

(b) Si traduca lo schema concettuale in uno schema relazionale grafico, definendo gli attributi delle chiavi primarie e delle chiavi esterne (Figura 2).

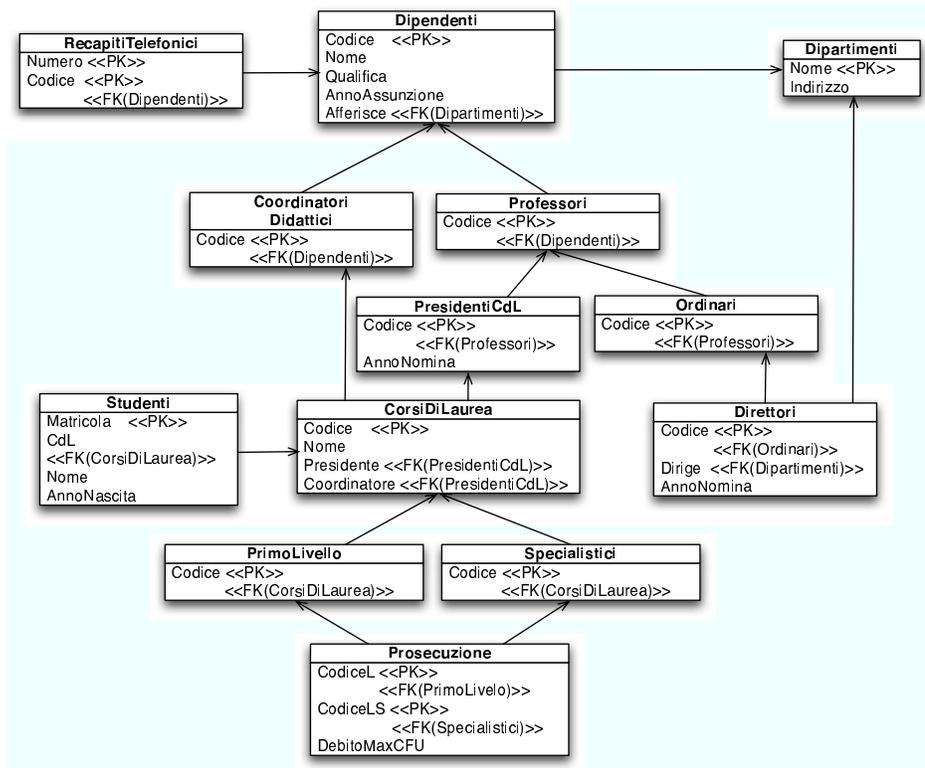


Figura 2: Schema relazionale grafico

2. Si consideri il seguente schema relazionale:

Impiegati (IdImp, Nome, AnnoNascita, Sesso, Stipendio)
 Allocazioni (IdImp*, IdProg*)
 Progetti(IdProg, DataInizio, DataFine, Budget)

(a) Si rappresenti graficamente lo schema relazionale della base di dati (Figura 3).



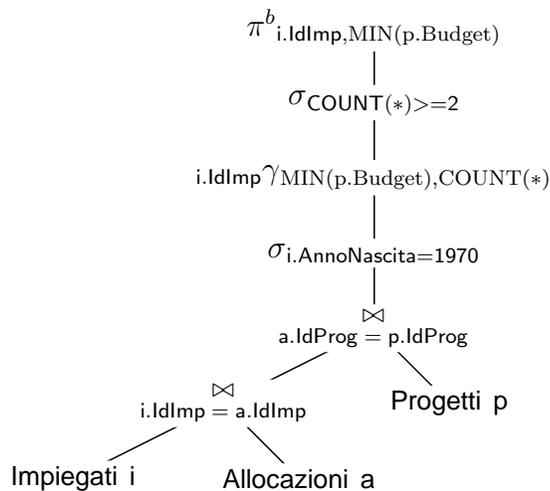
Figura 3: Schema relazionale grafico della BD

(b) Si disegni l'albero di sintassi astratta di un'espressione algebrica (albero logico) per la seguente interrogazione:

```

SELECT    i.IdImp, MIN(p.Budget)
FROM      Impiegati i, Allocazioni a, Progetti p
WHERE     i.IdImp = a.IdImp AND a.IdProg = p.IdProg
          AND i.AnnoNascita = 1970
GROUP BY i.IdImp
HAVING    COUNT(*) >= 2;
  
```

Albero logico



(c) Si considerino le relazioni:

Allocazioni

<i>IdImp</i>	<i>IdProg</i>
i1	p1
i1	p2
i1	p3
i2	p2
i2	p3
i3	p3

Progetti

<i>IdProg</i>	<i>DataInizio</i>	<i>DataFine</i>	<i>Budget</i>
p1	10
p2	5
p3	20

Si mostri il risultato delle seguenti interrogazioni:

```
SELECT a.IdImp AS I, p.IdProg AS P, Budget
FROM   Allocazioni a, Progetti p
WHERE  a.IdProg = p.IdProg AND Budget <= 10;
```

