

Esame di Basi di Dati

A.A. 2021/2022 – Appello del 07/02/2022

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione relativo al sistema informativo di un centro spaziale, in accordo ai seguenti requisiti. Di ogni astronauta interessano il codice (identificativo), il nome, il cognome, il sesso e le missioni nelle quali ha fatto parte del relativo equipaggio, con i relativi ruoli (almeno uno) ricoperti (ad esempio, capitano, responsabile di rotta, responsabile delle comunicazioni, ecc.). Alcuni astronauti sono senior e di essi interessa anche la nazione di nascita. Di ogni missione interessano il codice (identificativo), la descrizione (una stringa) degli obiettivi e gli astronauti (almeno uno) del relativo equipaggio. Alcune missioni sono speciali e di esse interessa anche la base spaziale di partenza. Si noti che l'equipaggio delle missioni speciali è composto solo da un astronauta, che è necessariamente un astronauta senior. Infine, di ogni missione interessano le varie fasi di cui è composta e che caratterizzano le sue attività. Di ogni fase interessano la data di inizio ed il tipo di attività previste e, nel caso in cui la fase sia la fase finale della missione (una missione ha al massimo una fase finale) interessa anche il numero del relativo rapporto di valutazione. Si noti che una missione cambia fase non più di una volta al giorno e che, ovviamente, per una missione non sono ammesse fasi che iniziano in data posteriore a quella dell'eventuale fase finale.

Problema 2

Si richiede di effettuare la progettazione logica per il sistema informativo citato nel problema 1, tenendo conto dell'indicazione che quando si accede alle informazioni su una missione si vuole sempre sapere se essa è speciale oppure no, ed in caso positivo si vuole sempre sapere la base spaziale di partenza e l'astronauta senior che costituisce il suo equipaggio.

Problema 3

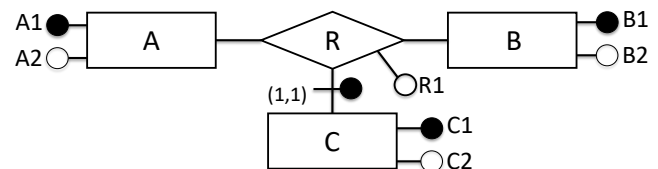
Si consideri una base di dati che comprende le tabelle **Persona**(codice, età, professione), dove sono ammessi valori nulli per gli attributi **età** e **professione**, e **Contratto**(codice, azienda, anno), dove invece non sono ammessi valori nulli. La prima tabella memorizza il codice (chiave primaria), l'età e la professione di un insieme di persone. Ogni tupla della seconda tabella rappresenta un contratto stipulato da una persona con una azienda in un certo anno. Si noti che nulla vieta che un contratto c abbia un "gemello", ossia un contratto stipulato dalla stessa persona di c con la stessa azienda di c nello stesso anno di c . È noto che la base di dati soddisfa il vincolo di integrità referenziale da **Contratto**[codice] a **Persona**[codice]. Scrivere una query in SQL che calcoli quanti sono i contratti nella tabella **Contratto** che non hanno alcun gemello.

Problema 4

Riferendosi ancora alla base di dati menzionata nel problema 3, scrivere una query in algebra relazionale che calcoli il codice delle persone che hanno stipulato almeno un contratto dopo il 2000 e per le quali se non è nota la professione è nota l'età.

Problema 5

Si consideri lo schema concettuale S_1 mostrato qui a destra e si svolga la progettazione logica a partire da esso, illustrando il risultato dell'applicazione di tutti i passi previsti nella metodologia e considerando l'indicazione che quando si accede ad una istanza di **C** non sono in genere di interesse le informazioni sulla sua partecipazione ad **R**.



