

Esame di Stato Istituto Tecnico Industriale

Proposta di soluzione della seconda prova

Indirizzo: INFORMATICA Tema: INFORMATICA

Anno Scolastico: 2003-2004

1. Schema concettuale della base di dati

Lo schema concettuale (o statico) è uno dei due schemi del progetto concettuale di un sistema informativo.

Fatti elementari

Partendo dalle specifiche, contenute nel testo del problema, i fatti elementari del sistema informativo in fase di progetto sono i seguenti.

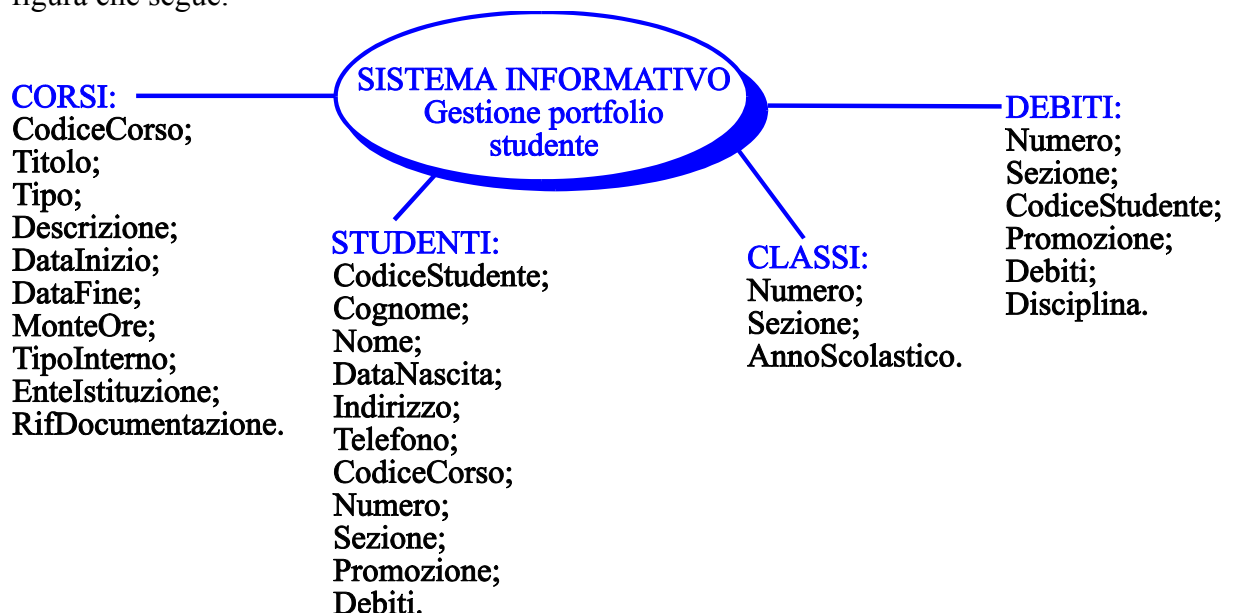
- | |
|--|
| ■ Uno studente può frequentare più corsi e un corso può riguardare più studenti. |
| ■ Una classe si riferisce a più studenti e uno studente può frequentare più classi, in anni scolastici differenti. |
| ■ Uno studente può avere più debiti e lo stesso tipo di debito può riguardare più studenti. |

Tipi di entità e attributi

I tipi di entità caratteristici del sistema sono:

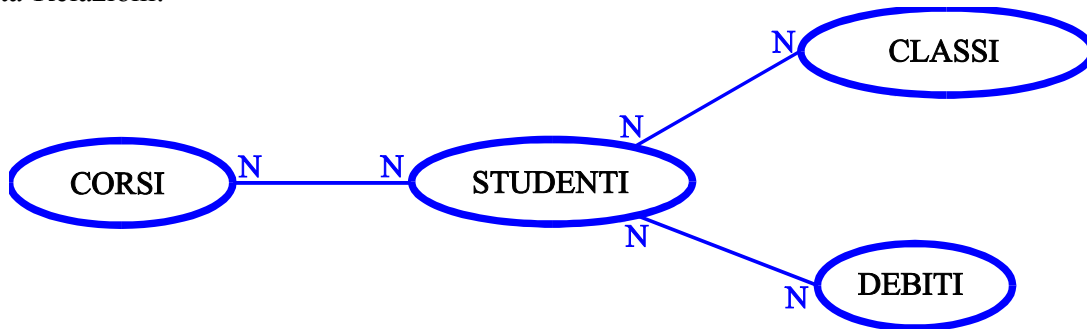
Studenti
Corsi
Classi
Debiti

I tipi di entità vengono individuati dai gruppi di attributi illustrati in modo dettagliato nella figura che segue.



Le relazioni

Le relazioni tra i tipi di entità nel modello dei dati sono individuate nel seguente diagramma Entità-Relazioni.



