

BD: prima prova di verifica del 4/11/2013

1. (Obbligatorio) Si consideri la seguente interrogazione sulla relazione S(A int, B string, C int):

SELECT	B, MAX(C) AS M	S			
FROM	S	RID	A	B	C
WHERE	A > 1	1	1	a	3
GROUP BY	B	2	2	b	2
HAVING	COUNT(*) > 1;	3	3	c	5
		4	1	a	6
		5	3	c	10

Si diano (1) l'albero logico dell'interrogazione, (2) il tipo del risultato e (3) il valore del risultato.

2. Una catena di negozi di abbigliamento vuole tenere traccia dei capi in ciascun negozio e delle vendite. La catena tratta ad ogni stagione un insieme di nuovi modelli, identificati da un codice, per ciascuno dei quali interessa conoscere un nome, un tipo, e l'insieme dei colori e delle taglie disponibili. I colori sono scelti da un insieme predefinito. Ogni negozio tiene traccia di ogni singolo capo che tratta. Per ogni capo interessano il modello, la taglia, il colore. Per i capi non ancora venduti interessano anche il negozio in cui il capo si trova al momento ed il prezzo di listino. Per quelli già venduti interessano la data di vendita, il prezzo effettivo di vendita ed il dipendente che ne ha curato la vendita. Di ogni dipendente interessa il nome, il cognome, la data di assunzione. Ogni dipendente può essere trasferito da un negozio all'altro ed interessa tenere traccia di tutti i periodi che ciascun dipendente ha passato in ciascuno dei negozi nei quali ha lavorato, tenendo conto, per ciascun periodo, della data di inizio e fine del lavoro.
- (a) Si definisca lo schema concettuale della base di dati
 - (b) Si traduca lo schema concettuale in uno schema relazionale grafico
 - (c) Si elenchino per ogni relazione gli attributi, con particolare attenzione alle chiavi primarie ed esterne.

3. Si consideri il seguente schema relazionale:

Ordini(IdOrdine, Data, IdFornitore*)
RigheOrdini(IdOrdine*, NumeroRiga, Quantità, IdProdotto*, Ammontare)
Prodotti(IdProdotto, Nome, Prezzo, IdProduttore*)
Fornitori(IdFornitore, Nome, Città, Nazione)
Produttori(IdProduttore, Nome, Nazione)

Scrivere le interrogazioni SQL che restituiscono le seguenti informazioni:

- (a) L'IdOrdine e la Data di tutti gli ordini che hanno un prodotto di nome 'KK'
- (b) Per ogni ordine il cui ammontare totale (la somma dell'ammontare di tutte le righe) supera 1000, l'IdOrdine e la Data, il numero di righe, e l'Ammontare totale.
- (c) Per ogni ordine in cui almeno l'Ammontare di una riga supera 100, l'IdOrdine e la Data, il numero di righe, e l'Ammontare totale.
- (d) Per ogni ordine in cui l'Ammontare di ciascuna riga supera 100, l'IdOrdine e la Data, il numero di righe, e l'Ammontare totale.
- (e) Il nome e la nazione dei fornitori per i quali, in ogni ordine a loro inviato, appare almeno un prodotto dal prezzo superiore a 100.
- (f) (Opzionale) (facoltativo) Il nome di ciascun fornitore F e produttore P tali che al fornitore F è stato ordinato ogni prodotto del produttore P.

BD: prima prova di verifica del 4/11/2013. Soluzione

1. (Obbligatorio)

Si consideri la seguente interrogazione sulla relazione S(A int, B string, C int):

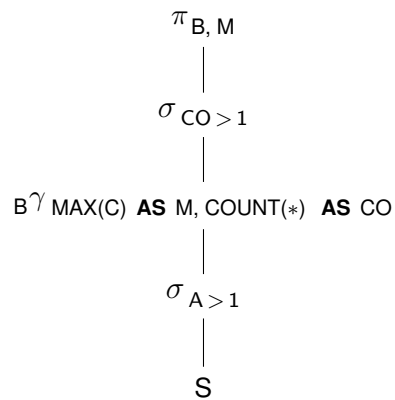
```

SELECT  B, MAX(C) AS M
FROM    S
WHERE   A > 1
GROUP BY B
HAVING  COUNT(*) > 1;
    
```

S			
RID	A	B	C
1	1	a	3
2	2	b	2
3	3	c	5
4	1	a	6
5	3	c	10

Si diano (1) l'albero logico dell'interrogazione, (2) il tipo del risultato e (3) il valore del risultato.

