Data Transformation and Integration
Part 2: Data Fusion

Dr Tom Diethe

tom.diethe@bristol.ac.uk

University of Bristol

Recap and Outlook

Lec 1 Data Formats

Lec 3 Databases

Lec 4 Data Wrangling

- Demonstration
- Cleaning data
- Missing data

Lec 5 **Data Fusion**

- Demonstration (continued)
- Merging Datasets
- Data Aggregation

Lec 6 Data Exploration

Lec 7 Data Visualisation

Data Integration

- Combine data from different sources *Information silos* $\longrightarrow$ unified view
  - Commercial (e.g. similar companies)
  - Scientific (e.g. bioinformatics)

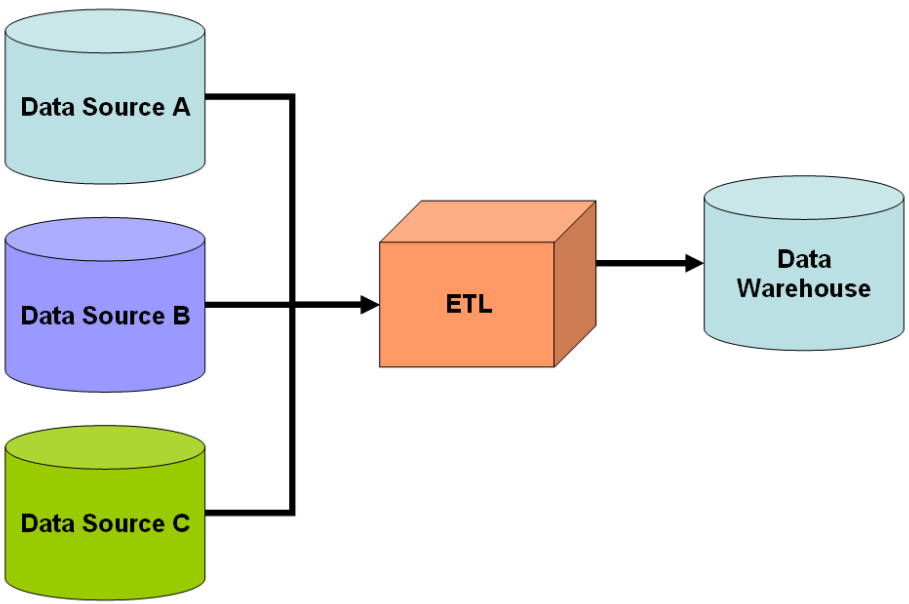


Figure: Extract, Transform, Load (ETL)

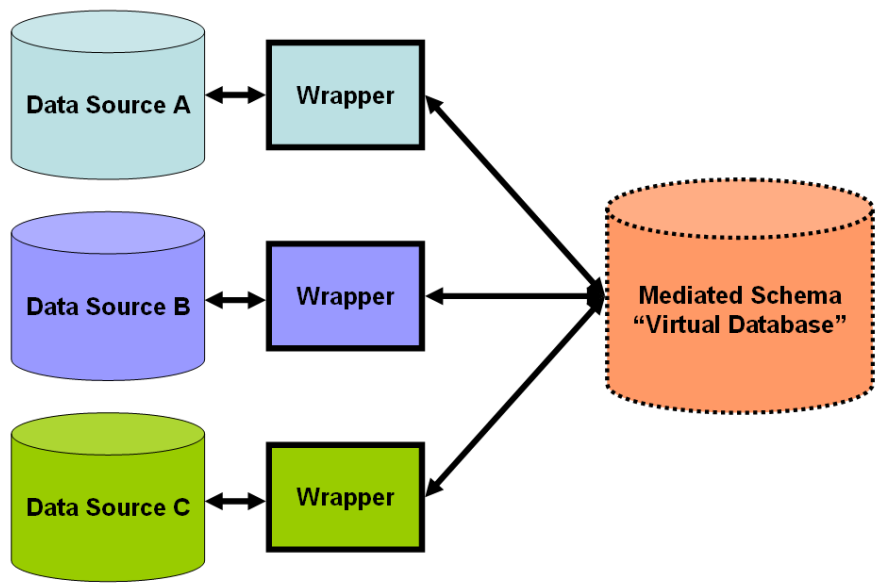


Figure: Data-integration solution

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Challenges

- Heterogeneity: Datasets may be different in terms of:
  - Resolution (e.g. temporal/spatial)
  - Coding (e.g. country names)
  - Imputation of missing values
  - Methods of collection
  - License
- Bad Data
- Lack of Storage Capacity
- Original source(s) may vanish
- Read the specifications!

Simple integration example

- Worldbank data integration:

`https://github.com/njtwomey/ADS/blob/master/03_data_transformation_and_integration/02_worldbank.ipynb`

- See also:

`https://goo.gl/2U2ZUs`

More complex integration example

- CASAS wrangling (continued from February 22nd)
- Start at “Merging categorical and numeric data together”

`https://github.com/njtwomey/ADS/blob/master/03_data_transformation_and_integration/01_wrangling_casas.ipynb`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

Object Relational Mapping (ORM)

- Converting data between incompatible type systems in OO languages
- “virtual object database”

+ves Speeds-up development; overcomes SQL differences

-ves Less control; execution speed; difficult to do complex queries

- Design queries for least number of round-trips with the server
- Compare queries with the actual ones being executed in SQL server profiler
- Examples
  - Hibernate (Java)
  - Django, mongoengine (python)
  - SQLAlchemy (python)
  - LINQ to SQL (.NET languages)
  - More: `https://en.wikipedia.org/wiki/List_of_object-relational_mapping_software`