Nella fase di realizzazione del progetto concettuale le relazioni molti a molti tra i tipi di entità *Corsi-Studenti*, *Studenti-Classi* e *Studenti-Debiti*, dovranno essere trasformate in relazioni uno a molti e molti a uno, introducendo dei tipi di entità ausiliari.

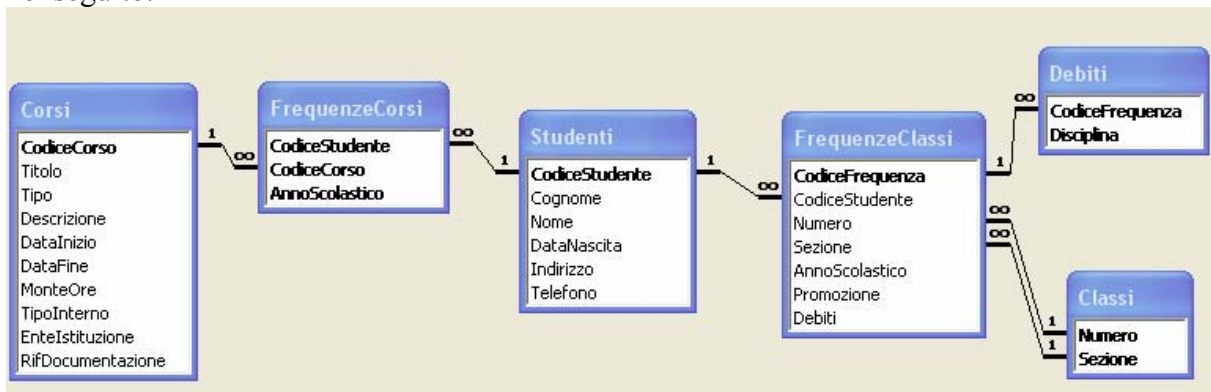
2. Schema logico della base di dati

Passaggio dallo schema statico del progetto concettuale al modello relazionale

Per la creazione dello schema logico della base di dati relazionale effettueremo i seguenti due passi.

1. Traduzione dello schema statico concettuale in un modello relazionale preliminare.
2. Applicazione delle tre forme normali per verificare se lo schema preliminare può essere ulteriormente ottimizzato.

Nella fase di realizzazione del progetto concettuale, le relazioni molti a molti *Corsi-Studenti*, *Studenti-Classi* e *Studenti-Debiti* devono essere trasformate in due associazioni uno a molti e molti a uno, introducendo, rispettivamente, le due tabelle ausiliarie: *FrequenzeCorsi* e *FrequenzeClassi*. Lo schema logico relazionale del database diventa quindi quello riportato nel seguito.



3. Definizione delle relazioni della base di dati in linguaggio SQL

Nel seguito è riportato il codice sorgente interpretato da un RDBMS (script), scritto nel linguaggio standard SQL, per la creazione dello schema logico del database.

```
CREATE DATABASE GestionePortfolio

CREATE TABLE Studenti
(
  CodiceStudente CHAR(7) NOT NULL,
  Cognome        CHAR(20) NOT NULL,
  Nome           CHAR(20) NOT NULL,
  DataNascita    DATE NOT NULL,
  Indirizzo      CHAR(50) NOT NULL,
```

```

Telefono      CHAR(15) NOT NULL,
PRIMARY KEY (CodiceStudente),
CHECK (CodiceStudente LIKE 'CS_____')
)
CREATE TABLE Corsi
(
CodiceCorso   CHAR(7) NOT NULL,
Titolo       CHAR(100) NOT NULL,
Tipo        CHAR(7) NOT NULL,
Descrizione  CHAR(200),
DataInizio  DATE NOT NULL,
DataFine    DATE NOT NULL,
MonteOre    INTEGER,
TipoInterno CHAR(13),
EnteIstituzione CHAR(50),
RifDocumentazione CHAR(100),
PRIMARY KEY (CodiceCorso),
CHECK (CodiceCorso LIKE 'CC_____'),
CHECK (Tipo IN 'Interno', 'Esterno'),
CHECK (TipoInterno IN 'Recupero', 'Sostegno', 'Arricchimento' )
)
CREATE TABLE FrequenzeCorsi
(
CodiceStudente CHAR(7) NOT NULL,
CodiceCorso    CHAR(7) NOT NULL,
AnnoScolastico CHAR(9) NOT NULL,
PRIMARY KEY (CodiceStudente, CodiceCorso, AnnoScolastico),
FOREIGN KEY (CodiceStudente) REFERENCES Studenti (CodiceStudente),
FOREIGN KEY (CodiceCorso) REFERENCES Corsi (CodiceCorso)
)
CREATE TABLE Classi
(
Numero      CHAR(3) NOT NULL,
Sezione     CHAR(1) NOT NULL,
PRIMARY KEY (Numero, Sezione),
CHECK (Numero IN 'I', 'II', 'III', 'IV', 'V'),
CHECK (Sezione IN 'A', 'B', 'C', 'D', 'E', 'F', 'G')
)
CREATE TABLE Debiti
(
CodiceFrequenza INTEGER NOT NULL,
Disciplina     CHAR(50) NOT NULL,
PRIMARY KEY (CodiceFrequenza, Disciplina)
)
CREATE TABLE FrequenzeClassi
(
CodiceFrequenza INTEGER IDENTITY(1,1),
CodiceStudente CHAR(7) NOT NULL,
Numero        CHAR(3) NOT NULL,
Sezione       CHAR(1) NOT NULL,
AnnoScolastico CHAR(9) NOT NULL,
Promozione    BIT,
Debiti       BIT,
PRIMARY KEY (CodiceFrequenza),
FOREIGN KEY (CodiceStudente) REFERENCES Studenti (CodiceStudente),
FOREIGN KEY (Numero, Sezione) REFERENCES Classi (Numero, Sezione),
FOREIGN KEY (CodiceFrequenza) REFERENCES Debiti (CodiceFrequenza)
)

