

PROBLEMA. *Un albergo di una grande città intende gestire in modo automatizzato sia le prenotazioni sia i soggiorni e realizzare un database. Ogni cliente viene individuato, tra l'altro, con i dati anagrafici, il numero di telefono e l'eventuale e-mail. Per quanto riguarda le prenotazioni occorre indicare il periodo, i dati relativi alla persona che prenota, il numero di camera assegnato, l'eventuale disdetta, il tipo di trattamento: mezza pensione (Half Board: HB), pensione completa (Full Board: FB), pernottamento/prima colazione (Bed & Breakfast: B&B). A ogni prenotazione non disdetta corrisponde un soggiorno nell'albergo per il quale si devono registrare i dati di tutte le persone che occuperanno la camera. Ogni cliente che soggiorna deve anche fornire un documento di identità di cui si deve memorizzare una copia in formato digitale (eseguita, ad esempio, con uno scanner). Della suddetta realtà, dopo aver condotto le fasi di analisi (con la stesura delle specifiche) e di progettazione concettuale (schema statico e schema dinamico), realizzare il sistema informativo automatizzato presentando:*

- *la definizione dello schema logico relazionale del database;*
- *l'implementazione dei seguenti servizi espressi nel linguaggio SQL:*
 1. *l'elenco ordinato, in modo alfabetico, dei clienti che hanno prenotato in un certo giorno dell'anno;*
 2. *la lista delle camere occupate (ordinate per numero) in un determinato giorno;*
 3. *l'elenco dei clienti, ordinato per cognome e nome, che soggiornano in un dato giorno;*
 4. *il numero di soggiorni fatti da ogni cliente registrato in un certo anno.*

Il sistema informativo deve essere di tipo centralizzato e accessibile mediante un computer server installato nella rete locale dell'albergo.

■ ■ ■ ANALISI

SPECIFICHE CONCETTUALI PER I DATI

Il database deve raccogliere e memorizzare tutti i dati relativi:

- ai **clienti**, che prenotano le camere;
- alle **camere**, prenotate dai clienti;
- alle **prenotazioni**, effettuate dai clienti per le camere.

SPECIFICHE CONCETTUALI PER LE OPERAZIONI

Il sistema informativo automatizzato dell'albergo deve offrire questi servizi.

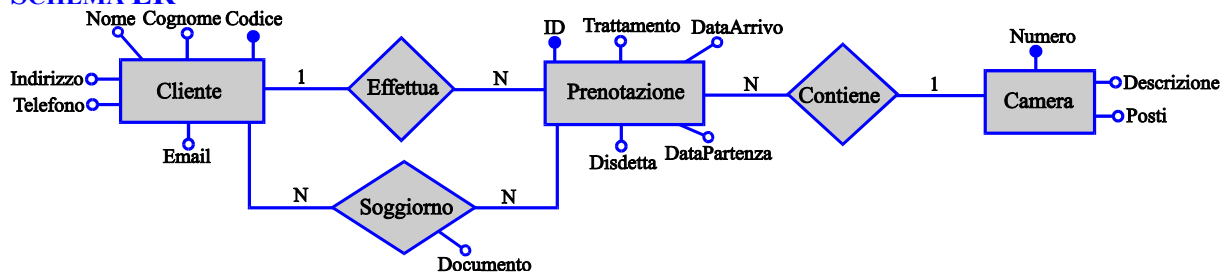
- l'elenco ordinato, in modo alfabetico, dei clienti che hanno prenotato in un certo giorno dell'anno;
- la lista delle camere occupate (ordinate per numero) in un determinato giorno;
- l'elenco dei clienti, ordinato per cognome e nome, che soggiornano in un dato giorno;
- il numero di soggiorni fatti da ogni cliente registrato in un certo anno.

SPECIFICHE TECNOLOGICHE

Il sistema informativo centralizzato deve essere realizzato con un server di database, al quale possono accedere diversi utenti mediante personal computer, inseriti nella rete locale dell'albergo. Si ipotizza che l'albergo disponga di 40 camere, mentre si stima che il numero medio di clienti dell'albergo sia 1000. Il sistema deve essere in grado di gestire i dati relativi all'albergo per almeno un anno.

