evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

INTEGRACE DAT

2011/12

Zdroj: H. Garcia-Molina, J.D.Ullman, J.Widom:
Database systems the Complete Book, 2009, chapter 21

Integrace informací

- Integrace informací je proces, při kterém vezmeme několik databází nebo jiných zdrojů informací a umožníme, aby se k datům přistupovalo jako kdyby byla uložena v jedné databázi.
- Zdroje mohou být tradiční databáze, ale může to také být například kolekce informací na webu.
- Integrovaná databáze může být
  - fyzická (datový sklad),
  - virtuální (mediátor - prostředník, nad kterým lze realizovat dotazy aniž by fyzicky existoval).

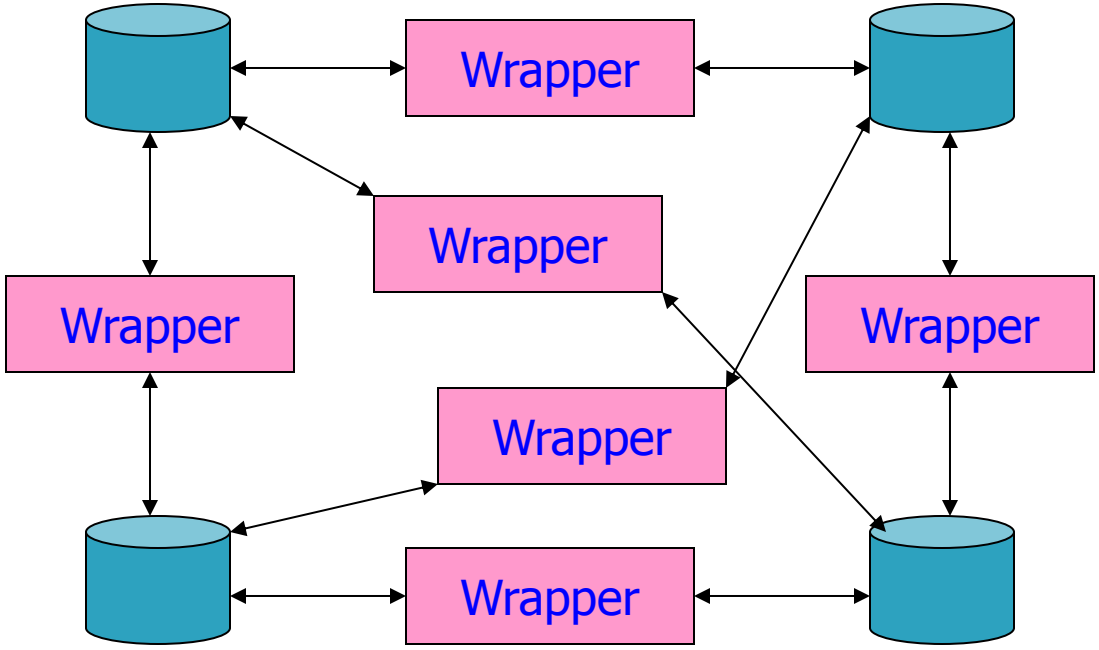
Problémy

- Již existující databáze
  - nelze měnit jejich strukturu
- Nekompatibilita různého druhu i mezi zdánlivě velmi podobnými databázemi:
  - lexikální – sloupec stejného významu může být v různých databázích různě pojmenován,
  - v interpretaci dat – například číselný údaj udávající teplotu může vzbudit pochybnosti, zda je to v 0C nebo 0 F
  - sémantická – je význam např. tabulky Zaměstnanci stejný? Někdo může být zaměstnán v HPP, jiný na DPČ nebo na DPP – je to řešeno v různých databázích stejně?

Přístupy k integraci informací

- Federativní databáze:
  - zdroje jsou nezávislé, ale každý může realizovat dotazy nad ostatními zdroji
- Datové sklady:
  - Kopie dat (většinou vhodně upravené) z různých zdrojů jsou uloženy v jedné databázi a periodicky aktualizované (často v noci).
- Mediátor – podporuje pohledy které integrují několik zdrojů velmi podobným způsobem jako materializované pohledy. Protože ale mediátor nemá žádná vlastní data, musí relevantní data získat z původních zdrojů.

Federativní databáze



Wrapper (adaptér) transformuje příchozí dotazy a odchozí odpovědi.

Federace - charakteristiky

- Nejjednodušší architektura
- Je potřebné implementovat propojení mezi každými dvěma databázemi, pokud si mají vzájemně poskytovat data.
- Toto propojení umožňuje, aby jeden DBS vznášel dotazy na data druhého DBS.
- Pokud každá z n databází v systému potřebuje přistupovat k datům ostatních databází, tak potřebujeme implementovat n(n-1) částí kódu na podporu dotazů.

