ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΚΕΝΤΡΟ ΕΠΙΜΟΡΦΩΣΗΣ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗΣ

CODING FACTORY

OLAP & DATAWAREHOUSE

Χρυσόστομος Καπέτης

OLTP & OLAP

- On Line Transaction Processing – *OLTP*
  - Υποστηρίζει τις καθημερινές λειτουργίες μιας βάσης δεδομένων η οποία αποτελεί ακριβές μοντέλο μιας πραγματικής επιχείρησης. Χαρακτηριστικά:
    - Σύντομες και απλές συναλλαγές
    - Σχετικά συχνές ενημερώσεις
    - Οι συναλλαγές επηρεάζουν ένα μικρό σύνολο των δεδομένων της βάσης.
- On Line Analytic Processing – *OLAP*
  - Αξιοποιεί πληροφορίες της βάσης δεδομένων για την υποστήριξη στρατηγικών αποφάσεων:
    - Σύνθετα ερωτήματα
    - Σπάνιες ενημερώσεις
    - Οι συναλαγές προσπελαύνουν σημαντικό όγκο δεδομένων
    - Τα δεδομένα δεν χρειάζεται να είναι ενημερωμένα (up-to-date).

Ηλεκτρονικό Οπωροπωλείο

- OLTP-style transaction:
  - Ο Γιάννης αγόρασε ένα καφάσι ντομάτες από την κεντρική αποθήκη των Αθηνών.
    - Χρέωσε τον λογαριασμό του.
    - Παρέδωσε το καφάσι με τις ντομάτες.
    - Ενημέρωσε το απόθεμα της κεντρικής αποθήκης;
- OLAP-style transaction:
  - Πόσα καφάσια ντομάτες πουλήθηκαν από όλες τις αποθήκες της Αθήνας τα έτη 2021 και 2022;

OLAP: Traditional vs Newer Applications

- Traditional OLAP queries
  - Χρήση δεδομένων που συλλέγει η επιχείρηση μέσω ενός συστήματος OLTP
  - Ερωτήματα κατά περίπτωση (ad-hoc)
- Newer Applications (e.g., Internet companies)
  - Στοχευμένη συλλογή δεδομένων (ίσως και αγορά).
  - Εξελιγμένα ερωτήματα σχεδιασμένα από επαγγελματίες.

Ηλεκτρονικό Οπωροπωλείο

- Traditional
  - Πόσα καφάσια ντομάτες πουλήθηκαν σε όλες τις αποθήκες της Αθήνας στην διάρκεια των ετών 2021 και 2022;
- Newer
  - Δημιούργησε ένα προφίλ αγορών για τον Γιάννη για τα έτη 2021 και 2022, ώστε να μπορέσουμε να προσαρμόσουμε το marketing στις ανάγκες του και να επωφεληθούμε περισσότερο από την επιχείρησή του.

Αποθήκες Δεομένων (Data Warehouses)

- Τα συστήματα OLAP αποθηκεύονται σε ειδικούς διακομιστές οι οποίοι καλούνται αποθήκες δεδομένων (***data warehouses)***:
  - Μπορούν να φιλοξενήσουνε τεράστιο όγκο δεδομένων που συνήθως παράγονται από τα συστήματα OLTP
  - Επιτρέπουν την εκτέλεση σύνθετων επερωτήσεων δίχως να επηρεάζεται η απόδοση των συστημάτων OLTP.

