

Esame di Stato Istituto Tecnico Industriale
Soluzione della Seconda Prova
Indirizzo: INFORMATICA
Tema: INFORMATICA
Anno Scolastico: 2002-2003

Il primo punto richiede l'analisi e lo sviluppo del progetto concettuale del sistema informativo e la sua realizzazione mediante uno schema logico, utilizzando il linguaggio standard SQL.

Schema concettuale della base di dati

Tipi di entità

I tipi di entità caratteristici del sistema sono:

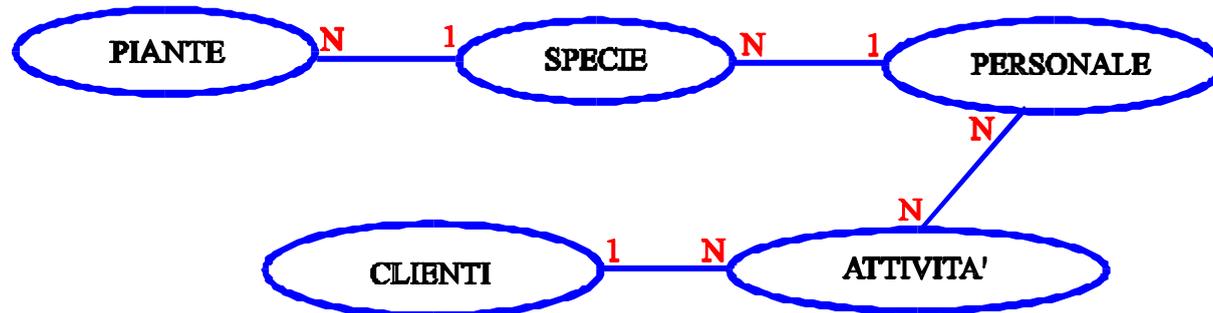
Piante
Specie
Personale
Attività
Clienti

Attributi

Sistema informativo Gestione Vivaio	
Tipo di entità	Attributi
Piante	IDPianta, IDSpecie, Nome, Immagine, Tipo, Numero, StagioneFioritura, PrezzoVendita
Specie	IDSpecie, IDPersona, Descrizione, ModoColtivazione, CaratteristicheEsposizione
Personale	IDPersona, Categoria, Cognome, Nome, DataNascita, Qualifica, CostoOrario
Attività	IDAttività, Tipo, IDCliente, Nome, DataPrenotazione, DataEffettuazione, NecessitàPianta, Evaso
Clienti	IDCliente, Cognome, Nome, Tipo, Telefono

Relazioni

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



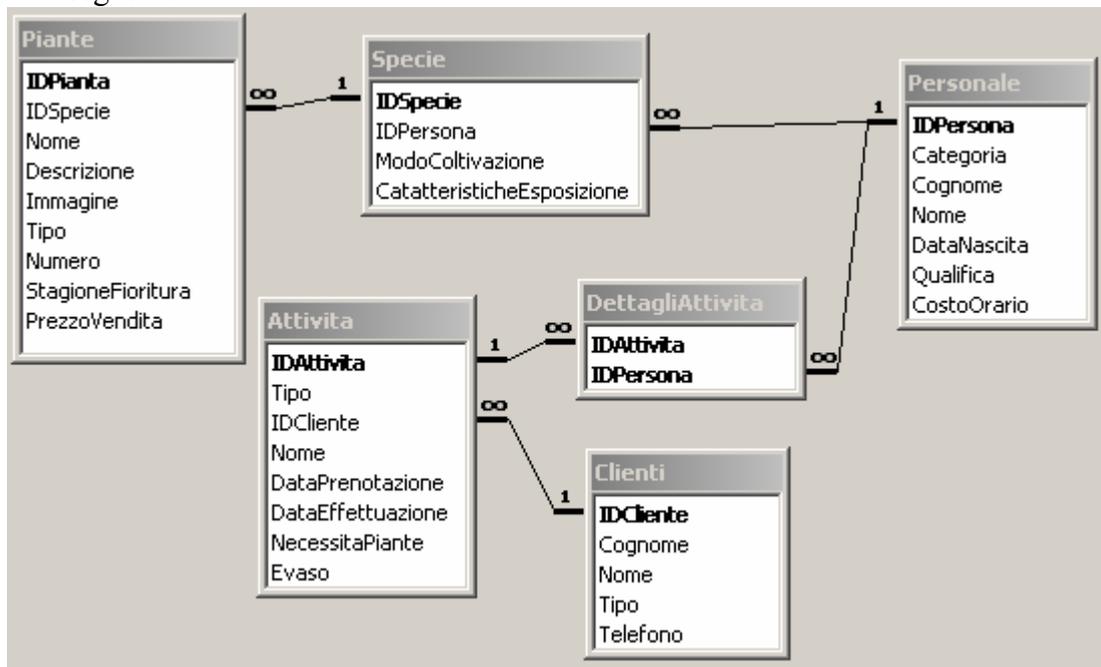
Schema logico della base di dati

Come tipo di schema logico, dovendo poi realizzarlo nello standard SQL, scegliamo quello relazionale.

Nel diagramma Entità-Relazioni precedente possiamo osservare che esiste una relazione N a N tra i tipi di entità *Personale* e *Attività*. Applicando il processo di normalizzazione (definito dalle forme normali) dobbiamo quindi inserire una tabella ausiliaria, che chiameremo

DettagliAttivita, che trasforma la relazione N a N in due relazioni: 1 a N, tra il *Personale* e *DettagliAttivita*, e N a 1 tra *DettagliAttivita* e *Clients*.

Lo schema relazionale del database del sistema informativo per la gestione del vivaio diventa quindi il seguente.