SPECIFICHE PER IL TESTING

Per il testing del sistema informativo centralizzato, ipotizzare una serie di dati iniziali ed eseguire le operazioni realizzate, verificando se i risultati ottenuti sono coerenti con lo stato del database.

■ ■ ■ PROGETTO CONCETTUALE**SCHEMA ER**

Lo schema ER deve seguire queste **regole**.

- R1. L'attributo *Codice* di un *Cliente* segue il formato "C-XXX" con X cifra decimale (adatto per 1000 clienti con valori da "C-000" a "C-999").
- R2. L'attributo *Documento* di un *Soggiorno* contiene il percorso e il nome del file grafico con la copia digitale (fatta con uno scanner) di un documento di identità.
- R3. L'attributo *ID* di una *Prenotazione* è un contatore (numero intero che si incrementa automaticamente).
- R4. L'attributo *Disdetta* di una *Prenotazione* può assumere i valori logici Sì oppure No.
- R5. L'attributo *Trattamento* di una *Prenotazione* assume solo i valori "HB", "FB" e "B&B".
- R6. L'attributo *Numero* di una *Camera* è un numero intero compreso tra 1 e 40.

SCHEMA DINAMICO

Operazione: prenotazioni clienti			
Descrizione: l'elenco ordinato, in modo alfabetico, dei clienti che hanno prenotato in un certo giorno dell'anno;			
Tipo	Utenti	Parametri di ingresso	Risultati
Ricerca e ordinamento	Addetti alla reception, direttore albergo	La data di un giorno compresa tra una <i>DataArrivo</i> e una <i>DataPartenza</i> di una <i>Prenotazione</i>	<i>Cognome</i> , <i>Nome</i> per ogni <i>Cliente</i>

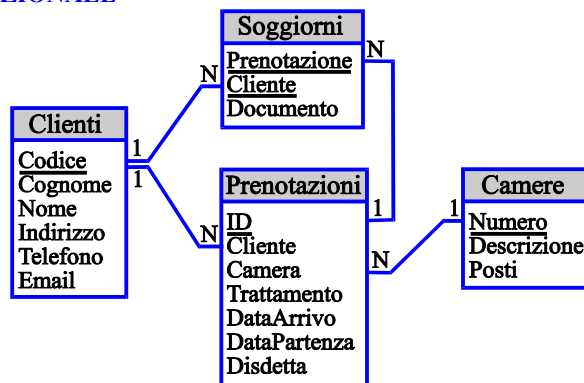
Operazione: camere occupate			
Descrizione: la lista delle camere occupate (ordinate per numero) in un determinato giorno;			
Tipo	Utenti	Parametri di ingresso	Risultati
Ricerca e ordinamento	Addetti alla reception, direttore albergo	La data di un giorno compresa tra una <i>DataArrivo</i> e una <i>DataPartenza</i> di una <i>Prenotazione</i>	<i>Numero</i> e <i>Posti</i> per ogni <i>Camera</i>

Operazione: soggiorni clienti			
Descrizione: l'elenco dei clienti, ordinato per cognome e nome, che soggiornano in un dato giorno;			
Tipo	Utenti	Parametri di ingresso	Risultati
Ricerca e ordinamento	Addetti alla reception, direttore albergo	La data di un giorno compresa tra una <i>DataArrivo</i> e una <i>DataPartenza</i> di una <i>Prenotazione</i>	<i>Cognome</i> , <i>Nome</i> per ogni <i>Cliente</i>

Operazione: soggiorni per anno			
Descrizione: il numero di soggiorni fatti da ogni cliente registrato in un certo anno;			
Tipo	Utenti	Parametri di ingresso	Risultati
Ricerca, ordinamento ed elaborazione	Addetti alla reception, direttore albergo	Un anno	Cognome e Nome di un Cliente e il conteggio del numero di soggiorni

■ ■ ■ REALIZZAZIONE

SCHEMA LOGICO RELAZIONALE



Lo schema logico relazionale **rispetta le forme normali** ed è soggetto a questi **vincoli di integrità referenziale**.

