- Αδυναμία σχεσιακού μοντέλου
- Βασικές έννοιες αντικειμενοστρεφών ΣΔΒΔ
- Πλεονεκτήματα και μειονεκτήματα

Αδυναμία Σχεσιακού Μοντέλου

- Το σχεσιακό μοντέλο αποτελεί ακόμη και σήμερα το πιο διαδεδομένο μοντέλο που υποστηρίζει τις περισσότερες εφαρμογές διαχειριστικού τύπου (τραπεζικά συστήματα, συστήματα κράτησης θέσεων, κλπ).
- Ωστόσο, υπάρχουν σύγχρονες εφαρμογές που θέτουν μεγάλες απαιτήσεις με αποτέλεσμα η υποστήριξη των εφαρμογών αυτών από το σχεσιακό μοντέλο να είναι αρκετά δύσκολη.
- Παραδείγματα σύγχρονων εφαρμογών είναι τα συστήματα CAD (computer-aided design), GIS (geographical information systems) και εφαρμογές πολυμέσων (multimedia applications).

Αδυναμία Σχεσιακού Μοντέλου

- Οι σύγχρονες εφαρμογές παρουσιάζουν σημαντικές διαφορές σε σχέση με τις παραδοσιακές:
  - τα δεδομένα χαρακτηρίζονται από μεγαλύτερο όγκο και μεγαλύτερη πολυπλοκότητα,
  - η διαχείριση των δεδομένων απαιτεί την υλοποίηση πολύπλοκων αλγορίθμων (π.χ. την εύρεση της συντομότερης διαδρομής σε ένα χάρτη),
  - τα ερωτήματα είναι πιο πολύπλοκα με αποτέλεσμα να απαιτούν περισσότερο χρόνο για την επεξεργασία τους.

Αδυναμία Σχεσιακού Μοντέλου

- Έχει επιβεβαιωθεί πολλές φορές ότι τα σχεσιακά ΣΔΒΔ δεν επαρκούν για την ικανοποίηση των σύγχρονων απαιτήσεων.
- Το ερώτημα που προκύπτει εύλογα είναι γιατί τα σχεσιακά ΣΔΒΔ αδυνατούν να καλύψουν τις ανάγκες των σύγχρονων εφαρμογών;
- Στη συνέχεια περιγράφονται μερικά από τα προβλήματα που εμφανίζονται χρησιμοποιώντας ένα σχεσιακό ΣΔΒΔ.

Αδυναμία Σχεσιακού Μοντέλου

- **Αδυναμία αναπαράστασης του πραγματικού κόσμου**. Η διαδικασία της κανονικοποίησης οδηγεί στη δημιουργία σχέσεων που δεν αντιστοιχούν σε οντότητες του πραγματικού κόσμου.
- Η κατατεμάχιση των δεδομένων σε πολλούς πίνακες οδηγεί στην εκτέλεση **πολλών χρονοβόρων πράξεων σύνδεσης**.

Αδυναμία Σχεσιακού Μοντέλου

- **Ομοιογένεια**. Το σχεσιακό μοντέλο ορίζει ότι κάθε γραμμή ενός πίνακα πρέπει να αποτελείται από τις ίδιες στήλες, ενώ κάθε στήλη του πίνακα πρέπει να δέχεται τιμές από το ίδιο πεδίο ορισμού. Οι δύο αυτές ιδιότητες καλούνται ***οριζόντια*** και ***κάθετη ομοιογένεια***.
- **Η δομή** αυτή **του πίνακα** είναι **αρκετά περιοριστική** για **αντικείμενα** του πραγματικού κόσμου τα οποία έχουν **πολύπλοκη δομή**.

Αδυναμία Σχεσιακού Μοντέλου

- **Περιορισμένες λειτουργίες**. Τα σχεσιακά ΣΔΒΔ διαθέτουν ένα **περιορισμένο σύνολο λειτουργιών επί των δεδομένων**, το οποίο προσδιορίζεται από την **SQL**.
- ***Σήμερα απαιτείται η δυνατότητα ορισμού νέων τύπων δεδομένων*** και λειτουργιών:
  - μία εφαρμογή GIS χρειάζεται υποστήριξη για γεωμετρικά αντικείμενα (γραμμές, πολύγωνα) και πράξεις μεταξύ των γεωμετρικών αντικειμένων (τομές γραμμών, επικάλυψη πολυγώνων κλπ.).
  - τα παραδοσιακά σχεσιακά ΣΔΒΔ δεν επιτρέπουν τον ορισμό νέων τύπων δεδομένων και λειτουργιών, οπότε η διαχείριση των αντικειμένων εμπεριέχεται στη λογική της εφαρμογής και όχι στο ΣΔΒΔ.

Αδυναμία Σχεσιακού Μοντέλου

- **Συναλλαγές μικρής διάρκειας**. Οι συναλλαγές στις παραδοσιακές εφαρμογές είναι συνήθως μικρής διάρκειας με αποτέλεσμα οι μηχανισμοί ταυτόχρονης εκτέλεσης (πχ. κλείδωμα δύο φάσεων, χρονικές σφραγίδες) να είναι επαρκείς.
- Αν οι συναλλαγές έχουν μεγάλη διάρκεια, πράγμα που παρατηρείται σε περιπτώσεις πολύπλοκων αντικειμένων, οι ανωτέρω μηχανισμοί δεν επαρκούν.

Αδυναμία Σχεσιακού Μοντέλου

- **Οι αλλαγές στο σχήμα της ΒΔ είναι χρονοβόρες**. Αν απαιτηθεί αλλαγή στο σχήμα της ΒΔ θα πρέπει ο διαχειριστής να διακόψει προσωρινά τη λειτουργία του συστήματος και επιπλέον τα προγράμματα εφαρμογής πρέπει να διαμορφωθούν αναλόγως.

