

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Ακαδημαϊκό έτος: 2020-2021

Μάθημα: Σχεδιασμός Βάσεων Δεδομένων

Project – Part B

Όνομ/πώνυμο: Βασίλειος-Ηλίας Δρούζας

ΑΜ: 3180051

Ζήτημα 1ο

Α. Αρχικά φτιάχνουμε τη βάση δεδομένων CAMPDW, δημιουργούμε τον πίνακα CAMPDATA:

--1a.Table 'campdata' creation

```
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  
)
```

και στη συνέχεια εισάγουμε τα δεδομένα:

--1b.Insert data in the table

```
BULK INSERT campdata  
FROM 'C:\Users\user\Desktop\CAMPDATA.TXT'  
WITH (FIRSTROW =2, FIELDTERMINATOR='|', ROWTERMINATOR = '\n');
```

B.

Φτιάχνουμε σιγά σιγά τους πίνακες μας. Κατά σύμβαση, χρησιμοποιώ μπροστά από το όνομα του πίνακα το γράμμα f ή d ανάλογα με το αν πρόκειται για fact table ή dimension table αντίστοιχα.

1. Πίνακας *d_countries* (περιέχει στα στοιχεία των χωρών)

```
--2.  
  
--d: Dimension Table  
--f: Fact Table  
  
CREATE TABLE d_countries(  
  cid int primary key,  
  country varchar(30)  
)
```

2. Πίνακας *d_camping* (περιέχει τα στοιχεία των κατασκηνώσεων)

```
CREATE TABLE d_camping(  
  campcode char(3),  
  campName varchar(50),  
  empno int,  
  primary key (campcode, empno)  
)
```

3. Πίνακας *d_customers* (περιέχει τα στοιχεία των πελατών)

```
CREATE TABLE d_customers(  
  custid int primary key,  
  fname varchar(30),  
  lname varchar(30),  
  cid int,  
  country varchar(30)  
)
```

4. Πίνακας *d_seats* (περιέχει τα στοιχεία των θέσεων)

```
CREATE TABLE d_seats(  
  catcode char(1) primary key,  
  category varchar(20)  
)
```

5. Πίνακας *d_timeinfo* (περιλαμβάνει μετρικές στη μονάδα του χρόνου: ημέρα, μήνας, χρόνος)

```
CREATE TABLE d_timeinfo(  
  startdate date primary key,  
  t_dayofyear int,  
  t_month int,  
  t_year int  
)
```

6. Fact πίνακας *f_bookings* (περιλαμβάνει τα στοιχεία των κρατήσεων)

```
CREATE TABLE f_bookings(  
  bookid int,  
  cid int,  
  campcode char(3),  
  empno int,  
  custid int,  
  catcode char(1),  
  bookdate date,  
  unitcost numeric(4,2),  
  startdate date,  
  overnights int,  
  persons int,  
  
  primary key(bookid,cid,campcode,empno,custid,catcode,startdate),  
  foreign key(cid) references d_countries(cid),  
  foreign key(campcode,empno) references d_camping(campcode,empno),  
  foreign key(custid) references d_customers(custid),  
  foreign key(catcode) references d_seats(catcode),  
  foreign key (startdate) references d_timeinfo(startdate)  
)
```

Γ.