<i>I</i>	<i>P</i>	<i>Budget</i>
i1	p1	10
i1	p2	5
i2	p2	5

```
SELECT      a.IdImp AS I, MIN(Budget) AS minBudget
FROM        Allocazioni a, Progetti p
WHERE       a.IdProg = p.IdProg AND Budget <= 10
GROUP BY   a.IdImp
HAVING      COUNT(*) >= 2;
```

<i>I</i>	<i>minBudget</i>
i1	5

(d) Si scrivano le interrogazioni SQL che restituiscono le seguenti informazioni, senza duplicazione dei risultati:

- i. **(Compito A)** Identificatore e nome di tutti gli impiegati che sono allocati su almeno un progetto che è finito prima del 1/1/2000 oppure che ha un Budget < 100.

```
SELECT DISTINCT i.IdImp, i.Nome
FROM   Impiegati i, Allocations a, Progetti p
WHERE  i.IdImp = a.IdImp AND a.IdProg = p.IdProg
      AND (p.DataFine < '1/1/2000' OR Budget <= 100);
```

- (Compito B)** Identificatore e data di inizio di tutti i progetti su cui è allocato almeno un impiegato che è di sesso femminile oppure che è nato prima del 1960.

```
SELECT DISTINCT p.IdProg, p.DataInizio
FROM   Impiegati i, Allocations a, Progetti p
WHERE  i.IdImp = a.IdImp AND a.IdProg = p.IdProg
      AND (i.Sesso = 'F' OR i.DataNascita < 1960);
```

- ii. **(Compito A)** Per ciascun progetto con almeno 5 impiegati allocati, l'identificatore del progetto e la data di inizio del progetto, il numero di impiegati allocati ed il loro stipendio medio.

```
SELECT      p.IdProg, p.DataInizio, COUNT(*), AVG(i.Stipendio)
FROM        Impiegati i, Allocations a, Progetti p
WHERE       i.IdImp = a.IdImp AND a.IdProg = p.IdProg
GROUP BY    p.IdProg, p.DataInizio
HAVING      COUNT(*) >= 5;
```

- (Compito B)** Per ciascun impiegato allocato su almeno 3 progetti, l'identificatore e il nome dell'impiegato, il numero di progetti su cui è allocato ed il budget medio di tali progetti.

```
SELECT      i.IdImp, i.Nome, COUNT(*), AVG(p.Budget)
FROM        Impiegati i, Allocations a, Progetti p
WHERE       i.IdImp = a.IdImp AND a.IdProg = p.IdProg
GROUP BY    i.IdImp, i.Nome
HAVING      COUNT(*) >= 3;
```

- iii. **(Compito A)** Identificatore e nome di tutti gli impiegati che sono allocati solo su progetti che sono finiti prima del 1/1/2000 oppure che hanno un Budget < 100.

```
SELECT i.IdImp, i.Nome
FROM   Impiegati i
WHERE  FOR ALL a IN Allocations, p IN Progetti
       WHERE a.IdProg = p.IdProg AND i.IdImp = a.IdImp :
       p.DataFine < '1/1/2000' OR Budget <= 100;
```

```
SELECT i.IdImp, i.Nome
FROM   Impiegati i
WHERE  NOT EXISTS(
  SELECT *
  FROM   Allocations a, Progetti p
  WHERE  a.IdProg = p.IdProg AND i.IdImp = a.IdImp
        AND NOT (p.DataFine < '1/1/2000' OR Budget <= 100) );
```

- (Compito B)** L'identificatore e data di inizio di tutti i progetti su cui sono allocati solo impiegati che sono di sesso femminile oppure sono nati prima del 1960.

```
SELECT p.IdProg, p.DataInizio
FROM   Progetti p
WHERE  FOR ALL a IN Allocations, i IN Impiegati
       WHERE a.IdImp = i.IdImp AND a.IdProg = p.IdProg :
       i.Sesso = 'F' OR i.DataNascita < 1960;
```

```
SELECT p.IdProg, p.DataInizio
FROM   Progetti p
WHERE  NOT EXISTS(
  SELECT *
  FROM   Allocations a, Impiegati i
  WHERE  a.IdImp = i.IdImp AND a.IdProg = p.IdProg
        AND NOT (i.Sesso = 'F' OR i.DataNascita < 1960) );
```

- iv. **(Compito A)** L'identificatore e data di inizio di ciascun progetto in cui tutte le donne che lavorano al progetto sono nate prima del 1970.

```
SELECT p.IdProg, p.DataInizio
FROM   Progetti p
WHERE  FOR ALL a IN Allocations, i IN Impiegati
       WHERE a.IdProg = p.IdProg AND i.IdImp = a.IdImp AND i.Sesso = 'F' :
       i.AnnoNascita < 1970;
```

```
SELECT p.IdProg, p.DataInizio
FROM   Progetti p
WHERE  NOT EXISTS(
  SELECT *
  FROM   Allocations a, Impiegati i
  WHERE  a.IdProg = p.IdProg AND i.IdImp = a.IdImp AND i.Sesso = 'F'
        AND NOT (i.AnnoNascita < 1970) );
```