Τάσεις

- Τα προαναφερθέντα προβλήματα οδήγησαν στη μελέτη νέων μεθόδων μοντελοποίησης και διαχείρισης ώστε να είναι εφικτή η αποτελεσματική και αποδοτική διαχείριση των σύγχρονων δεδομένων.
- Προς αυτήν την κατεύθυνση παρατηρούνται δύο τάσεις που στηρίζονται σε διαφορετική φιλοσοφία:
  - στην εφαρμογή του **αντικειμενοστρεφούς μοντέλου δεδομένων** (object-oriented data model) και την υλοποίηση **αντικειμενοστρεφών ΒΔ** (object-oriented databases),
  - στην **επέκταση** των σχεσιακών ΒΔ ώστε **μία σχεσιακή βάση να είναι σε θέση να διαχειρίζεται πολύπλοκα αντικείμενα**. Η τάση αυτή οδήγησε στη δημιουργία των **αντικειμενοσχεσιακών ΒΔ** (object-relational databases)

Αντικειμενοστρεφές Μοντέλο

- Το αντικειμενοστρεφές μοντέλο δεδομένων έχει εφαρμοσθεί με επιτυχία στις αντικειμενοστρεφείς γλώσσες προγραμματισμού όπως C++ και Java.
- Η υλοποίηση μίας εφαρμογής με μία αντικειμενοστρεφή γλώσσα προγραμματισμού διαφέρει σημαντικά από την υλοποίηση της ίδιας εφαρμογής σε μία διαδικαστική γλώσσα. Επιπλέον, οι μέθοδοι ανάλυσης και σχεδίασης διαφέρουν από τις αντίστοιχες κλασικές μεθόδους.

Αντικειμενοστρεφές Μοντέλο

- Ένα **αντικείμενο** αποτελεί μία οντότητα του πραγματικού κόσμου (σε αντιστοιχία με το μοντέλο ΟΣ).
- Κάθε αντικείμενο αποτελείται από **χαρακτηριστικά**. Το σύνολο των τιμών των χαρακτηριστικών καλείται **κατάσταση** του αντικειμένου.
- *Ένα αντικείμενο καθορίζει και τις επιτρεπτές ενέργειες που μπορούν να εφαρμοσθούν στο αντικείμενο*. Αυτό πραγματοποιείται με τη βοήθεια **μεθόδων** με δυνατότητα προσπέλασης στα χαρακτηριστικά του αντικειμένου. Το σύνολο των μεθόδων καλείται **συμπεριφορά** του αντικειμένου.

Αντικειμενοστρεφές Μοντέλο

- Παρατηρούμε μία σημαντική διαφορά σε σχέση με τις διαδικαστικές γλώσσες προγραμματισμού, όπου **δεδομένα και μέθοδοι ορίζονται ξεχωριστά**. Η ιδιότητα αυτή των αντικειμένων καλείται **ενθυλάκωση** (encapsulation).
- **Η προσπέλαση των δεδομένων** ενός αντικειμένου πραγματοποιείται **μόνο μέσω των μεθόδων**. Έτσι αποτρέπεται η αλλαγή των χαρακτηριστικών απευθείας από άλλο αντικείμενο και **επιτυγχάνεται απόκρυψη πληροφοριών** (information hiding).

Αντικειμενοστρεφές Μοντέλο

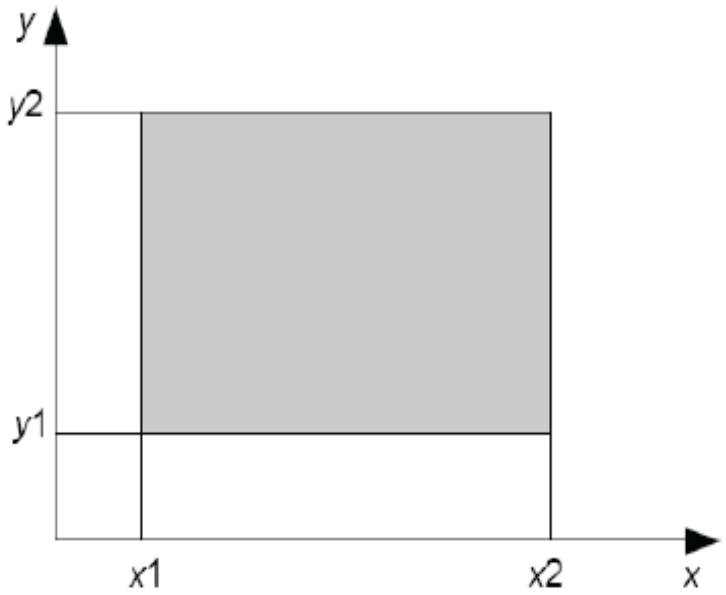
- *Αντικείμενα με τα ίδια χαρακτηριστικά και μεθόδους σχηματίζουν μία κλάση αντικειμένων* (object class).
- Το κάθε αντικείμενο αποτελεί ένα **στιγμιότυπο** της **κλάσης** όπου ανήκει.
- Η διάκριση των αντικειμένων μεταξύ τους πραγματοποιείται με τη χρήση της ταυτότητας αντικειμένου (object identifier), η οποία είναι μοναδική για κάθε αντικείμενο και ποτέ δεν επαναχρησιμοποιείται.

Αντικειμενοστρεφές Μοντέλο

χαρακτηριστικά

μέθοδοι

```
class rectangle :
{
    int x1;
    int y1;
    int x2;
    int y2;

public:
    setRectangle (int x1, int y1, int x2, int y2);
    resizeRectangle (int dx, int dy);
    moveRectangle (int newx1, int newy1);
};
```



