

Informationssysteme

Skriptum zur Vorlesung

Dipl.-Ing. Paul Panhofer BSc.^{1*}

1 ZID, TU Wien, Taubstummengasse 11, 1040, Wien, Austria

Abstract: Ein Informationssystem ist ein soziotechnisches System, das die Abarbeitung von Informationsnachfrage zur Aufgabe hat. Es handelt sich um ein Mensch/Aufgabe/Technik-System, das Daten produziert, beschafft, verteilt und verarbeitet.

Daneben bezeichnen Informationssysteme im allgemeineren Sinne Systeme von Informationen, die in einem wechselseitigen Zusammenhang stehen.

Die Begriffe Informationssystem und Anwendungssystem werden häufig synonym verwendet. Dabei werden Informationssysteme im engeren Sinne als computergestützte Anwendungssysteme verstanden. Es ist jedoch wichtig zu verstehen, dass ein Anwendungssystem mit Anwendungssoftware und Datenbank nur Teil eines Informationssystems sind.

MSC: paul.panhofer@gmail.com

Keywords:

Contents		
1. SQL - Data Query Lanuguage	4	
1.1. SQL Grundlagen	4	
1.1.1. 1.Beispiel - select Klausel	4	
1.1.2. 2.Beispiel - II Operator	5	
1.1.3. 3.Beispiel - where Klausel	5	
1.1.4. 4.Beispiel - like Operator	5	
1.1.5. 5.Beispiel - in Operator	6	
1.1.6. 6.Beispiel - 3 wertige Logik	6	
1.1.7. 7.Beispiel - between Operator	7	
1.1.8. 8.Beispiel - between Operator	7	
1.1.9. 9.Beispiel - Order By Klausel	7	
1.1.10. 10.Beispiel - Fetch Klausel	8	
1.2. Datenaggregation	8	
1.2.1. 1.Beispiel - Datenaggregation	8	
1.2.2. 2.Beispiel - Inner Join	9	
1.2.3. 3.Beispiel - Inner Join	9	
1.2.4. 4.Beispiel - Inner Join	10	
1.2.5. 5.Beispiel - Cross Join	10	
1.2.6. 6.Beispiel - Cross Join	10	
1.2.7. 7.Beispiel - Cross Join	11	
1.2.8. 8.Beispiel - Left Join	11	
1.2.9. 9.Beispiel - Left Join	11	
1.2.10. 10.Beispiel - Left Join	12	
1.3. Aggregatfunktionen	13	
1.3.1. 1.Beispiel - Aggregatfunktionen	13	
1.3.2. 2.Beispiel - Group By Klausel	13	
1.3.3. 3.Beispiel - Group By Klausel	13	
1.3.4. 4.Beispiel - Group By Klausel	14	
1.3.5. 5.Beispiel - Group By Klausel	14	
1.3.6. 6.Beispiel - Having Klausel	15	
1.3.7. 7.Beispiel - Having Klausel	15	
1.3.8. 8.Beispiel - Group By Klausel	16	
1.4. Subselect	16	
1.4.1. 1.Beispiel - Subselect	16	

