

Háttéralkalmazások

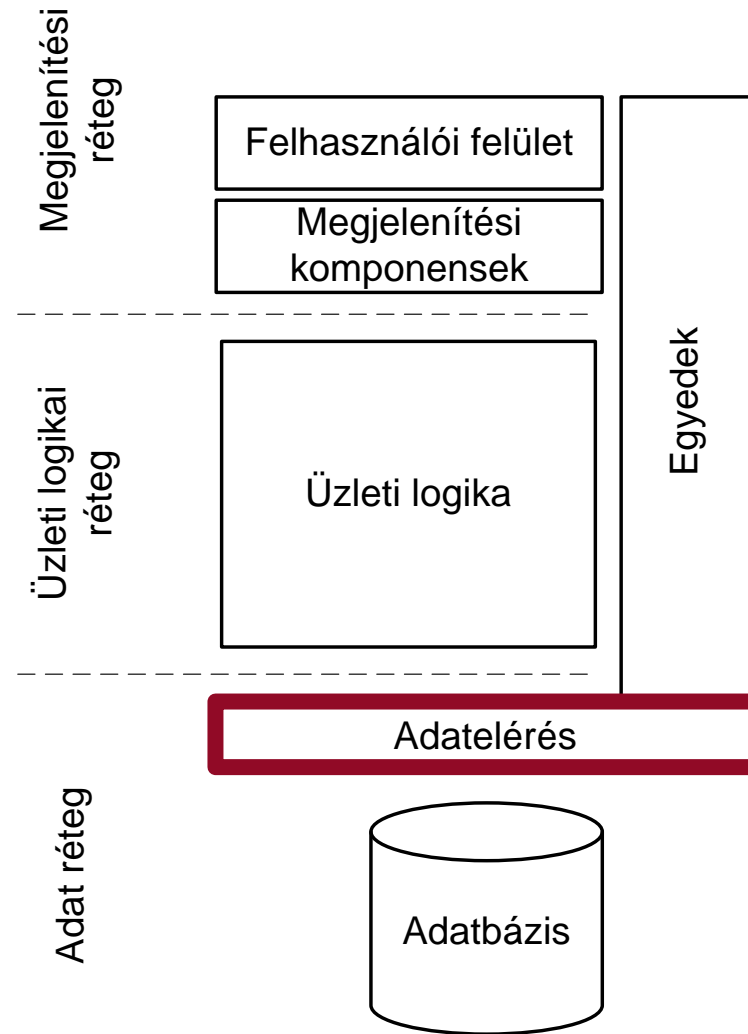
Adatelérési osztálykönyvtárak
Objektum-relációs leképzés elméleti
alapjai



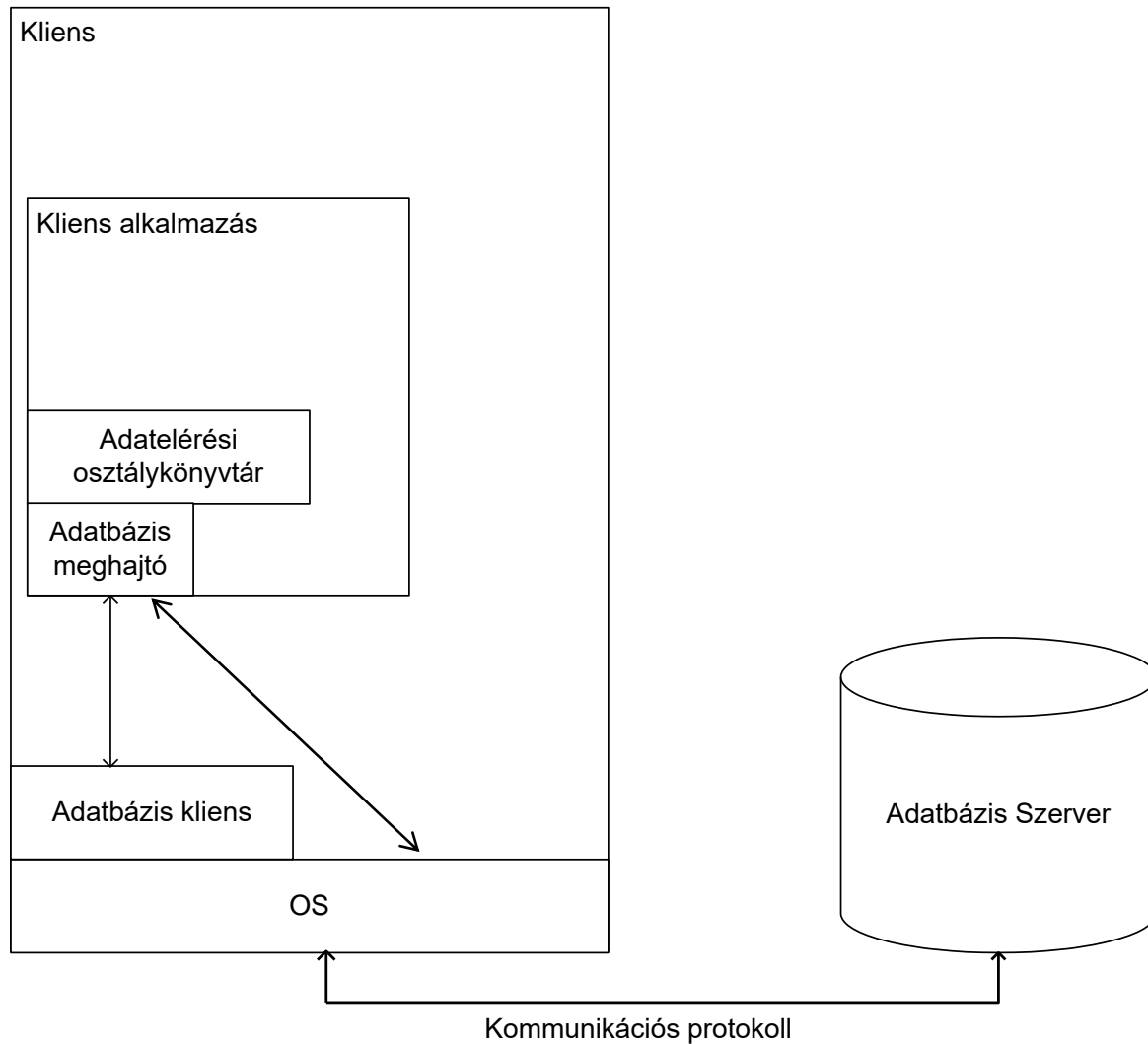
Automatizálási és
Alkalmazott
Informatikai Tanszék

