRelazione = Attributi.NomeRelazione

```

```

select   NomeUtente
from     Diritti,Attributi
where    NomeAttributo = "XXX"
           and Diritti.NomeRelazione = Attributi.NomeRelazione

```

Supponendo di aggiungere la dipendenza  $\text{NomeAttributo} \rightarrow \text{Dominio}$  alla relazione  $\text{Attributi}$ , tale relazione non è più in 2FN, poiché la chiave resta  $\text{NomeAttributo}, \text{NomeRelazione}$ , e la nuova dipendenza è una dipendenza parziale.

- 13) Le dipendenze delle due relazioni sono descritte dalle seguenti forme canoniche:

COMPUTER:

$\text{codComputer} \rightarrow \text{modello}$ ,  
 $\text{modello} \rightarrow \text{marca}$ ,  
 $\text{marca} \rightarrow \text{indirizzoFornitore}$

INSTALLAZIONE:

$\text{codComputer}, \text{codSoftware} \rightarrow \text{dataInstallaz}$ ,  
 $\text{codSoftware} \rightarrow \text{descrizSoftware}$

La seguente decomposizione della relazione COMPUTER è in BCNF e preserva dati e dipendenze:

$\text{FORNITORI}(\underline{\text{marca}}, \text{indirizzoFornitore})$   
 $\text{MODELLI}(\underline{\text{modello}}, \text{marca})$   
 $\text{COMPUTER}(\underline{\text{codComputer}}, \text{modello})$

$\text{SOFTWARE}(\underline{\text{codSoftware}}, \text{descrizSoftware})$   
 $\text{INSTALLAZIONI}(\underline{\text{codComputer}}, \underline{\text{codSoftware}}, \text{dataInstallaz})$

Una decomposizione  $R_1 \langle T_i \rangle_{i=1..n}$  di  $R \langle T, F \rangle$  preserva le dipendenze se:

(\*)  $(\cup_{i=1..n}\{X \rightarrow Y \mid X \rightarrow Y \in F^+, X, Y \subseteq T_i\})^+ \supseteq F^+$   
(l'altra inclusione è ovvia). In questo caso si ha che:

$$(\cup_{i=1..n}\{X \rightarrow Y \mid X \rightarrow Y \in F, X, Y \subseteq T_i\}) = F$$

da cui:

$$(\cup_{i=1..n}\{X \rightarrow Y \mid X \rightarrow Y \in F^+, X, Y \subseteq T_i\}) \supseteq F$$

da cui si deduce subito (\*).

```
b) SELECT indirizzoFornitore
FROM FORNITORI, MODELLI, COMPUTER, INSTALLAZIONI
WHERE FORNITORI.marca = MODELLI.marca
AND MODELLI.modello = COMPUTER.modello
AND INSTALLAZIONI.codCompute=COMPUTER.codCompute
AND dataInstallaz ≥ 1-1-1995
```

14) Da fare.

15) Da fare.

#### 4. NORMALIZZAZIONE DI SCHEMI RELAZIONALI

- 1) Da fare.
- 2)  $R\langle T, F \rangle$  (con  $F$  canonica) è in 3FN sse per ogni dipendenza  $X \rightarrow A$  in  $F$  o  $X$  è una chiave o  $A$  è primo.  
 $R\langle T, F \rangle$  è in FNBC sse per ogni dipendenza  $X \rightarrow A$  in  $F$   $X$  è una chiave.  
Sia  $Y$  l'unica chiave di  $R\langle T, F \rangle$ . Supponiamo che  $R\langle T, F \rangle$  sia in 3FN ma non in FNBC.  
Dato che non è in FNBC, allora esiste  $X \rightarrow A$  in  $F$  dove  $X$  non è chiave, ma, essendo  $R$  in 3FN, allora  $A$  deve essere primo; riassumendo:  $X \rightarrow A$ ,  $A \notin X$  ( $F$  canonico) ed  $A \in Y$  ( $A$  primo). Assurdo: poiché  $A$  è implicato dagli altri attributi, allora esiste una chiave candidata che non contiene  $A$ , e che è perciò diversa da  $Y$ .  
Formalmente,  $X \rightarrow A \mid -$  (arricchimento su  $T-A$ )  $(T-A)X \rightarrow (T-A)A$  ovvero  $T-A \rightarrow T$ , per cui  $T-A$  contiene una chiave candidata  $Y'$ , diversa da  $Y$  poiché  $Y'$  non contiene  $A$ .