Albero logico:



Tipo risultato: {(B string, M int)}

Risultato:

B	M
C	10

2. Una catena di negozi di abbigliamento vuole tenere traccia dei capi in ciascun negozio e delle vendite. La catena tratta ad ogni stagione un insieme di nuovi modelli, identificati da un codice, per ciascuno dei quali interessa conoscere un nome, un tipo, e l'insieme dei colori e delle taglie disponibili. I colori sono scelti da un insieme predefinito. Ogni negozio tiene traccia di ogni singolo capo che tratta. Per ogni capo interessano il modello, la taglia, il colore. Per i capi non ancora venduti interessano anche il negozio in cui il capo si trova al momento ed il prezzo di listino. Per quelli già venduti interessano la data di vendita, il prezzo effettivo di vendita ed il dipendente che ne ha curato la vendita. Di ogni dipendente interessa il nome, il cognome, la data di assunzione. Ogni dipendente può essere trasferito da un negozio all'altro ed interessa tenere traccia di tutti i periodi che ciascun dipendente ha passato in ciascuno dei negozi nei quali ha lavorato, tenendo conto, per ciascun periodo, della data di inizio e fine del lavoro.

(a) Si definisca lo schema concettuale della base di dati.

(b) Si traduca lo schema concettuale in uno schema relazionale grafico.

Si elenchino per ogni relazione gli attributi, con particolare attenzione alle chiavi primarie ed esterne: gli attributi di una chiave primaria sono marcati con PK e quelli di una chiave sono marcati con K).

Lo schema concettuale si trova in Figura 1. In tale schema i colori sono trattati come una classe perché il committente richiede che siano scelti da un insieme predeterminato. Sarebbe stata certamente accettabile anche una soluzione in cui le taglie fossero state trattate anch'esse come i colori.

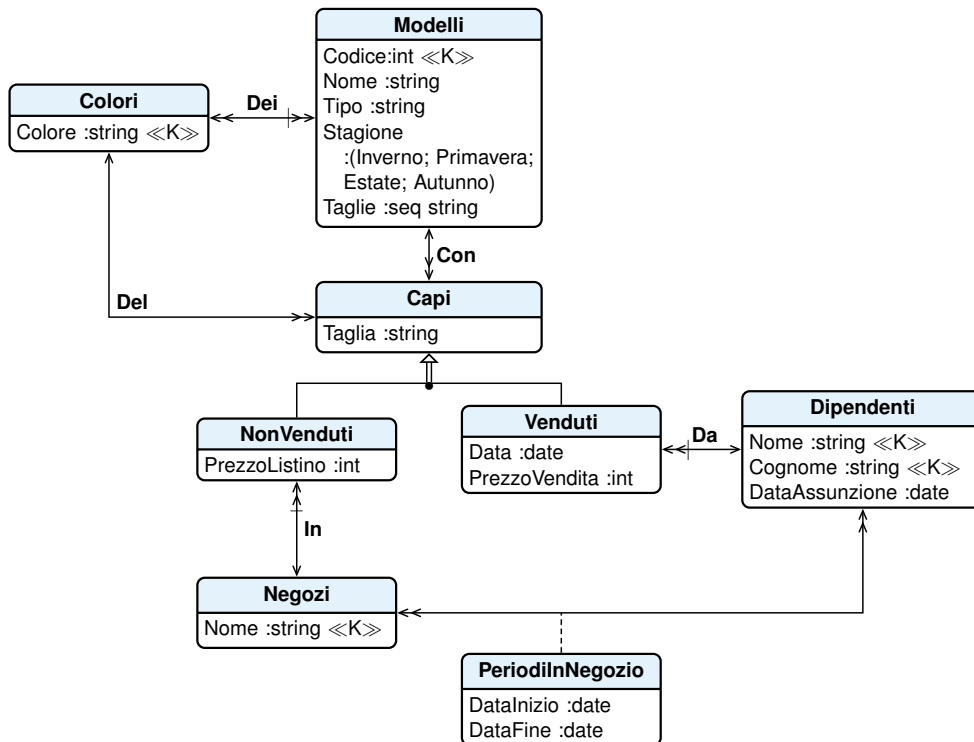


Figura 1: Schema concettuale

Lo schema relazionale si trova in Figura 2.