*E-mail: paul.panhofer@tuwien.ac.at

1.4.2. 2.Beispiel - Subselect	16	4.1.5. 5.Beispiel - Transformationsprozesse	43
1.4.3. 3.Beispiel - Subselect	17	4.1.6. 6.Beispiel - Transformationsprozesse	43
1.4.4. 4.Beispiel - Subselect	17	4.1.7. 7.Beispiel - Transformationsprozesse	44
1.4.5. 5.Beispiel - Subselect, from Klausel	18	4.2. Prozedurale Programmierung	44
1.4.6. 6.Beispiel - Subselect, from Klausel	18	4.2.1. 1.Beispiel - Transformationsprozesse	44
1.4.7. 7.Beispiel - Subselect, from Klausel	19	4.2.2. 2.Beispiel - Transformationsprozesse	45
1.4.8. 8.Beispiel - Subselect, from Klausel	19	4.2.3. 3.Beispiel - Transformationsprozesse	45
1.4.9. 9.Beispiel - Subselect	20	4.2.4. 4.Beispiel - Transformationsprozesse	46
1.5. Zeilenfunktionen	20	4.2.5. 5.Beispiel - Transformationsprozesse	46
1.5.1. 1.Beispiel - Datumsfunktionen	20	4.2.6. 6.Beispiel - Transformationsprozesse	47
1.5.2. 2.Beispiel - Datumsfunktionen	21	4.2.7. 7.Beispiel - Transformationsprozesse	47
1.5.3. 3.Beispiel - Datumsfunktionen	21		
1.5.4. 4.Beispiel - Textfunktionen	22	5. NoSQL - Informationssysteme	48
1.5.5. 5.Beispiel - Textfunktionen	22	5.1. NoSQL Informationssysteme	48
1.5.6. 6.Beispiel - Numerische Funktionen	23	5.1.1. 1.Beispiel - Sql vs. NoSql System	48
		5.1.2. 2.Beispiel - CAP Theorem	48
2. XML Schema	24	5.1.3. 3.Beispiel - CAP Theorem	49
2.1. Einfache Datentypen	24	5.2. Konsistenzmodelle	49
2.1.1. 1.Beispiel - Einfache Datentypen	24	5.2.1. 1.Beispiel - Transaktion	49
2.1.2. 2.Beispiel - Einfache Datentypen	25	5.2.2. 2.Beispiel - Konsistenzmodelle	50
2.1.3. 3.Beispiel - Einfache Datentypen	25	5.2.3. 3.Beispiel - Sperren	50
2.2. Komplexe Datentypen	26	5.2.4. 4.Beispiel - BASE Transaktionen	50
2.2.1. 1.Beispiel - Komplexe Datentypen	26	5.2.5. 5.Beispiel - MVCC Verfahren	50
2.2.2. 2.Beispiel - Komplexe Datentypen	27		
2.2.3. 3.Beispiel - Komplexe Datentypen	28	6. MongoDB	52
2.2.4. 4.Beispiel - Komplexe Datentypen	28	6.1. Modellierung	52
2.2.5. 5.Beispiel - Komplexe Datentypen	29	6.1.1. 1.Beispiel) Modellierung	52
2.2.6. 6.Beispiel - Komplexe Datentypen	30	6.1.2. 2.Beispiel) Dokumente einfügen	53
2.2.7. 7.Beispiel - Komplexe Datentypen	31	6.1.3. 3.Beispiel) Dokumente bearbeiten	54
		6.2. Abfragen	54
3. XML - XPath	32	6.2.1. 1.Beispiel) Query Kriterien	54
3.1. XPath - Grundlagen	32	6.2.2. 2.Beispiel) Query Kriterien	55
3.1.1. 1.Beispiel - Knotenbaum	32	6.2.3. 3.Beispiel) Query Kriterien	55
3.1.2. 2.Beispiel - Knotenbaum	33		
3.1.3. 3.Beispiel - Stringdarstellung	33		
3.2. Lokalisierungspfade	34		
3.2.1. 1.Beispiel - Lokalisierungspfade	34		
3.2.2. 2.Beispiel - Lokalisierungspfade	35		
3.2.3. 3.Beispiel - Lösungsobjekte	35		
3.2.4. 4.Beispiel - Prädikate	36		
3.2.5. 5.Beispiel - Prädikate	36		
3.2.6. 6.Beispiel - Lokalisierungspfade	37		
3.3. Standardpfadangaben	38		
3.3.1. 1.Beispiel - Standardschreibweise	38		
3.3.2. 2.Beispiel - Standardschreibweise	38		
3.3.3. 3.Beispiel - Standardschreibweise	39		
4. XML - XSLT	40		
4.1. Deklarative Templates	40		
4.1.1. 1.Beispiel - Transformationsprozesse	40		
4.1.2. 2.Beispiel - Transformationsprozesse	41		
4.1.3. 3.Beispiel - Transformationsprozesse	41		
4.1.4. 4.Beispiel - Transformationsprozesse	42		

1. SQL - Data Query Language

01

SQL Grundlagen

01.	SQL Grundlagen	4
02.	Datenaggregation	8
03.	Aggregatfunktionen	13
04.	Subselect	16
05.	Zeilenfunktionen	20

1.1. SQL Grundlagen

1.Beispiel - select Klausel



Beispielbeschreibung

- **Schwerpunkt:** SQL GRUNDLAGEN, SELECT KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 44

► Aufgabenstellung: Select Klausel

- GEBEN SIE FÜR ALLE EMPLOYEES FOLGENDE SPALTEN AUS:
 - FIRST_NAME, LAST_NAME, SALARY UND EMAIL
- SORTIEREN SIE DAS ERGEBNIS NACH LAST_NAME UND FIRST_NAME.