Κάνουμε insert τα δεδομένα στους πίνακες:

```
--3.

insert into d_countries
select distinct cid, country
from campdata

insert into d_camping
select distinct campcode, campName, empno
from campdata

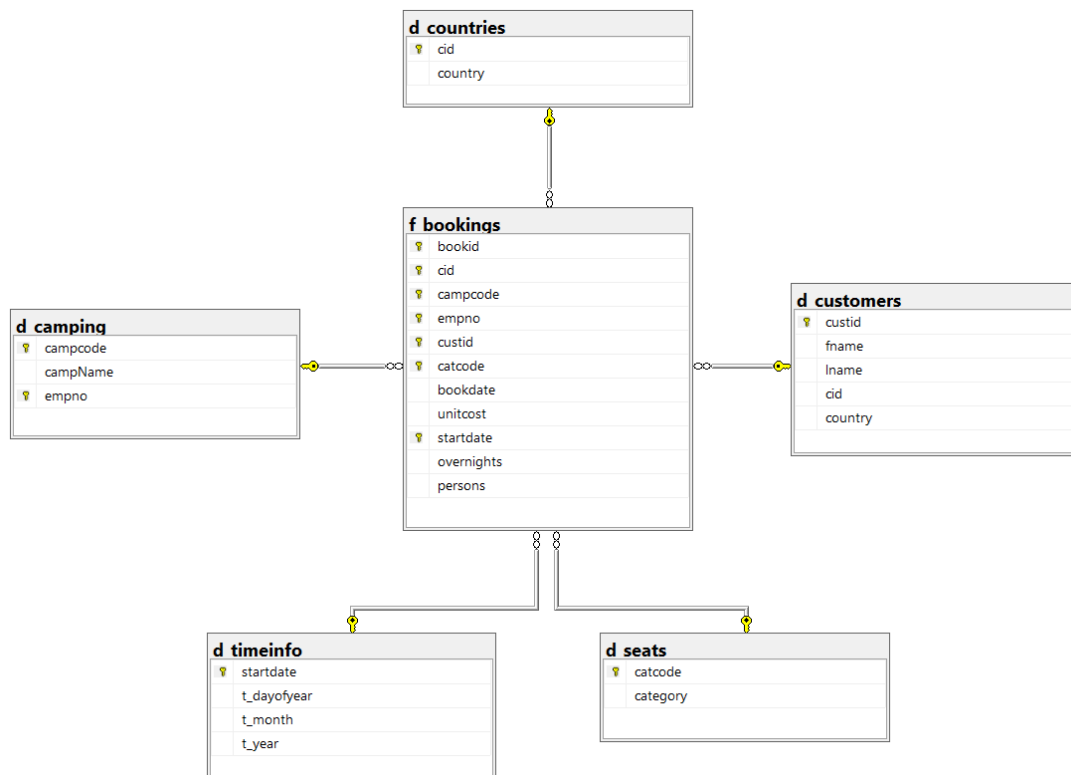
insert into d_customers
select distinct custid, fname, lname, cid, country
from campdata

insert into d_seats
select distinct catcode, category
from campdata


set datefirst 1;
insert into d_timeinfo
select distinct startdate, datepart(day, startdate),
datepart(month, startdate), datepart(year, startdate)
from campdata

insert into f_bookings
select bookid, cid, campcode, empno, custid, catcode, bookdate,
sum(unitcost), startdate, sum(overnights), sum(persons)
from campdata
group by bookid, cid, campcode, empno, custid, catcode, bookdate, startdate
```

Δ. Με τη χρήση του εργαλείου 'Database Diagrams' , έχουμε διαγραμματικά:



Ζήτημα 2ο

1.

```
--1.
SELECT TOP 100(sum(unitcost*overnights*persons)) as totalcost , fname,lname,d_countries.country
from f_bookings,d_customers,d_countries
WHERE d_customers.custid=f_bookings.custid
AND d_countries.cid=f_bookings.cid
GROUP BY fname,lname,d_countries.country
ORDER BY totalcost DESC
```

2.

```
--2.
```

```
SELECT DISTINCT campName,category, sum(unitcost*overnights*persons) as profit
FROM f_bookings , d_camping , d_seats,d_timeinfo
WHERE d_timeinfo.startdate=f_bookings.startdate
AND f_bookings.campcode=d_camping.campcode
AND d_seats.catcode=f_bookings.catcode
AND t_year=2000
GROUP BY campName,category
ORDER BY campName,category
```

3.

```
--3.
```

```
SELECT DISTINCT t_month,campName,sum(unitcost*overnights*persons) as profit
FROM f_bookings,d_camping,d_seats,d_timeinfo
WHERE d_timeinfo.startdate=f_bookings.startdate
AND f_bookings.campcode=d_camping.campcode
AND t_year=2018
GROUP BY campName,t_month
ORDER BY campName,t_month
```