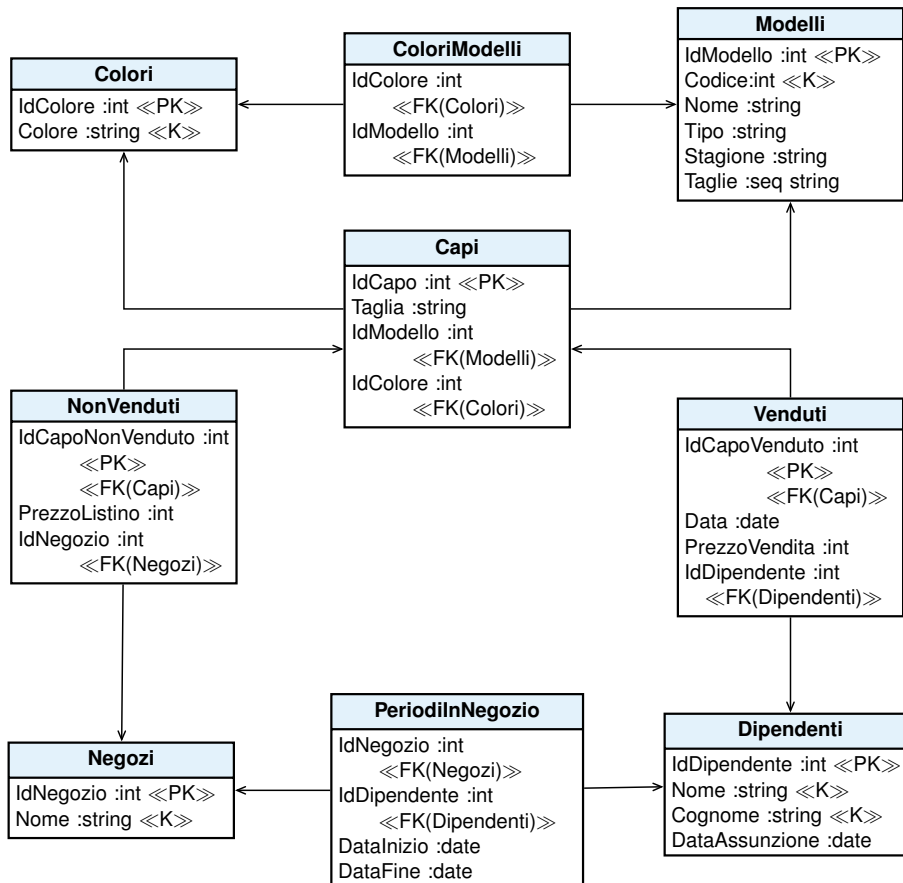


Figura 2: Schema relazionale

3. Si consideri il seguente schema relazionale:

Ordini(IdOrdine, Data, IdFornitore*)
 RigheOrdini(IdOrdine*, NumeroRiga, Quantità, IdProdotto*, Ammontare)
 Prodotti(IdProdotto, Nome, Prezzo, IdProduttore*)
 Fornitori(IdFornitore, Nome, Città, Nazione)
 Produttori(IdProduttore, Nome, Nazione)

Scrivere le interrogazioni SQL che restituiscono le seguenti informazioni:

- (a) L'IdOrdine e la Data di tutti gli ordini che hanno un prodotto di nome 'KK'

```
SELECT DISTINCT O.IdOrdine, O.Data
FROM Ordini O, RigheOrdini RO, Prodotti P
WHERE O.IdOrdine = RO.IdOrdine AND RO.IdProdotto = P.IdProdotto
AND P.Nome = 'KK';
```