Adatelérési osztálykönyvtárak

Adatelérési osztálykönyvtárak



Adatelérési osztálykönyvtárak működése



Adatelérési könyvtár feladata

- Adatbázis használat absztrakció
 - > Relációs adatbázisok elérése
 - > Egységes, adatbázis független kódolás
- Tipikus elmei
 - > Connection
 - > Command + Paraméterek
 - > ResultSet
 - > Exception

Adatelérési könyvtár használata

- Adatelérési rétegben az **adatelérési osztálykönyvtárak** feladata
 - > Adatelérési absztrakció biztosítása
 - > Elemi adatelérés biztosítása
 - Listázd az összes terméket!
 - Van még raktáron a 23-as termékből?
 - Rögzítsd egy új vevő adatait!
- Technológia specifikus kód
 - > Elrejtje az adatbázist, a kommunikációt, a különbségeket
 - > Platform/technológia specifikus a kód a rétegben

ADO.NET - kapcsolat

- *****Connection**
 - > Adatbázis szerver függően más-más osztály, pl. **SqlConnection**
 - > Műveletek: Open, Close, BeginTransaction
- **Connection string**
 - > Kiszolgáló függő szintaktika
 - > User ID=LOGIN;Password=PASSWORD;Persist Security Info=false;Initial Catalog=AdventureWorks;Data Source=DATASOURCE;Packet Size=4096
- **Connection pooling**
 - > Cache-elt kapcsolatok, újra felhasználhatóak
 - > Connection leak

Kapcsolat felépítése - példa

```
var builder = new
SqlConnectionStringBuilder();
builder.UserID = "User";
builder.Password = "Pw";
builder.DataSource = "database.server.hu";
builder.InitialCatalog = "hatteralk";

using (var con = new
    SqlConnection(builder.ConnectionString))
{
    con.Open();
    ...
    con.Close();
}
```


Adatbázis parancs használata

- *****Command**
 - > 3 különböző típus (CommandType)
 - Tárolt eljárás
 - Tábla teljes tartalma
 - SQL query
 - > A parancs szövege (CommandText)
 - > Az adatbázis kapcsolat (Connection)
 - > A tranzakció (Transaction)
 - > Timeout (CommandTimeout)
 - Alapértelmezett 30 sec
 - > Paraméterek
- **Végrehajtás**
 - > ExecuteScalar / ExecuteReader

```
using(var conn = ...)
{
    conn.Open();
    var cmd = new
        SqlCommand();
    cmd.CommandText = "...";
    cmd.Connection = conn;
    return cmd.ExecuteScalar();
}
```

JDBC - kapcsolat

- `java.sql.Connection`
 - > *Egységes* interfész minden relációsadatbázis-kezelőhöz
 - > A gyártóspecifikus meghajtók a `DriverManager`-nél regisztrálják magukat, az alkalmazások innen érhetik el
- `Connection URL`
 - > Kiszolgáló függő szintaktika, de mindig URL formátumú
 - > **`jdbc:sqlserver://<server>:<port>;databaseName=<dbname>;user=<user>;password=<password>`**
- Fontosabb műveletek:
 - > `open` nincs (a `DriverManager`-től csak nyitott kapcsolatot lehet kérni)
 - > `close`: meg kell hívni, ha már nem használjuk (finally blokkban!)
 - > `setAutoCommit(false)`: tranzakció indítása (nincs `begin`)
 - > `commit`, `rollback`
 - > `prepareStatement`: tetszőleges paraméterezett SQL utasítás létrehozására
- `Connection pooling`
 - > `javax.sql.DataSource` interfészen keresztül (lásd a JPA előadáson)

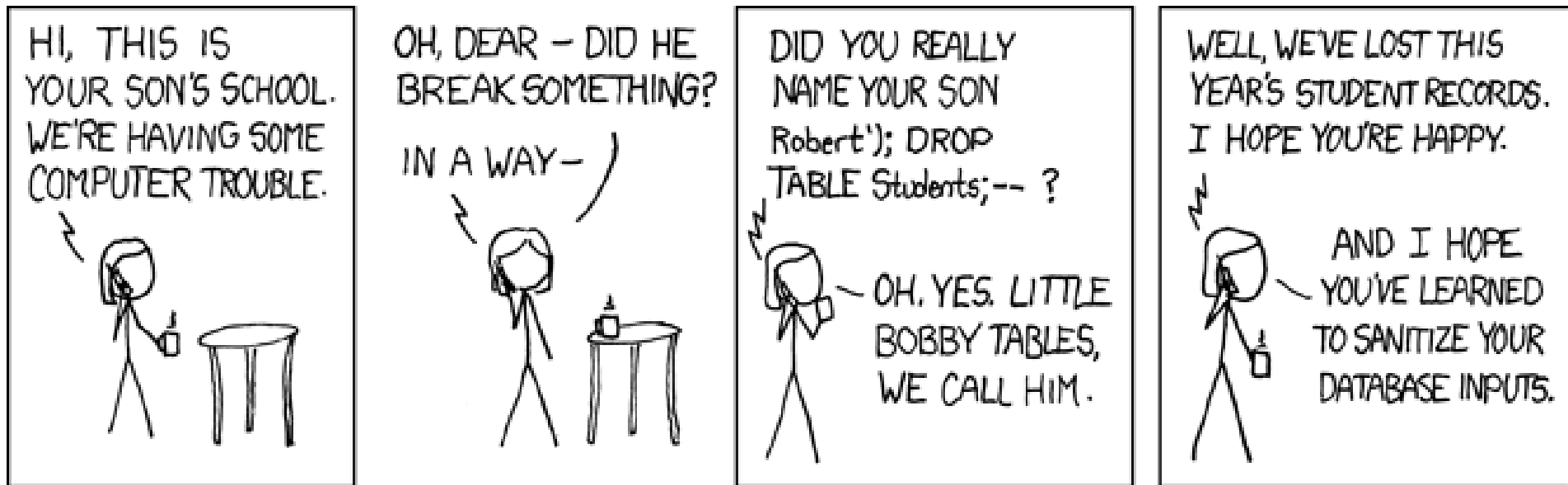
JDBC – kapcsolat felépítés példa

```
/*  
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver ");  
Csak JDBC 4-nél korábbi meghajtók előtt szükséges, akkor a driver osztályát így  
kellett betölteni, hogy regisztrálni tudja magát a DriverManagernél.  
A JDBC 4-es drivereket viszont enélkül is megtalálja a DriverManager.  
*/  
try (Connection conn =  
    DriverManager.getConnection("jdbc:sqlserver://localhost;databaseName=...");)  
{  
    //conn használata  
} catch (SQLException e) {  
    e.printStackTrace();  
} // automatikus finally, conn.close() hívással a try(...) miatt
```

