Project B

Δημήτριος Τσιομπίκας AM : 3180223

Ζήτημα Πρώτο

Άσκηση 1

Κώδικας βάσης :

```
CREATE TABLE campdata(
    custID int,
    fname varchar(30),
    lname varchar(30),
    cID int,
    country varchar(30),
    bookID int,
    bookDate date,
    campCode char(3),
    campName varchar(50),
    empno int,
    catCode char(1),
    category varchar(20),
    unitCost numeric(4,2),
    startDate date,
    overnights int,
    persons int
);

BULK INSERT campdata
FROM 'G:\AUEB\6thsemester\Database Design\Project B\CAMPDATA.TXT'
WITH (FIRSTROW =2, FIELDTERMINATOR='|', ROWTERMINATOR = '\n');
```

Άσκηση 2

Επέλεξα για λογικό πλάνο τους εξής πίνακες :

Dimension tables : customer, Camp , Emplacement , timeinfo

Fact table : Booking

Εντολές :

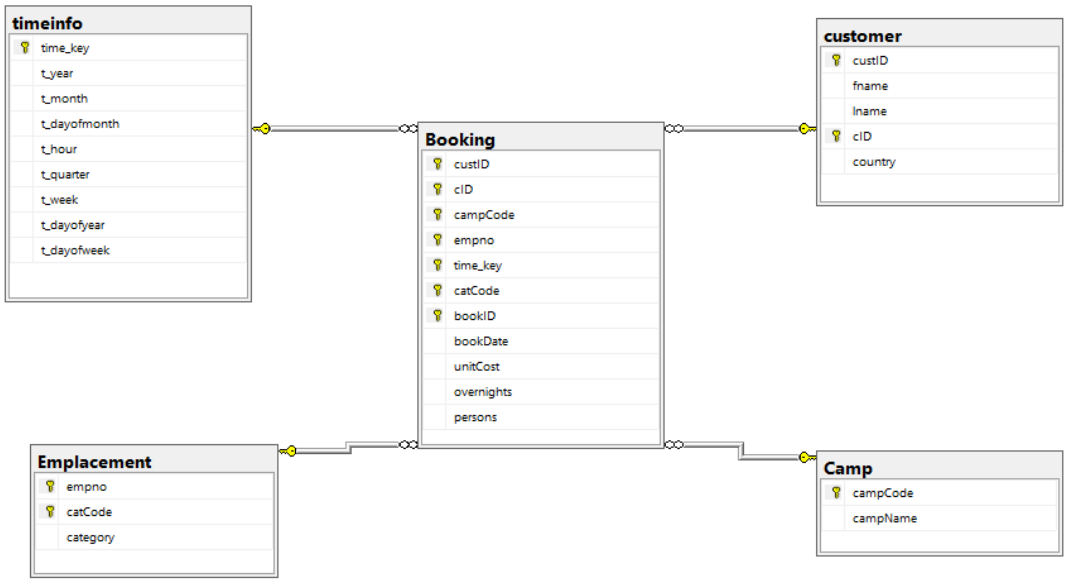
```sql
CREATE TABLE customer(
    custID int,
    fname varchar(30),
    lname varchar(30),
    cID int,
    country varchar(30),
    PRIMARY KEY(custID,cID)
);

CREATE TABLE Camp(
    campCode char(3) primary key,
    campName varchar(50)
);

CREATE TABLE Emplacement(
    empno int,
    catCode char(1),
    category varchar(20),
    PRIMARY KEY(empno,catCode)
);

CREATE TABLE timeinfo(
    time_key datetime primary key,
    t_year int,
    t_month int,
    t_dayofmonth int,
    t_hour int,
    t_quarter int,
    t_week int,
    t_dayofyear int,
    t_dayofweek int
);
```

```sql
CREATE TABLE Booking( /* Fact table */
    custID int,
    cID int,
    campCode char(3),
    empno int,
    time_key datetime,
    catCode char(1),
    bookID int,
    bookDate datetime,
    unitCost numeric(4,2),
    overnights int,
    persons int,

    PRIMARY KEY(custID,cID,campCode,empno,time_key,catCode,bookID),
    FOREIGN KEY(custID,cID) REFERENCES customer(custID,cID),
    FOREIGN KEY(campCode) REFERENCES Camp(campCode),
    FOREIGN KEY(empno,catCode) REFERENCES Emplacement(empno,catCode),
    FOREIGN KEY(time_key) REFERENCES timeinfo(time_key)
);
```

Άσκηση 3

```sql
/* Gemisma pinakwn */

INSERT INTO customer
SELECT DISTINCT custID,fname,lname,cID,country
FROM campdata;

INSERT INTO Camp
SELECT DISTINCT campCode,campName
FROM campdata;


INSERT INTO Emplacement
SELECT DISTINCT empno,catCode,category
FROM campdata;

SET datefirst 1;

INSERT INTO timeinfo
SELECT DISTINCT startDate,DATEPART(year,startDate),DATEPART(month,startDate),
DATEPART(day,startDate),DATEPART(hour,startDate),DATEPART(quarter,startDate),
DATEPART(week,startDate),DATEPART(dayofyear,startDate),DATEPART(dw,startDate)
FROM campdata;


INSERT INTO Booking
SELECT DISTINCT custID,cID,campCode,empno,startDate,catCode,bookID,bookDate,
unitCost,overnights,persons
FROM campdata;
```