4.

```
--ROLLUP
```

```
SELECT COUNT(d_customers.custid) as totalcustomers , t_year,campName,category
FROM d_customers,f_bookings,d_camping,d_timeinfo,d_seats
WHERE d_customers.custid=f_bookings.custid
AND d_timeinfo.startdate=f_bookings.startdate
AND d_camping.campcode=f_bookings.campcode
AND d_camping.empno=f_bookings.empno
AND d_seats.catcode=f_bookings.catcode
GROUP BY ROLLUP(t_year,campName,category)
ORDER BY t_year,campName,category
```

5.

```
--5.
```

```
CREATE VIEW v1 AS
SELECT campName,COUNT(d_customers.custid) AS totalcustomers,t_year
FROM d_camping,d_customers,f_bookings,d_timeinfo
WHERE d_timeinfo.startdate=f_bookings.startdate
AND d_camping.campcode=f_bookings.campcode
AND d_camping.empno=f_bookings.empno
AND d_customers.custid=f_bookings.custid
GROUP BY campName,t_year

SELECT A.campName, A.totalcustomers
FROM v1 AS A,v1 as B
WHERE A.campName=B.campName
AND A.t_year=2018 AND B.t_year=2017
AND A.totalcustomers>B.totalcustomers
```

Ζήτημα 3ο

A. Το ερώτημα που έφτιαξα βρίσκει τον αριθμό των διανυκτερεύσεων (count *) ανάλογα με

1. Αριθμό ατόμων (persons)
2. Ονομα Κατασκήνωσης (campName)
3. Κατηγορία θέσης (category)

```
--3rd Task
```

```
SELECT campName,category,persons,count(*) as totalnights
FROM d_camping,d_seats,f_bookings
WHERE d_camping.campcode=f_bookings.campcode
AND d_seats.catcode=f_bookings.catcode
GROUP BY CUBE(campName,category,persons)
```


* Σε πραγματικές συνθήκες και με πιο ρεαλιστικά δεδομένα (πχ. αν υπήρχε αλλαγή στην τιμή με βάση την ημερολογιακή διαφορά ανάμεσα στην ημερομηνία κράτησης και την πρώτη διανυκτέρευση) θα είχε περισσότερο νόημα να γράψουμε ένα cube που θα προέβλεπε το κόστος της κράτησης για κάθε πελάτη ανάλογα με μετρικές όπως 1. Αριθμό ατόμων , 2.CampName , 3. Κατηγορία θέσης. Αυτή ήταν και η πρώτη μου σκέψη για την άσκηση αυτή, αλλά μετά την άφηση, αφού εφόσον στο παράδειγμα μας το κόστος δεν επηρεάζεται από τον παράγοντα του χρόνου, ο κύβος αυτός δεν θα είχε πρακτική αξία.

B. Τα **group-bys** στο παράδειγμά μας έχουν ως εξής:

- (campName, category, persons)
- (campName, category)
- (campName, persons)
- (category, persons)
- (campName)
- (category)
- (persons)
- ()

Ζήτημα 4ο

- Base Table :

| Row | Country |
|-----|---------|
| 69 | Spain |
| 70 | Spain |
| 71 | Denmark |
| 72 | Austria |
| 73 | Austria |
| 74 | Germany |
| 75 | Germany |
| 76 | Germany |

- Ενώ το ζητούμενο bitmap country Index μας είναι:

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

Σημείωση: Όπως ζητούνταν από την εκφώνηση, έχω συμπεριλάβει όλες τις χώρες με τις οποίες συνεργάζεται η εταιρεία. Οι πρώτες τέσσερις (Spain,Denmark,Austria,Germany) λαμβάνουν βάσει του query σε κάποιες εγγραφές τη τιμή 1 ενώ οι υπόλοιπες έξι έχουν πάντα τη τιμή 0 , καθώς δεν επιστρέφονται από το συγκεκριμένο query.