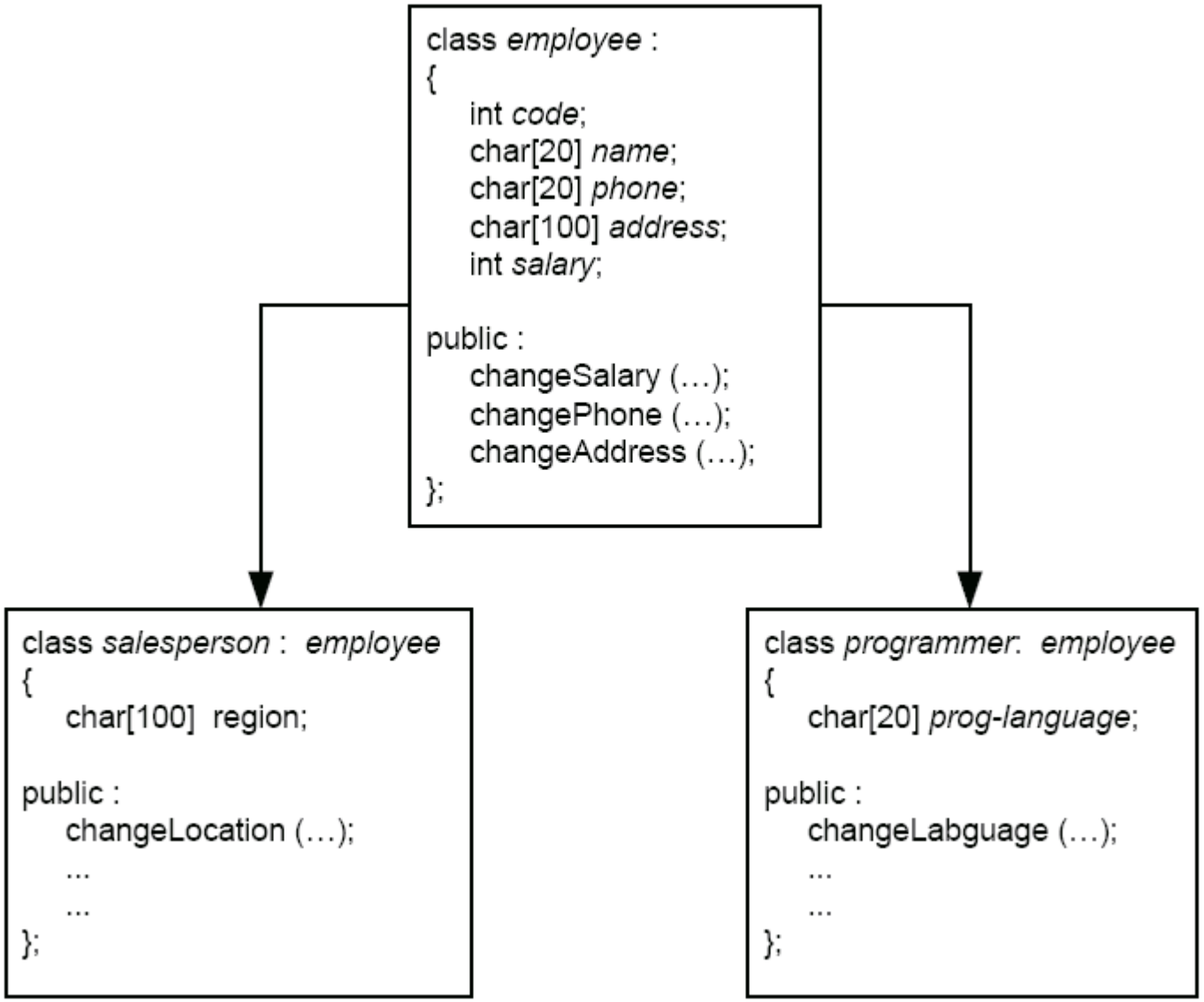
Αντικειμενοστρεφές Μοντέλο

- Για τον ορισμό ενός σχήματος ΒΔ απαιτούνται συνήθως πολλές κλάσεις αντικειμένων. Αν και μπορεί δύο ή περισσότερες κλάσεις να είναι διαφορετικές μεταξύ τους, είναι δυνατόν να έχουν και αρκετά κοινά χαρακτηριστικά και μεθόδους.
- Η **κληρονομικότητα** (inheritance) *επιτρέπει σε δύο κλάσεις να μοιράζονται χαρακτηριστικά και μεθόδους*. Όταν μία κλάση Β κληρονομεί την κλάση Α, τότε το σύνολο των χαρακτηριστικών της Β είναι υπερσύνολο του συνόλου των χαρακτηριστικών της Α (το ίδιο ισχύει και για τις μεθόδους).

Αντικειμενοστρεφές Μοντέλο

- Για παράδειγμα, οι κλάσεις Υπάλληλος (employee), Πωλητής (salesperson)και Προγραμματιστής (programmer) μίας ΒΔ που χρησιμοποιείται για την εσωτερική οργάνωση της εταιρείας.
- Προφανώς ένας πωλητής είναι επίσης και υπάλληλος, άρα οι κλάσεις Υπάλληλος και Πωλητής θα έχουν αρκετά κοινά χαρακτηριστικά και μεθόδους. Το ίδιο θα ισχύει για τις κλάσεις Υπάλληλος και Προγραμματιστής.