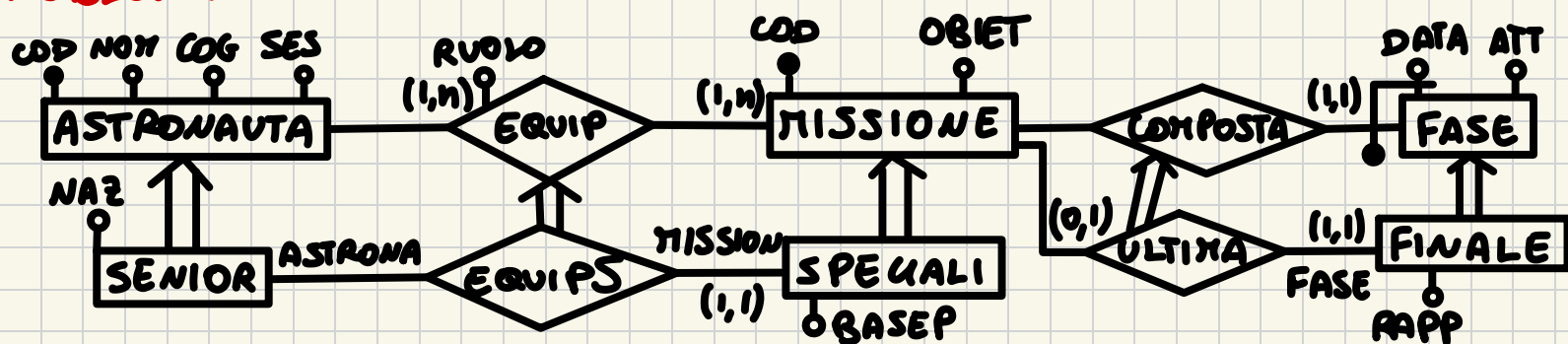
Problema 6

Si consideri lo schema logico L descritto qui di seguito (in cui il valore di default per l'attributo **A** di **R** è 1 e in cui sia **A** sia **B** sono chiavi per **R**):

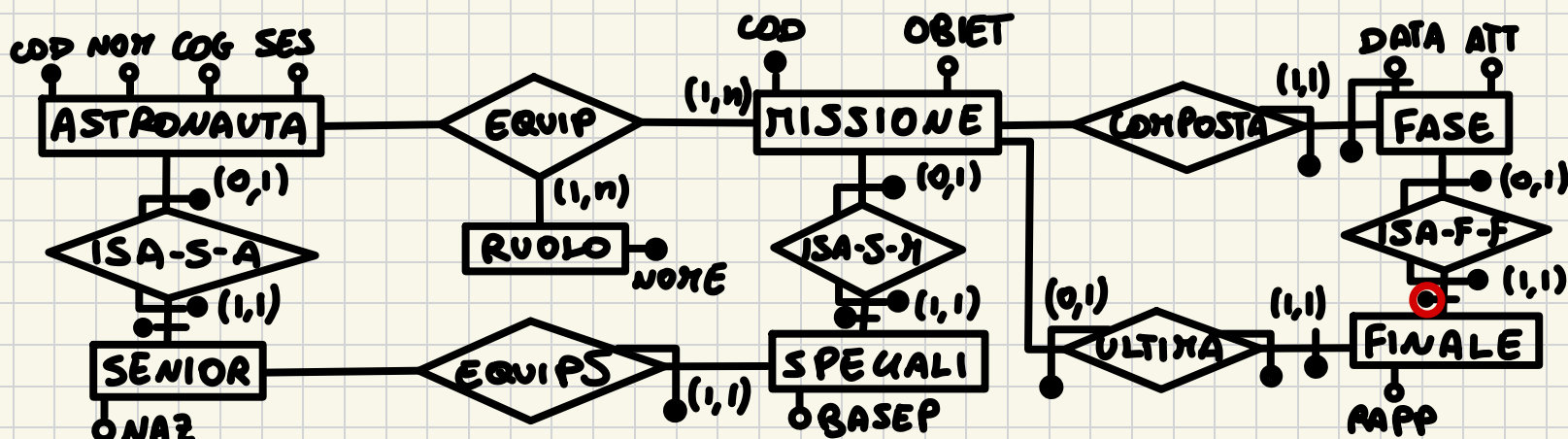
R(A, **B**) default **A** = 1, foreign key **R**[**B**] \subseteq **S**[**C**] on delete set default, chiave **B**
S(C, **D**) foreign key **S**[**D**] \subseteq **R**[**A**] on delete cascade

e si risponda alla seguente domanda: esiste una base di dati G coerente con L tale che la cancellazione dalla base di dati G di una tupla t di **R** provoca un errore? Se la risposta è negativa, si motivi la risposta e se la risposta è positiva, si illustri una tale base di dati G e la corrispondente t e si descriva il motivo dell'errore.