- (b) Per ogni ordine il cui ammontare totale (la somma dell'ammontare di tutte le righe) supera 1000, l'IdOrdine e la Data, il numero di righe, e l'Ammontare totale.

```
SELECT O.IdOrdine, O.Data, COUNT(*), SUM(RO.Ammontare)
FROM Ordini O, RigheOrdini RO
WHERE O.IdOrdine = RO.IdOrdine
GROUP BY O.IdOrdine, O.Data
HAVING SUM(RO.Ammontare) > 1000;
```

- (c) Per ogni ordine in cui almeno l'Ammontare di una riga supera 100, l'IdOrdine e la Data, il numero di righe, e l'Ammontare totale.

```
SELECT O.IdOrdine, O.Data, COUNT(*), SUM(RO.Ammontare)
FROM Ordini O, RigheOrdini RO
WHERE O.IdOrdine = RO.IdOrdine
AND EXISTS
(
SELECT *
FROM RigheOrdini RO2
WHERE O.IdOrdine = RO2.IdOrdine AND RO2.Ammontare > 100
)
GROUP BY O.IdOrdine, O.Data;
```

Altra soluzione con il **GROUP BY**.

```
SELECT O.IdOrdine, O.Data, COUNT(*), SUM(RO.Ammontare)
FROM Ordini O, RigheOrdini RO
WHERE O.IdOrdine = RO.IdOrdine
GROUP BY O.IdOrdine, O.Data
HAVING MAX(RO.Ammontare) > 100;
```

- (d) Per ogni ordine in cui l'Ammontare di ciascuna riga supera 100, l'IdOrdine e la Data, il numero di righe, e l'Ammontare totale.

```
SELECT O.IdOrdine, O.Data, COUNT(*), SUM(RO.Ammontare)
FROM Ordini O, RigheOrdini RO
WHERE O.IdOrdine = RO.IdOrdine
AND (FOR ALL R IN RigheOrdini WHERE O.IdOrdine = R.IdOrdine
: R.Ammontare > 100)
GROUP BY O.IdOrdine, O.Data;
```

```

SELECT O.IdOrdine, O.Data, COUNT(*), SUM(RO.Ammontare)
FROM Ordini O, RigheOrdini RO
WHERE O.IdOrdine = RO.IdOrdine
AND NOT EXISTS
(
SELECT *
FROM RigheOrdini R
WHERE O.IdOrdine = R.IdOrdine AND NOT( R.Ammontare > 100)
)
GROUP BY O.IdOrdine, O.Data;

```

Altra soluzione con il **GROUP BY**.

```

SELECT O.IdOrdine, O.Data, COUNT(*), SUM(RO.Ammontare)
FROM Ordini O, RigheOrdini RO
WHERE O.IdOrdine = RO.IdOrdine
GROUP BY O.IdOrdine, O.Data
HAVING MIN(RO.Ammontare) > 100;

```

- (e) Il nome e la nazione dei fornitori per i quali, in ogni ordine a loro inviato, appare almeno un prodotto dal prezzo superiore a 100.

```

SELECT F.Nome, F.Nazione
FROM Fornitori F
WHERE FOR ALL O IN Ordini WHERE O.IdFornitore = F.IdFornitore
      : ( FOR SOME RO IN RigheOrdini, P IN Prodotti
          WHERE RO.IdProdotto = P.IdProdotto AND RO.IdOrdine = O.IdOrdine
          : P.Prezzo > 100 );

```

```

SELECT F.Nome, F.Nazione
FROM Fornitori F
WHERE NOT EXISTS
(
SELECT *
FROM Ordini O
WHERE O.IdFornitore = F.IdFornitore
AND NOT EXISTS (
SELECT *
FROM RigheOrdini RO, Prodotti P
WHERE RO.IdProdotto = P.IdProdotto AND RO.IdOrdine = O.IdOrdine
AND P.Prezzo > 100 )
);

```

La soluzione precedente restituisce anche i fornitori che non hanno ricevuto alcun ordine. Per restringersi ai soli fornitori che abbiano ricevuto qualche ordine, è possibile aggiungere una giunzione al livello più esterno.


```

SELECT DISTINCT F.Nome, F.Nazione
FROM Fornitori F, Ordini O
WHERE O.IdFornitore = F.IdFornitore AND
FOR ALL O2 IN Ordini WHERE O2.IdFornitore = F.IdFornitore
      : ( FOR SOME RO IN RigheOrdini, P IN Prodotti
          WHERE RO.IdProdotto = P.IdProdotto AND RO.IdOrdine = O.IdOrdine
          : P.Prezzo > 100 );