JDBC – lekérdezés végrehajtása

```
String sql = "SELECT id, name FROM users WHERE id = ?";  
List<User> users = new ArrayList<>();  
try (PreparedStatement ps = conn.prepareStatement(sql);) {  
    ps.setInt(1, 1234);  
    ResultSet rs = ps.executeQuery();  
    while (rs.next()) {  
        users.add(new User(rs.getInt("id"), rs.getString("name")));  
    }  
}
```

SQL Injection - 1



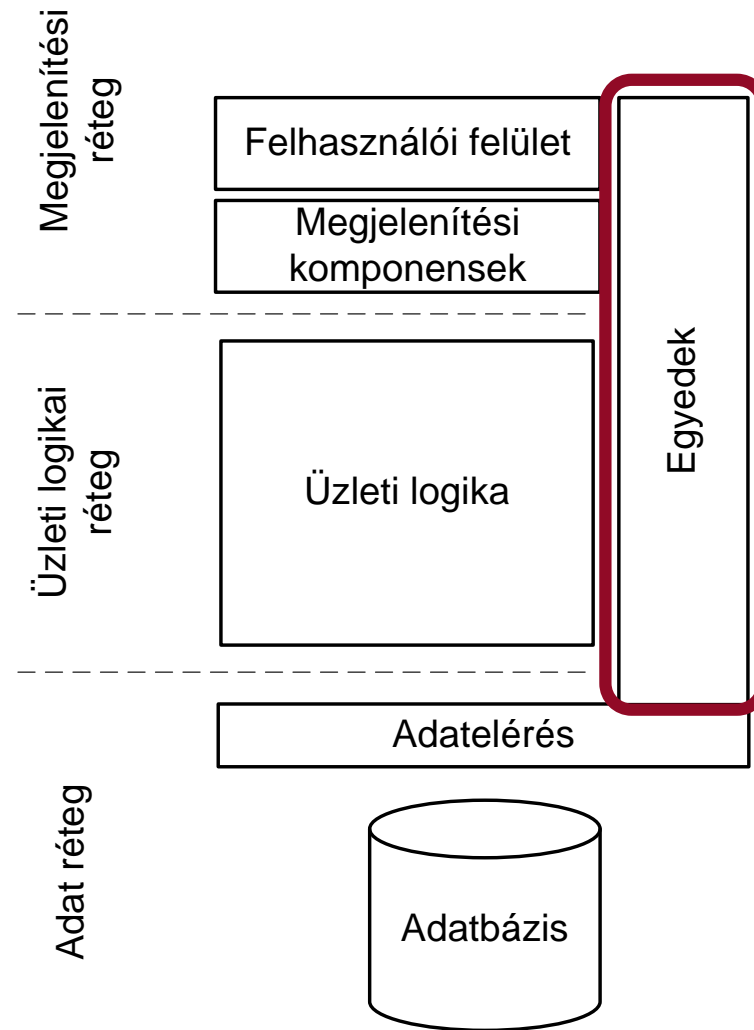
SQL Injection – 2

- Biztonsági probléma
 - > Input adatok ellenőrizetlen felhasználása
- SQL utasítás string jellegű összefűzése
 - > “Select * from product where name =” + Name.Text;
 - > Name.Text-ben bármi szerepelhet
 - 1; drop table product;--
- Súlyos hiba!
 - > Paraméterek használata

Objektum-relációs leképzés

Probléma felvetés

Háromrétegű architektúra



Termék
Vevő
Megrendelés

...

Modellezés

- Üzleti logika
 - > OO modellezés
 - > UML
 - > Tervezési minták (Design Patterns)
 - > Nem csak statikus tagok, folyamatok is
- Adatréteg
 - > E/K diagramok
 - > UML data modelling profile
 - > Statikus, attribútum szemlélet

O/R leképzés feladata

- Object Relational Mapping
 - > Üzleti objektumok leképezése relációs adatmodellre
 - > Adattárolás és üzleti folyamatok összekötése
- Problémák
 - > Eltérő koncepciók
 - > Öröklődés
 - > Shadow információk
 - > Kapcsolatok leképezése