Γεγονότα (Fact Tables)

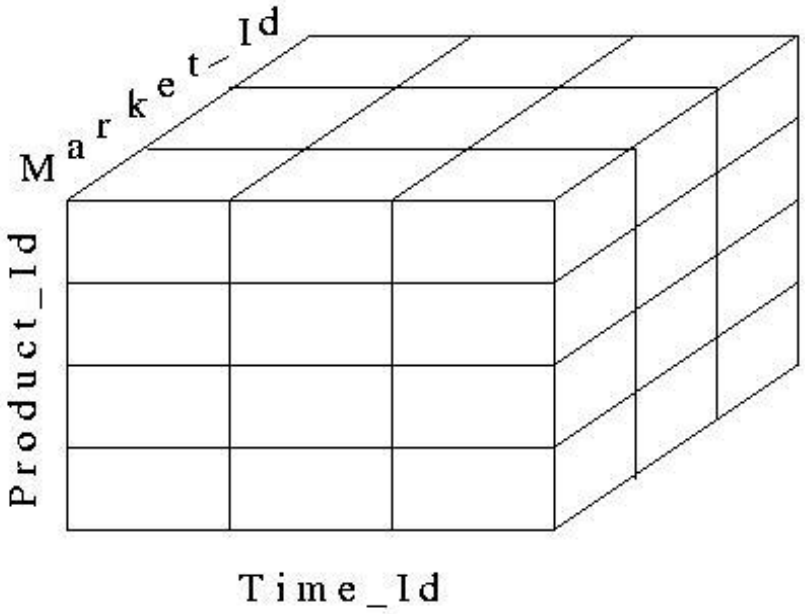
- Πολλά συστήματα OLAP βασίζονται σε έναν πίνακα ο οποίος καλείται ***fact table*** και περιέχει τα μετρήσιμα μεγέθη.
- Για παράδειγμα μια εφαμογή ενός Super-Market μπορεί να βασίζεται στο παρακάτω fact table:

  Sales (*Market_Id*, *Product_Id*, *Time_Id*, *Sales_Amt*)

- Ο παραπάνω πίνακας μπορεί να θεωρηθεί πολυδιάστατος
  - *Market_Id*, *Product_Id*, *Time_Id* είναι διαστάσεις οι οποίες αντιπροσωπέυουν συγκεκριμένα καταστήματα, προϊόντα και χρονικές περιόδους.
  - *Sales_Amt* μετρήσιμο μέγεθος (ως συνάρτηση των διαστάσεων).

Data Cube

- Ο πίνακας με τα γεγονότα (fact table) μπορεί να θεωρηθεί ως ένας πολυδιάστατος (3-διαστάσεων στο παράδειγμά μας) κύβος δεδομένων (data cube).
- Οι καταχωρήσεις στον κύβο είναι οι τιμές του πεδίου Sales_Amts



Διαστάσεις (Dimension Tables)

- Οι διαστάσεις του **fact table** αναλύονται στους **πίνακες διαστάσεων** (***dimension tables)***
- **Fact table:**

  Sales (*Market_id*, *Product_Id*, *Time_Id*, *Sales_Amt*)

- **Dimension Tables:**

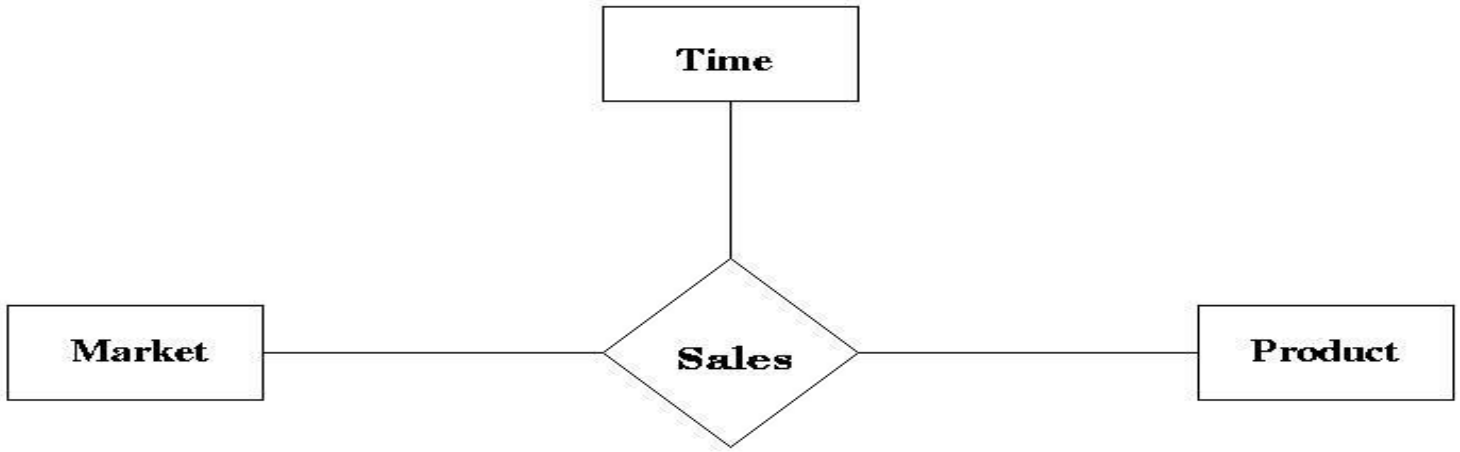
  Market (*Market_Id*, *City*, *State*, *Region*)