Federace - příklad

- Automobilka má mnoho dealerů a každý z nich si udržuje svoji vlastní relační databázi automobilů, které má na skladě. Bylo rozhodnuto, že databáze dealerů se budou integrovat, aby každý dealer měl možnost v případě potřeby zjistit, zda požadovaný model nemá na skladě některý z dealerů v blízkosti.
- Struktura jednotlivých databází je ale rozdílná:
  - Dealer1 používá pouze jeden soubor VOZIDLA(Cislo, Model, Barva, AutPrevod) kde AutPrevod jenom indikuje zda ano či ne.
  - Dealer2 používá dva soubory AUTA(SerioveC, Model, Barva), VYBAVA(SerioveC, Popis)

Příklad – pokračování

- Aby Dealer1 mohl přistupovat k datům Dealera2, potřebuje adaptovat dotazy na strukturu Dealera2.
- Předpokládejme, že Dealer1 potřebuje získat informace o autech specifikovaných v tabulce

  PozadovanaVozidla(Model, Barva, AutPrevod)

V principu bude kód v T-SQL realizující dotaz na Dealera2 následující:

Pro každý řádek hodnot @Model, @Barva, @AutPrevod v tabulce PozadovanaVozidla budeme potřebovat dotaz

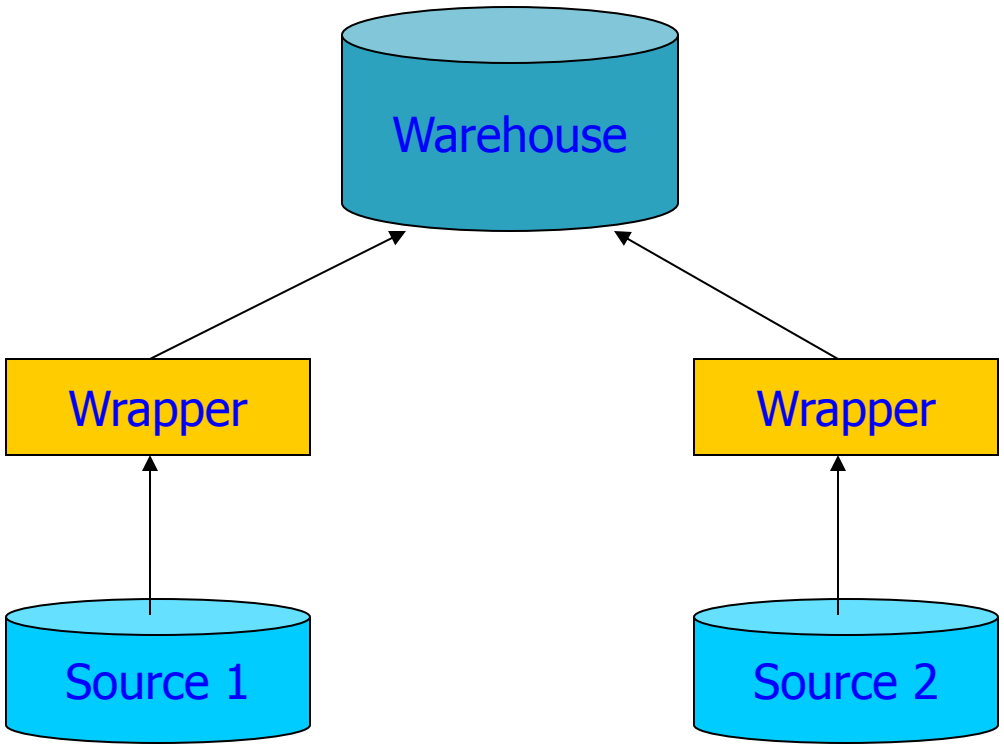
Příklad - pokračování


```
if @AutPrevod = 1
Begin
USE Dealer2
SELECT Auta.SerioveC
FROM Auta JOIN Vybava ON
   Auta.SerioveC=Vybava.SerioveC
WHERE Vybava.Popis='AutPrevod' AND
   Auta.Model=@model AND Auta.Barva=@barva
END
ELSE
```

Příklad - pokračování


```
BEGIN
   USE Dealer2
   SELECT Auta.SerioveC
   FROM Auta
   WHERE Auta.Model = @model
   AND Auta.Barva = @barva
   AND not EXISTS
   ( SELECT *
   FROM VYBAVA
   WHERE SerioveC = Auta.SerioveC AND popis=
    'AutPrevod')
END
```

Datový sklad



Datový sklad

- V datovém skladu se při integraci dat z různých zdrojů vytváří globální schéma. Data se poté uloží v datovém skladu, který vypadá stejně jako standardní databáze. Jsou-li již data ve skladu, tak dotazy se realizují obvyklým způsobem.
- Aktualizace datového skladu se dělá obvykle jedním ze dvou způsobů:
  - Celý datový sklad se vytvoří znovu z aktuálních dat; v té době je sklad pro dotazy nepřístupný.
  - Datový sklad se periodicky upravuje na základě změn v původních databázích, které nastaly od poslední aktualizace skladu. Tento přístup pracuje s menším objemem dat, je ale výrazně složitější.