► Lösung: Select Klausel

```

1  -----
2  -- Loesung: Select Klausel
3  -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32  .

```



2.Beispiel - || Operator



Beispielbeschreibung ▾

- **Schwerpunkt:** SQL GRUNDLAGEN, SELECT KLAUSEL, || OPERATOR, SPALTENALIAS
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 45

▸ Aufgabenstellung: || Operator ▾

- GEBEN SIE FÜR ALLE **EMPLOYEES** FOLGENDE SPALTEN AUS:
 - **FIRST_NAME, LAST_NAME, SALARY UND EMAIL**
- SORTIEREN SIE DAS ERGEBNIS NACH **LAST_NAME** UND **FIRST_NAME**.
- DAS ERGEBNIS SOLL ALS EINZELNE SPALTE DARGESTELLT WERDEN.
- VERWENDEN SIE DEN BEGRIFF **EMPLOYEE_DATA** ALS SPALTENBEZEICHNUNG.

▸ Lösung: || Operator ▾

```

1  -- -----
2  -- Lösung: || Operator
3  -- -----
4
5
6
7
8
9
10
11
12
13
14  .

```



3.Beispiel - where Klausel



Beispielbeschreibung ▾

- **Schwerpunkt:** SQL BEFEHL, WHERE KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 47

▸ Aufgabenstellung: where Klausel ▾

- FINDEN SIE ALLE **EMPLOYEES** DIE IM UNTERNEHMEN ALS **IT_PROG** ARBEITEN. BEACHTEN SIE DASS NUR **EMPLOYEES** AUSGEGEBEN WERDEN SOLLEN DIE MEHR ALS 12 000€ VERDIENEN.
- GEBEN SIE FÜR DIE **EMPLOYEES** **FIRST_NAME, LAST_NAME, JOB_ID** UND **SALARY** AUS.
- SORTIEREN SIE DAS ERGEBNIS NACH **FIRST_NAME** UND **LAST_NAME**.

▸ Lösung: where Klausel ▾

```

1  -- -----
2  -- Lösung: where Klausel
3  -- -----
4
5
6
7
8
9  .

```



4.Beispiel - like Operator



Beispielbeschreibung ▾

- **Schwerpunkt:** SQL GRUNDLAGEN, WHERE KLAUSEL, LIKE OPERATOR
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 48

▸ Aufgabenstellung: like Operator ▾

- GEBEN SIE ALLE **EMPLOYEES** AUS DEREN **LAST_NAME** MIT H BEGINNT ABER NICHT MIT R ENDET.
- GEBEN SIE FÜR DIE **EMPLOYEES** **FIRST_NAME, LAST_NAME** UND **DEPARTMENT_ID** AUS.

▸ Lösung: like Operator ▾

```

1  -- -----
2  -- Lösung: like Operator
3  -- -----
4
5
6
7
8
9  .

```



5.Beispiel - in Operator



Beispielbeschreibung ▼

- **Schwerpunkt:** SQL GRUNDLAGEN, WHERE KLAUSEL, IN OPERATOR
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 48

► Aufgabenstellung: in Operator ▼

- FINDEN SIE ALLE **EMPLOYEES** DIE ENTWEDER ALS **SA_MAN**, **SA_REP** ODER ALS **ST_CLERK** ARBEITEN.
- ES SOLLEN ABER KEINE **IT_PROG** AUSGEGEBEN WERDEN.
- GEBEN SIE DEN **LAST_NAME** UND **FIRST_NAME** FÜR DIE **EMPLOYEES** AUS.

