

2^ο ΠΡΟΤΖΕΚΤ ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΚΑΡΑΓΕΩΡΓΗΣ ΠΛΑΤΩΝ p3180068

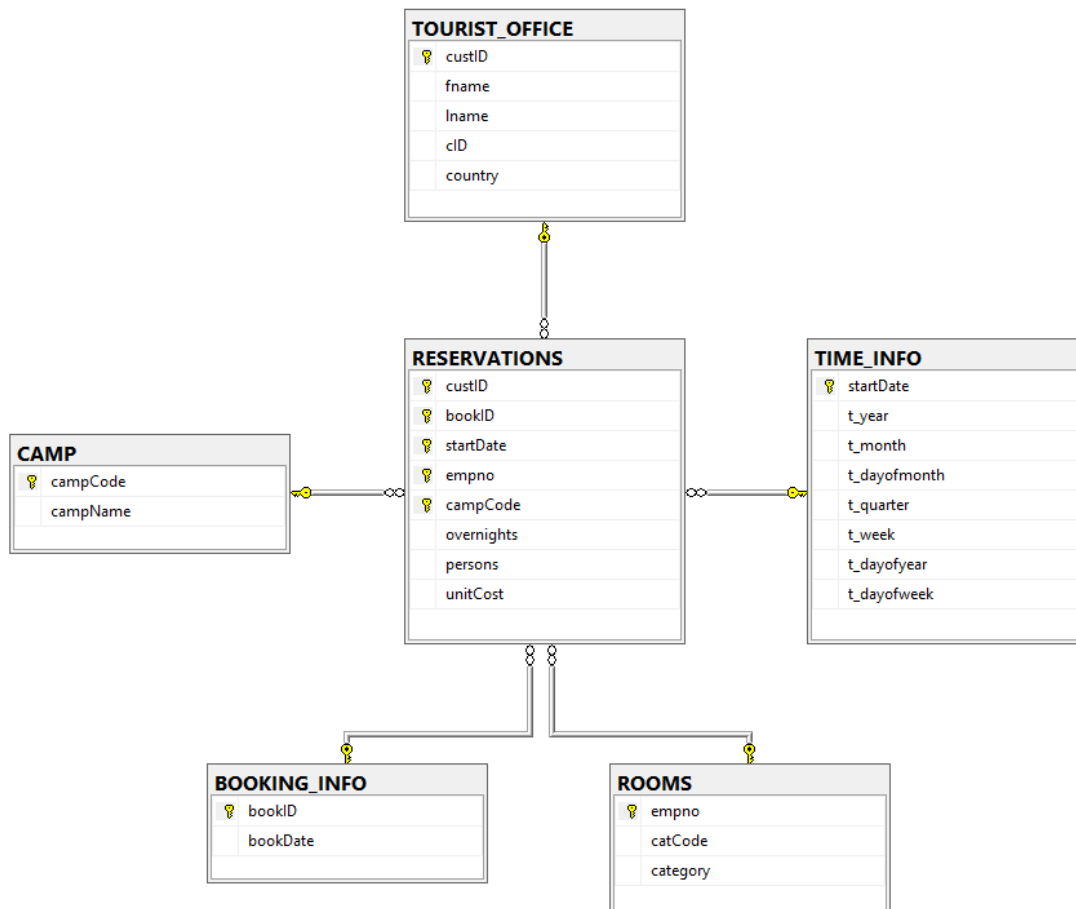
ΖΗΤΗΜΑ 1

Δημιουργώ τον πίνακα campdata όπως φαίνεται παρακάτω:

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

Έπειτα, έχοντας κατεβάσει το zip από το eclass φορτώνω τα δεδομένα του .txt στον πίνακα αυτόν με την εντολή που δίνεται στην εκφώνηση.