```

```

SELECT DISTINCT F.Nome, F.Nazione
FROM Fornitori F, Ordini O
WHERE O.IdFornitore = F.IdFornitore AND
NOT EXISTS
(
  SELECT *
FROM Ordini O2
WHERE O2.IdFornitore = F.IdFornitore
AND NOT EXISTS (
    SELECT *
    FROM RigheOrdini RO, Prodotti P
    WHERE RO.IdProdotto = P.IdProdotto AND RO.IdOrdine = O2.IdOrdine
    AND P.Prezzo > 100 )
);

```

- (f) (facoltativo) Il nome di ciascun fornitore F e produttore P tali che al fornitore F è stato ordinato ogni prodotto del produttore P.

Mostriamo alcune diverse soluzioni, tra quelle che ritornano solo i produttori che producono almeno un prodotto.

Questa prima soluzione usa la giunzione più esterna per restringere l'attenzione alle coppie fornitore-produttore F-PP tali che il fornitore F ha ordinato almeno un prodotto al produttore PP. Per tali coppie, viene verificata la condizione espressa dalla query, ovvero:

$\forall P \in \text{Prodotti}$:

$P.\text{IdProduttore} = PP.\text{IdProduttore}$

$\Rightarrow \exists RO \in \text{RigheOrdini}, O \in \text{Ordini}$:

$F.\text{IdFornitore} = O.\text{IdFornitore} \text{ And } RO.\text{IdOrdine} = O.\text{IdOrdine}$

$\text{And } RO.\text{IdProdotto} = P.\text{IdProdotto}$

```

SELECT DISTINCT F.Nome, PP.Nome
FROM Fornitori F, Ordini O, RigheOrdini RO, Prodotti P, Produttori PP
WHERE O.IdFornitore = F.IdFornitore
      AND RO.IdOrdine = O.IdOrdine
      AND RO.IdProdotto = P.IdProdotto
      AND P.IdProduttore = PP.IdProduttore
AND FOR ALL P2 IN Prodotti WHERE P2.IdProduttore = PP.IdProduttore
      : ( FOR SOME RO2 IN RigheOrdini, O2 IN Ordini
        WHERE F.IdFornitore = O2.IdFornitore AND RO2.IdOrdine = O2.IdOrdine
        : RO2.IdProdotto = P2.IdProdotto );

```

```

SELECT DISTINCT F.Nome, PP.Nome
FROM Fornitori F, Ordini O, RigheOrdini RO, Prodotti P, Produttori PP
WHERE O.IdFornitore = F.IdFornitore
      AND RO.IdOrdine = O.IdOrdine
      AND RO.IdProdotto = P.IdProdotto
      AND P.IdProduttore = PP.IdProduttore
      AND NOT EXISTS
      (
        SELECT *
        FROM Prodotti P2
        WHERE P2.IdProduttore = PP.IdProduttore
          AND NOT EXISTS
          (
            SELECT *
            FROM RigheOrdini RO2, Ordini O2
            WHERE F.IdFornitore = O2.IdFornitore
              AND RO2.IdOrdine = O2.IdOrdine
              AND RO2.IdProdotto = P2.IdProdotto )
          );

```

In realtà non è necessario restringersi alle coppie F-PP che sono correlate, ma è sufficiente restituire tutte le coppie F-PP che soddisfano la condizione della query, come esemplificato dalla soluzione successiva. In questo caso al livello più esterno lasciamo la sola giunzione tra prodotti e produttori che serve a garantirci che il produttore PP abbia almeno un prodotto. Si osservi che la congiunzione di questa proprietà con quella verificata internamente garantisce che il fornitore abbia almeno un ordine con un prodotto del produttore PP, per cui questa soluzione equivale del tutto alla precedente.

```

SELECT DISTINCT F.Nome, PP.Nome
FROM Fornitori F, Prodotti P, Produttori PP
WHERE P.IdProduttore = PP.IdProduttore AND
FOR ALL P2 IN Prodotti WHERE P2.IdProduttore = PP.IdProduttore
      : ( FOR SOME RO IN RigheOrdini, O IN Ordini
        WHERE F.IdFornitore = O.IdFornitore AND RO.IdOrdine = O.IdOrdine
        : RO.IdProdotto = P.IdProdotto );