  Product (*Product_Id*, *Name*, *Category*, *Price*)

  Time (*Time_Id*, *Week*, *Month*, *Quarter*)

Σχήμα Αστέρα (Star Schema)

- Οι συσχετίσεις ανάμεσα στο πίνακα fact και στους πίνακες των διαστάσεων μπορούν να αποτυπωθούν σε ένα διάγραμμα E-R το οποίο έχει διάταξη αστέρα (***star schema).***



Συναθροίσεις (Aggregation)

- Πολλά ερωτήματα OLAP περιλαμβάνουν συναθροίσεις δεδομένων στον πίνακα **fact table**
- Παράδειγμα, για να βρούμε τις συνολικές πωλήσεις (με την πάροδο του χρόνου) για κάθε προιόν, σε κάθε κατάστημα μπορούμε να χρησιμοποιήσουμε την παρακάτω εντολή:

```
SELECT    S.Market_Id, S.Product_Id, SUM (S.Sales_Amt)
FROM      Sales S
GROUP BY  S.Market_Id, S.Product_Id
```

- Η συνάρτηση γίνεται για όλη την χρονική περίοδο (διάσταση χρόνος) και έτσι παράγει μια δισδιάστατη προβολή των δεδομένων.

Aggregation over Time

- Το αποτέλεσμα της προηγούμενης επερώτησης

Market_Id

SUM(*Sales_Amt*)	M1	M2	M3	M4
P1	3003	1503	…	
P2	6003	2402	…	
P3	4503	3	…	
P4	7503	7000	…	
P5	…	…	…	

Product_Id

Drilling Down and Rolling Up

- Τα γνωρίσματα ορισμένων διαστάσεων μπορεί να σχηματίζουν ιεραρχίες (***aggregation hierarchy***)

  $Market\_Id \rightarrow City \rightarrow State \rightarrow Region$

- Η εκτέλεση μιας σειράς επερωτήσεων που κινούνται προς τα κατώτερα επίπεδα μιας ιεραρχίας (π.χ. από συνάθροιση ανα περιοχή σε συνάθροιση ανα πόλη) καλείται ***drilling down***
  - Απαιτείται η χρήση πληροφοριών από τον fact_table
- Η εκτέλεση μιας σειράς επερωτήσεων που κινούνται προς τα ανώτερα επιπέδα μιας ιεραρχίας (π.χ. από συνάθροιση ανα πόλη σε συνάθροιση ανα περιοχή) καλείται ***rolling up***
  - Note: Σε μια λειτουργία rollup, γενικότερες συναθροίσεις (συναθροίσεις σε υψηλότερα επίπεδα της ιεραρχίας) μπορούν να υπολογιστούν από ειδικότερες συναθροίσεις (συναθροίσεις σε κατώτερα επίπεδα της ιεραρχίας).