Αντικειμενοστρεφές Μοντέλο



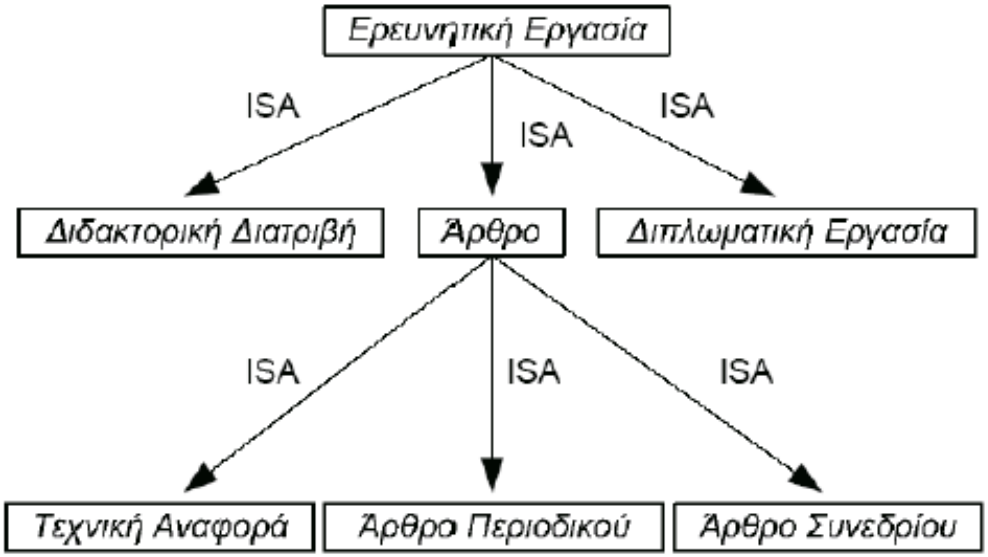
Αντικειμενοστρεφές Μοντέλο

- Με τη χρήση της κληρονομικότητας μπορούμε να κατασκευάσουμε μία **ιεραρχία εξειδίκευσης** (specia-lization hierarchy), η οποία συνδέει τις κλάσεις ως προς την κληρονομικότητα.
- Για να δηλώσουμε ότι η κλάση Β είναι ειδική περίπτωση της κλάσης Α χρησιμοποιείται η λέξη **ISA** (Β ISA Α), όπως ακριβώς και στο μοντέλο ΟΣ.

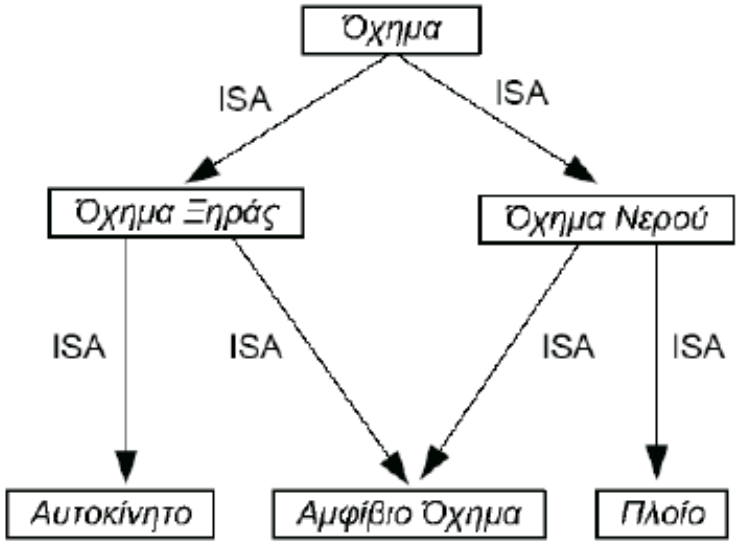
Αντικειμενοστρεφές Μοντέλο

- **Πολλαπλή κληρονομικότητα**. Μία ιεραρχία εξειδίκευσης μπορεί να γίνει πιο πολύπλοκη αν μια κλάση μπορεί να κληρονομεί χαρακτηριστικά και μεθόδους από πολλές κλάσεις.

Αντικειμενοστρεφές Μοντέλο



(α) απλή κληρονομικότητα



(β) πολλαπλή κληρονομικότητα

Αντικειμενοστρεφές Μοντέλο

- **Σύνθεση**. Ένας άλλος τύπος ιεραρχικής δομής που χρησιμοποιείται συχνά για την αναπαράσταση του πραγματικού κόσμου είναι η **ιεραρχία συμπερίληψης** (containment hierarchy), η οποία *ορίζει ότι ένα αντικείμενο αποτελεί τμήμα ενός άλλου αντικειμένου* ***(αποτελείται από)***. Η σχέση αυτή μεταξύ των αντικειμένων δηλώνεται με το IS-PART-OF.
- Η ιεραρχία συμπερίληψης οδηγεί σε **σύνθετα** (complex) **αντικείμενα** με αποτέλεσμα κάποια από τα χαρακτηριστικά ενός αντικειμένου να είναι και αυτά αντικείμενα.

Αντικειμενοστρεφές Μοντέλο

- **DownCasting- UpCasting**

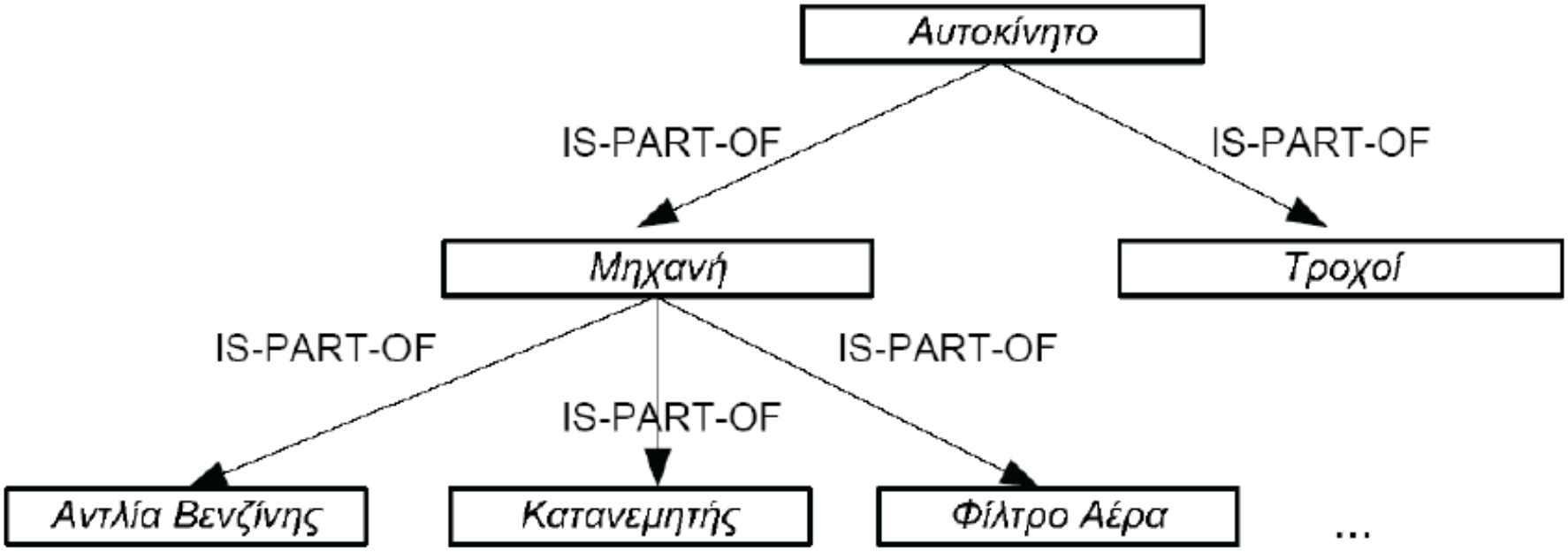
```
class Animal {
   int health = 100;
}

class Mammal extends Animal {
        public void print1(){
                System.out.println("mammal");
        }
}

class Cat extends Mammal {
        public void print2(){
                System.out.println("cat");
        }
}

class Dog extends Mammal {
        public void print3(){
                System.out.println("dog");
        }
}
```

