

Δημιουργία Αποθήκης Δεδομένων (Data Warehouse) με χρήση του DBMS SQL SERVER

Μία εμπορική εταιρεία δραστηριοποιείται στην πώληση ποδηλάτων, ανταλλακτικών και αξεσουάρ ποδηλασίας. Η εταιρεία διαθέτει ένα δίκτυο υποκαταστημάτων σε διάφορες πόλεις της Αμερικής και της Ευρώπης.

Τα υποκαταστήματα είναι συνδεδεμένα σε ένα κεντρικό πληροφοριακό σύστημα το οποίο υποστηρίζει την απαιτούμενη λειτουργικότητα των πωλήσεων. Πρόκειται για ένα σύστημα OLTP στο οποίο είναι συνδεδεμένα όλα τα τερματικά (υπολογιστές και ταμειακές μηχανές) των σημείων πώλησης της εταιρείας.

Το τμήμα πωλήσεων (marketing) ενδιαφέρεται να αναπτύξει μία αποθήκη δεδομένων, ώστε να είναι σε θέση να αναλύει τα δεδομένα και να παράγει στατιστικές αναφορές σχετικά με την πορεία των πωλήσεων και την συμπεριφορά των πελατών της.

Τα δεδομένα που θα τροφοδοτούν την αποθήκη είναι διαθέσιμα υπό την μορφή ενός γραμμογραφημένου αρχείου κειμένου, το οποίο παράγεται από το πληροφοριακό σύστημα της εταιρείας. Κάθε εγγραφή του αρχείου αποτελείται από 35 πεδία.

Καλείστε να σχεδιάσετε και να υλοποιήσετε την παραπάνω αποθήκη δεδομένων προκειμένου να αυξήσετε την αποτελεσματικότητα της διεξαγωγής χρήσιμων στατιστικών στοιχείων, μειώνοντας ταυτόχρονα τον χρόνο εκτέλεσης των επερωτήσεων. Στην συνέχεια να τροφοδοτήσετε την αποθήκη με τα δεδομένα του αρχείου "salesData.txt" και να εκτελέσετε ορισμένες επερωτήσεις για την παραγωγή χρήσιμων στατιστικών στοιχείων.

Περιγραφή Αρχείου salesData.txt

Το αρχείο salesData.txt περιέχει 58189 εγγραφές πωλήσεων. Κάθε εγγραφή αποτελείται από 35 πεδία τα οποία διαχωρίζονται με τον χαρακτήρα "|" (pipe).

salesData.txt	
productKey	Κωδικός προϊόντος
orderDate	Ημερομηνία παραγγελίας/πώλησης
salesTerritoryKey	Κωδικός περιοχής
salesOrderNum	Κωδικός παραγγελίας/πώλησης
salesOrderLineNum	Γραμμή παραγγελίας
orderQuantity	Παραγγελθείσα ποσότητα
unitPrice	Τιμή μονάδας
totalProductCost	Συνολικό κόστος παραγωγής
salesAmount	Ποσό πώλησης
taxAmt	Φόρος

firstName	Όνομα πελάτη
lastName	Επώνυμο Πελάτη
gender	Φύλο πελάτη
occupation	Επάγγελμα πελάτη
productName	Όνομα προϊόντος
subCategory	Υποκατηγορία προϊόντος
category	Κατηγορία προϊόντος
standardCost	Κόστος προϊόντος
listPrice	Τιμή καταλόγου
productLine	Γραμμή παραγωγής
modelName	Μοντέλο
region	Περιοχή
country	Χώρα
s_year	Έτος πώλησης
s_quarter	Τρίμηνο πώλησης
monthNum	Αριθμός μήνα πώλησης
s_month	Μήνας πώλησης (συντομογραφία)
fiscalYear	Οικονομικό έτος πώλησης
fiscalQuarter	Τρίμηνο οικονομικού έτους
fiscalMonthNum	Αριθμός μήνα οικονομικού έτους
fiscalMonth	Συντομογραφία μήνα οικονομικού έτους
weekDayNum	Αριθμός μέρας της εβδομάδας
s_weekDay	Ημέρα εβδομάδας
weekDayWeekend	Ενδειξη ημέρας καθημερινή ή Σαββατοκύριακο (Weekday/Weekend).