PROBLEMA 1



PROBLEMA 2



VINCOLI:

OGNI ISTANZA DI EQUIPS È ANCHE ISTANZA DI EQUIP
OGNI ISTANZA DI ULTIMA È ANCHE ISTANZA DI COMPOSTA

ASTRONAUTA (COD, NOM, COGNOME, SESSO)

SENIOR (ASTRONAUTA, NAZIONE)

FOREIGN KEY: S[AST] \subseteq AST[COD]

MISSIONE (COD, OBIETTIVI)

INCLUSIONE: M[COD] \subseteq EQUIP[MISSIONE]

SPECIALI (MISSIONE, BASEP)

FOREIGN KEY: SP[MISSIONE] \subseteq MISSIONE[COD]

FOREIGN KEY: SP[MISSIONE] \subseteq EQUIPS[SPECIALI]

RUOLO (NOME)

INCLUSIONE: R[NOME] \subseteq EQUIP[RUOLO]

EQUIPAGGIO (ASTRONAUTA, MISSIONE, RUOLO)

FOREIGN KEY: EQUIP[ASTR] \subseteq ASTR[COD]

FOREIGN KEY: EQUIP[MISS] \subseteq MISS[COD]

FOREIGN KEY: EQUIP[RUOLO] \subseteq RUOLO[NOME]

EQUIPAGGIOSENIOR (SPECIALI, SENIOR)

FOREIGN KEY: EQUIPS[SPECIALI] \subseteq SPECIALI[MISSIONE]

FOREIGN KEY: EQUIPS[SENIOR] \subseteq SENIOR[ASTR]

INCLUSIONE: EQUIPS[SPECIALI, SENIOR] \subseteq EQUIP[SPECIALI, SENIOR]

FASE (MISSIONE, DATA, ATTIVITÀ)

FOREIGN KEY: FASE [MISSIONE] \subseteq MISSIONE [COD]

FINALE (MISSIONE, DATA, RAPPORTO)

FOREIGN KEY: FINALE [MISSIONE, DATA] \subseteq FASE [MISSIONE, DATA]

FOREIGN KEY: FINALE [MISSIONE, DATA] \subseteq ULTIMA [MISSIONE, DATA]

ULTIMA (MISSIONE, DATA)

FOREIGN KEY: ULTIMA [MISSIONE, DATA] \subseteq FINALE [MISSIONE, DATA]



ACCORPIAMO MISSIONE E SPECIALE CON UNA FLAG

MISSIONE (COD, OBIETTIVI, BASEP^{*})

INCLUSIONE: M [COD] \subseteq EQUIP [MISSIONE]

FOREIGN KEY: MISSIONE [COD] \subseteq EQUIPS [MISSIONE]

ACCORPIAMO CON EQUIPS

MISSIONE (COD, OBIETTIVI, BASEP^{*}, SENIOR^{*})

INCLUSIONE: MISS [SENIOR, COD] \subseteq EQUIP [SENIOR, COD]

FOREIGN KEY: MISS [SENIOR] \subseteq SENIOR [ASTR]

BASEP È NULL \Leftrightarrow SENIOR È NULL

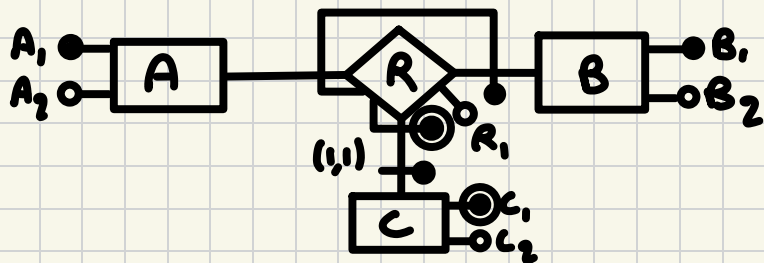
PROBLEMA 3

```
SELECT COUNT(*)  
FROM (SELECT *  
      FROM CONTRATTO  
      EXCEPT  
      SELECT C.CODICE, C.AZIENDA, C.ANNO  
      FROM CONTRATTO C  
      GROUP BY (C.CODICE, C.AZIENDA, C.ANNO) AS X  
      HAVING COUNT(X) > 1  
)
```

PROBLEMA 4

PROJ_{CODICE} (SEL_{ANNO > 2000} (CONTRATTO) JOIN SEL_{PROF IS NOT NULL OR ETÀ IS NOT NULL} (PERSONA))