```

Lo script deve essere eseguito (una sola volta) in un RDBMS per la creazione delle tabelle e delle relazioni della base di dati.

4. Implementazione delle operazioni in linguaggio SQL

Operazione A: *data una classe, quali studenti di quella classe hanno frequentato corsi e di che tipo.*

Codice sorgente SQL

```
SELECT DISTINCT Studenti.Cognome, Studenti.Nome, Corsi.Titolo,
                Corsi.Tipo
FROM Corsi INNER JOIN (
    (Studenti INNER JOIN FrequenzeClassi
     ON Studenti.CodiceStudente = FrequenzeClassi.CodiceStudente)
  INNER JOIN FrequenzeCorsi
   ON Studenti.CodiceStudente = FrequenzeCorsi.CodiceStudente)
ON Corsi.CodiceCorso = FrequenzeCorsi.CodiceCorso
WHERE FrequenzeClassi.Numero = [Numero Classe:] AND
      FrequenzeClassi.Sezione =[Sezione classe:]
```

Nel codice precedente, così come in quelli dei servizi successivi, i nomi racchiusi tra parentesi quadre (ad esempio *[Numero classe:]*) rappresentano parametri di ingresso il cui valore viene richiesto dal RDBMS prima dell'esecuzione dell'operazione.

Operazione B: *dato uno studente, quali corsi ha frequentato, di che tipo, per quale monte ore e in quale anno scolastico.*

Codice sorgente SQL

```
SELECT Corsi.Titolo, Corsi.Tipo, Corsi.MonteOre,
       FrequenzeCorsi.AnnoScolastico
FROM Corsi INNER JOIN FrequenzeCorsi
  ON Corsi.CodiceCorso = FrequenzeCorsi.CodiceCorso
WHERE FrequenzeCorsi.CodiceStudente = [Codice studente:]
```

Operazione C: *dato un anno scolastico, quali corsi interni sono stati attivati e da quali studenti sono stati seguiti.*

Codice sorgente SQL

```
SELECT Corsi.Titolo, Studenti.Cognome, Studenti.Nome
FROM Studenti INNER JOIN
  (Corsi INNER JOIN FrequenzeCorsi
   ON Corsi.CodiceCorso = FrequenzeCorsi.CodiceCorso)
ON Studenti.CodiceStudente = FrequenzeCorsi.CodiceStudente
WHERE FrequenzeCorsi.AnnoScolastico = [Anno scolastico:] AND
      Corsi.Tipo = "Interno"
```

Operazione D: *dato un corso, quali sono i dati relativi ad esso e per quali anni scolastici è stato attivato.*

Codice sorgente SQL

```
SELECT DISTINCT FrequenzeCorsi.AnnoScolastico, Corsi.Tipo,
                Corsi.Titolo
FROM Corsi INNER JOIN FrequenzeCorsi
  ON Corsi.CodiceCorso = FrequenzeCorsi.CodiceCorso
WHERE Corsi.CodiceCorso = [Codice corso:]
```

Operazione E: *dato uno studente quali classi ha frequentato, in quali anni scolastici e con quali esiti finali.*

Codice sorgente SQL

```
SELECT FrequenzeClassi.Numero, FrequenzeClassi.Sezione,  
       FrequenzeClassi.AnnoScolastico, FrequenzeClassi.Promozione,  
       FrequenzeClassi.Debiti  
FROM Studenti INNER JOIN  
     (Classi INNER JOIN FrequenzeClassi  
      ON (Classi.Sezione = FrequenzeClassi.Sezione AND  
          Classi.Numero = FrequenzeClassi.Numero) )  
     ON Studenti.CodiceStudente = FrequenzeClassi.CodiceStudente  
WHERE Studenti.CodiceStudente = [Codice studente:]
```