ΔΗΜΙΟΥΡΓΙΑ ΑΠΟΘΗΚΗΣ ΣΤΟΝ SQL SERVER

Βήμα 1: Από το περιβάλλον του Microsoft Sql Server Management Studio δημιουργούμε μια βάση δεδομένων με όνομα salesDW.

Βήμα 2: Δημιουργούμε τον πίνακα salesData και φορτώνουμε τα δεδομένα του αρχείου "salesData.txt".

```
CREATE TABLE SalesData
(
    productKey int,
    orderDate date,
    customerKey int,
    salesTerritoryKey int,
    salesOrderNum varchar(10),
    salesOrderLineNum int,
    orderQuantity int,
    unitPrice numeric(10,4),
    totalProductCost numeric(10,4),
    salesAmount numeric(10,4),
    taxAmt numeric(10,4),
    firstName varchar(30),
    lastName varchar(30),
    gender varchar(1),
    occupation varchar(50),
    productName varchar(100),
    subCategory varchar(50),
    category varchar(50),
    standardCost numeric(10,4),
    listPrice numeric(10,4),
    productLine varchar(50),
    modelName varchar(50),
    region varchar(30),
    country varchar(30),
    s_year int,
    s_quarter varchar(2),
    monthnum int,
    s_month varchar(5),
    fiscalYear varchar(10),
    fiscalQuarter varchar(3),
    fiscalMonthNum int,
    fiscalMonth varchar(5),
    weekDayNum int,
    s_weekDay varchar(5),
    weekDayWeekend varchar(20)
);

bulk insert SalesData
from 'C:\OLAP\SalesData.txt'
with (firstrow=2, fieldterminator='|', rowterminator='\n');
```

Βήμα 3: Δημιουργούμε το σχήμα της αποθήκης δεδομένων. Στην συγκεκριμένη περίπτωση επιλέξαμε ένα μοντέλο αστέρα (star schema) με fact table τον πίνακα των πωλήσεων (Sales) και τέσσερις πίνακες διαστάσεων (Customers, Products, Territories, Calendar).

```
CREATE TABLE Customers
```

```
(
    customerKey int Primary key ,
    firstName varchar(30),
    lastName varchar(30),
    gender varchar(1),
    occupation varchar(50),
);
```

```
CREATE TABLE Products
```

```
(productKey int Primary key,
    productName varchar(100),
    subCategory varchar(50),
    category varchar(50),
    standardCost numeric(10,4),
    listPrice numeric(10,4),
    productLine varchar(50),
    modelName varchar(50)
);
```

```
CREATE TABLE Territories
```

```
(salesTerritoryKey int primary key,
    region varchar(30),
    country varchar(30),
);
```