Příklad – vytvoření datového skladu

Uvažujme opět předchozí příklad, kde

- Dealer1
  - VOZIDLA(Cislo, Model, Barva, AutPrevod)
- Dealer2
  - AUTA(SerioveC, Model, Barva), VYBAVA(SerioveC, Popis)
- Máme vytvořit datový sklad s následujícím globálním schématem:
  - AUTA_SKLAD(SerioveC, Model, Barva, AutPrevod, Dealer)

Příklad – vložení dat do datového skladu

Kód, který vloží data do datového skladu může vypadat následovně:

- *Vložení dat z databáze Dealera1:*

```
INSERT INTO AUTA_SKLAD(SerioveC, Model, Barva,
AutPrevod, Dealer)
SELECT Cislo, Model, AutPrevod, 'Dealer 1'
FROM Vozidla
```

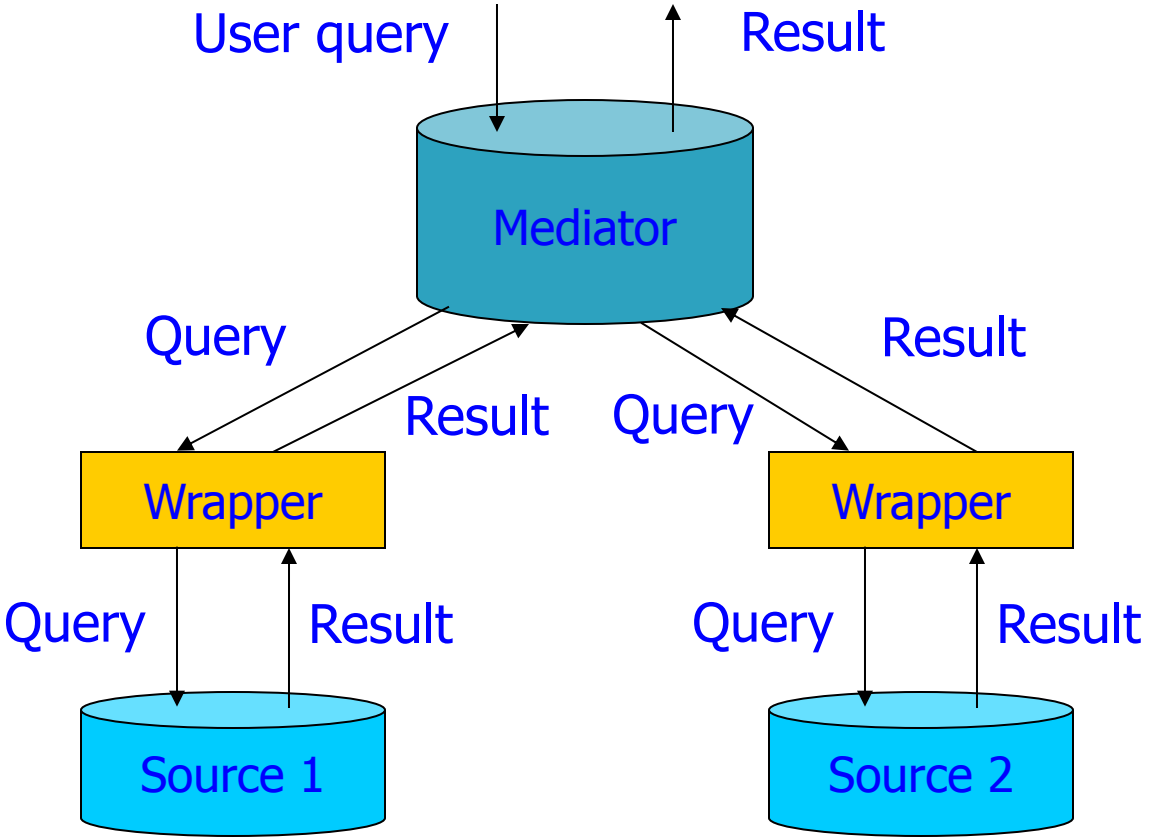
Data Dealera1 lze v tomto případě rovnou přidat do datového skladu; často se ale požaduje před přidáním dat do skladu spojení relací nebo výpočet různých agregačních funkcí.

