Ergebnisse Extraktion und Datenmodell

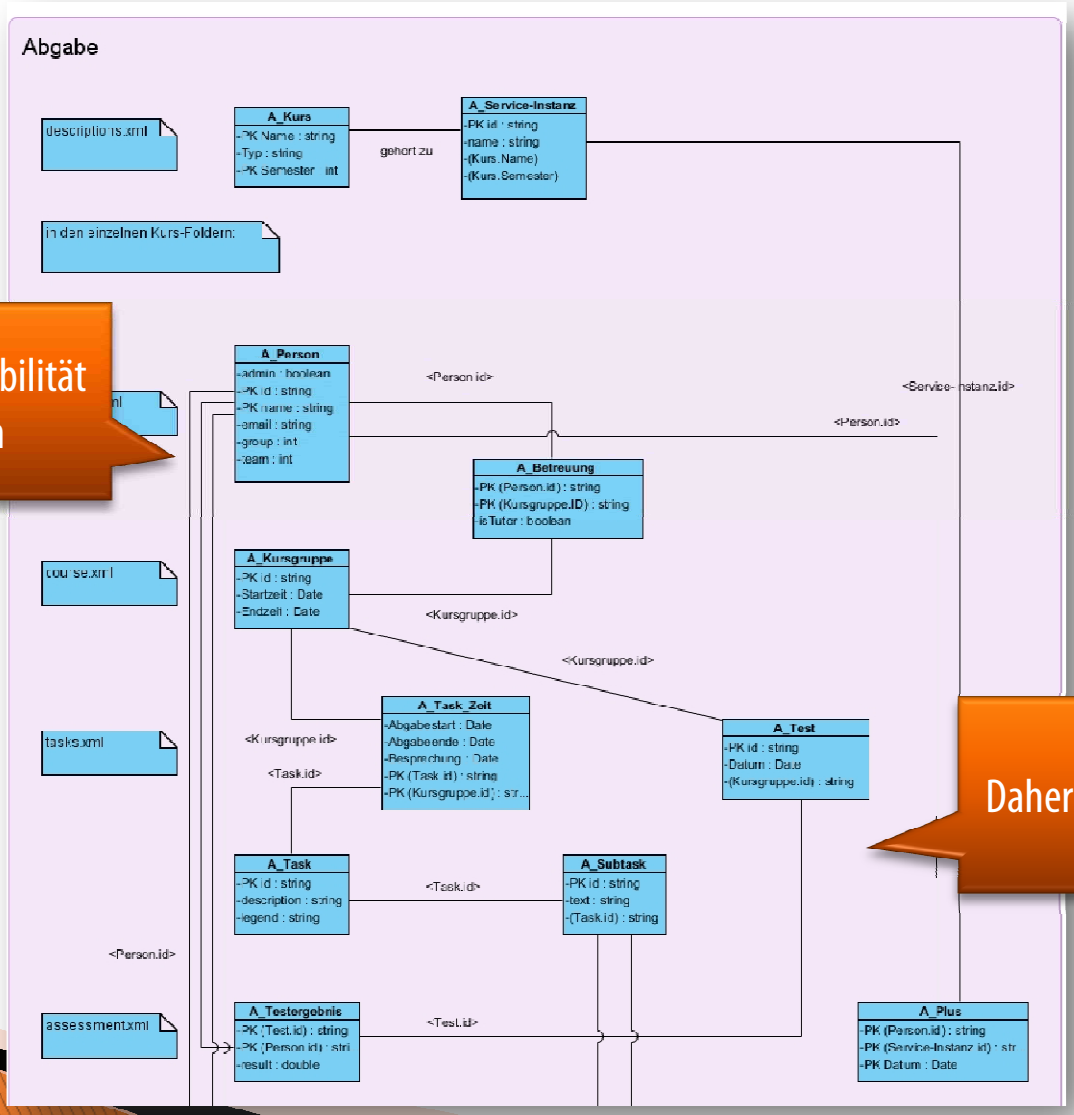
VU Business Intelligence II

Xin Li, Stefan Schwarzenberger, Christian Sieberer

Inhalt

1. **Datenmodell**
2. **Extraktion**
    1. Methode Pentaho
    2. Methode PHP
    3. Methode Data Warehouse
3. **Next Steps**

