PROIECT BD

Gestionarea excursiilor in Romania printr-o agentie de turism

Prunel Marian-Alexandru 334AA

Descrierea cerintelor de proiectare ale bazei de date:

- De fiecare client se ocupa cate un angajat al agentiei, un angajat se poate ocupa de mai multi clienti;
- Pentru a dispune de serviciile agentiei, clientul trebuie sa se prezinte personal la sediu;
- La inregistrarea unui client in baza de date a companiei, acestuia i se adauga si angajatul care l-a luat in primire;
- Un client poate participa la mai multe excursii, iar la o excursie se pot alatura mai multi clienti;
- Fiecare excursie are alocata un singur transport, respectiv o singura cazare;
- Nu se pot adauga noi transporturi, respectiv noi cazari, daca fiecare excursie are alocata o cazare, respectiv un transport;

Descrierea tabelelor bazei de date:

Baza de date contine 6 tabele:Angajati,Clienti,Excursie,Transport,Cazare,Vacanta.

```
CREATE TABLE `Angajati` (
  `IDAngajat` int(3) NOT NULL,
  `Nume` varchar(30) NOT NULL,
  `Prenume` varchar(30) NOT NULL,
  `Mail` varchar(35) NOT NULL,
  `Passwd` varchar(25) NOT NULL,
  `Salariu` int(5) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `Clienti` (
  `IDClient` int(3) NOT NULL,
  `Nume` varchar(30) NOT NULL,
  `Prenume` varchar(30) NOT NULL,
  `Varsta` int(2) NOT NULL,
  `IDAngajat` int(3) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `Excursie` (
  `IDExcursie` int(3) NOT NULL,
  `Destinatie` varchar(40) NOT NULL,
  `DataPlecare` date NOT NULL,
  `DataSosire` date NOT NULL,
  `Pret` int(5) NOT NULL DEFAULT '0',
  `IDCazare` int(3) NOT NULL DEFAULT '0',
  `IDTransport` int(3) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `Transport` (
  `IDTransport` int(3) NOT NULL,
  `Mijloc` varchar(15) NOT NULL DEFAULT 'Fara',
  `Firma` varchar(20) NOT NULL DEFAULT 'Fara',
  `Pret` int(5) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


CREATE TABLE `Cazare` (
  `IDCazare` int(3) NOT NULL,
  `Locatie` varchar(30) NOT NULL DEFAULT 'Fara',
  `Hotel` varchar(20) NOT NULL DEFAULT 'Fara',
  `Pret` int(5) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `Vacanta` (
  `IDVacanta` int(3) NOT NULL,
  `IDClient` int(3) NOT NULL DEFAULT '0',
  `IDExcursie` int(3) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Relatiile dintre tabele:

Angajati-Clienti: relatie de tip 1:N, in care un angajat se poate ocupa de mai multi clienti;
Clienti-Excursie: relatie de tip N:N; un client se poate alatura mai multor excursii, iar o excursie este realizata cu mai multi clienti;
Excursie-Cazare: relatie de tip 1:N; fiecare excursie are alocata o cazare,iar o cazare poate fi comuna mai multor excursii;
Excursie-Transport: relatie de tip 1:N; fiecare excursie are alocat un transport, iar un transport poate fi destinat mai multor excursii;
Tabela Vacanta: realizeaza realatia N:N dintre Clienti si Excursie.

Constrangeri de integritate:

- PRIMARY KEY
- FOREIGN KEY
- UNIQUE

```
ALTER TABLE `Angajati`
  ADD PRIMARY KEY (`IDAngajat`);

ALTER TABLE `Cazare`
  ADD PRIMARY KEY (`IDCazare`),
  ADD UNIQUE KEY `Locatie` (`Locatie`); → am folosit-o pentru a evita duplicatele

ALTER TABLE `Clienti`
  ADD PRIMARY KEY (`IDClient`),
  ADD KEY `IDAngajat` (`IDAngajat`),

ALTER TABLE `Excursie`
  ADD PRIMARY KEY (`IDExcursie`),
  ADD KEY `IDCazare` (`IDCazare`),
  ADD KEY `IDTransport` (`IDTransport`),