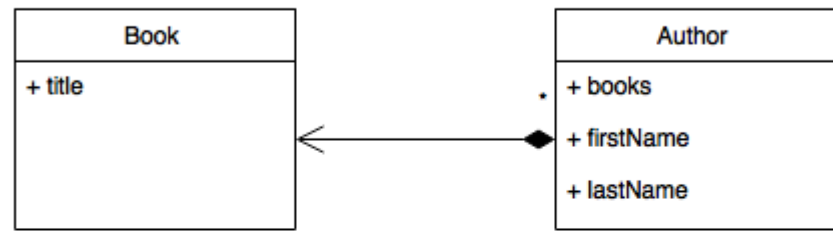
Objektum-relációs leképzés

Alap koncepció

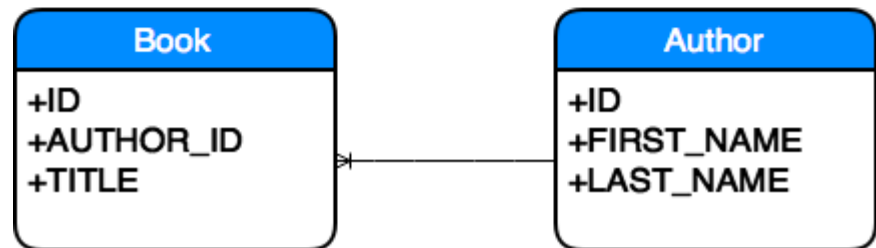
Megközelítés

- Alap ötlet
 - > Osztály → Tábla
 - > Adattag → Oszlop
 - > Kapcsolat → FK

Objektum-orientált

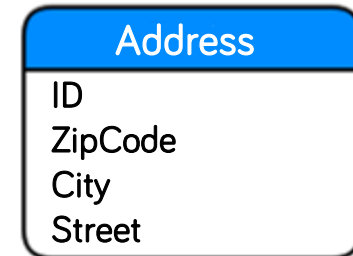
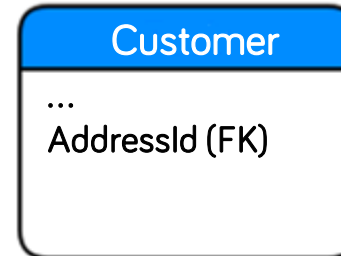
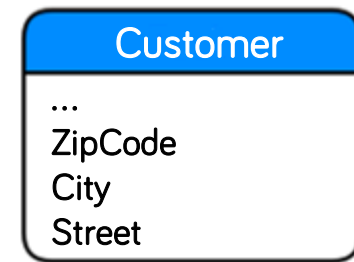


Relációs séma

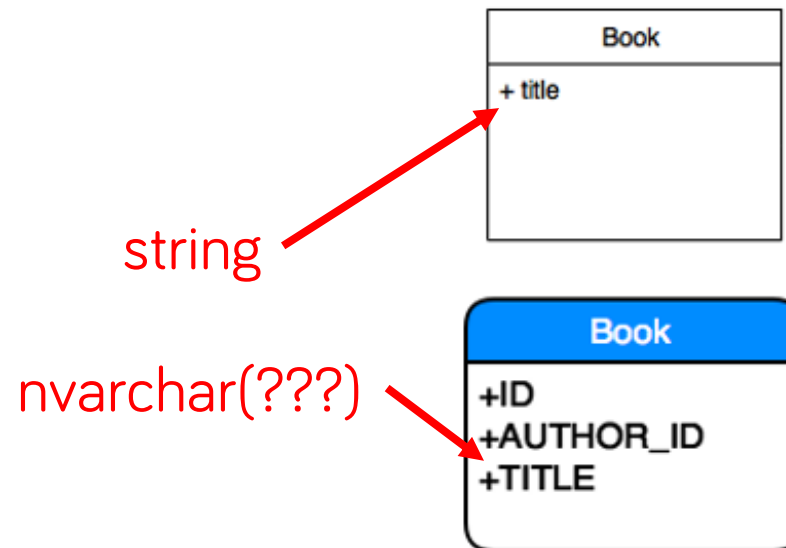


Problémák

- Összetett mezők
 - > Vásárló
 - Cím (Irányítószám, Város, Utca)

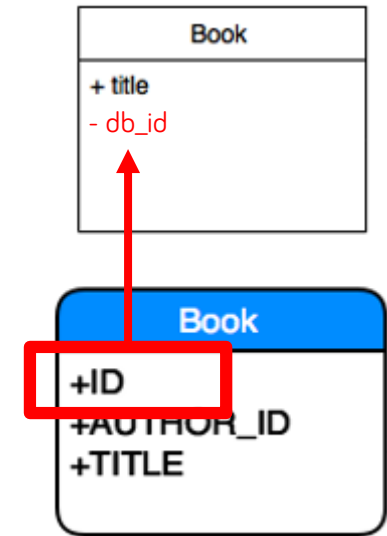


- Eltérő adattípusok
 - > Konverzió!



Shadow információk

- Szükségesek a perzisztencia megvalósításához
 - > Kulcsok
 - > Időbélyegek (optimista konkurencia kezelés)
- Nem szükséges az üzleti objektumba helyezni, de kezelni kell valahogy