► Lösung: in Operator ▼

```

1  -- -----
2  -- Loesung: in Operator
3  -- -----
4
5
6
7
8
9
10
11
12
13
14  .

```



6.Beispiel - 3 wertige Logik



Beispielbeschreibung ▼

- **Schwerpunkt:** 3 WERTIGE LOGIK
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 49

► Aufgabenstellung: 3 wertige Logik ▼

- WERTEN SIE DIE FOLGENDEN LOGISCHEN TERME AUS.
- JEDE DER SPALTEN KANN AUCH **NULL** WERTE ENTHALTEN.



► Lösung: 3 wertige Logik ▼

```

1  -- -----
2  -- Loesung: 3 wertige Logik
3  -- -----
4  not(index > 100)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22  real = 10 and not(index = 100)
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42  not(real >= 32 or not (index < 0 and deptno = 20))
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58  .

```

7. Beispiel - between Operator



Beispielbeschreibung ▼

- **Schwerpunkt:** SQL GRUNDLAGEN, WHERE KLAUSEL, BETWEEN OPERATOR
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 49

► Aufgabenstellung: between Operator ▼

- FINDEN SIE ALLE **ANGESTELLTEN** DIE NICHT WENIGER ALS 10000 UND NICHT MEHR ALS 17000 VERDIENEN.
- BESCHRÄNKEN SIE SICH BEI DER FORMULIERUNG IHRER QUERY AUF DIE **EMPLOYEES** TABELLE.

► Lösung: between Operator ▼

```

1  -----
2  -- Loesung: between Operator
3  -----
4
5
6
7
8
9
10 .

```



8. Beispiel - between Operator



Beispielbeschreibung ▼

- **Schwerpunkt:** SQL GRUNDLAGEN, WHERE KLAUSEL, BETWEEN OPERATOR
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 49

► Aufgabenstellung: between Operator ▼

- FINDEN SIE ALLE **ANGESTELLTEN** DIE 5 JAHRE ODER LÄNGER IM UNTERNEHMEN ARBEITEN.
- MIT DER **TO_DATE** FUNKTION KÖNNEN SIE DATUMSWERTE ANGEBEN.

► Lösung: between Operator ▼

```

1  -----
2  -- Loesung: between Operator
3  -----
4  to_date('15.05.2012', 'dd.mm.yyyy')
5
6
7
8
9
10
11
12 .

```



9. Beispiel - Order By Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** SQL GRUNDLAGEN, WHERE KLAUSEL, FETCH KLAUSEL, ORDER BY KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 49, 50

► Aufgabenstellung: Order By Klausel ▼

- FINDEN SIE ALLE **EMPLOYEES** DES UNTERNEHMENS
- GEBEN SIE DEN **FIRST_NAME** UND **LAST_NAME** DER ANGESTELLTEN AN.
- ORDNET SIE DAS ERGEBNIS DER ABFRAGE NACH DEM **FIRST_NAME** ABSTEIGEN UND DEM **LAST_NAME** AUFSTIEGEND.
- GEBEN SIE NUR DIE ERSTEN 10 ANGESTELLTEN AUS.

► Lösung: Order by Klausel ▼

```

1  -----
2  -- Loesung: Order By Klausel
3  -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17 .

```



10. Beispiel - Fetch Klausel



Beispielbeschreibung ▾

- **Schwerpunkt:** SQL GRUNDLAGEN, WHERE KLAUSEL, FETCH KLAUSEL, ORDER BY KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 49, 50, 51

► Aufgabenstellung: Fetch Klausel ▾

- FINDEN SIE ALLE **EMPLOYEES** DES UNTERNEHMENS.
- GEBEN SIE DEN **FIRST_NAME** UND **LAST_NAME** DER **EMPLOYEES** AUS.

► Lösung: Fetch Klausel ▾

```

1  -- -----
2  -- Loesung: Fetch Klausel
3  -- -----
4  -- Geben Sie nur die ersten 10% der employees aus
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 -- Formulieren Sie eine Paginierung. Geben Sie die
22 -- Angestellten 90 - 110 aus.
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39 .