```

```

SELECT DISTINCT F.Nome, PP.Nome
FROM Fornitori F, Prodotti P, Produttori PP
WHERE P.IdProduttore = PP.IdProduttore AND
NOT EXISTS
(
SELECT *
FROM Prodotti P2
WHERE P2.IdProduttore = PP.IdProduttore
AND NOT EXISTS
(
SELECT *
FROM RigheOrdini RO, Ordini O
WHERE F.IdFornitore = O.IdFornitore
AND RO.IdOrdine = O.IdOrdine
AND RO.IdProdotto = P2.IdProdotto )
);

```

La prossima soluzione verifica la stessa condizione verificando che il numero di prodotti distinti di PP ordinati da F coincida con il numero totale di prodotti di PP (Mas-saro).

```

SELECT    F.Nome, PP.Nome
FROM      Fornitori F, Ordini O, RigheOrdini RO, Prodotti P, Produttori PP
WHERE     O.IdFornitore = F.IdFornitore
           AND RO.IdOrdine = O.IdOrdine
           AND RO.IdProdotto = P.IdProdotto
           AND P.IdProduttore = PP.IdProduttore
GROUP BY F.IdFornitore, F.Nome, PP.IdProduttore, PP.Nome
HAVING   COUNT(DISTINCT P.IdProdotto) =
           (
           SELECT COUNT(*)
           FROM    Prodotti P2
           WHERE  P2.IdProduttore = PP.IdProduttore
           );

```

La prossima soluzione è equivalente, ma usa viste temporanee dichiarate con l'operatore **WITH** (Aceto).

```

WITH      NoProdottiDiProduttore AS
           (
           SELECT    P.IdProduttore AS IdProduttore
                   , COUNT(IdProdotto) AS NProdottiDiProduttore
           FROM      Prodotti P, Produttori PP
           WHERE     P.IdProduttore = PP.IdProduttore
           GROUP BY P.IdProduttore
           )
           , ProdDiProdottoOrdinatiAF AS
           (
           SELECT    F.Nome AS FNome
                   , PP2.IdProduttore AS IdProduttore
                   , PP2.Nome AS NomeProduttore
                   , COUNT(DISTINCT P2.IdProdotto) AS NPF
           FROM      Fornitori F, Ordini O, RigheOrdini RO, Prodotti P2, Produttori PP2
           WHERE     O.IdFornitore = F.IdFornitore
                   AND RO.IdOrdine = O.IdOrdine
                   AND RO.IdProdotto = P2.IdProdotto
                   AND P2.IdProduttore = PP2.IdProduttore
           GROUP BY F.IdFornitore, F.Nome, PP2.IdProduttore, PP2.Nome
           )
SELECT     R.FNome AS NomeFornitore, NomeProduttore
FROM       ProdDiProdottoOrdinatiAF R, NoProdottiDiProduttore S
WHERE      R.IdProduttore = S.IdProduttore AND NProdottiDiProduttore = NPF;

```

Infine, osserviamo che il committente non ha chiesto espressamente che PP produca almeno un prodotto, per cui era accettabile anche la seguente soluzione. Questa differisce dalle altre nel senso che un produttore PP senza prodotti sarà restituito in coppia con ogni fornitore F, perché la condizione ‘ogni prodotto di PP è presente in un ordine inviato ad F’ è verificata in maniera banale dalla coppia F-PP. Presentiamo solo la versione in OQL, dato che la traduzione in SQL non presenta elementi nuovi.

```
SELECT DISTINCT F.Nome, PP.Nome
FROM Fornitori F, Produttori PP
WHERE FOR ALL P2 IN Prodotti WHERE P2.IdProduttore = PP.IdProduttore
      : ( FOR SOME RO IN RigheOrdini, O IN Ordini
          WHERE F.IdFornitore = O.IdFornitore AND RO.IdOrdine = O.IdOrdine
          : RO.IdProdotto = P.IdProdotto );
```