- 3) PRIMALITA': Un algoritmo per valutare la primalità di un attributo  $A$  consiste nel generare tutti i sottoinsiemi di  $T$  che contengono  $A$  e verificare se uno di essi sia una chiave. Questo algoritmo ha complessità esponenziale rispetto al numero di attributi, poiché il numero di sottoinsiemi di un insieme di " $a$ " elementi è  $2^a$ , e verificare se un sottoinsieme di un insieme di " $a$ " attributi sia una chiave, rispetto ad un insieme di " $p$ " dipendenze, ha complessità polinomiale  $O(ap)$ .

TEST 3FN: Un modo per determinare se uno schema  $R\langle T, F \rangle$  sia in 3FN consiste nel portare lo schema in una forma canonica  $G$  e poi nel verificare se, per ogni dipendenza in  $X \rightarrow A$  in  $G$ , se  $X$  non è una chiave, allora  $A$  è primo. Per portare  $F$  in forma canonica è sufficiente:

- a) dividere le dipendenze in modo che ogni membro destro sia composto di un solo attributo;
- b) eliminare, da ogni membro sinistro, gli attributi ridondanti;
- c) eliminare le dipendenze ridondanti.

Tutte queste operazioni si possono effettuare in tempo polinomiale.

Data poi una dipendenza  $X \rightarrow A$  in  $G$ , verificare se  $X$  sia chiave richiede un tempo polinomiale, poiché significa verificare se  $X^+ = T$ ; tuttavia, quando  $X$  non è chiave, verificare se  $A$  sia primo richiede un tempo esponenziale. Quindi questo algoritmo ha complessità esponenziale.

TEST FNBC: Questo problema ha complessità polinomiale; basta utilizzare lo stesso algoritmo visto al punto precedente per portare  $F$  in forma canonica  $G$ . Per ogni  $X \rightarrow A$  in  $G$  bisogna poi verificare se  $X$  sia una chiave, e questa operazione è polinomiale.

TEST FNBC DI SOTTOSHEMA: Per risolvere questo problema si può operare come segue: prima si proietta  $F$  su  $X$  e poi si verifica se  $R\langle X, \prod_X F \rangle$  è in FNBC. Questo algoritmo è esponenziale, poiché per proiettare  $F$  su  $X$  è necessario valutare  $F^+$ .

- 4) L'algoritmo più semplice verifica le seguenti condizioni:

- a) Tutti i membri destri sono formati da un solo attributo: costo  $O(p)$   
 b) Per ogni  $X \rightarrow A \in F$ , per ogni  $B \in X$ ,  $A \notin (X-B)^+$ : costo  $p \times a \times O(ap) = O(a^2p^2)$   
 c) Per ogni  $X \rightarrow A \in F$ ,  $A \notin X^+$ , dove la chiusura è calcolata rispetto ad  
 $F - \{X \rightarrow A\}$ : costo  $p \times O(ap) = O(ap^2)$

Costo totale dell'algoritmo:  $O(a^2p^2)$ .

- 5)  $r$  soddisfa  $X \rightarrow Y$  significa che  $e_1 \in r$ ,  $e_2 \in r$  ed  $e_1[X]=e_2[X]$  implicano  $e_1[Y]=e_2[Y]$ .  
 Anche  $r'$  soddisfa  $X \rightarrow Y$ , poiché:  
 $e_1 \in r'$ ,  $e_2 \in r'$ ,  $e_1[X]=e_2[X] \Rightarrow$  (per l'ipotesi di inclusione)  $e_1 \in r$ ,  $e_2 \in r$ ,  $e_1[X]=e_2[X]$   
 $\Rightarrow e_1[Y] = e_2[Y]$ .
- 6)  $r$  soddisfa  $X \rightarrow Y$  significa che  $e_1 \in r$ ,  $e_2 \in r$  ed  $e_1[X]=e_2[X]$  implicano  $e_1[Y]=e_2[Y]$ .  
 Quindi  $r \cap s$  soddisfa  $X \rightarrow Y$ , poiché:  
 $e_1 \in r \cap s$ ,  $e_2 \in r \cap s$ ,  $e_1[X]=e_2[X] \Rightarrow e_1 \in r$ ,  $e_2 \in r$ ,  $e_1[X]=e_2[X] \Rightarrow e_1[Y] = e_2[Y]$ .