ALTER TABLE `Transport`
  ADD PRIMARY KEY (`IDTransport`);

ALTER TABLE `Vacanta`
  ADD PRIMARY KEY (`IDVacanta`),
  ADD KEY `IDClient` (`IDClient`),
  ADD KEY `IDExcursie` (`IDExcursie`),
```

Functionarea aplicatiei:

- Aplicatia are o interfata simpla;
- Aplicatia iti ofera acces total asupra bazei de date(INSERT, UPDATE, DELETE);
- Nu se pot insera inregistrari in tabelele Cazare, respectiv Transport daca:
  1) Nu exista excursii disponibile;
  2) Numarul inregistrariilor din tabele este egal cu numarul Excursiilor disponibile;
- Initial fiecare excursie nu va dispune de transport si cazare, urmand ca dupa introducerea unei cazari, respectiv a unui transport, se pot adauga in tabela Excursie in zona de EDIT.
- Nu se pot introduce duplicate in tabela Cazari;
- Nu se pot introduce duplicate de genul Client-Excursie;

Interogari:

Interogari simple:

```
1)SELECT C.IDCazare,E.Destinatie AS Locatie,C.Hotel,C.Pret
  FROM Cazare C LEFT JOIN Excursie E ON C.IDCazare = E.IDCazare
  ORDER BY Locatie,C.IDCazare;
```

```
2)SELECT  IDClient,  C.Nume,C.Prenume,C.Varsta,A.Nume  AS  Numea,A.Prenume  AS
                                                                   Prenumea
   FROM Clienti C LEFT JOIN Angajati A ON C.IDAngajat=A.IDAngajat;
```

```
3)SELECT E.IDExcursie,E.Destinatie,E.DataPlecare,E.DataSosire,C.Hotel,T.Firma,E.Pret
FROM Excursie E LEFT JOIN Cazare C ON C.IDCazare = E.IDCazare
                    LEFT JOIN Transport T ON T.IDTransport=E.IDTransport
ORDER BY E.DataPlecare;
```

```
4)SELECT C.Nume,C.Prenume,C.Varsta,E.Destinatie
FROM Clienti C INNER JOIN Vacanta V ON C.IDClient=V.IDClieNt
                 INNER JOIN Excursie E ON E.IDExcursie=V.IDExcursie;
```

```
5)SELECT C.Nume,C.Prenume,H.Locatie,T.Firma
FROM Clienti C LEFT JOIN Vacanta V ON C.IDClient=V.IDClient
                 LEFT JOIN Excursie E ON E.IDExcursie=V.IDExcursie
                 LEFT JOIN Cazare H ON H.IDCazare=E.IDCazare
                 LEFT JOIN Transport T ON T.IDTransport=E.IDTransport
ORDER BY C.Nume,C.Prenume,H.Locatie;
```

```
6)SELECT T.IDTransport, T.Mijloc,T.Firma,E.Destinatie AS dest, T.Pret
  FROM Transport T LEFT JOIN Excursie E ON T.IDTransport = E.IDTransport;
```

Interogari complexe:

```
1)SELECT A.IDAngajat, A.Nume, A.Prenume, A.Mail, A.Passwd, A.Salariu,
                   (SELECT COUNT(*)
                   FROM Clienti WHERE IDAngajat=A.IDAngajat) AS Nr_Clienti
  FROM Angajati A
  ORDER BY Nr_Clienti;

2)SELECT Nume,Prenume, Salariu
  FROM Angajati JOIN (SELECT DISTINCT Salariu Sal
                       FROM Angajati
                       ORDER BY Salariu DESC
                       LIMIT 2, 1) x
ON Salariu >= Sal
ORDER by Salariu DESC,Nume;

3)DELETE FROM Vacanta
WHERE IDVacanta IN (SELECT * FROM (SELECT IDVacanta FROM Vacanta
                      GROUP BY IDClient,IDExcursie
                      HAVING (COUNT(*) > 1) ) AS A);

4)SELECT SUM(Pret) AS Pret FROM
              (SELECT Cazare.Pret FROM Cazare WHERE IDCazare ='".$idc."'
               UNION
               SELECT Transport.Pret FROM Transport WHERE IDTransport='".$idt."') T;
```