Definizione delle relazioni della base di dati in linguaggio SQL

Lo schema logico relazionale precedente (tabelle, relazione e applicazione dei vincoli di integrità dei dati e referenziale) può essere creato eseguendo il seguente script sviluppato con il linguaggio SQL.

```
CREATE DATABASE GestioneVivaio
CREATE TABLE Personale
(
  IDPersona      CHAR (5) NOT NULL,
  Categoria      CHAR (14) NOT NULL,
  Cognome        CHAR (20) NOT NULL,
  Nome           CHAR (20) NOT NULL,
  DataNascita    DATE NOT NULL,
  Qualifica      CHAR (15) NOT NULL,
  CostoOrario    DOUBLE PRECISION,
  PRIMARY KEY (IDPersona),
  CHECK (IDPersona LIKE 'CP '),
  CHECK (Categoria IN ('Agronomo','Operaio','Amministrativo'))
)
CREATE TABLE Specie
(
  IDSpecie      CHAR (5) NOT NULL,
  IDPersona     CHAR (5) NOT NULL,
  ModoColtivazione CHAR (20),
  CaratteristicheEsposizione CHAR (20),
  PRIMARY KEY (IDSpecie),
  CHECK (IDSpecie LIKE 'CS '),
  FOREIGN KEY (IDPersona)
```

```

        REFERENCES Personale(IDPersona)
    )
CREATE TABLE Piante
(
    IDPianta          CHAR (5) NOT NULL,
    IDSpecie          CHAR (5)  NOT NULL,
    Nome              CHAR (50) NOT NULL,
    Descrizione       CHAR (50) NOT NULL,
    Immagine          CHAR (20),
    Tipo              CHAR (7) NOT NULL,
    Numero            INTEGER NOT NULL,
    StagioneFioritura CHAR (9) NOT NULL,
    PrezzoVendita     DOUBLE PRECISION NOT NULL,
    PRIMARY KEY (IDPianta),
    CHECK (IDPianta LIKE 'PP  '),
    CHECK (StagioneFioritura IN
            ('Primavera','Estate','Autunno', 'Inverno') ),
    CHECK (Tipo IN ('Interno', 'Esterno') ),
    FOREIGN KEY (IDSpecie) REFERENCES Specie(IDSpecie)
)
CREATE TABLE Clienti
(
    IDCliente CHAR (5) NOT NULL,
    Cognome   CHAR (20) NOT NULL,
    Nome      CHAR (20) NOT NULL,
    Tipo      CHAR (7) NOT NULL,
    Telefono  CHAR (11) NOT NULL,
    PRIMARY KEY (IDCliente),
    CHECK (IDCliente LIKE 'CC  '),
    CHECK (Tipo IN ('Privati','Aziende') )
)
CREATE TABLE Attivita
(
    IDAttivita      CHAR (5) NOT NULL,
    Tipo            CHAR (20) NOT NULL,
    IDCliente       CHAR (5)  NOT NULL,
    Nome           CHAR (50),
    DataPrenotazione DATE NOT NULL,
    DataEffettuazione DATE,
    NecessitaPiante BIT NOT NULL,
    Evaso          BIT NOT NULL,
    PRIMARY KEY (IDAttivita),
    CHECK (IDAttivita LIKE 'CA  '),
    FOREIGN KEY (IDCliente)
        REFERENCES Clienti(IDCliente)
)
CREATE TABLE DettagliAttivita
(
    IDAttivita CHAR (5) NOT NULL,
    IDPersona  CHAR (5) NOT NULL,
    PRIMARY KEY (IDAttivita, IDPersona),

```

```

FOREIGN KEY (IDAttivita)
REFERENCES Attivita(IDAttivita),
FOREIGN KEY (IDPersona)
REFERENCES Personale(IDAttivita)
)

```

La colonna *Immagine* della tabella Piante può contenere (in alternativa):

- il percorso (o l'URL) del file che memorizza l'immagine della pianta;
- direttamente un oggetto immagine, se il RDBMS in cui è realizzato il database dispone di questo tipo di dato.

Realizzazione delle interrogazioni espresse in linguaggio SQL

1. Interrogazione

```

SELECT Numero
FROM Piante
WHERE Nome = [Nome pianta:]

```

Nella query precedente abbiamo codificato un parametro di input scrivendolo tra parentesi quadre.

2. Interrogazione

```

SELECT Nome
FROM Piante
WHERE StagioneFioritura = [Stagione fioritura:]

```

3. Interrogazione

```

SELECT Clienti.Cognome, Clienti.Nome, Clienti.Telefono
FROM Clienti INNER JOIN Attivita
ON Clienti.IDCliente = Attivita.IDCliente
WHERE (Attivita.Tipo = [Tipo intervento:]) AND
(Attivita.DataPrenotazione
BETWEEN [Data iniziale:] AND [Data finale:])

```

4. Interrogazione

```

SELECT COUNT(Piante.IDPianta) AS [Numero esemplari piante]
FROM (Personale INNER JOIN Specie
ON Personale.IDPersona = Specie.IDPersona)
INNER JOIN Piante ON Specie.IDSpecie = Piante.IDSpecie
WHERE (Personale.Cognome = [Cognome agronomo:])
AND (Personale.Nome = [Nome agronomo:])
AND (Personale.Categoria = "Agronomo");

```

5. Interrogazione

```

SELECT Nome, Descrizione, Numero
FROM Piante
WHERE PrezzoVendita = ANY
(SELECT MIN(PrezzoVendita) FROM Piante)

```

6. Interrogazione

```

SELECT Attivita.Nome, Clienti.Cognome, Clienti.Nome,
Clienti.Telefono
FROM Clienti INNER JOIN Attivita

```

```
ON Clienti.IDCliente = Attivita.IDCliente  
WHERE Attivita.Evaso = 0
```