Στη συνέχεια, το κυριότερο κομμάτι θεωρώ της εργασίας ζητά να σχεδιάσουμε μία αποθήκη δεδομένων που να ακολουθεί το Star Schema. Έχοντας κάνει τα ερωτήματα 1β και 1γ παραθέτω το σχήμα από την επιλογή “Database Diagrams” και στη συνέχεια θα εξηγήσω τη λογική την οποία ακολούθησα προκειμένου να καταλήξω στο αποτέλεσμα της φωτογραφίας.



Το πρώτο ζητούμενο με τη δημιουργία ενός Star Schema είναι να καταλήξουμε σε ένα fact table. Εγώ ήθελα το fact table να έχει μία σύνοψη όλων των δεδομένων και να έχει κάποιο νόημα, να μην έχει δηλαδή απλά columns που να εξασφαλίζουν την μοναδικότητα των rows με τα δεδομένα που έχουμε ως τώρα. Ο fact table εδώ είναι ο πίνακας “Reservations” δηλαδή ο πίνακας των κρατήσεων. Μία κράτηση, χρησιμοποιώντας την κοινή λογική για να είναι μοναδική και να έχει και τις απαραίτητες πληροφορίες, χρειάζεται τις δύο πλευρές που έρχονται σε συμφωνία, δηλαδή την κατασκήνωση και το τουριστικό γραφείο, τον κωδικό κράτησης, την ημερομηνία που θα πραγματοποιηθεί η διαμονή και το «δωμάτιο». Είναι σχεδιαστικό λάθος να υπάρχουν δύο ίδιες τέτοιες πλειάδες, γιατί αν γινόταν τότε θα ήταν καταστροφικό για τον οικοδεσπότη, στην προκειμένη την κατασκήνωση. Με μία πρώτη ματιά θα χρειαζόμασταν λοιπόν τα custID, bookID, startDate, campCode, empno. Το startDate μας εξασφαλίζει όπως λέει και η εκφώνηση, την ημερομηνία, το empno το «δωμάτιο», το bookID τον κωδικό της κράτησης και τέλος τα custID, campCode είναι οι δύο πλευρές που επέρχονται σε συμφωνία.

Αφού λοιπόν κατέληξα σε αυτό το fact table και τα primary keys του, τότε το μόνο που έμενε ήταν να δημιουργήσω αντίστοιχα dimension tables με primary keys τις τιμές που έχει ορίσει ως foreign το fact table. Το μόνο που είχε μία παραπάνω δυσκολία, ήταν το TIME_INFO στο οποίο επέλεξα να δώσω αναλυτικά σε κάθε στήλη τα στοιχεία της ημερομηνίας startDate για απλότητα στα μελλοντικά queries. Τέλος, αποφάσισα να προσθέσω τις στήλες overnights, persons, unitCost στον πίνακα Reservations επειδή αφενός μεν θεώρησα ότι ταίριαζαν εκεί περισσότερο αφετέρου δε επιτάχυναν τα queries του ζητήματος 2, καθώς με αυτόν τον τρόπο γλίτωνα περιττά joins.