Αντικειμενοστρεφές Μοντέλο

- **DownCasting- UpCasting**

```
public class UpDownCasting2 {
    public static void main(String[] args) {
        Cat c = new Cat();
        System.out.println(c.health);
        Dog d = new Dog();
        System.out.println(d.health);
        // Upcasting
        Mammal m1 = new Cat();
        m1.print1();
        // m.print2(); Δεν μπορεί να καλεστεί
        ((Cat)m1).print2(); // Έγκυρο - Downcasting
        // Downcasting
        // Check the object's class
        Mammal m2 = new Dog();
        if (m2 instanceof Dog){ //Safe
            Dog d2 = (Dog)m2;
            d2.print1();
            d2.print3();
        }
    }
}
```

Αντικειμενοστρεφές Μοντέλο



Αντικειμενοστρεφές Μοντέλο

- *Ένα αντικειμενοστρεφές ΣΔΒΔ (OODBMS) είναι ένα ΣΔΒΔ που χρησιμοποιεί ως βάση το αντικειμενοστρεφές μοντέλο δεδομένων.*
- Η έρευνα και η ανάπτυξη των συστημάτων αυτών **έχει επηρεασθεί** σε μεγάλο βαθμό **από το χώρο των αντικειμενοστρεφών γλωσσών προγραμματισμού**.
- Η ανάπτυξη των αντικειμενοστρεφών ΣΔΒΔ πραγματοποιείται με την προσθήκη της κατάλληλης λειτουργικότητας σε μία αντικειμενοστρεφή γλώσσα προγραμματισμού.

Αντικειμενοστρεφές Μοντέλο

- Οι βασικές δυνατότητες που πρέπει να έχει **ένα αντικειμενοστρεφές ΣΔΒΔ** προσδιορίζονται σύμφωνα με δύο κριτήρια:
  - πρέπει το σύστημα να είναι αντικειμενοστρεφές, και
  - πρέπει να παρέχει όλες τις δυνατότητες ενός ΣΔΒΔ.
- Οι δυνατότητες αυτές αναλύονται στο object-oriented database system manifesto[Atkinson1989] και είναι συνολικά δεκατρείς. Οι πρώτες οκτώ αφορούν στην υποστήριξη του αντικειμενοστρεφούς μοντέλου δεδομένων. Οι υπόλοιπες πέντε δυνατότητες αφορούν στη λειτουργικότητα του ΣΔΒΔ.

Αντικειμενοστρεφές Μοντέλο

- Αποθήκευση αντικειμένων.
- Διαχείριση μεγάλων ΒΔ (απαιτείται αποτελεσματική διαχείριση δευτερεύουσας μνήμης).
- Επεξεργασία ερωτημάτων.
- Συγχρονισμός ταυτόχρονων προσπελάσεων σε αντικείμενα.
- Επανάκτηση δεδομένων.

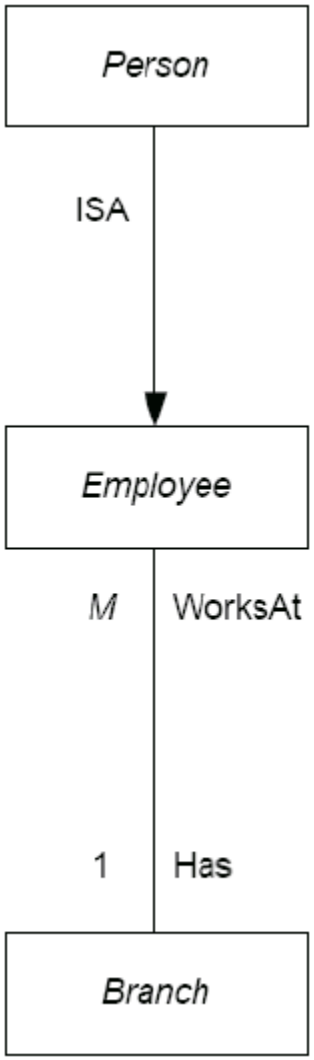
Αντικειμενοστρεφές Μοντέλο

- Μία αντικειμενοστρεφής γλώσσα προγραμματισμού **δεν έχει δυνατότητα διατήρησης** (persistence) **των αντικειμένων**. Αυτό σημαίνει ότι μετά τον τερματισμό της εφαρμογής **τα αντικείμενα καταστρέφονται** και δεν είναι διαθέσιμα σε άλλη εφαρμογή.
- Το γεγονός αυτό οδήγησε στη **μελέτη διατηρήσιμων γλωσσών προγραμματισμού με δυνατότητες διατήρησης των αντικειμένων**, ώστε να είναι διαθέσιμα μετά το τέλος της εφαρμογής.
- Αυτό επιτρέπει σε πολλές εφαρμογές να έχουν πρόσβαση στα ίδια αντικείμενα διευκολύνοντας την ανάπτυξη αντικειμενοστρεφών ΣΔΒΔ.