Příklad – vložení dat do datového skladu

Vložení dat z databáze Dealera2:

```
INSERT INTO Auta_DWH (SerioveC, Model, Barva, AutPrevod, Dealer)
SELECT AUTA.SerioveC, Model, Barva, 1  AutPrevod, 'Dealer2'  Dealer
FROM AUTA WHERE EXISTS
(SELECT * FROM Vybava WHERE Vybava.SerioveC = Auta.SerioveC
AND Popis='AutPrevod')
UNION
SELECT Auta.SerioveC, Model, Barva, 0, 'Dealer2'
FROM Auta WHERE NOT EXISTS
(SELECT * FROM Vybava WHERE Vybava.SerioveC = AUTA.SerioveC
    AND Popis='AutPrevod')
```

Mediátor


User query
Result
Mediator
Query
Result
Result
Query
Wrapper
Wrapper
Query
Result
Query
Result
Source 1
Source 2

Mediátor - charakteristika

- Mediátor je založen na vytvoření množiny pohledů, které integrují několik zdrojů velmi podobným způsobem jak je tomu u datového skladu; mediátor ale neukládá žádná data. Proto je mechanizmus mediátorů a datových skladů rozdílný.
- Uživatel adresuje dotaz na mediátora; ten musí získat data z původních zdrojů (protože sám žádná data nemá) a použít je (zkombinovat) na vytvoření odpovědi.
- Mediátor může poslat i více dotazů na některé wrappery, ale nemusí se vždy dotazovat všech. Postup může být i takový, že nejdříve vyhodnotí výsledky dotazu na podmnožinu zdrojů a až poté případně pokračuje v dotazovaní dále.
- Wrapper musí být schopen přijímat různorodé dotazy od mediátora a adaptovat je na dotazy nad strukturami zdrojů.

Vytváření adaptérů (wrapperů)

- V systémech založených na mediátorech jsou potřebné mnohem složitější adaptéry než v systémech pro datové sklady.
- Systematický přístup k vytváření adaptérů vychází z určité klasifikace očekávaných dotazů a vytváří šablony dotazů s parametry.
- Mediátor poskytuje potom do šablony dotazu příslušnou konstantu a adaptér potom realizuje dotaz s touto hodnotou. Počet šablon dotazů může nepřiměřeně narůstat; existují metody zjednodušení, které jsou založeny na tom, že přidávají wrapperům více schopností – např. wrappery umí filtrovat data nebo kombinovat různé šablony.

Uživatelsky definované funkce

- Pohledy neumožňují pracovat s parametry; lze ale využít uživatelsky definované funkce:
  - buď přímé tabulkové (inline-table-valued) funkce
  - nebo vícepříkazové tabulkové (multi-statement table-valued) funkce.
- Přímá tabulková funkce vrací sadu výsledků na základě jediného příkazu SELECT, jímž se definují vracené řádky a sloupce.
- Na rozdíl od uložené procedury se na přímou UDF dá odkazovat v klauzuli FROM dotazu, nebo ji spojit s jinými tabulkami.
- Přímá UDF může přebírat parametry.

Uživatelsky definované funkce

- Vícepříkazová tabulková UDF také vrací sadu výsledků a lze se na ni odkazovat v klauzuli FROM dotazu. Není ale limitovaná jediným příkazem SELECT v těle funkce.
- UDF mají dobrý výkon, protože jejich plány vykonání se ukládají do cache, aby se mohly opětovně využívat.

Příklad

- Uvažujme opět příklad s dealery aut - mediátor integruje stejné zdroje jako v případě datového skladu:
  - AUTA_MEDIATOR(SerioveC, Model, Barva, AutPrevod, Dealer)
- Předpokládejme, že uživatel položí dotaz na mediátora

```
SELECT SerioveC, Model, Barva, AutPrevod, Dealer
FROM AUTA_MEDIATOR
WHERE barva = ‘Zelená’
```

- Mediátor v reakci na tento dotaz položí stejný dotaz na wrapper jedné i druhé databáze.
- Příslušný wrapper adaptuje dotaz na strukturu základní databáze a vrátí výsledky mediátoru.

Dotaz na Dealera1 v T-SQL


```
CREATE  FUNCTION GetAuta1_barva(@barva varchar(20))
RETURNS @NalezenaVozidla TABLE
    (Cislo  char(10),
    Model  char(10),
    Barva  varchar(20) ,
    AutPrevod bit,
    Dealer varchar(10) )
BEGIN
     Insert @NalezenaVozidla
     Select  Cislo, Model, Barva, AutPrevod, ‘Dealer1'
     FROM Vozidla
     WHERE barva=@barva
RETURN
END
```

Použití UDF:

```
select * from GetAuta1_barva('Zelená')
```

Příklad - pokračování

- V dalším kroku mediátor pošle dotaz adaptovaný na strukturu databáze Dealera2.
- Nakonec mediátor vytvoří sjednocení výsledku obou dotazů a předá výsledek uživateli.