Objektum-relációs leképzés

Öröklés

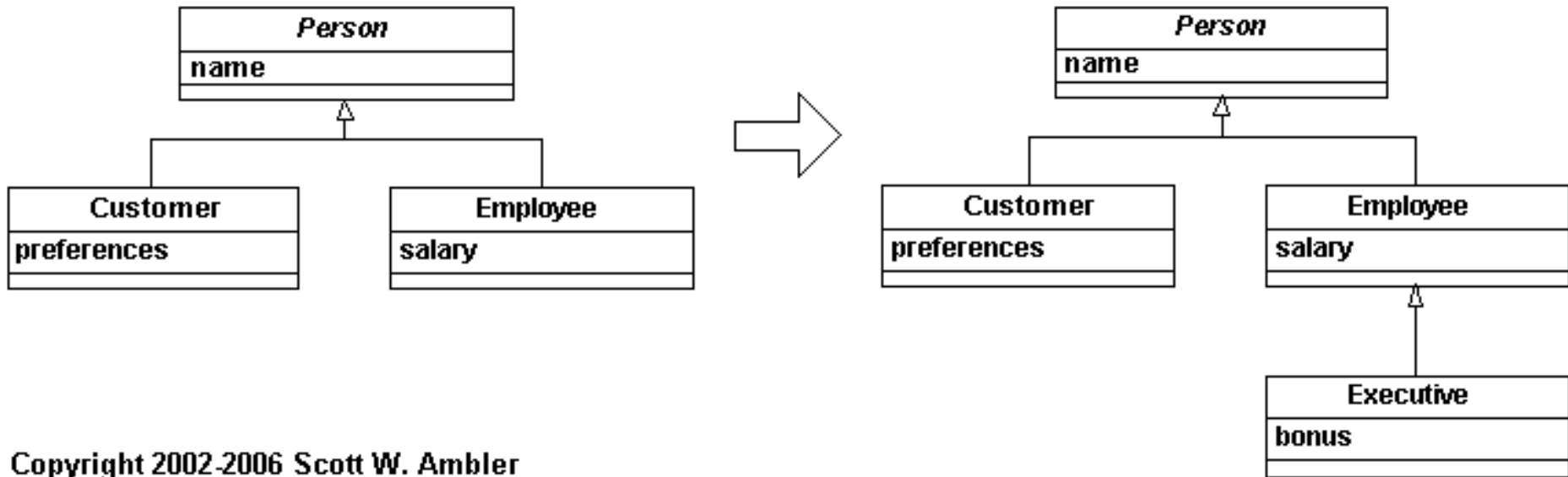
Irodalom

- Ábrák, példák:

> <http://www.agiledata.org/essays/mappingObjects.html>

Öröklés – Mintapélda

- Absztrakt *Person* osztály
- Több implementáció
- Új funkciót építünk az alkalmazásba -> új leszármazott: *Executive*



Copyright 2002-2006 Scott W. Ambler

Öröklődés modellezése

- Hierarchia leképzése egy közös táblába
- Minden valós osztály leképzése saját táblába
 - > Akikből objektum példányok képződhetnek
- Minden osztály leképzése saját táblába
 - > Absztrakt osztályok is
- Osztályok és hierarchia szintek általános leképzése

Egy táblába történő leképezés – 1

- Összes attribútum felsorolása a hierarchiát bejárva
- Típus azonosítás
 - > Egy oszlopban kódolt értékkel
 - > IsCustomer, IsEmployee,... oszlopokkal, vagy egyetlen pl. Type oszloppal
- Bővítés kezelése
 - > Új attribútumok felvitele

Egy táblába történő leképezés – 2

Person tábla

ID	Name	Prefer.	Salary	IsCustomer	IsEmployee
1	X Y	NULL	1234	0	1
2	W Z	xxxxx	NULL	1	0



Executive bevezetése

ID	Name	Prefer.	Salary	IsCustomer	IsEmployee	Bonus	IsExecutive
1	X Y	NULL	1234	0	1	NULL	0
2	W Z	xxxxx	NULL	1	0	NULL	0
3	Q Q	NULL	456	0	1	999	1

Egy táblába történő leképezés – 3

- Előnyök
 - > Egyszerű
 - > Könnyű új osztályt bevenni a hierarchiába
 - > Objektum példány szerepének változása könnyen követhető
 - Employee → Executive
 - Employee és Customer egyszerre
- Hátrányok
 - > Helypazarlás
 - > Egy osztály változása miatt az összes tárolása megváltozik
 - > Komplex struktúra esetén nehezen áttekinthető
 - > NOT NULL constraintek nem használhatók
- Célszerű használni
 - > Egyszerű hierarchiák esetén

Valós osztályok leképzése táblába – 1

- Osztályonként egy tábla
- Osztály összes attribútumának eltárolása
- Példányazonosító
- Változás követése
 - > Új osztály → új tábla
 - > Attribútum változás → Hierarchia szint mentén végig kell vinni

Valós osztályok leképzése táblába - 2

Customer tábla

ID	Name	Prefer.
2	W Z	xxxxx

Employee tábla

ID	Name	Salary
1	X Y	1234



Executive bevezetése

Executive tábla

ID	Name	Salary	Bonus
4	QQ	456	999

Valós osztályok leképzése táblába – 3

- Előnyök
 - > Átláthatóbb
 - > Jobban illeszkedik az objektum modellhez
 - > Gyors adatelérés
- Hátrányok
 - > Osztály módosítása → Több táblát is érinthet
 - > Több szerepet is betöltő példányok kezelése
 - Employee → Executive
 - Employee és Customer
- Célszerű használni
 - > Ritkán változó struktúrák esetén

Összes osztály leképezése táblába – 1

- Osztály hierarchiát követik a táblák
- Szülő – gyerek viszony leképezése idegen kulccsal
- Példány azonosító