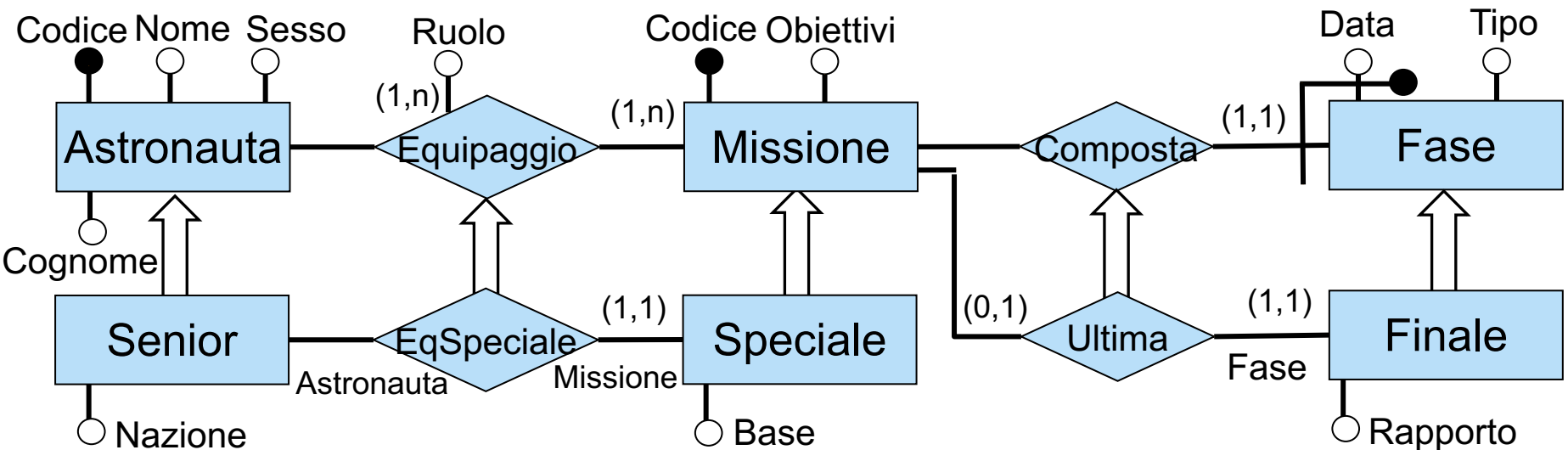
PROBLEMA 5



$A(A_1, A_2)$
 $B(\underline{B_1}, B_2)$
 $C(\underline{C_1}, C_2)$
 $FK: C[C_1] \subseteq R[C_1]$
 $R(A, B, \underline{C}, R_1)$
 $FK: R[A] \subseteq A[A_1]$
 $FK: R[B] \subseteq B[B_1]$
 $FK: R[C] \subseteq C[C_1]$
 $UNIQUE: A, B$

Problema 1 – Schema concettuale

Schema concettuale:

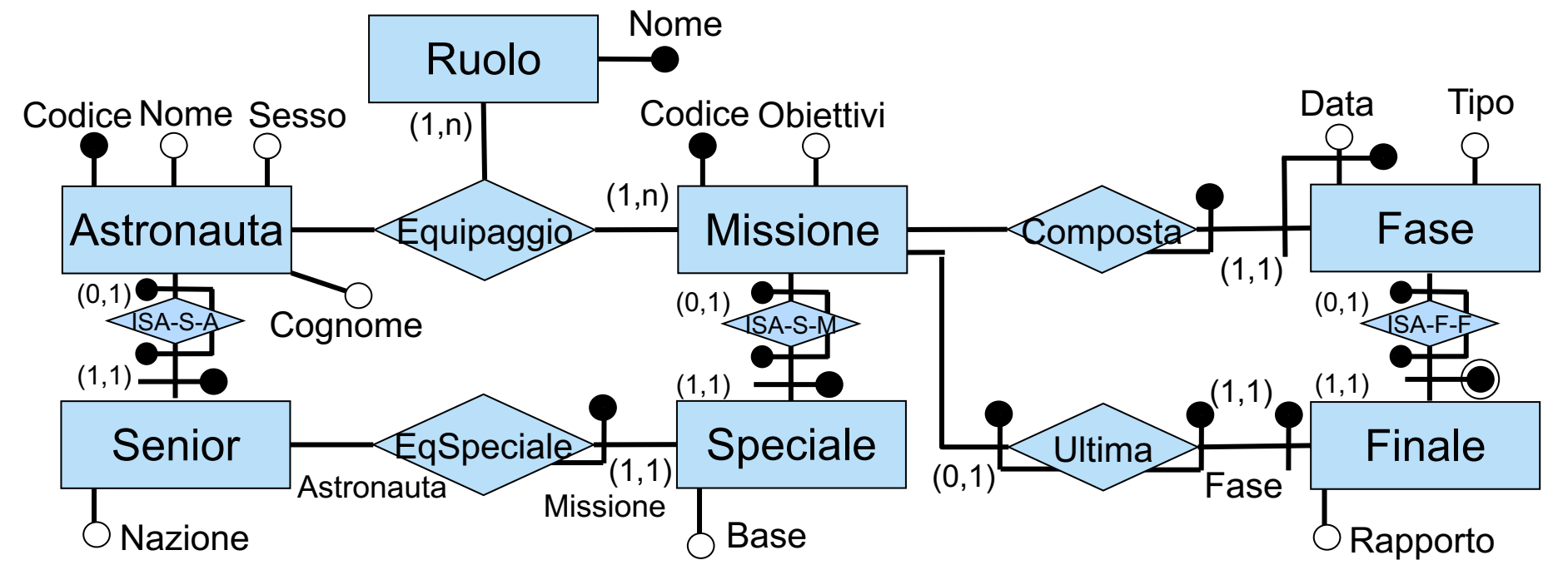


Vincolo esterno – Per ogni istanza I dello schema concettuale:

Per ogni $m, f1, f2, d1, d2$, se $\langle \text{Missione}:m, \text{Fase}:f1 \rangle$ è in $\text{Istanze}(I, \text{Composta})$, $\langle \text{Missione}:m, \text{Fase}:f2 \rangle$ è in $\text{Istanze}(I, \text{Ultima})$ e $\langle f1, d1 \rangle$ e $\langle f2, d2 \rangle$ sono in $\text{Istanze}(I, \text{Data})$, allora $d2 > d1$.

Problema 2 – Schema concettuale ristrutturato

Schema concettuale ristrutturato:



Vincoli esterni – Per ogni istanza I dello schema concettuale:

1. Per ogni s,p,a,m, se $\langle \text{Astronauta:s}, \text{Missione:p} \rangle$ è in $\text{Istanze}(I, \text{Eq-Speciale})$, $\langle \text{Senior:s}, \text{Astronauta:a} \rangle$ è in $\text{Istanze}(I, \text{ISA-S-A})$ e $\langle \text{Speciale:p}, \text{Missione:m} \rangle$ è in $\text{Istanze}(I, \text{ISA-S-M})$, allora esiste almeno un r in $\text{Istanze}(I, \text{Ruolo})$ tale che $\langle \text{Astronauta:a}, \text{Missione:m}, \text{Ruolo:r} \rangle$ è in $\text{Istanze}(I, \text{Equipaggio})$.
2. Per ogni m,f,s, se $\langle \text{Missione:m}, \text{Fase:f} \rangle$ è in $\text{Istanze}(I, \text{Ultima})$ e $\langle \text{Finale:f}, \text{Fase:s} \rangle$ è in $\text{Istanze}(I, \text{ISA-F-F})$, allora $\langle \text{Missione:m}, \text{Fase:s} \rangle$ è in $\text{Istanze}(I, \text{Composta})$.
3. Per ogni m,f1,f2,d1,d2, se $\langle \text{Missione:m}, \text{Fase:f1} \rangle$ è in $\text{Istanze}(I, \text{Composta})$, $\langle \text{Missione:m}, \text{Fase:f2} \rangle$ è in $\text{Istanze}(I, \text{Ultima})$, $\langle \text{Finale:f2}, \text{Fase:f3} \rangle$ è in $\text{Istanze}(I, \text{ISA-F-F})$ e $\langle f1, d1 \rangle$ e $\langle f3, d2 \rangle$ sono in $\text{Istanze}(I, \text{Data})$, allora $d2 > d1$.