- V1. La chiave esterna *Cliente* della tabella *Prenotazioni* è in relazione con la tabella *Clienti* mediante la chiave primaria *Codice*.
- V2. La chiave esterna *Cliente* della tabella *Soggiorni* è in relazione con la tabella *Clienti* mediante la chiave primaria *Codice*.
- V3. La chiave esterna *Prenotazione* della tabella *Soggiorni* è in relazione con la tabella *Prenotazioni* mediante la chiave primaria *ID*.
- V3. La chiave esterna *Camera* della tabella *Prenotazioni* è in relazione con la tabella *Camere* mediante la chiave primaria *Numero*.

CODIFICA

Per tradurre nel DBMS lo **schema logico del database** si deve eseguire (una sola volta) questo codice sorgente SQL.

```
CREATE DATABASE DBAlbergo
```

```
CREATE TABLE Clienti
```

```
(
  Codice      CHAR(6) CHECK(Codice LIKE 'C-____'),
  Cognome     CHAR(20) NOT NULL,
  Nome        CHAR(20) NOT NULL,
  Indirizzo   CHAR(60),
  Telefono    CHAR(15) NOT NULL,
  Email       CHAR(30),
  PRIMARY KEY (Codice)
)
```

```
CREATE TABLE Camere
```

```
(
  Numero      INTEGER CHECK(Numero BETWEEN 1 AND 40),
  Descrizione CHAR(100),
  Posti       INTEGER NOT NULL,
  PRIMARY KEY (Numero)
)
```

```

CREATE TABLE Prenotazioni
(
  ID                INTEGER IDENTITY(1,1),
  Cliente           CHAR(6),
  Camera           INTEGER,
  Trattamento      CHAR(3) CHECK(Trattamento IN ('HB','FB','B&B') ) NOT NULL,
  DataArrivo       DATE NOT NULL,
  DataPartenza    DATE NOT NULL,
  Disdetta         BIT DEFAULT 0,
  PRIMARY KEY (ID),
  FOREIGN KEY (Cliente) REFERENCES Clienti(Codice),
  FOREIGN KEY (Camera) REFERENCES Camere(Numero)
)

CREATE TABLE Soggiorni
(
  Prenotazione     INTEGER,
  Cliente          CHAR(6),
  Documento       CHAR(60),
  PRIMARY KEY (Prenotazione, Cliente),
  FOREIGN KEY (Cliente) REFERENCES Clienti(Codice),
  FOREIGN KEY (Prenotazione) REFERENCES Prenotazioni(ID)
)

```

Le **interrogazioni sul database** sono realizzate mediante questi codici sorgente SQL che devono essere eseguiti in un DBMS.

Interrogazione: prenotazioni clienti

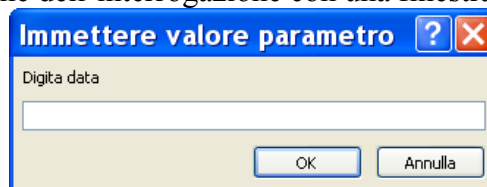
Stampare l'elenco ordinato, in modo alfabetico, dei clienti che hanno prenotato in un certo giorno dell'anno.

```

SELECT DISTINCT Clienti.Cognome, Clienti.Nome
  FROM Clienti INNER JOIN Prenotazioni
    ON Clienti.Codice = Prenotazioni.Cliente
 WHERE Prenotazioni.Disdetta = FALSE AND
        [Digita data] BETWEEN Prenotazioni.DataArrivo AND Prenotazioni.DataPartenza
 ORDER BY Clienti.Cognome, Clienti.Nome

```

Nel codice SQL, *[Digita data]* è un parametro di ingresso il cui valore viene richiesto dal DBMS prima dell'esecuzione dell'interrogazione con una finestra di dialogo di questo tipo.