Drilling Down

- Drilling down on market: από περιοχή σε πόλη

Sales (*Market_Id, Product_Id, Time_Id, Sales_Amt*)

Market (*Market_Id, City, State, Region*)

1.
```
SELECT    S.Product_Id, M.Region, SUM (S.Sales_Amt)
FROM      Sales S, Market M
WHERE     M.Market_Id = S.Market_Id
GROUP BY  S.Product_Id, M.Region
```

2.
```
SELECT    S.Product_Id, M.State, SUM (S.Sales_Amt)
FROM      Sales S, Market M
WHERE     M.Market_Id = S.Market_Id
GROUP BY  S.Product_Id, M.State,
```

Rolling Up

- Rolling up on market, από πόλη σε περιοχή
  - Αν έχουμε ήδη δημιουργήσει έναν πίνακα (ή όψη), State_Sales, για να αποθκεύσουμε το αποτέλεσμα της παρτακάτω εντολής:

```
1.  SELECT   S.Product_Id,  M.State, SUM (S.Sales_Amt)
    FROM     Sales  S,   Market  M
    WHERE    M.Market_Id = S.Market_Id
    GROUP BY  S.Product_Id,  M.State
```

τότε μπορούμε να κάνουμε rollup με την παρκάτω εντολή:

```
2.  SELECT   T.Product_Id,  M.Region, SUM (T.Sales_Amt)
    FROM     State_Sales  T,  Market  M
    WHERE    M.State = T.State
    GROUP BY T.Product_Id,  M.Region
```

Pivoting

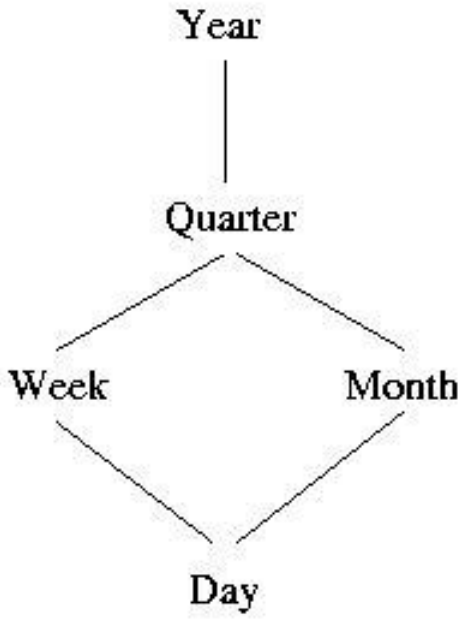
- Αν θεωρήσουμε τα δεδομένα ως πολυδιάστατο κύβο και εκτλέσουμε μια ομαδοποίηση ως προς ένα υποσύνολο των αξόνων, λέμε ότι εκτελούμε μία περιστροφή σε αυτούς τους άξονες **(pivoting).**
  - Pivoting στις διαστάσεις $D_1,\ldots,D_k$ ενός κύβου δεδομένων $D_1,\ldots,D_k,D_{k+1},\ldots,D_n$ σημαίνει ότι ομαδοποιούμε (GROUP BY) με τα γνωρίσματα $A_1,\ldots,A_k$ και συναθροίζουμε με τα γνωρίσματα $A_{k+1},\ldots A_n$, όπου A_i είναι ένα γνώρισμα της διάστασης D_i
  - *Παράδειγμα*: Το Pivoting στις διαστάσεις Product και Time αντιστοιχεί σε ομαδοποίηση με βάση το πεδία *Product_id* και *Quarter* και συνάθροιση του πεδίου *Sales_Amt* για όλα τα καταστήματα (over *Market_id):*

```
SELECT    S.Product_Id, T.Quarter, SUM (S.Sales_Amt)
FROM      Sales S, Time T
WHERE     T.Time_Id = S.Time_Id
GROUP BY  S.Product_Id, T.Quarter
```

Pivot

Πλέγμα (Lattice) ιεραρχίας χρόνου

- Δεν είναι όλες οι ιεραρχίες γραμμικές.
  - Ο χρόνος σχηματίζει μια ιεραρχία υπό την μορφή πλέγματος (γράφου).
    - Οι εβδομάδες δεν περιέχονται σε μήνες.
    - Μπορούμε να κάνουμε rollup τις ημέρες σε εβδομάδες ή μήνες αλλά μπορούμε να κανουμε μόνο rollup τις εβδομάδες σε τρίμηνα.