```



1.2. Datenaggregation ▾

1. Beispiel - Datenaggregation



Beispielbeschreibung ▾

- **Schwerpunkt:** DATENAGGREGATION
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 49, 50

► Aufgabenstellung: Datenaggregation ▾

- ERKLÄREN SIE DEN BEGRIFF DER **Datenaggregation**. GEBEN SIE EIN BEISPIEL FÜR DEN VORGANG DER DATENAGGREGATION.

► Lösung: Datenaggregation ▾

```

1  -- -----
2  -- Loesung: Datenaggregation
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37 .

```



2.Beispiel - Inner Join



Beispielbeschreibung ▼

- **Schwerpunkt:** DATENAGGREGATION, INNER JOIN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 55, 56, 57

► Aufgabenstellung: Inner Join ▼

- FINDEN SIE ALLE **EMPLOYEES** DIE IN DER IT ABTEILUNG ARBEITEN.
- GEBEN SIE FÜR DIE DATENSÄTZEN DIE FOLGENDEN SPALTEN AUS: **FIRST_NAME**, **LAST_NAME**, **DEPARTMENT_NAME**, **COUNTRY_NAME**.

► Lösung: Inner Join ▼

```

1  -- -----
2  -- Loesung: Datenaggregation
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34 .

```



3.Beispiel - Inner Join



Beispielbeschreibung ▼

- **Schwerpunkt:** DATENAGGREGATION, INNER JOIN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 55, 56, 57

► Aufgabenstellung: Inner Join ▼

- GILT DIE FOLGENDE **mengentheoretische Äquivalenz**? BEGRÜNDEN SIE IHRE ANTWORT.

$A \text{ inner join } B \Leftrightarrow B \text{ inner join } A$

- MIT WELCHER **Mengenoperation** KANN DER **Inner Join** VERGLICHEN WERDEN?

► Lösung: Inner Join ▼

```

1  -- -----
2  -- Loesung: Datenaggregation
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```



4.Beispiel - Inner Join



Beispielbeschreibung ▾

- **Schwerpunkt:** DATENAGGREGATION, INNER JOIN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 55, 56, 57

▸ Aufgabenstellung: Inner Join ▾

- DIE TABELLEN A UND B WERDEN ÜBER EINE **Fremdschlüsselspalte** MITEINANDER GEJOINED.
- WELCHE **Beziehung**¹ MUSS ZWISCHEN A UND B VORHERSCHEN DAMIT DAS ERGEBNIS DES **Inner Joins genauso-viele** ZEILEN ENTHÄLT WIE DIE TABELLE A.
- WELCHE **Beziehung** MUSS ZWISCHEN A UND B VORHERSCHEN DAMIT DAS ERGEBNIS DES **Innner Joins weniger** ZEILEN ENTHÄLT ALS DIE TABELLE A.
- WELCHE **Beziehung** MUSS ZWISCHEN A UND B VORHERSCHEN DAMIT DAS ERGEBNIS DES **Innner Joins mehr** ZEILEN ENTHÄLT ALS DIE TABELLE A.

▸ Lösung: Inner Join ▾

```

1  -----
2  -- Loesung: Datenaggregation
3  -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25  .

```



5.Beispiel - Cross Join



Beispielbeschreibung ▾

- **Schwerpunkt:** DATENAGGREGATION, CROSS JOIN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 55, 56, 57

▸ Aufgabenstellung: Cross Join ▾

- GEGEBEN IST EINE TABELLE A MIT 5 ZEILEN UND EINE TABELLE B MIT 10 ZEILEN, WIEVIEL ZEILEN HAT DER **Cross Join** ZWISCHEN A UND B.

▸ Lösung: Cross Join ▾

```

1  -----
2  -- Loesung: Cross Join
3  -----
4
5
6
7
8  .

```



6.Beispiel - Cross Join



Beispielbeschreibung ▾

- **Schwerpunkt:** DATENAGGREGATION, CROSS JOIN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 55, 56, 57

▸ Aufgabenstellung: Cross Join ▾

- GILT DIE FOLGENDE **mengentheoretische Äquivalenz?** BEGRÜNDEN SIE IHRE ANTWORT.

$$A \text{ cross join } B \Leftrightarrow B \text{ cross join } A$$

▸ Lösung: Cross Join ▾