ORM Example

```
// Standard version
string sql = "SELECT * FROM persons WHERE id = 10";
DbCommand cmd = new DbCommand(connection, sql);
Result res = cmd.Execute();
string name = res[0]["FIRST_NAME"];

// ORM version
Person p = repository.GetPerson(10);
string name = p.getFirstName();
```

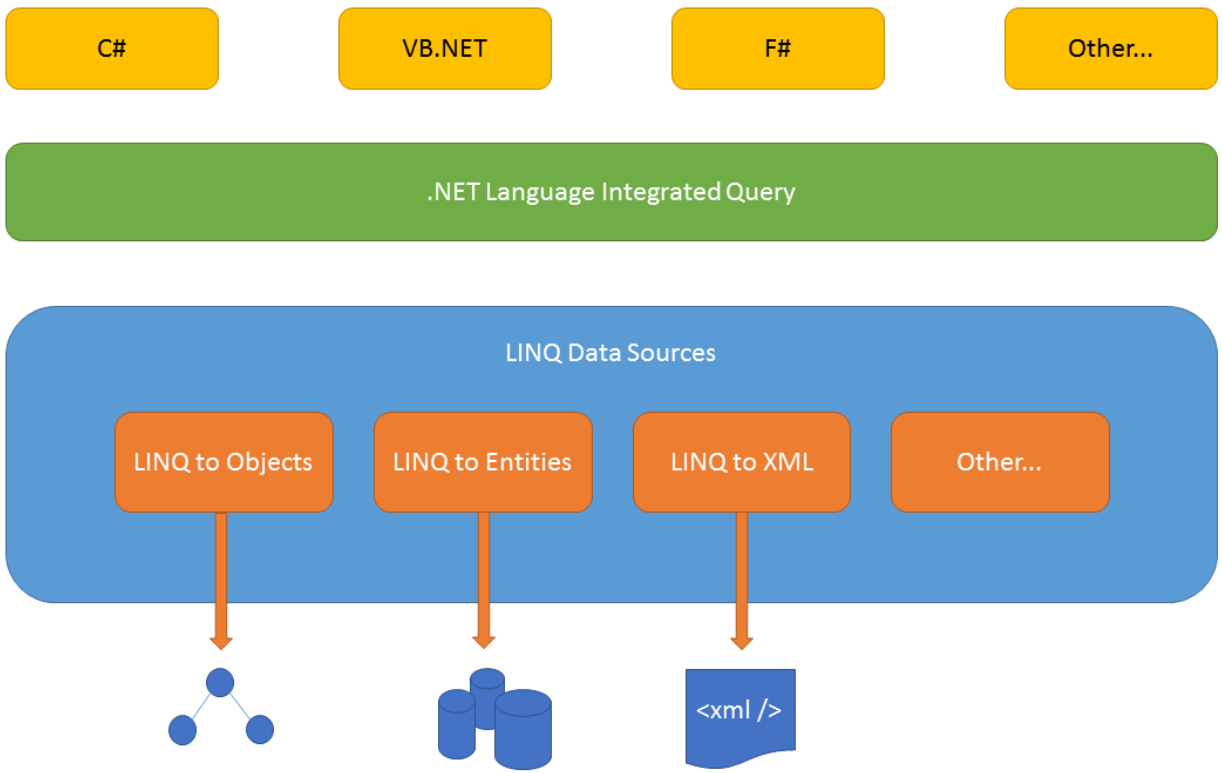
ORM Example

```
// Standard version
string sql = "SELECT * FROM persons WHERE id = 10";
DbCommand cmd = new DbCommand(connection, sql);
Result res = cmd.Execute();
string name = res[0]["FIRST_NAME"];

// ORM version
Person p = repository.GetPerson(10);
string name = p.getFirstName();
```

Language Integrated Query (LINQ)

- .NET based abstraction layer for working with data
- Enumerations, set-based operations, projections, filters, etc
- Providers for relational data (LINQ → SQL, → Datasets, → Entities)



Query structure

```
1  // Query syntax
2  int[] numbers = { 7, 53, 45, 99 };
3  var res = from n in numbers
4        where n > 50
5        orderby n
6        select n.ToString();
7
8  // Lambda syntax
9  int[] numbers = { 7, 53, 45, 99 };
10 var res = numbers.Where(n => n > 50)
11               .OrderBy(n => n)
12               .Select(n => n.ToString());
```

Query structure

```
1  // Query syntax
2  int[] numbers = { 7, 53, 45, 99 };
3  var res = from n in numbers
4        where n > 50
5        orderby n
6        select n.ToString();
7
8  // Lambda syntax
9  int[] numbers = { 7, 53, 45, 99 };
10 var res = numbers.Where(n => n > 50)
11               .OrderBy(n => n)
12               .Select(n => n.ToString());
```

LINQ Examples

- LINQ to Database C# demo:

`https://github.com/njtwomey/ADS/tree/master/03_data_transformation_and_integration/Linq2dbDemo`

Summary

- Data integration
  - Challenges
- Data aggregation
  - Object Relational Mapping (ORM)
  - Language Integrated Query (LINQ)
  - Some resources:
    - pandas documentation `http://pandas.pydata.org/`
    - Chapter 9 of Python for Data Analysis
    - Effective Django ORM: `http://www.effectivedjango.com/orm.html`
    - Edulinq: `https://codeblog.jonskeet.uk/category/edulinq/` - an eBook about LINQ by Jon Skeet