Problema 2 – Traduzione diretta

Schema logico prodotto
dalla traduzione diretta:

Astronauta(codice, nome, cognome, sesso)
Senior(codice, nazione)
 foreign key: Senior[codice] \subseteq Astronauta[codice]
Equipaggio(astronauta, missione, ruolo)
 foreign key: Equipaggio[astronauta] \subseteq Astronauta[codice]
 foreign key: Equipaggio[missione] \subseteq Missione[codice]
 foreign key: Equipaggio[ruolo] \subseteq Ruolo[nome]
Missione(codice, obiettivi)
 inclusione: Missione[codice] \subseteq Equipaggio[missione]
Ruolo(nome)
 inclusione: Ruolo[nome] \subseteq Equipaggio[ruolo]
Speciale(codice, base)
 foreign key: Speciale[codice] \subseteq Missione[codice]
 foreign key: Speciale[codice] \subseteq EqSpeciale[codice]
EqSpeciale(astronauta, missione)
 foreign key: EqSpeciale[missione] \subseteq Speciale[codice]
 foreign key: EqSpeciale[astronauta] \subseteq Senior[codice]
 inclusione: EqSpeciale[astronauta, missione] \subseteq Equipaggio[astronauta, missione]
Fase(data, missione, tipo)
 foreign key: Fase[missione] \subseteq Missione[codice]
Finale(data, missione, rapporto)
 foreign key: Finale[data, missione] \subseteq Fase[data, missione]
 vincolo esterno data > all (select data from Fase where Fase.missione = missione and
 (Fase.data, Fase.missione) not in (select data, missione from Finale)

Problema 2 – Schema logico ristrutturato

Schema logico ristrutturato (primo passo):

(1) accorpamento tra le tabelle Missione e Speciale debolmente accoppiate

Astronauta(codice, nome, cognome, sesso)
Senior(codice, nazione)
foreign key: Senior[codice] \subseteq Astronauta[codice]
Equipaggio(astronauta, missione, ruolo)
foreign key: Equipaggio[astronauta] \subseteq Astronauta[codice]
foreign key: Equipaggio[missione] \subseteq Missione[codice]
foreign key: Equipaggio[ruolo] \subseteq Ruolo[nome]
Missione(codice, obiettivi, base*)
inclusione: Missione[codice] \subseteq Equipaggio[missione]
Ruolo(nome)
inclusione: Ruolo[nome] \subseteq Equipaggio[ruolo]
EqSpeciale(astronauta, missione)
foreign key: EqSpeciale[astronauta] \subseteq Senior[codice]
inclusione: EqSpeciale[missione] \subseteq (select codice from Missione where base is not null)
inclusione: EqSpeciale[astronauta, missione] \subseteq Equipaggio[astronauta, missione]
Fase(data, missione, tipo)
foreign key: Fase[missione] \subseteq Missione[codice]
Finale(data, missione, rapporto)
foreign key: Finale[data, missione] \subseteq Fase[data, missione]
vincolo esterno data > all (select data from Fase where Fase.missione = missione and (Fase.data, Fase.missione) not in (select data, missione from Finale))

Vista Speciale = (select codice, base from Missione where base is not null)

Problema 2 – Schema logico ristrutturato

Schema logico ristrutturato (primo e secondo passo):

(1) accorpamento tra le tabelle Missione e Speciale debolmente accoppiate

(2) ulteriore accorpamento tra le tabelle EqSpeciale e Missione debolmente accoppiate

Astronauta(codice,nome,cognome,sex)

Senior(codice,nazione)

foreign key: Senior[codice] \subseteq Astronauta[codice]

Equipaggio(astronauta,missione,ruolo)

foreign key: Equipaggio[astronauta] \subseteq Astronauta[codice]

foreign key: Equipaggio[missione] \subseteq Missione[codice]

foreign key: Equipaggio[ruolo] \subseteq Ruolo[nome]

Missione(codice,obiettivi,base*,senior*)

foreign key: Missione[senior,codice] \subseteq Equipaggio[astronauta,missione]

foreign key: Missione[senior] \subseteq Senior[codice]

vincolo di tupla: base è null se e solo se senior è null

Ruolo(nome)

inclusione: Ruolo[nome] \subseteq Equipaggio[ruolo]

Fase(data,missione,tipo)

foreign key: Fase[data,missione] \subseteq Fase[data,missione]

Finale(data,missione,rapporto)

foreign key: Finale[missione] \subseteq Missione[codice]

vincolo esterno data > all (select data from Fase where Fase.missione = missione and
(Fase.data,Fase.missione) not in (select data, missione from Finale))

Vista Speciale = (select codice,base from Missione where base is not null)

Vista EqSpeciale = (select astronauta,codice from Missione where base is not null)