Παραθέτω τις εντολές που εκτέλεσα στο ζήτημα αυτό:

```
CREATE TABLE TOURIST_OFFICE(
  custID int,
  fname VARCHAR(30),
  lname VARCHAR(30),
  cID int,
  country VARCHAR(30)
  PRIMARY KEY(custID));

CREATE TABLE RESERVATIONS(
  custID int,
  bookID int,
  startDate date,
  empno int,
  campCode char(3),
  overnights int,
  persons int,
  unitCost numeric(4,2)
  PRIMARY KEY(custID,bookID,empno,startDate,campCode),
  FOREIGN KEY(custID) REFERENCES TOURIST_OFFICE(custID),
  FOREIGN KEY(bookID) REFERENCES BOOKING_INFO(bookID),
  FOREIGN KEY(empno) REFERENCES ROOMS(empno),
  FOREIGN KEY(startDate) REFERENCES TIME_INFO(startDate),
  FOREIGN KEY(campCode) REFERENCES CAMP(campCode));

CREATE TABLE BOOKING_INFO(
  bookID int,
  bookDate date
  PRIMARY KEY(bookID));

CREATE TABLE ROOMS(
  empno int,
  catCode char(1),
  category VARCHAR(20)
  PRIMARY KEY(empno));

CREATE TABLE CAMP(
  campCode char(3),
  campName VARCHAR(50)
  PRIMARY KEY(campCode));

CREATE TABLE TIME_INFO(
  startDate date,
  t_year int,
  t_month int,
  t_dayofmonth int,
  t_quarter int,
  t_week int,
  t_dayofyear int,
  t_dayofweek int
  PRIMARY KEY(startDate));

INSERT INTO TOURIST_OFFICE
  SELECT DISTINCT custID, fname, lname, cID, country
  FROM campdata;
INSERT INTO CAMP
  SELECT DISTINCT campCode, campName
  FROM campdata;
INSERT INTO ROOMS
  SELECT DISTINCT empno, catCode, category
  FROM campdata;
INSERT INTO BOOKING_INFO
  SELECT DISTINCT bookID, bookDate
  FROM campdata;
INSERT INTO RESERVATIONS
  SELECT custID, bookID, startDate, empno, campCode, overnights, persons, unitCost
  FROM campdata;

SET DATEFIRST 1;
INSERT INTO TIME_INFO
  SELECT DISTINCT startDate, DATEPART(YEAR, startDate), DATEPART(MONTH, startDate),
  DATEPART(DAY, startDate), DATEPART(QUARTER, startDate),
  DATEPART(WEEK, startDate), DATEPART(DAYOFYEAR, startDate),
  DATEPART(dw, startDate)
  FROM campdata;
```

ZHTHMA 2

1)

```
SELECT TOP 100 country, fname, lname, SUM(unitCost*persons*overnights) AS total_cost
FROM TOURIST_OFFICE, RESERVATIONS
WHERE RESERVATIONS.custID = TOURIST_OFFICE.custID
GROUP BY country, fname, lname
ORDER BY total_cost DESC;
```

2)

```
SELECT campName, category, SUM(unitCost*persons*overnights) AS total_cost
FROM RESERVATIONS, CAMP, ROOMS, TIME_INFO
WHERE RESERVATIONS.empno = ROOMS.empno AND RESERVATIONS.campCode =
CAMP.campCode AND RESERVATIONS.startDate = TIME_INFO.startDate AND t_year = 2000
GROUP BY campName, category
ORDER BY campName;
```

3)

```
SELECT campName, t_month, SUM(unitCost*persons*overnights) AS total_cost
FROM RESERVATIONS, CAMP, ROOMS, TIME_INFO
WHERE RESERVATIONS.empno = ROOMS.empno AND RESERVATIONS.campCode =
CAMP.campCode AND RESERVATIONS.startDate = TIME_INFO.startDate AND t_year = 2018
GROUP BY campName, t_month
ORDER BY campName;
```

4)

```
SELECT t_year, campName, category, SUM(persons) as total_tenants
FROM RESERVATIONS, TIME_INFO, CAMP, ROOMS
WHERE RESERVATIONS.startDate = TIME_INFO.startDate AND RESERVATIONS.campCode =
CAMP.campCode AND RESERVATIONS.empno = ROOMS.empno
GROUP BY ROLLUP(t_year, campName, category);
```

5)

```
CREATE VIEW [PER_YEAR_TENANTS] AS
SELECT campName, t_year, SUM(persons) as total_tenants
FROM RESERVATIONS, TIME_INFO, CAMP
WHERE RESERVATIONS.startDate = TIME_INFO.startDate AND RESERVATIONS.campCode =
CAMP.campCode
GROUP BY campName, t_year;
```