- (Compito B)** L'identificatore e nome di ciascun impiegato per cui tutti i progetti su cui lavora hanno un budget < 500.

```
SELECT i.IdImp, i.Nome
FROM   Impiegati i
WHERE  FOR ALL a IN Allocations, p IN Progetti
       WHERE a.IdProg = p.IdProg AND i.IdImp = a.IdImp :
       p.Budget < 500;
```

```
SELECT i.IdImp, i.Nome
FROM   Impiegati i
WHERE  NOT EXISTS(
  SELECT *
  FROM   Allocations a, Progetti p
  WHERE  a.IdProg = p.IdProg AND i.IdImp = a.IdImp
        AND NOT (p.Budget < 500) );
```

- v. **(Compito A)** (Opzionale) Per ciascun progetto con almeno 5 impiegati allocati di sesso femminile, l'identificatore e data di inizio del progetto, il numero totale di impiegati allocati (indipendentemente dal sesso) ed il loro stipendio medio.

```

SELECT  p.IdProg, p.DataInizio, COUNT(*), AVG(i.Stipendio)
FROM    Impiegati i, Allocations a, Progetti p
WHERE   i.IdImp = a.IdImp AND a.IdProg = p.IdProg AND
        5 <= ( SELECT COUNT(*)
                FROM    Impiegati i2, Allocations a2
                WHERE   i2.IdImp = a2.IdImp AND a2.IdProg = p.IdProg
                AND    i2.Sesso = 'F' )
GROUP BY p.IdProg, p.DataInizio;

```

- (Compito B)** (Opzionale) Gli identificatori i1 e i2 per ogni coppia di impiegati tali che i1 è nato (strettamente) prima di i2, ed inoltre su ogni progetto su cui è allocato i1 è allocato anche i2.

```

SELECT  i1.IdImp, i2.IdImp
FROM    Impiegati i1, Impiegati i2
WHERE   i1.DataNascita < i2.DataNascita AND
        FOR ALL a1 IN Allocations WHERE a1.IdImp = i1.IdImp :
        FOR SOME a2 IN Allocations WHERE a2.IdImp = i2.IdImp :
        a1.IdProg = a2.IdProg;

SELECT  i1.IdImp, i2.IdImp
FROM    Impiegati i1, Impiegati i2
WHERE   i1.DataNascita < i2.DataNascita AND
        NOT EXISTS(
        SELECT  *
        FROM    Allocations a1
        WHERE   a1.IdImp = i1.IdImp AND
                NOT EXISTS(
                SELECT  *
                FROM    Allocations a2
                WHERE   a2.IdImp = i2.IdImp AND a1.IdProg = a2.IdProg));

```

- vi. **(Compito A)** (Opzionale) Gli identificatori p1 e p2 per ogni coppia di progetti tali che il budget di p1 è strettamente maggiore di quello di p2 e ogni impiegato allocato su p1 è allocato anche su p2.

```

SELECT p1.IdProg, p2.IdProg
FROM   Progetti p1, Progetti p2
WHERE  p1.Budget > p2.Budget AND
       FOR ALL a1 IN Allocations WHERE a1.IdProg = p1.IdProg :
           FOR SOME a2 IN Allocations WHERE a2.IdProg = p2.IdProg :
               a1.IdImp = a2.IdImp;

```

```

SELECT p1.IdProg, p2.IdProg
FROM   Progetti p1, Progetti p2
WHERE  p1.Budget > p2.Budget AND
       NOT EXISTS(
         SELECT *
         FROM   Allocations a1
         WHERE  a1.IdProg = p1.IdProg AND
               NOT EXISTS(
                 SELECT *
                 FROM   Allocations a2
                 WHERE  a2.IdProg = p2.IdProg AND a1.IdImp = a2.IdImp));

```

- (Compito B)** (Opzionale) Per ciascun impiegato allocato su almeno 3 progetti con budget superiore a 10000, identificatore e nome dell'impiegato, il numero di progetti su cui è allocato (indipendentemente dal budget), ed il budget medio di tali progetti.

```

SELECT  i.IdImp, i.Nome, COUNT(*), AVG(p.Budget)
FROM    Impiegati i, Allocations a, Progetti p
WHERE   i.IdImp = a.IdImp AND a.IdProg = p.IdProg AND
       3 <= ( SELECT COUNT(*)
              FROM    Progetti p2, Allocations a2
              WHERE   p2.IdProg = a2.IdProg AND a2.IdImp = i.IdImp
                  AND p2.Budget > 10000)
GROUP BY i.IdImp, i.Nome;

```