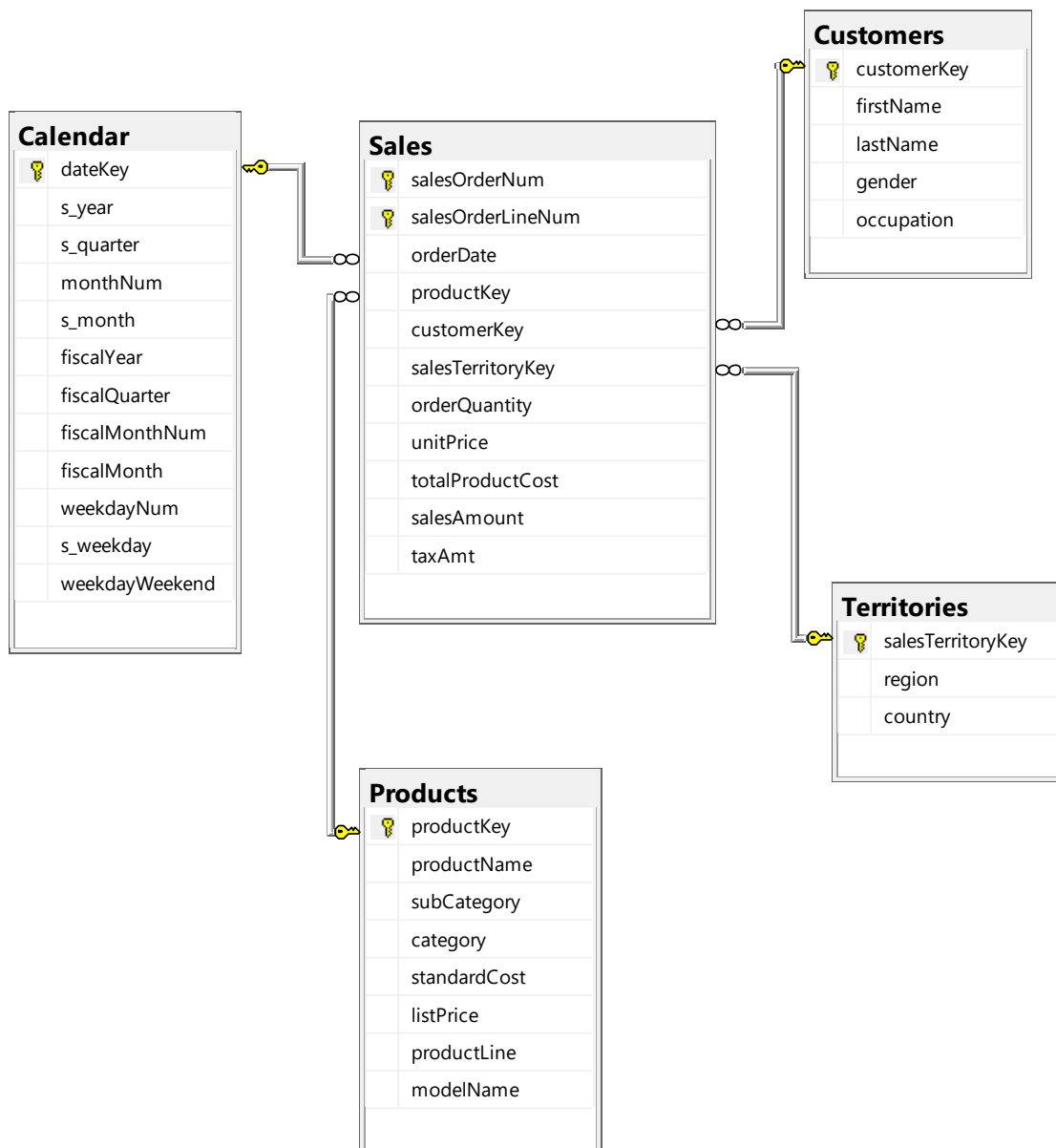
```
CREATE TABLE Calendar
```

```
(dateKey date primary key,
    s_year int,
    s_quarter varchar(2),
    monthNum int,
    s_month varchar(3),
    fiscalYear varchar(10),
    fiscalQuarter varchar(3),
    fiscalMonthNum int,
    fiscalMonth varchar(5),
    weekdayNum int,
    s_weekday varchar(5),
    weekdayWeekend varchar(20)
);
```

```
CREATE TABLE Sales
```

```
(salesOrderNum varchar(10),
    salesOrderLineNum int,
    orderDate date foreign key references Calendar (dateKey),
    productKey int foreign key references Products(productKey),
    customerKey int foreign key references Customers(customerKey),
    salesTerritoryKey int foreign key references Territories(salesTerritoryKey),
    orderQuantity int,
    unitPrice numeric(10,4),
    totalProductCost numeric(10,4),
    salesAmount numeric(10,4),
    taxAmt numeric(10,4)
    PRIMARY KEY (SalesOrderNum, SalesOrderLineNum)
);
```

Διαγραμματική αναπαράσταση του σχήματος αστέρα της αποθήκης δεδομένων.