Αντικειμενοστρεφές Μοντέλο

- Ομάδα Τυποποίησης (standards group): ODMG = Object Data Management Group.
  - Στόχος: Οι παραγωγοίOO-DBMS να υλοποιούν μία ΟΟ γλώσσα , π.χ.C++, με επεκτάσεις(OQL) ώστε να μεταφέρονται δεδομένα μεταξύ ΒΔ και προγραμμάτων απευθείας.
- ODL = Object Definition Language
  - Αντιστοιχεί στο κομμάτι της SQL όπως οι εντολές CREATE TABLE κλπ..
- OQL = Object Query Language
  - Προσπαθεί να μιμηθεί το κομμάτι ερωτημάτων τηςSQL σε ΟΟ πλαίσιο.

Αντικειμενοστρεφές Μοντέλο




```
class Person
{
attribute struct person_name {string first_name, string last_name} name;
};


class Employee extends Person
(extent employee)
{
attribute string emp_no;
attribute enum sex_type {M, F} sex;
attribute enum pos_type {Manager, Supervisor, Assistant} position;
attribute date date_birth;
attribute float salary;

int getAge ();
void increaseSalary (in float salary_incr);
};

class Branch
(extent branches)
{
attribute string branch_name;
attribute struct branch_address {string street, string city, string postal_code} address;

relationship set<Employee> Has inverse Employee::WorksAt;
};
```

Γλώσσες Αντικειμένων

- Η **γλώσσα ερωτημάτων αντικειμένων** (OQL, object query language) είναι μία δηλωτική γλώσσα χειρισμού δεδομένων, η οποία μοιάζει πολύ στη σύνταξη της γλώσσας SQLτων σχεσιακών ΣΔΒΔ.
- Η OQL μπορεί να χρησιμοποιηθεί ως αυτόνομη γλώσσα ερωτημάτων ή να ενσωματωθεί σε κάποια άλλη γλώσσα προγραμματισμού (C++, Java).
- Βασικός στόχος της OQL είναι η απλή διατύπωση ερωτημάτων σε αντικειμενοστρεφή ΒΔ με τρόπο παρόμοιο με αυτόν της SQL.

Γλώσσες Αντικειμένων

- Η OQL μπορεί να χρησιμοποιηθεί για συσχετιστική προσπέλαση ή για πλοηγητική προσπέλαση:
  - η **συσχετιστική προσπέλαση** (associative access) επιστρέφει ένα σύνολο αντικειμένων της ΒΔ. Ο τρόπος εντοπισμού και επιστροφής των αντικειμένων ανήκει στην ευθύνη του αντικειμενοστρεφούς ΣΔΒΔ.
  - κατά την **πλοηγητική προσπέλαση** (navigational access) κάθε φορά πραγματοποιείται προσπέλαση σε ένα αντικείμενο και χρησιμοποιούνται οι δεσμοί που υπάρχουν μεταξύ των αντικειμένων για να μεταβούμε από ένα αντικείμενο σε άλλο.

Γλώσσες Αντικειμένων

- Έστω ότι ζητούμε τα ονόματα και τις οδούς για όλα τα υποκαταστήματα που βρίσκονται στη Θεσσαλονίκη. Η ομοιότητα της OQL με την SQL είναι σαφής.

```
SELECT b.name,b.address
FROM b IN branches
WHERE b.address.city='Θεσσαλονίκη';
```

Γλώσσες Αντικειμένων

- Ας εξετάσουμε ένα παράδειγμα με περισσότερες από μία κλάσεις αντικειμένων. Έστω ότι αναζητούμε τα επίθετα των υπαλλήλων που εργάζονται σε όλα τα υποκαταστήματα της Αλεξανδρούπολης. Το αντίστοιχο ερώτημα έχει ως εξής:

```
SELECT e.name.last-name
FROM b IN branches, e IN branches.WorksAt
WHERE b.address.city = 'Αλεξανδρούπολη';
```