Összes osztály leképezése táblába – 2

Person tábla

ID	Name
1	XY
2	WZ

Customer tábla

CID	PID (FK)	Prefer.
11	2	xxxxx

Employee tábla

EID	PID (FK)	Salary
21	1	1234



Executive bevezetése

Executive tábla

XID	EID (FK)	Bonus
41	22	999

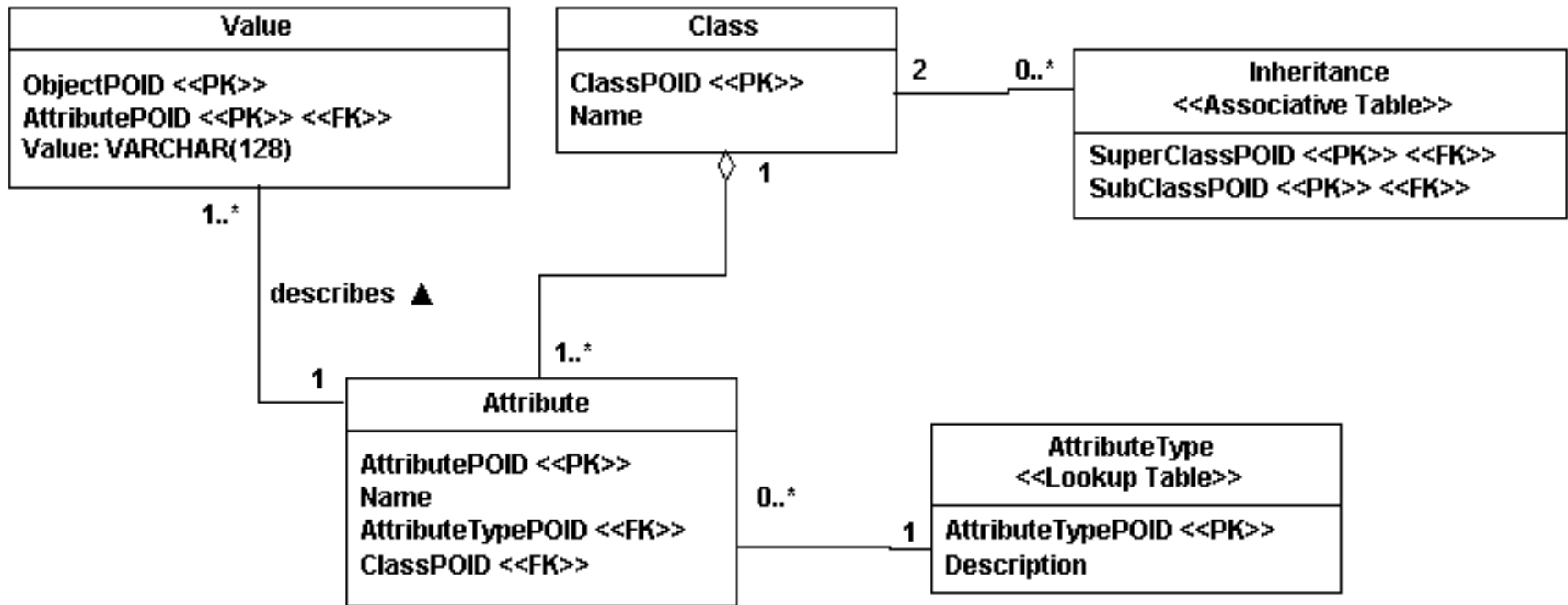
Összes osztály leképezése táblába – 3

- Előnyök
 - > Könnyű megérteni (egy az egyben leképezés)
 - > Könnyű módosítani a szülő osztályok struktúráját
- Hátrányok
 - > Összetett adatbázis séma
 - > Egy példány adatai több táblában vannak
 - Összetett lekérdezés
 - Szükséges join → lassabb
- Célszerű használni
 - > Komplex hierarchia esetén
 - > Változó struktúra esetén

Leképezés általános struktúrába – 1

- Meta data driven megoldás
- Általános séma
 - > Tetszőleges hierarchia leírható
 - > Független a konkrét osztályoktól
 - Osztály hierarchia → Meta adat
 - Osztály példányok → Attribútumok manifesztálódása

Leképezés általános struktúrába - 2



Copyright 2002-2006 Scott W. Ambler

Leképezés általános struktúrába - 3

Class tábla

CID	Name
1	Person
2	Employee
3	Executive

Attribute tábla

AID	Name	CID (FK)	TID (FK)
11	Name	1	21
12	Salary	2	22
13	Bonus	3	23

Value tábla

OID	AID (FK)	Value
101	11	X Y
101	12	456
101	13	999

Inheritance tábla

Super	Sub
1	2
2	3

AttributeType tábla

ATID	Desc.
21	First name and last name.
22	Salary of every employee
23	Bonus of executive employees.

Leképezés általános struktúrába – 4

- Előnyök
 - > Flexibilis
 - > „Bármilyen” leírható benne
- Hátrányok
 - > Elsőre nehéz „megemészteni”
 - > Nehéz „összeszedni” egy objektum példány adatait
 - > Nagy adatmennyiség esetén nem hatékony
- Célszerű használni
 - > Komplex alkalmazások
 - > Kis mennyiségű adatok
 - > „Minden változhat” akár futási időben is