Slicing-and-Dicing

- Όταν χρησιμοποιούμε τον προσδιοριστή **WHERE** για να ορίσουμε μια συγκεκριμένη τιμή για έναν ή περισσότερτους άξονες τότε λέμε ότι εκτελούμε την λειτουργία της τμηματοποίησης (**slice)**
  - Τμηματοποίηση (Slicing) του κύβουμε βάση την διάσταση Time (επιλέγουμε μόνο της πωλήσεις της εβδομάδας 12) και πεστροφή (pivoting) με βάση το πεδίο *Product_id* (συνάθροιση για όλα τα καταστήματα, market id)

```
SELECT   S.Product_Id,  SUM (Sales_Amt)
FROM     Sales S, Time T
WHERE    T.Time_Id = S.Time_Id  AND  T.Week = 'Wk-12'
GROUP BY  S. Product_Id
```

Slice

Pivot

Ο Τελεστής CUBE

- Για την δημιουργία του παρακάτω πίνακα απαιτείται η εκτέλεση των τριών επερωτήσεων της επόμενης διαφάνειας.

SUM(*Sales_Amt*)	*Market_Id* M1	M2	M3	*Total*
Product_Id P1	3003	1503	…	…
P2	6003	2402	…	…
P3	4503	3	…	…
P4	7503	7000	…	…
Total	…	…	…	…

Οι Τρείς Επερωτήσεις

- Για τα περιεχόμενα του πίανακα δίχως τα συγκεντρωτικά ποσά (aggregation on time)

```
SELECT    S.Market_Id, S.Product_Id, SUM (S.Sales_Amt)
FROM      Sales S
GROUP BY  S.Market_Id, S.Product_Id
```

- Για τον υπολογισμό των συνόλων ανα γραμμές (aggregation on time and supermarkets)

```
SELECT    S.Product_Id, SUM (S.Sales_Amt)
FROM      Sales S
GROUP BY  S.Product_Id
```

- Για τον υπολογισμό των συνόλων των στηλών (aggregation on time and products)

```
SELECT    S.Market_Id, SUM (S.Sales)
FROM      Sales S
GROUP BY  S.Market_Id
```

Ορισμός του Τελεστή CUBE

- Η εκτέλεση των προηγούμενων τριών επερωτήσεων είναι σπατάλη χρόνου.
  - Η πρώτη επερώτηση κάνει μέρος της δουειάς των άλλων δύο: αν μπορούμε να αποθηκεύσουμε το αποτέλεσμα και να συναθροίσουμε για τα καταστήματα (market_id) και τα προϊόντα (product_id) μπορούμε να υπολογίσουμε τα επερωτήματα αποτελεσματικότερα.
- Ο τελεστής CUBE είναι μέρος της SQL:1999
  - GROUP BY CUBE (v1, v2, …, vn)
  - Ισοδυναμεί με ένα σύνολο από of GROUP BYs, ένα για κάθε έναν από τους 2^n συνδυασμούς των v1, v2, …, vn.

Παράδειγμα Χρήσης του Τελεστή CUBE.