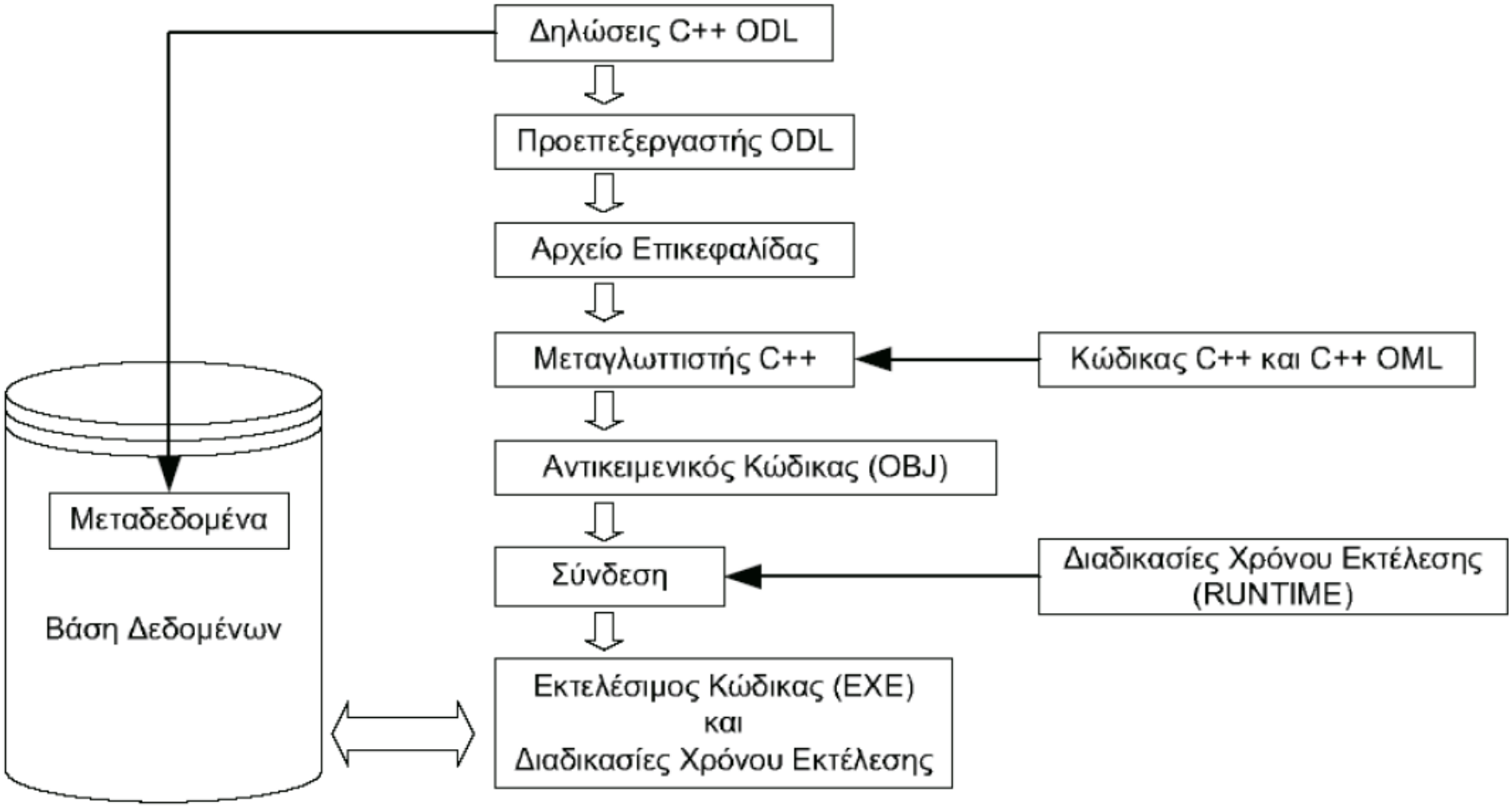
Δεσμεύσεις ODMG C++

- Οι **δεσμεύσεις** (bindings) ορίζουν την **αντιστοίχηση** των δομών της γλώσσας ορισμού αντικειμένων στις δομές μίας συγκεκριμένης γλώσσας προγραμματισμού.
- Οι δεσμεύσεις διευκολύνουν τον προγραμματιστή εφαρμογών διότι έτσι χρησιμοποιεί μία γλώσσα προγραμματισμού και όχι δύο διαφορετικές.
- Οι γλώσσες για τις οποίες παρέχονται δεσμεύσεις από την ODMG είναι οι Smalltalk, C++ και Java. Εδώ εστιάζουμε στις δεσμεύσεις για τη γλώσσα C++.

Δεσμεύσεις ODMG C++

- Παρέχεται μία βιβλιοθήκη κλάσεων C++ για την υλοποίηση των δομών της ODMG ODL.
- Η βιβλιοθήκη χρησιμοποιείται από το πρόγραμμα εφαρμογής για την προσπέλαση των αντικειμένων ανάλογα με τη λογική της εφαρμογής.
- Η χρήση της βιβλιοθήκης αποτρέπει τη σημαντική αλλαγή της δομής και του συντακτικού της γλώσσας C++ για την υποστήριξη της διατηρησιμότητας των αντικειμένων.
- Το πρόγραμμα εφαρμογής, το οποίο είναι γραμμένο σε C++, χρησιμοποιεί τη γλώσσα χειρισμού αντικειμένων (object manipulation language, OML), η οποία καθορίζει τον τρόπο προσπέλασης των αντικειμένων της ΒΔ.

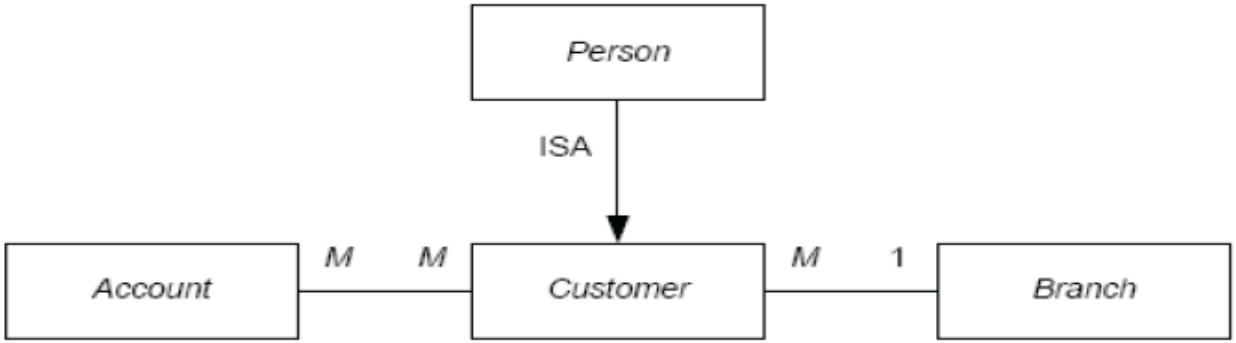
Δεσμεύσεις ODMG C++



Δεσμεύσεις ODMG C++

- Η βιβλιοθήκη παρέχει νέους τύπους δεδομένων για να επιτευχθεί η διατηρησιμότητα των αντικειμένων.
  - d_String ισοδυναμεί με τη String
  - d_Float ισοδυναμεί με τη Float
  - d_Date ισοδυναμεί με τη Date
  - d_Ref χρησιμοποιείται για να ορίσει συσχετίσεις μεταξύ αντικειμένων
- Οι νέοι τύποι έχουν το πρόθεμα d_
- Κάθε κλάση αντικειμένων πρέπει να κληρονομεί από την κλάση d_Object

Δεσμεύσεις ODMG C++



(α) συσχετισμοί μεταξύ κλάσεων

```
class Branch : public d_Object
{
   ...
   ...
};

class Person : public d_Object
{
   d_String name;
   d_String address;
};

class Account : public d_Object
{
   d_Float balance;
public :
   d_Set<d_Ref<Customer> > owners;

   float getBalance ();
   void setBalance (float new_balance);
};
```

```
class Customer : public Person
{
public :
   d_Long customer_id;
   d_Ref<Branch> home_branch;
   d_Set<d_Ref<Account>> accounts;
};
```