Objektum-relációs leképzés

Kapcsolatok

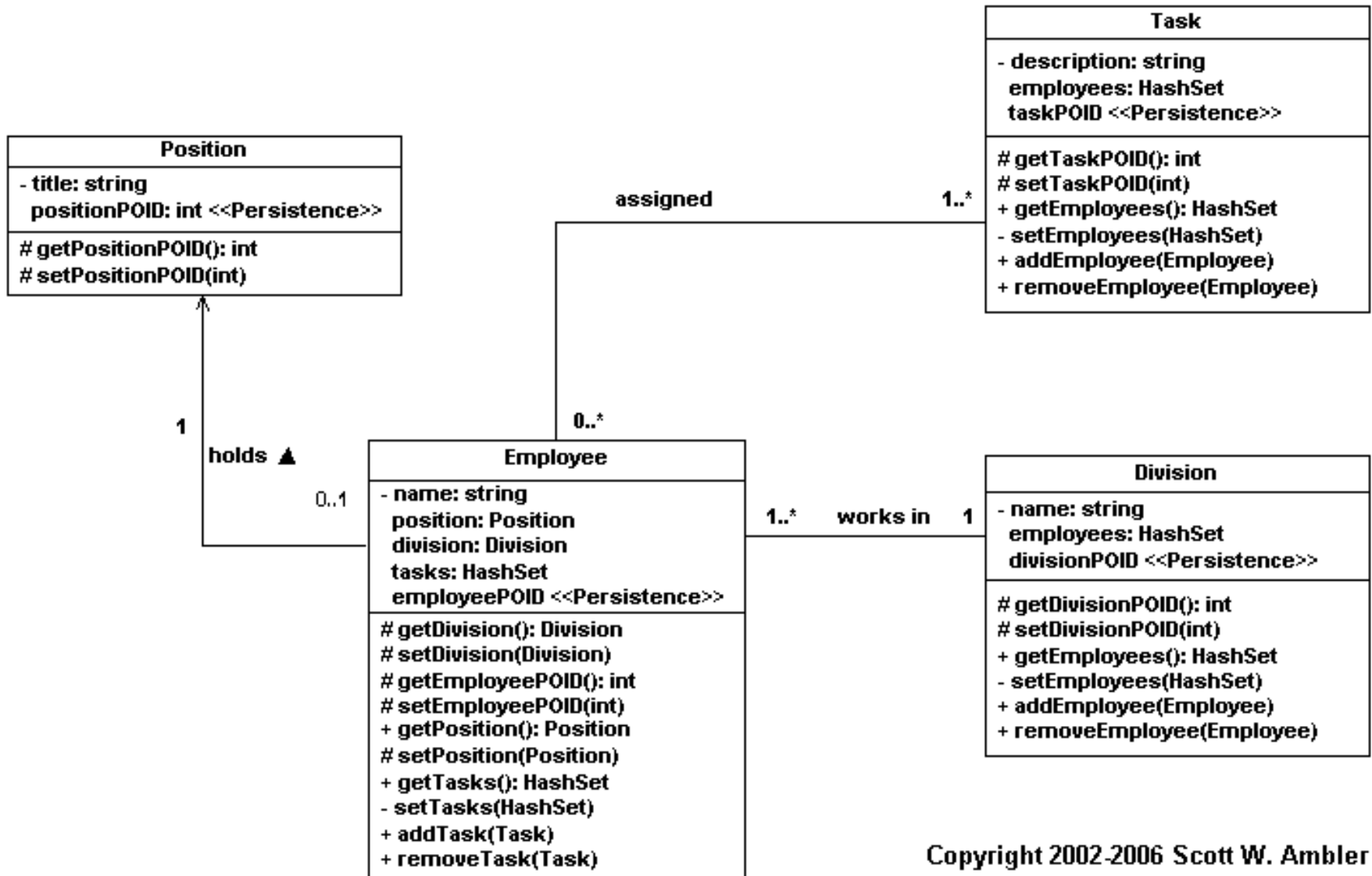
Objektum kapcsolatok leképzése – 1

- Kapcsolatok
 - > Asszociáció
 - > Aggregáció
 - > Kompozíció
 - Típusok
 - > Egy – Egy
 - > Egy – Több
 - > Több – Több
 - Irány
 - > Egy irányú
 - > Két irányú
- Referenciális integritás
- Nem képezhető le

Objektum kapcsolatok leképzése – 2

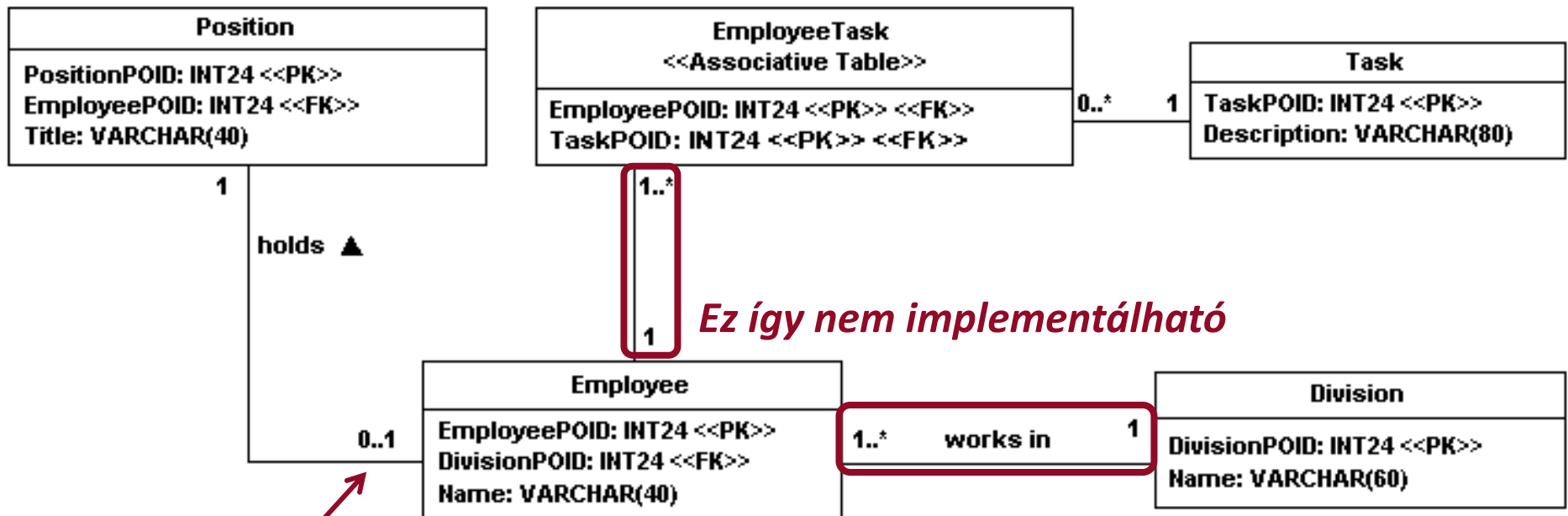
- Egy – Egy kapcsolat
 - > Külső kulcs az egyik táblába
 - Egy – több lehetőségét magában hordja
- Egy – Több kapcsolat
 - > Külső kulcs az „egy”-re
- Több – Több kapcsolat
 - > Közvetlenül nem képezhető le
 - > Kapcsoló tábla használata
- Kardinalitások
 - > Mindkét oldal kötelezőt nem célszerű leképezni
 - > Elindulási probléma adatbázis szinten
 - > Számolás adatréteg szinten: 0, 1, több

Objektum kapcsolatok leképzése – 3



Copyright 2002-2006 Scott W. Ambler

Objektum kapcsolatok leképzése – 4



Ez így nem implementálható

Egy – több kapcsolat is lehetne az adatréteg alapján

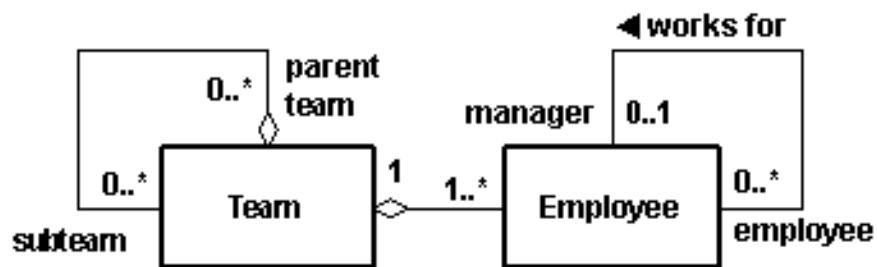
Copyright 2002-2006 Scott W. Ambler

Rekurzió – 1

- Más néven reflexió
- Kapcsolat kezdő és végpontja ugyan az az entitás
- Hasonlóan a többi kapcsolathoz
 - > Több – több leképezésben kis eltérés