Βήμα 4: Τροφοδοτούμε με δεδομένα τους πίνακες της αποθήκης.

```
insert into customers
  select distinct customerKey, firstName, LastName, gender, occupation
  FROM SalesData
```

```
insert into Products
  select distinct productKey, productName, subCategory, category,
  standardCost, listPrice, productLine, modelName
  from salesData
```

```
insert into Territories
  select distinct salesTerritoryKey, region, country from salesdata
```

```
insert into Calendar
  select distinct orderdate, s_year, s_quarter, monthNum, s_month,
  fiscalYear, fiscalQuarter, fiscalMonthNum,
  fiscalMonth, weekdayNum, s_weekday, weekdayWeekend
  from salesData
```

```
insert into Sales
  select salesOrdernum, salesOrderLineNum, orderDate, productKey,
  customerKey, salesTerritoryKey, orderQuantity,
  unitPrice, totalProductCost, salesAmount, taxAmt
  from salesData
```

ΕΠΕΡΩΤΗΣΕΙΣ

1. Παράδειγμα επερώτησης με χρήση της διάστασης customers

Συνολική αξία πωλήσεων ανά φύλο (F=female M=male).

```
select gender, sum(salesAmount)
  from Customers, Sales
 where Customers.customerKey=Sales.customerKey
 group by gender
```

2. Παράδειγμα επερώτησης με χρήση της διάστασης Territories

Συνολική αξία των πωλήσεων ανά Χώρα.

```
select country, sum(salesAmount)
  from territories, Sales
 where Territories.salesTerritoryKey=Sales.salesTerritoryKey
 group by country
```

3. Παράδειγμα επερώτησης με χρήση της διάστασης products

Συνολικές πωλήσεις ανά κατηγορία προϊόντων.

```
select category, sum(salesAmount)
  from products, Sales
 where products.productKey=Sales.productKey
 group by category
```

4. Παράδειγμα επερώτησης με χρήση της διάστασης calendar

Συνολική αξία πωλήσεων ανά έτος.

```
select s_year, sum(salesAmount)
  from calendar, Sales
 where dateKey=orderDate
 group by s_year
 order by s_year
```

5. Παραδείγματα επερωτήσεων με συνδυασμό διαστάσεων.

Συνολική αξία των πωλήσεων ανά χώρα και έτος.

```
select country, s_year, sum(salesAmount)
  from Territories, calendar, Sales
 where dateKey=orderDate and
        Territories.salesTerritoryKey=Sales.salesTerritoryKey
 group by country, s_year
 order by s_year
```

6. Αριθμός πωλήσεων ανά έτος με αξία μεγαλύτερη των \$2500.

```
create view v1 as
  select s_year, salesOrderNum, sum(salesAmount) as totalAmount
    from calendar, Sales
     where dateKey=orderDate
     group by s_year, salesOrderNum
     having sum(salesAmount) > 2500

select s_year, count(*) from v1 group by s_year
```

7. Παράδειγμα επερώτησης με χρήση του τελεστή ROLLUP

Ανάλυση πωλήσεων ανά έτος, τοποθεσία, και κατηγορία προϊόντων.

-- Ανάλυση εσόδων ανά έτος, Χώρα και κατηγορία προϊόντων

```
select s_year, country, category, sum(salesAmount)
from Calendar, Territories, Products, Sales
where dateKey=orderDate and
      Territories.salesTerritoryKey=Sales.salesTerritoryKey and
      products.productKey=Sales.productKey
group by rollup (s_year, country, category)
```

Το αποτέλεσμα της επερώτησης είναι η ομαδοποίηση των εσόδων με βάση τους παρακάτω συνδυασμούς:

- s_year, country_category (group by s_year, country, category)
- s_year, country, null (group by s_year, country)
- s_year, null, null (group by s_year)
- null, null, null **Συνολικό ποσό** (group by none)

8. Data Cube

```
select s_year, country, category, sum(salesAmount)
from Calendar, Territories, Products, Sales
where dateKey=orderDate and
      Territories.salesTerritoryKey=Sales.salesTerritoryKey and
      products.productKey=Sales.productKey
group by cube (s_year, country, category)
```

Το αποτέλεσμα της επερώτησης είναι η δημιουργία ενός κύβου, κάθε κελί του οποίου περιέχει τις πωλήσεις για έναν συνδυασμό τιμών (έτους, χώρας, κατηγορίας).

- s_year, country, category (group by s_year, country, category)
- s_year, country, null (group by s_year, country)
- s_year, null, category (group by s_year, category)
- null, country, category (group by country, category)
- s_year, null, null (group by s_year)
- null, country, null (group by country)
- null, null, category (group by category)
- null, null, null (group by none)

Σημείωση: Η τιμή **null** αναπαριστά την τιμή **ALL**.