(β) ορισμός των κλάσεων σε ODMG C++ ODL

Δεσμεύσεις ODMG C++

- Την κλάση d_Database μπορούμε να τη χρησιμοποιήσουμε για:
  - Άνοιγμα ΒΔ (open(dbname))
  - Ονομασία αντικειμένου (set_object_name(object,name))
  - Αναζήτηση αντικειμένων (lookup_object(name))
  - Μετονομασία αντικειμένου
  - Κλείσιμο ΒΔ (close(dbname))

Δεσμεύσεις ODMG C++

- Παράδειγμα χρήσης δεσμεύσεων για εισαγωγή νέου branch.

```
1.  d_Database accountDB;
2.  accountDB->open("ACCOUNT DATABASE");
3.  d_Transaction MyTransaction;
4.  MyTransaction.begin();
5.  d_Ref<Branch>branch=new (accountDB,"Branch")Branch;
6.  branch->name="New Branch";
7.  branch->address.city="New City";
8.  branch->address.street="New Street";
9.  branch->address.postal_code="54321";
10. MyTransaction.end();
```

Πλεονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- **Ενισχυμένες δυνατότητες μοντελοποίησης**:
  1. το αντικειμενοστρεφές μοντέλο δεδομένων επιτρέπει την αναπαράσταση του πραγματικού κόσμου με μεγαλύτερη ακρίβεια από το σχεσιακό μοντέλο,
  2. η έννοια του αντικειμένου που περιλαμβάνει δεδομένα και μεθόδους ανταποκρίνεται περισσότερο στην πραγματικότητα,
  3. με τις ιεραρχίες εξειδίκευσης και συμπερίληψης μοντελοποιούνται σχετικά εύκολα ακόμη και πολύπλοκες οντότητες του πραγματικού κόσμου.

Πλεονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- **Επεκτασιμότητα**:
  - σε ένα αντικειμενοστρεφές ΣΔΒΔ μπορούμε να ορίσουμε νέες κλάσεις αντικειμένων χρησιμοποιώντας τις υπάρχουσες,
  - με τη χρήση της απλής και της πολλαπλής κληρονομικότητας και την εφαρμογή του πολυμορφισμού μπορούμε να παράγουμε νέες κλάσεις αντικειμένων που κληρονομούν χαρακτηριστικά και μεθόδους άλλων κλάσεων,
  - έτσι αποφεύγεται η επανάληψη του ορισμού χαρακτηριστικών και μεθόδων που έχουν ήδη ορισθεί σε άλλες κλάσεις.

Πλεονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- **Εξέλιξη σχήματος ΒΔ**:
  - οι δυνατότητες του αντικειμενοστρεφούς μοντέλου δεδομένων και οι ευκολίες που παρέχει το ΣΔΒΔ μπορούν να χρησιμοποιηθούν για την εξέλιξη του σχήματος (schema evolution) της ΒΔ ,
  - η εξέλιξη του σχήματος είναι αρκετά δύσκολη στα σχεσιακά συστήματα, διότι επιφέρει πολλές αλλαγές στη λογική των εφαρμογών.

Πλεονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- **Υποστήριξη συναλλαγών μεγάλης διάρκειας**:
  - ο τρόπος διαχείρισης κατά την εκτέλεση των ταυτόχρονων συναλλαγών από τα σχεσιακά συστήματα οδηγεί σε περιορισμένη απόδοση όταν οι συναλλαγές έχουν μεγάλη διάρκεια,
  - τα αντικειμενοστρεφή συστήματα υλοποιούν διαφορετικούς μηχανισμούς με αποτέλεσμα οι συναλλαγές μεγάλης διάρκειας να εκτελούνται πιο αποδοτικά.

Μειονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- **Έλλειψη καθορισμένου μοντέλου**:
  - παρά τα πλεονεκτήματα του αντικειμενοστρεφούς μοντέλου δεν υπάρχει κάποιο συγκεκριμένο πρότυπο που να ακολουθείται από όλους τους κατασκευαστές συστημάτων,
  - το μοντέλο που έχει προτείνει η ODMG έχει καθιερωθεί ως de facto πρότυπο.

Μειονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- **Πολυπλοκότητα**:
  - οι μηχανισμοί αποθήκευσης αντικειμένων, διαχείρισης συναλλαγών, εξέλιξης σχήματος της βάσης και επεξεργασίας ερωτημάτων είναι αρκετά πολύπλοκοι,
  - η πολυπλοκότητα αυτή οδηγεί στην υλοποίηση συστημάτων που είναι πιο απαιτητικά και δυσκολότερα στη διαχείρισή τους.

Μειονεκτήματα Αντικειμενοστρεφών ΣΔΒΔ

- Έλλειψη υποστήριξης όψεων:
  - πολλά από τα σύγχρονα αντικειμενοστρεφή συστήματα δεν υποστηρίζουν μηχανισμό όψεων παρά τα πολλά πλεονεκτήματα που προσφέρουν.

Σύνοψη

- Η αδυναμία του σχεσιακού μοντέλου για επακριβή αναπαράσταση του πραγματικού κόσμου και υποστήριξη σύγχρονων εφαρμογών οδήγησε στην υιοθέτηση άλλων μοντέλων.
- Το αντικειμενοστρεφές μοντέλο βασίζεται στις αντικειμενοστρεφείς γλώσσες προγραμματισμού και χρησιμοποιείται για τη μοντελοποίηση σύνθετων οντοτήτων του πραγματικού κόσμου.
- Τα αντικειμενοστρεφή ΣΔΒΔ στηρίζουν το αντικειμενοστρεφές μοντέλο δεδομένων και έχουν πολλές δυνατότητες που λείπουν από τα σχεσιακά ΣΔΒΔ.
- Ωστόσο, παρουσιάζουν και σοβαρά μειονεκτήματα που δυσκολεύουν την ευρεία χρήση τους: απουσία καθορισμένου μοντέλου, αυξημένη πολυπλοκότητα, απουσία υποστήριξης όψεων.