```

1  -----
2  -- Loesung: Cross Join
3  -----
4
5
6  .

```



¹ Relation

7.Beispiel - Cross Join



Beispielbeschreibung ▼

- **Schwerpunkt:** DATENAGGREGATION, CROSS JOIN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 55, 56, 57

► Aufgabenstellung: Cross Join ▼

- FORMULIEREN SIE DIE FOLGENDEN 2 QUERIES MIT HILFE VON **cross joins**.

► Lösung: Cross Join ▼

```

1  -- -----
2  -- Loesung: Cross Join
3  -- -----
4  SELECT e.last_name,
5         e.first_name,
6         d.department_name
7  FROM employees e JOIN departements d
8  ON e.department_id = d.department_id;
9
10
11
12
13
14
15
16
17
18
19
20
21
22 SELECT e.last_name, e.first_name,
23        d.department_name,
24        c.country_name
25 FROM employees e JOIN departements d
26 ON e.department_id = d.department_id
27 JOIN locations l ON d.location_id = l.location_id
28 JOIN countries c ON l.country_id = c.country_id;
29
30
31
32
33
34
35
36
37
38
39
40
41 .

```



8.Beispiel - Left Join



Beispielbeschreibung ▼

- **Schwerpunkt:** DATENAGGREGATION, LEFT JOIN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 58

► Aufgabenstellung: Left Join ▼

- GEBEN SIE FÜR JEDEN **EMPLOYEE** AUS IN WELCHER ABTEILUNG ER ARBEITET.
- FÜR JEDEN DATENSATZ SOLL DAZU DER **FIRST_NAME**, **LAST_NAME** UND **DEPARTMENT_NAME** AUSGEGEBEN WERDEN.
- IST EIN **EMPLOYEE** KEINER ABTEILUNG ZUGEORDNET SOLL DIE ZEICHENKETTE **NO DEPARTMENT** AUSGEGEBEN WERDEN.

► Lösung: Left Join ▼

```

1  -- -----
2  -- Loesung: Left Join
3  -- -----
4
5
6
7
8
9
10
11
12 .

```



9.Beispiel - Left Join



Beispielbeschreibung ▼

- **Schwerpunkt:** DATENAGGREGATION, LEFT JOIN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 58

► Aufgabenstellung: Left Join ▼

- GILT DIE FOLGENDE **mengentheoretische Äquivalenz?** BEGRÜNDEN SIE IHRE ANTWORT.

A left join B ⇔ B left join A

- MIT WELCHER Mengenoperation KANN DER Left Join VERGlichen WERDEN?

► Lösung: left Join ▼

```

1  -- -----
2  -- Loesung: Left Join
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 .

```



10. Beispiel - Left Join



Beispielbeschreibung ▼

- **Schwerpunkt:** DATENAGGREGATION, LEFT JOIN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 58

► Aufgabenstellung: Left Join ▼

- BESCHREIBEN DIE FOLGENDEN QUERIES DIESELBE ABFRAGE.
- IHRE ANTWORT MUSS UNABHÄNGIG VOM DATENBESTAND DER DATENBANK GELTEN.

► Lösung: Left Join ▼

```

1  -- -----
2  -- Loesung: Left Join
3  -- -----
4  SELECT e.last_name, e.first_name,
5         d.department_name, c.country_name
6  FROM employees e LEFT JOIN departements d
7  ON e.department_id = d.department_id
8  LEFT JOIN locations l ON d.location_id =
   l.location_id
9  LEFT JOIN countries c ON l.country_id =
   c.country_id;

```