- Το παρακάτω επερώτημα επιστρέφει όλες τις πληροφορίες που απαιτούνται για την δημιουργία του προηγούμενου πίνακα (products/markets):

```
SELECT S.Market_Id, S.Product_Id, SUM (S.Sales_Amt)
FROM Sales S
GROUP BY CUBE (S.Market_Id, S.Product_Id)
```

Ο Τελεστής ROLLUP

- Ο τελστής **ROLLUP** είναι παρόμοιος με τον τελεστή **CUBE** με την διαφορά ότι αντί να συναθροίζει για όλα τα υποσύνολα των ορισμάτων, δημιουργεί υποσύνολα από δεξιά προς τα αριστερά:
- GROUP BY ROLLUP (A_1,A_2,…,A_n) ισοδυναμεί με τις παρακάτω ομαδοποιήσεις:
  - GROUP BY A_1 ,…, A_{n-1} ,A_n
  - GROUP BY A_1 ,…, A_{n-1}
  - … … …
  - GROUP BY A_1, A_2
  - GROUP BY A_1
  - *No* GROUP BY

Παράδειγμα Χρήσης του Τελεστή ROLLUP

SELECT S.*Market_Id*, S.*Product_Id*, SUM (S.*Sales_Amt*)

FROM Sales S

GROUP BY ROLLUP (S.*Market_Id*, S. *Product_Id*)

- first aggregates with the finest granularity:

  GROUP BY S.*Market_Id*, S.*Product_Id*

- then with the next level of granularity:

  GROUP BY S.*Market_Id*

- then the grand total is computed with *no* GROUP BY clause

ROLLUP vs. CUBE

- Η ίδια επερώτηση με τον τελστή CUBE:

  - first aggregates with the finest granularity:

    GROUP BY S.*Market_Id*, S.*Product_Id*

  - then with the next level of granularity:

    GROUP BY S.*Market_Id*

    and

    GROUP BY *S.Product_Id*

  - then the grand total with *no* GROUP BY

Materialized Views

Ο τελστής CUBE χρησιμοποιείται συχνά για τον υπολογισμό συναθροίσεων σε όλες τις διαστάσεις ενός πίνακα γεγονότων και στη συνέχεια για την αποθήκευσή τους ως υλοποιημένες όψεις (Materialized Views) για την επιτάχυνση μελλοντικών επερωτήσεων.

ROLAP and MOLAP

- Relational OLAP: ROLAP
  - Τα δεδομένα OLAP αποθηκεύονται σε μια σχεσιακή βάση δεδομένων. Ο κύβος δεδομένων είναι μία εννοιολογική θεώρηση για έναν πίνακα γεγονότων (fact table).
- Multidimensional OLAP: MOLAP
  - Ορισμένοι κατασκευαστές παρέχουν υπηρεσίες OLAP οι οποίες υλοποιούν έναν πίνακα γεγονότων (fact table) ως κύβο δεδομένων, χρησιμοποιώντας ειδικές πολυδιάστατες (μη σχεσιακές) δομές δεδομένων.