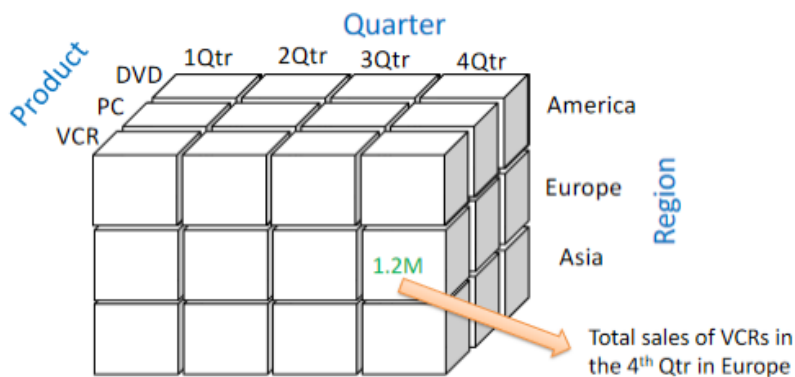
```
SELECT DISTINCT table1.campName
FROM [PER_YEAR_TENANTS] as table1, [PER_YEAR_TENANTS] as table2
WHERE table1.campName = table2.campName AND table1.t_year = 2018 AND table2.t_year =
2017 AND (table1.total_tenants > table2.total_tenants);
```

ZΗΤΗΜΑ 3

Το query που θα χρησιμοποιήσω για το ζήτημα 3 είναι το παρακάτω:

```
SELECT RESERVATIONS.campCode, catCode, t_year, SUM(persons) as total_tenants
FROM RESERVATIONS, CAMP, ROOMS, TIME_INFO
WHERE RESERVATIONS.campCode = CAMP.campCode AND RESERVATIONS.empno =
ROOMS.empno AND RESERVATIONS.startDate = TIME_INFO.startDate
GROUP BY CUBE(RESERVATIONS.campCode,catCode,t_year);
```

A) Παραθέτω ένα screenshot από τις διαλέξεις για να εξηγήσω την απάντησή μου:



Αν υποθέσουμε ότι ο ένας άξονας αντί για Product έχει Rooms, ο δεύτερος άξονας αντί για Quarter έχει Camp και ο τρίτος άξονας αντί για Region έχει τον πίνακα Time_Info, τότε το κελί της φωτογραφίας θα περιέχει π.χ. το σύνολο των tenants στο δωμάτιο με catCode A, στη κατασκήνωση με campCode APL και το έτος 2000. Δηλαδή, κάθε κελί είναι η τιμή του column που βρίσκεται στο aggregate function, η οποία προκύπτει από τον συνδυασμό των τριών τιμών στις διαστάσεις x, y, x του κύβου.

B) Τα GROUP BY είναι συνολικά 8 διότι ο κύβος έχει 3 διαστάσεις και άρα $2^3=8$ περιπτώσεις. Αυτές τις βλέπουμε παρακάτω:

1^η Περίπτωση : campCode, catCode, t_year (GROUP BY campCode, catCode, t_year)

2^η Περίπτωση : NULL, catCode, t_year (GROUP BY catCode, t_year)

3^η Περίπτωση : campCode, NULL, t_year (GROUP BY campCode, t_year)

4^η Περίπτωση : campCode, catCode, NULL (GROUP BY campCode ,catCode)

5^η Περίπτωση : NULL, NULL, t_year (GROUP BY t_year)

6^η Περίπτωση : campCode, NULL, NULL (GROUP BY campCode)

7^η Περίπτωση : NULL, catCode , NULL (GROUP BY catCode)

8^η Περίπτωση : NULL, NULL, NULL (GROUP BY NONE)

ΖΗΤΗΜΑ 4

Το bitmap για όλες τις χώρες της στήλης countries θα έχει την ακόλουθη μορφή:

	Finland	Sweden	Italy	Holland	Germany	Austria	France	Belgium	Spain	Denmark
Row										
1	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	1

Με βάση αυτή την «κωδικοποίηση», το περιεχόμενο του ευρετηρίου για τις 8 δειγματοληπτικές εγγραφές που δίνει η εκφώνηση θα είναι:

	Finland	Sweden	Italy	Holland	Germany	Austria	France	Belgium	Spain	Denmark
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0