$r \cup s$  non soddisfa necessariamente  $X \rightarrow Y$ . Si ponga  $X=A$  ed  $Y=B$  e si considerino le seguenti due relazioni, che consistono di una tupla ciascuna:

$A \ B$	$A \ B$	$A \ B$
$r: \ 1 \ 1$	$s: \ 1 \ 2$	$r \cup s: \ 1 \ 1$
		$1 \ 2$

- 7) *Algoritmo:*  
 $X := T$ ;  
**for**  $A$  **in**  $T$  **do** **if**  $(X-A)^+ = T$  **then**  $X := X-A$ ;  
**return**( $X$ );

Il ciclo for viene ripetuto "a" volte, eseguendo ogni volta un'operazione di chiusura di costo  $O(ap)$ , per un costo totale  $O(a^2p)$ .

Per dimostrare la correttezza dell'algoritmo, si osservi che il ciclo mantiene le seguenti invarianti:

- 1) In ogni momento  $X^+ = T$
- 2) Dopo aver eseguito il corpo del for sugli attributi  $A_1 \dots A_i$ , tutti gli attributi in  $X \cap \{A_1 \dots A_i\}$  (gli attributi di  $X$  già esaminati) sono necessari.

Formalmente:  $\forall A \in X \cap \{A_1 \dots A_i\}. (X-A)^+ \neq T$

Se ne deduce che al termine del ciclo  $X^+ = T$  e tutti gli attributi di  $X$  sono necessari.

- 8) Una copertura canonica per  $F$  è data da:

$$G = \{B \rightarrow C, B \rightarrow D, B \rightarrow E, C \rightarrow B\}.$$

Si osservi che:

- ogni chiave deve contenere  $A$ , poiché  $A$  non è implicata da altri attributi;

- ogni chiave deve contenere B o C, poiché l'insieme degli altri tre attributi (ADE) non è una chiave.

Ne segue tutte le chiavi sono contenute in ABC; da una semplice verifica segue che le chiavi sono esattamente AB ed AC.

Lo schema non è quindi in 3FN a causa delle dipendenze  $C \rightarrow D$ ,  $B \rightarrow E$ , poiché D ed E non sono attributi primi e, C e B non sono chiavi.

Applicando l'algoritmo elementare di sintesi a G otteniamo la seguente decomposizione, che non è comunque la sola possibile: {BCDE, AB (o AC)}.

Si noti che AB (o AC) va aggiunto perché altrimenti lo schema risultante non conterrebbe alcuna relazione che contenga una chiave dello schema originale (e inoltre andrebbe perduto l'attributo A).

- 9) Uno schema  $R \langle T, F \rangle$  è in BCNF quando per ogni dipendenza non banale  $X \rightarrow Y$  in  $F^+$  X è una superchiave. Per verificare se  $R \langle T, F \rangle$  è in BCNF non conviene utilizzare la definizione, ma conviene procedere così: (a) trasformare F in una forma canonica G; (b) verificare che per ogni dipendenza  $X \rightarrow A$  in G, X è una superchiave (o, equivalentemente, una chiave). A sua volta, l'operazione a) si divide in due fasi: (a1) eliminare gli attributi estranei e (a2) eliminare le dipendenze superflue. (a1) si esegue verificando, per ogni  $X \rightarrow A$  in F, e per ogni B in X, se  $A \in (X-B)^+_F$ , nel qual caso B è estraneo; costo:  $O(a^2p^2)$ . (a2) si esegue verificando, per ogni  $X \rightarrow A$  in F se  $A \in (X)^+_{(F-X \rightarrow A)}$ , nel qual caso  $X \rightarrow A$  è superflua; costo:  $O(ap^2)$ ; (b) si esegue verificando, per ogni  $X \rightarrow A$  in G se  $X^+_G = T$ , nel qual caso X è una superchiave; costo:  $O(ap^2)$ . Il costo complessivo della verifica è quindi  $O(a^2p^2)$ .

L'algoritmo di decomposizione individua uno schema  $R_i \langle T_i, F_i \rangle$  ed una dipendenza  $X \rightarrow A$  in  $F_i$  tale che X non sia chiave, scompone  $T_i$  in  $T_i' = XA$ ,  $T_i'' = T_i - A$ , proietta  $F_i$  su  $T_i'$  e  $T_i''$ , e riprende la ricerca di una dipendenza che violi la BCNF, fino a che tutte queste dipendenze non sono state individuate. Lo schema risultante preserva, rispetto allo schema iniziale, i dati ma non necessariamente le dipendenze. Questo algoritmo ha un costo esponenziale, dovuto al costo della proiezione delle dipendenze su  $T'$  e  $T''$ .