```

9  SELECT e.last_name, e.first_name,
10         d.department_name,
11         c.country_name
12 FROM employees e
13 LEFT JOIN departements d ON e.department_id =
   d.department_id
14 JOIN locations l ON d.location_id = l.location_id
15 JOIN countries c ON l.country_id = c.country_id;
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67 .

```

1.3. Aggregatfunktionen ▼

1.Beispiel - Aggregatfunktionen



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 70, 76

► Aufgabenstellung: Aggregatfunktionen ▼

- GEBEN SIE DAS MAXIMALE UND MINIMALE GEHALT DER EMPLOYEES IM UNTERNEHMEN AN.
- WARUM WIRD ZUR LÖSUNG DER AUFGABE KEINE GROUP BY KLAUSEL VERWENDET.
- WARUM KANN NICHT EBENFALLS DER LAST_NAME DER ANGESTELLTEN AUSGEGEBEN WERDEN?

► Lösung: Aggregatfunktionen ▼

```

1  -- -----
2  -- Loesung: Aggregatfunktionen
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33  .

```



2.Beispiel - Group By Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, GROUP BY KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 70, 71, 72, 73, 74, 75

► Aufgabenstellung: Group By Klausel ▼

- GEBEN SIE FÜR JEDES DEPARTMENT DES UNTERNEHMENS DAS MINIMALE UND MAXIMALE GEHALT AN. GEBEN SIE EBENFALLS DEN DEPARTMENT_NAME AN.

► Lösung: Group By Klausel ▼

```

1  -- -----
2  -- Loesung: Group By Klausel
3  -- -----
4
5
6
7
8
9
10
11
12
13
14  .

```



3.Beispiel - Group By Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, GROUP BY KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 70, 71, 72, 73, 74, 75

► Aufgabenstellung: Group By Klausel ▼

- GEBEN SIE FÜR JEDES DEPARTMENT DIE ANZAHL DER MITARBEITER AN, DIE DORT ARBEITEN.
- NACH WELCHEN WERTEN KANN DAS ERGEBNIS SORTIERT WERDEN?

► Lösung: Group By Klausel ▼

```

1  -----
2  -- Loesung: Group By Klausel
3  -----
4
5
6
7
8
9
10
11 .

```



4.Beispiel - Group By Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, GROUP BY KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 70, 71, 72, 73, 74, 75

► Aufgabenstellung: Group By Klausel ▼

- GEBEN SIE FÜR DIE ABTEILUNGEN, JEDES LANDES DIE ANZAHL DER MITARBEITER AN, DIE DORT ARBEITEN.

► Lösung: Group By Klausel ▼

```

1  -----
2  -- Loesung: Group By Klausel
3  -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 .

```



5.Beispiel - Group By Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, GROUP BY KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 70, 71, 72, 73, 74, 75

► Aufgabenstellung: Group By Klausel ▼

- BESCHREIBEN DIE BEIDEN QUERIES DIESELBE ABFRAGE? ERKLÄREN SIE IHRE LÖSUNG.

► Lösung: Group By Klausel ▼

```

1  -----
2  -- Loesung: Group By Klausel
3  -----
4  SELECT location_id, department_id,
5         count(e.employee_id)
6  FROM employees e JOIN departemts d on
7         e.department_id = d.department_id
8  JOIN locations l on d.location_id = l.location_id
9  GROUP BY d.department_id, l.location_id;
10
11 SELECT location_id, department_id,
12         count(e.employee_id)
13  FROM employees e JOIN departemts d on
14         e.department_id = d.department_id
15  JOIN locations l on d.location_id = l.location_id
16  GROUP BY l.location_id, d.department_id;
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35 .

```



6.Beispiel - Having Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, HAVING KLAUSEL, GROUP BY KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 75

► Aufgabenstellung: Having Klausel ▼

- GEBEN SIE JENE ABTEILUNGEN AUS, IN DENEN MEHR ALS 5 MITARBEITER BESCHÄFTIGT SIND.
- GEBEN SIE DEN `DEPARTMENT_NAME`, UND DIE ANZAHL DER MITARBEITER AN.
- ALLE `IT PROGRAMMIERER` SOLLEN DABEI NICHT BERÜCKSICHTIGT WERDEN.

► Lösung: Having Klausel ▼