Interrogazione: camere occupate

Visualizzare la lista delle camere occupate in un determinato giorno.

```

SELECT Camere.Numero, Camere.Posti
  FROM Camere INNER JOIN Prenotazioni
    ON Camere.Numero = Prenotazioni.Camera
 WHERE Prenotazioni.Disdetta = FALSE AND
        [Digita data] BETWEEN Prenotazioni.DataArrivo AND Prenotazioni.DataPartenza
 ORDER BY Camere.Numero

```

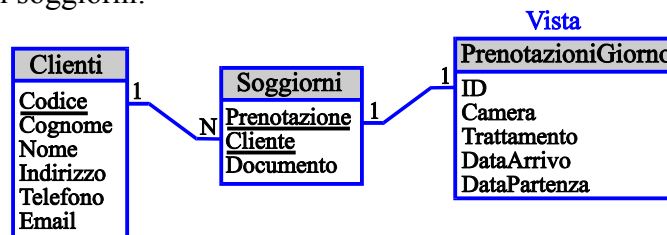
Interrogazione: soggiorni clienti

Stampare l'elenco dei clienti, ordinato per cognome e nome, che soggiornano in un dato giorno.

La realizzazione della prima operazione richiede la definizione della vista **PrenotazioniGiorno** che fornisce l'elenco delle prenotazioni non disdette relative a un certo giorno.

Schema logico	Query SQL
<p>Vista</p> <p>PrenotazioniGiorno</p> <p>ID Camera Trattamento DataArrivo DataPartenza</p>	<pre>CREATE VIEW PrenotazioniGiorno(ID, Camera, Trattamento, DataArrivo, DataPartenza) AS SELECT ID, Camera, Trattamento, DataArrivo, DataPartenza FROM Prenotazioni WHERE Disdetta = FALSE AND [Digita data] BETWEEN DataArrivo AND DataPartenza</pre>

L'operazione finale si ottiene richiamando la vista precedente messa in relazione con le tabelle dei clienti e dei soggiorni.



```
SELECT Clienti.Cognome, Clienti.Nome
FROM Clienti, Soggiorni, PrenotazioniGiorno
WHERE Clienti.Codice = Soggiorni.Cliente AND
      Soggiorni.Prenotazione = PrenotazioniGiorno.ID
ORDER BY Clienti.Cognome, Clienti.Nome
```

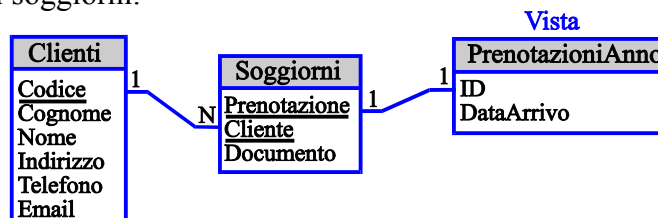
Interrogazione: soggiorni per anno

Stampare il numero di soggiorni fatti da ogni cliente registrato in un certo anno.

La realizzazione della prima operazione richiede la definizione della vista **PrenotazioniAnno** che fornisce l'elenco dei codici delle prenotazioni non disdette relative a un anno.

Schema logico	Query SQL
<p>Vista</p> <p>PrenotazioniAnno</p> <p>ID DataArrivo</p>	<pre>CREATE VIEW PrenotazioniAnno(ID, DataArrivo) AS SELECT ID, DataArrivo FROM Prenotazioni WHERE Disdetta = FALSE AND DataArrivo LIKE '*' + [Digita anno]</pre>

L'operazione finale si ottiene richiamando la vista precedente messa in relazione con le tabelle dei clienti e dei soggiorni.



```
SELECT Clienti.Cognome, Clienti.Nome, COUNT(*) AS 'Numero soggiorni'
FROM Clienti, Soggiorni, PrenotazioniAnno
WHERE Clienti.Codice = Soggiorni.Cliente AND
      Soggiorni.Prenotazione = PrenotazioniAnno.ID
GROUP BY Clienti.Cognome, Clienti.Nome
ORDER BY Clienti.Cognome, Clienti.Nome
```