- 10) La giunzione di due relazioni r1 ed r2 tali che r1 ha attributi T1 ed r2 ha attributi T2 si definisce come  $(t[T])$  denota la proiezione di una ennupla t sugli attributi T):

$$r1 \text{ join } r2 = \{t \mid t[T1] \in r1, t[T2] \in r2\}.$$

La stessa operazione si può definire, in modo più complesso e più operativo, come indicato sotto, supponendo che  $T1=XY$ ,  $T2=XZ$ ,  $X=\{x1, \dots, xn\}$ , Y e Z siano disgiunti, e infine che  $\{v1, \dots, vn\}$  non appartengano ad XYZ:

$$r1 \text{ join } r2 = \pi_{X,Y,Z}(\sigma_{x1=v1, \dots, xn=vn} ((r1 \text{ rename } x1 \text{ in } v1, \dots, xn \text{ in } vn) \text{ times } r2))$$

Una decomposizione  $r = \{R1 \langle T1, F1 \rangle, \dots, Rn \langle Tn, Fn \rangle\}$  di uno schema  $R \langle T, F \rangle$  preserva i dati quando, per ogni relazione r che soddisfa lo schema  $R \langle T, F \rangle$ , si ha:

$$r = \pi_{T1}(r) \text{ join } \dots \text{ join } \pi_{Tn}(r).$$

- 11) Da fare.
- 12) 1. Le chiavi di R sono A,E, BC, CD  
 2. La decomposizione conserva i dati perché l'attributo in comune A è una chiave per R1 e R2  
 3. R2 è in FNBC perché A e E sono chiavi e i determinanti delle DF  
 4. La decomposizione non conserva le dipendenze perché diverse dipendenze coinvolgono attributi di R1 e R2.
- 13) Da fare.
- 14) Da fare.
- 15) Da fare.
- 16) Finanziamenti(CodDipartimento(CD), IndirizzoDipartimento(ID), CodDocente(D),  
 TelDocente(TD), CodFinanziamento(CF), Ammontare(A), Scadenza(S))

1) Copertura canonica per le dipendenze funzionali valide su tale relazione

$F=\{CD \rightarrow ID; ID \rightarrow CD, D \rightarrow CD; D \rightarrow TD; CF \rightarrow A; CF \rightarrow S\}$

2) Elenco delle chiavi della relazione

D e CF non appaiono a destra di nessuna DF e quindi fanno parte di ogni chiave. Poiché  $\{D,CF\}^+$  coincide con l'insieme degli attributi, essa è l'unica chiave.

3) Tutte le DF violano sia 3FN che la FNBC.

4) Algoritmo di sintesi

I gruppi con lo stesso determinante sono:

$CD \rightarrow ID;$   
 $ID \rightarrow CD,$   
 $D \rightarrow CD; D \rightarrow TD;$   
 $CF \rightarrow A; CF \rightarrow S$

che producono gli schemi

$R1=(CD, ID)$   
 $R2=(ID, CD)$ , si elimina perchè uguale alla precedente.  
 $R3=(D, CD, TD)$   
 $R4=(CF, A, S)$

Poiché nessun schema contiene la chiave D,CF, si aggiunge un nuovo schema che la contiene per garantire la proprietà della conservazione dei dati

$R5=(D,CF)$

5) Algoritmo di analisi

Si considera  $CD \rightarrow ID$  e si decompone:

$R1=(CD,ID)$ , ok

$R2=(D,CD,TD,CF,A,S)$  che non è in FNBC per la DF  $D \rightarrow CD$

$R21=(D,CD,TD)$ , ok

$R22=(D,CF,A,S)$  che non è in FNBC per la DF  $CF \rightarrow A$ :

$R221=(CF, A, S)$ , ok

$R222=(D,CF)$ , ok

6) Come cambiano le dipendenze funzionali se si aggiunge il vincolo che ogni finanziamento riguarda un unico docente

$F=\{CD \rightarrow ID; ID \rightarrow CD, D \rightarrow CD; D \rightarrow TD; CF \rightarrow A; CF \rightarrow S, CF \rightarrow D\}$