Problema 3 – testo e soluzione

Si consideri una base di dati che comprende le tabelle Persona(codice,età,professione), dove sono ammessi valori nulli per gli attributi età e professione, e Contratto(codice,azienda,anno), dove invece non sono ammessi valori nulli. La prima tabella memorizza il codice (chiave primaria), l'età e la professione di un insieme di persone. Ogni tupla della seconda tabella rappresenta un contratto stipulato da una persona con una azienda in un certo anno. Si noti che nulla vieta che un contratto c abbia un “gemello”, ossia un contratto stipulato dalla stessa persona di c con la stessa azienda di c nello stesso anno di c. È noto che la base di dati soddisfa il vincolo di integrità referenziale da Contratto[codice] a Persona[codice]. Scrivere una query in SQL che calcoli quanti sono i contratti nella tabella Contratto che non hanno alcun gemello.

Possibile soluzione:

calcola quante sono le righe di t che hanno il valore dell'attributo “quantiduplicati” uguale ad 1

```
select count(*)  
from (select count(*) as quantiduplicati  
      from Contratto  
      group by codice, azienda, anno) t  
where t.quantiduplicati = 1
```

calcola la tabella t che ha una riga per ogni combinazione diversa di <codice,azienda,anno>, dove ogni riga contiene nell'attributo “quantiduplicati” un valore che indica quante duplicazioni ci sono di quella combinazione

Problema 4 – testo e soluzione

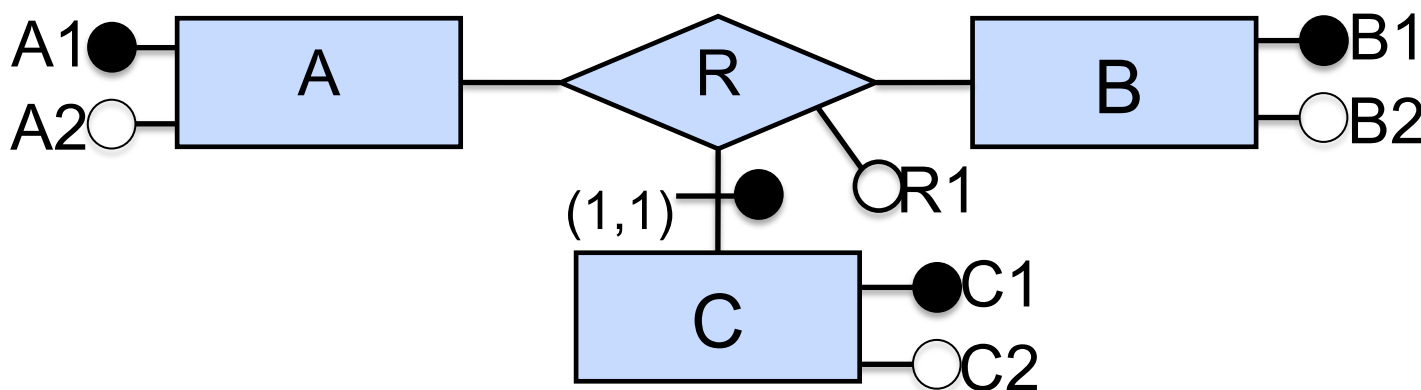
Riferendosi ancora alla base di dati menzionata nel problema 3, scrivere una query in algebra relazionale che calcoli il codice delle persone che hanno stipulato almeno un contratto dopo il 2000 e per le quali se non è nota la professione è nota l'età.

Soluzione:

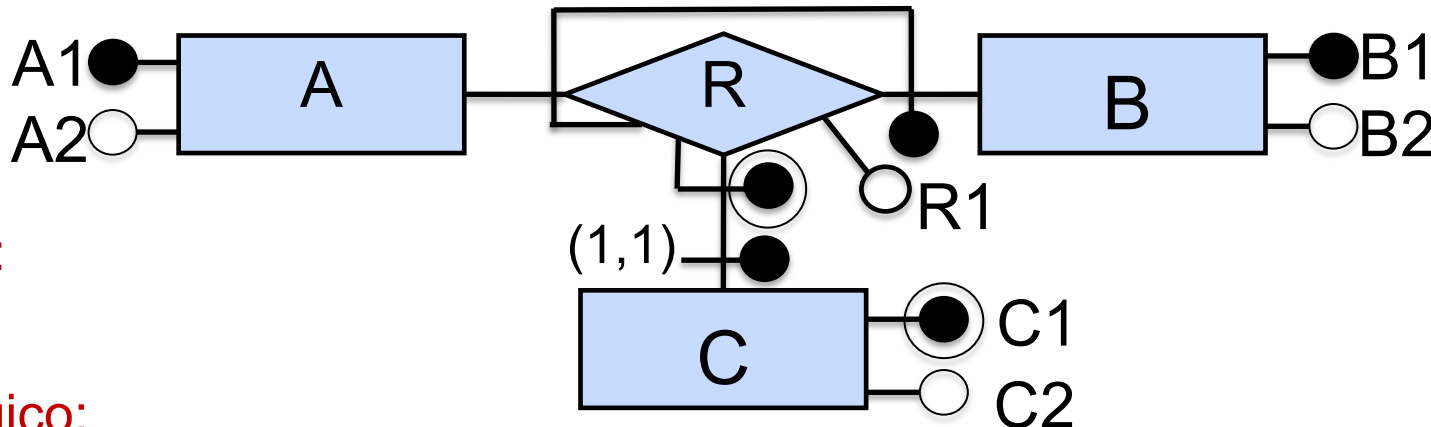
$\text{SEL}_{\text{professione is not null OR età is not null}}(\text{Persona}) \text{ JOIN } \text{SEL}_{\text{anno} > 2000}(\text{Contratto})$

Problema 5 – soluzione

Schema concettuale di partenza:



Schema concettuale ristrutturato:



Schema logico:

$A(\underline{A1}, A2)$	$R(A, B, \underline{C}, R1)$
$B(\underline{B1}, B2)$	foreign key: $R[A] \subseteq A[A1]$
$C(\underline{C1}, C2)$	foreign key: $R[B] \subseteq B[B1]$
foreign key: $C[C1] \subseteq R[C]$	foreign key: $R[C] \subseteq C[C1]$
	chiave A,B

Problema 6 – soluzione

Si noti che:

1. Sia A sia B sono chiavi per R, ma solo A è la chiave primaria e quindi è soggetto al vincolo NOT NULL; si accettano, invece, valori nulli nell'attributo B, perché il vincolo unique non forza a valori diversi da null e sono anche ammesse diverse tuple con il valore null nell'attributo B.
2. Il vincolo di chiave esterna da R[B] a S[C] ha la politica “on delete default” ed il default per l'attributo B è null (visto che non è stato dichiarato alcun default esplicito).
3. Il vincolo di chiave esterna da S[D] a R[A] ha la politica “on delete cascade”.

Proprio il punto 2) sopra descritto fa capire che se si dovesse eseguire una cancellazione di una tupla di S e questa cancellazione lasciasse una tupla t di R che non soddisfa il vincolo di foreign key da R[B] a S[C], la politica “on delete default” porrebbe t.B a null. Siccome B è unique in R (e non primary key) questo valore è sempre ammesso ed inoltre non violerà mai il vincolo di foreign key da R[B] a S[C], perché il valore nullo non viola mai un vincolo di foreign key.

Consideriamo allora una base di dati G qualunque, coerente con lo schema L ed assumiamo che si elimini da G una tupla t di R. Indichiamo con X il valore dell'attributo A di t. Per il punto 3) sopra descritto, a fronte di “on delete cascade”, verranno eliminate tutte le tuple t' di S tali che il valore dell'attributo D di t' è X. Prendiamo adesso in considerazione tutte le tuple t'' di R tali che nell'attributo B c'è un valore che compare nell'attributo C di S di una delle tuple t' eliminate. Per il punto 2) sopra descritto, a fronte di “on delete set default”, in tutte queste tuple t'' il valore dell'attributo B viene modificato in null ed è chiaro che questa modifica non pone alcun problema di violazione di vincoli nella base di dati, come osservato nei punti 1) e 2) sopra descritti. Osserviamo che tutti i vincoli di chiave e tutti i vincoli di foreign key sono chiaramente soddisfatti. Concludiamo che a fronte della cancellazione di una tupla t di R le politiche di violazione dei vincoli definite in L operano altre cancellazioni in modo da portare sempre la base di dati in uno stato in cui tutti i vincoli sono soddisfatti. Quindi la risposta alla domanda è che non esiste alcuna base di dati G coerente con L tale che la cancellazione dalla base di dati G di una tupla di R provoca un errore.