1. Datenmodell



Viele Entitäten → Flexibilität bei Auswertungen

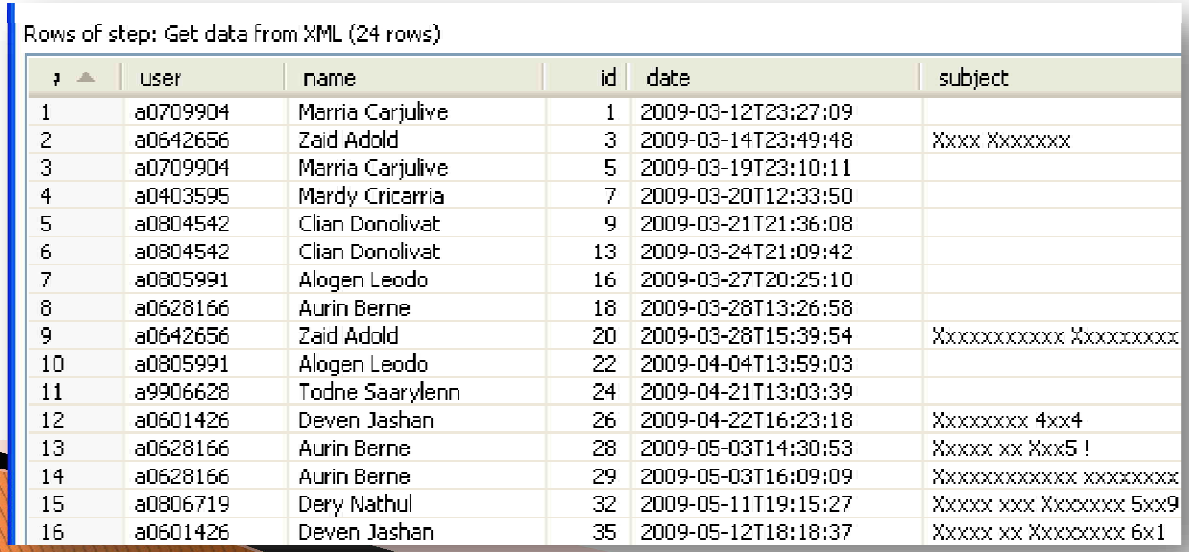
Daher sehr komplexes Modell

2. Extraction – Allgemeines

- Strikte **Aufteilung nach Datenmodell**
- Ein **CSV-File pro Tabelle**
- Folge: gleiche Entitäten in verschiedenen CSVs
  - Personen, Kurse, Kurse pro Semester
  - Bei Import in Pentaho zu integrieren
- **Verschiedene Ansätze** evaluiert
  - Direktimport von XML in Pentaho
  - Konversion per PHP
  - Konversion per Data Warehouse

2.1 Extraktion: Methode Pentaho

- **Pentaho Data Integration** (Kettle)
  - XML-Importfilter erlauben Direktimport
  - Einbindung von Codeblöcken (JavaScript, Java)
  - CSV/YAML-Export möglich
- **Aufwand** ähnlich wie bei PHP, aber weniger Flexibilität – daher nicht gewählt



Rows of step: Get data from XML (24 rows)

#	user	name	id	date	subject
1	a0709904	Marria Carjulive	1	2009-03-12T23:27:09	
2	a0642656	Zaid Adold	3	2009-03-14T23:49:48	Xxxx Xxxxxxx
3	a0709904	Marria Carjulive	5	2009-03-19T23:10:11	
4	a0403595	Mardy Cricarria	7	2009-03-20T12:33:50	
5	a0804542	Clian Donolivat	9	2009-03-21T21:36:08	
6	a0804542	Clian Donolivat	13	2009-03-24T21:09:42	
7	a0805991	Alogen Leodo	16	2009-03-27T20:25:10	
8	a0628166	Aurin Berne	18	2009-03-28T13:26:58	
9	a0642656	Zaid Adold	20	2009-03-28T15:39:54	Xxxxxxxxxx Xxxxxxxx
10	a0805991	Alogen Leodo	22	2009-04-04T13:59:03	
11	a9906628	Todne Saarylenn	24	2009-04-21T13:03:39	
12	a0601426	Deven Jashan	26	2009-04-22T16:23:18	Xxxxxxxx 4xx4
13	a0628166	Aurin Berne	28	2009-05-03T14:30:53	Xxxxx xx Xxx5 !
14	a0628166	Aurin Berne	29	2009-05-03T16:09:09	Xxxxxxxxxxx xxxxxxxx
15	a0806719	Dery Nathul	32	2009-05-11T19:15:27	Xxxxx xxx Xxxxxxx 5xx9
16	a0601426	Deven Jashan	35	2009-05-12T18:18:37	Xxxxx xx Xxxxxxx 6x1

2.2 Extraktion: Methode PHP

- „Straight-forward“ **code-getriebene Lösung**

XML Einlesen → **PHP**: Loops, SimpleXML, Xpath, Regex → **CSV** schreiben

- Herausforderung: **File-Encoding**

```
# Feedback aufbauen.
# Feedback: Task.id, Subtask.id, Student.id, Autor.id, Text.
foreach( $xml as $instance ) {.

        # Für jede Person.
        $atts = get_object_vars( $instance->attributes() ):.
        $atts = $atts['@attributes']:.
        .
        # Kommentare.
        $kommentare = $instance->xpath("comment"):.
        foreach( $kommentare as $sub_instance ) {.
                $kommentare_atts = get_object_vars( $sub_instance->attributes() ):.
                $kommentare_atts = $kommentare_atts['@attributes']:.
                $kommentar_text = str_replace( "\n", " ", (string) $sub_instance[0] ):.
                $feedback[] = array( $kommentare_atts['task'], $kommentare_atts['subtask'], $at
        }.
}.
.
$filename_out = "feedback_" . $subdir . ".csv":.
$filehandle = fopen( $filename_out, "w" ):
```

Eine Methode pro Entitätstyp

2.3 Extraktion: Methode Data-Warehouse

- **Daten-zentrierte Lösung**
- Konversion mittels SAP BW Real-Time Data Acquisition

- Aufwand ähnlich wie bei PHP

3. Next Steps

- **Vollständige Integration** der Daten
  - Entitäten, die an mehreren Orten vorkommen: Personen, Kurse, Betreuungszuordnungen, Kurse pro Semester
  - Einheitliches Encoding
- **Laden** der Daten **ins Data Warehouse**