Data Warehouse

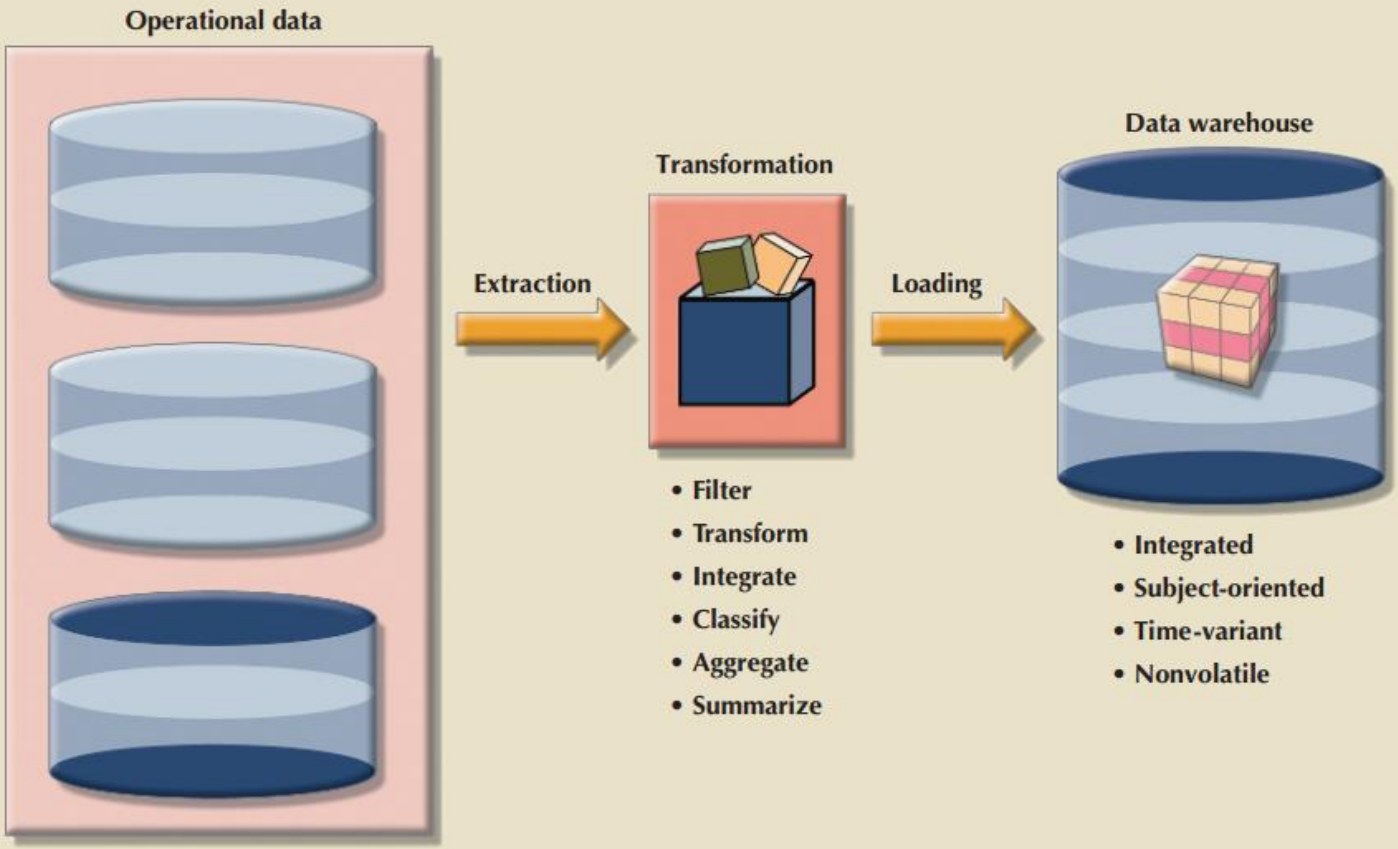
- Τα δεδομένα των συστημάτων OLAP αποθηκεύονται συνήθως σε μία βάση δεδομένων η οποία καλείται ***data warehouse***
- Οι αποθήκες δεδομένων (data warehouses) περιέχουν μεγάλο όγκο δεδομένων. Τα δεδομένα αυτά συλλέγονται σε διαφορετκούς χρόνους από βάσεις δεδομένων συστημάτων OLTP οι οποίες έχουν διαφορετικά σχήματα και διαχειρίζονται από διαφορετικά DBMS.

Ζητήματα Διαχείρισης Αποθηκών Δεδομένων

- *Μετασχηματισμοί (Transformations)*
  - *Syntactic*: η σύνταξη των εντολών μπορεί να διαφέρει μεταξύ διαφορετικών συστημάτων DMBS και διαφορετικών βάσεων δεδομένων:
    - Attribute names: SSN vs. Ssnum
    - Attribute domains: Integer vs. String
  - *Semantic*: διαφορετική σημασιολογία
    - Συνάθροιση πωλήσεων ανα ημέρα vs. Συνάθροιση πωλήσεων σε μηνιαία βάση.
- *Data Cleaning*
  - Απαλοιφή ασυνεπειών και λαθών στα δεδομένα.

Διαδικασία ETL

FIGURE 13.5 THE ETL PROCESS



Αρχιτεκτονική Αποθηκών Δεδομένων