Rekurzió - 2

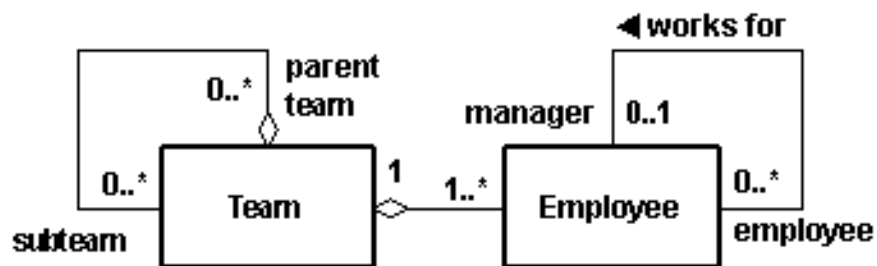
<<Class Model>>



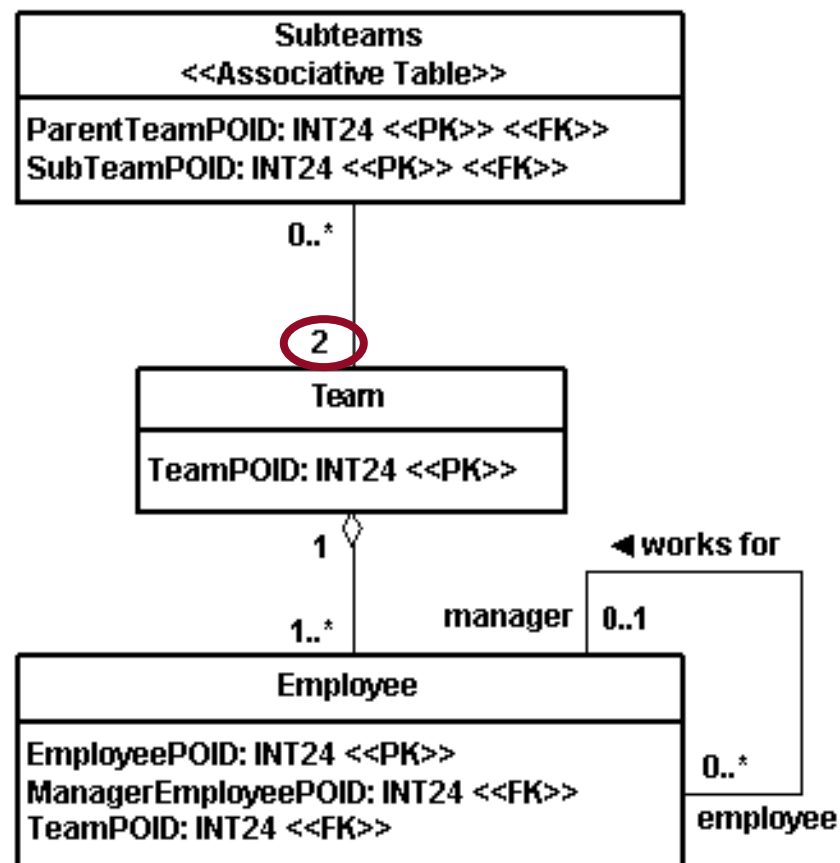
Copyright 2002-2006 Scott W. Ambler

Rekurzió - 2

<<Class Model>>



<<Physical Data Model>>







Copyright 2002-2006 Scott W. Ambler

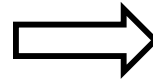
Objektum-relációs leképzés

Egyéb problémák

Rendezett gyűjtemények

Jegyzetek, előadások, gyakorlatok anyaga

-  Jegyzetek, gyakorlatok anyaga és házi feladatok
-  Bevezetés és a háromrétegű architektúra Modified 7/09/20, 13:05
-  Tranzakciók és az MSSQL platform Uploaded 3/09/20, 08:14
-  Microsoft SQL Server programozása Uploaded 3/09/20, 08:14



Files
CourseId
FileName
SequenceId
...

- Sorrendezést adó attribútum + sorrend helyesen olvassuk fel (order by ...)
- Sorrend változtatás hatása
 - > Több rekordot is módosítani kell
 - > Avagy: hézagos kiosztás
- Elem törlése
 - > Nem fontos újra „indexelni”
 - Ekkor csak sorrendet jelent pozíció indexet nem

Osztály szintű tulajdonságok – 1

- Osztályra jellemzők
- Nem kötődnek példányhoz
- Példa
 - > Következő számla sorszám
 - > Kedvezmény értéke bizonyos összeghatár felett
- Osztály szintű konstansok kezelése is hasonló

Osztály szintű tulajdonságok – 2

- Minden tulajdonságnak külön tábla
 - > Gyors
 - > Sok kicsi tábla → áttekinthetetlen adatmodell

InvoiceGlobals tábla

NextId

123

Sale tábla

Sale

2%

Osztály szintű tulajdonságok – 3

- Minden tulajdonság ugyanabban a táblában különböző oszlopokban
 - > Gyors
 - > Egyszerű (egy tábla az összesnek)
 - > Konkurencia (mivel sor alapú és nem oszlop)

„Globals” tábla

NextInvoiceId	Sale
123	2%

Osztály szintű tulajdonságok – 4

- Osztályonként egy tábla az értékek különböző oszlopokban
 - > Gyors
 - > Sok kicsi tábla → áttekinthetetlen adatmodell

InvoiceGlobals tábla

NextId	PaymentDue
123	30

Osztály szintű tulajdonságok – 5

- Általános megoldás
 - > Minden tulajdonság új rekord
 - Osztály
 - Tulajdonság név
 - Érték
 - > Adatkonverziót meg kell oldani
 - > Egyszerű bővíthetőség
 - Új tulajdonság → Új rekord

Class	Key	Value
Invoice	NextId	123
Invoice	Sale	2