Άσκηση 4



Ζήτημα Δεύτερο

Άσκηση 1

```sql
/* Query 1 */
SELECT TOP 100 country,fname,lname,SUM(unitCost * overnights * persons) AS BookingCost
FROM customer,Booking
WHERE customer.cID = Booking.cID AND customer.custID = Booking.custID
GROUP BY country,fname,lname
ORDER BY BookingCost DESC;
```

Άσκηση 2

```sql
/* Query 2 */
SELECT campName,category,SUM(unitCost * overnights * persons) AS BookingCost
FROM Emplacement,Camp,Booking,timeinfo
WHERE Emplacement.empno = Booking.empno AND Emplacement.catCode = Booking.catCode AND Camp.campCode = Booking.campCode
AND timeinfo.time_key = Booking.time_key AND t_year = 2000
GROUP BY category,campName;
```

Άσκηση 3

```sql
/* Query 3 */
SELECT campName,t_month,SUM(unitCost * overnights * persons) AS BookingCost
FROM Camp,timeinfo,Booking
WHERE Camp.campCode = Booking.campCode AND timeinfo.time_key = Booking.time_key AND t_year = 2018
GROUP BY campName,t_month
ORDER BY campName,t_month;
```

Άσκηση 4

```sql
/* Query 4 */
SELECT t_year,campName,category,COUNT(persons) AS totalPersonCount
FROM Booking,timeinfo,Camp,Emplacement
WHERE timeinfo.time_key = Booking.time_key AND Camp.campCode = Booking.campCode AND
Emplacement.empno = Booking.empno AND Emplacement.catCode = Booking.catCode
GROUP BY ROLLUP (t_year,campName,category);
```

- (t_year,campName,category) (group by t_year,campName,category)
- (t_year,campName,null) (group by t_year,campName)
- (t_year,null,null) (group by t_year)
- (null,null,null) (group by none)

Άσκηση 5

```sql
/* Query 5 */
CREATE VIEW v1 AS
SELECT t_year,campName,COUNT(persons) AS totalPersonCount
FROM timeinfo , Booking,Camp
WHERE timeinfo.time_key = Booking.time_key AND Camp.campCode = Booking.campCode
AND t_year = 2017
GROUP BY t_year,campName;

CREATE VIEW v2 AS
SELECT t_year,campName,COUNT(persons) AS totalPersonCount
FROM timeinfo , Booking,Camp
WHERE timeinfo.time_key = Booking.time_key AND Camp.campCode = Booking.campCode
AND t_year = 2018
GROUP BY t_year,campName;



SELECT v2.campName
FROM v1,v2
WHERE v1.campName = v2.campName AND v2.totalPersonCount > v1.totalPersonCount
GROUP BY v2.campName;
```

Ζήτημα Τρίτο

Επερώτηση : εμφάνισε το συνολικό ποσό πληρωμών κρατήσεων για έναν συνδυασμό στηλών (t_year , campName,category) , περισσότερα παρακάτω :

```sql
/* Ζήτημα Τρίτο */


SELECT t_year,campName,category,SUM(unitCost * overnights * persons) AS totalBookingCosts
FROM timeinfo,Camp,Booking,Emplacement
WHERE timeinfo.time_key = Booking.time_key AND Camp.campCode = Booking.campCode
AND Emplacement.catCode = Booking.catCode
GROUP BY CUBE(t_year,campName,category);
```

Το αποτέλεσμα της επερώτησης είναι η δημιουργία ενός κύβου , κάθε κελί του οποίου περιέχει τα συνολικά ποσά πληρωμών κρατήσεων για έναν συνδυασμό τιμών (έτους, Κατασκήνωσης, κατηγορίας θέσης) , πιο συγκεκριμένα :

- t_year,campName,category (group by t_year,campName,category)
- t_year,campName,null (group by t_year,campName)
- t_year,null,category (group by t_year,category)
- null, campName , category (group by campName,category)
- t_year,null,null (group by t_year)
- null,campName,null (group by campName)
- null,null,category (group by category)
- null,null,null (group by none)

Ζήτημα Τέταρτο

Ο πίνακας θα έχει την εξής μορφή (το ευρετήριο είναι στη στήλη country):

Row#	Austria	Denmark	Germany	Spain	Finland	Sweden	Italy	France	Netherlands	Belgium
69	0	0	0	1	0	0	0	0	0	0
70	0	0	0	1	0	0	0	0	0	0
71	0	1	0	0	0	0	0	0	0	0
72	1	0	0	0	0	0	0	0	0	0
73	1	0	0	0	0	0	0	0	0	0
74	0	0	1	0	0	0	0	0	0	0
75	0	0	1	0	0	0	0	0	0	0
76	0	0	1	0	0	0	0	0	0	0