```

1  -- -----
2  -- Loesung: Having Klausel
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30  .

```



7.Beispiel - Having Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, GROUP BY KLAUSEL, HAVING KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 75

► Aufgabenstellung: Having Klausel ▼

- FINDEN SIE ALLE LÄNDER IN DENEN SICH MEHR ALS 3 ABTEILUNGEN BEFINDEN.
- GEBEN SIE `COUNTRY_ID` UND DIE ANZAHL DER ABTEILUNGEN AUS.
- BEACHTEN SIE DAS `SALES` ABTEILUNGEN NICHT BERÜCKSICHTIGT WERDEN SOLLN.

► Lösung: Having Klausel ▼

```

1  -- -----
2  -- Loesung: Having Klausel
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31  .

```



8.Beispiel - Group By Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** AGGREGATFUNKTIONEN, GROUP BY KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 75

► Aufgabenstellung: Having Klausel ▼

- GEBEN SIE FÜR JEDES LAND AUS WIEVIELE ABTEILUNGEN ES DORT GIBT.
- GIBT ES IN EINEM LAND KEINE ABTEILUNGEN SOLL 0 AUSGEZEIGT WERDEN.

► Lösung: Having Klausel ▼

```

1  -- -----
2  -- Loesung: Group By Klausel
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37  .

```



1.4. Subselect

1.Beispiel - Subselect



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, WHERE KLAUSEL, HAVING KLAUSEL, SELECT KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 78, 79, 80

► Aufgabenstellung: Subselect ▼

- GEBEN SIE AN WIEVIEL MITARBEITER, DIE ABTEILUNG MIT DEN MEISTEN MITARBEITERN HAT.
- VERWENDEN SIE DAS SUBSELECT IN DER WHERE, HAVING ODER SELECT KLAUSEL.

► Lösung: Subselect ▼

```

1  -- -----
2  -- Loesung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12  .

```



2.Beispiel - Subselect



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, WHERE KLAUSEL, HAVING KLAUSEL, SELECT KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 78, 79, 80

► Aufgabenstellung: Subselect ▼

- GEBEN SIE DEN NAMEN DER ABTEILUNG MIT DEN MEISTEN MITARBEITERN AN.
- VERWENDEN SIE DAS SUBSELECT IN DER WHERE, HAVING ODER SELECT KLAUSEL.



► Lösung: Subselect ▼

```

1  -- -----
2  -- Loesung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12 .

```



3.Beispiel - Subselect



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, WHERE KLAUSEL, HAVING KLAUSEL, SELECT KLAUSEL
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 78, 79, 80

► Aufgabenstellung: Subselect ▼

- GEBEN SIE DEN COUNTRY_NAME DES LANDES AN, INDEM DIE MEISTEN MITARBEITER ARBEITEN.
- VERWENDEN SIE DAS SUBSELECT IN DER WHERE, HAVING ODER SELECT KLAUSEL.

► Lösung: Subselect ▼

```

1  -- -----
2  -- Loesung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 .

```



4.Beispiel - Subselect



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, WHERE KLAUSEL, HAVING KLAUSEL, SELECT KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 78, 79, 80

► Aufgabenstellung: Subselect ▼

- GEBEN SIE FÜR JEDE ABTEILUNG DEN MITARBEITER MIT DEM HÖCHSTEN EINKOMMEN AN.
- GEBEN SIE FÜR DEN DATENSATZ DEN DEPARTMENT_NAME, FIRST_NAME UND LAST_NAME AUS.
- VERWENDEN SIE DAS SUBSELECT IN DER WHERE, HAVING ODER SELECT KLAUSEL.

► Lösung: Subselect ▼

```

1  -- -----
2  -- Loesung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```



5.Beispiel - Subselect, from Klausel



Beispielbeschreibung ▾

- **Schwerpunkt:** SUBSELECT, FROM KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 81

▸ Aufgabenstellung: Subselect ▾

- GEBEN SIE AN WIEVIELE MITARBEITER, DIE ABTEILUNG MIT DEN MEISTEN MITARBEITERN HAT.
- GEBEN SIE FÜR DEN DATENSATZ DEN `DEPARTMENT_NAME` UND DIE ANZAHL DER MITARBEITER AN..
- SCHREIBEN SIE DIE QUERY OHNE DIE KOMPOSITION VON AGGREGATFUNKTIONEN.

▸ Lösung: Subselect ▾

```

1  -- -----
2  -- Lösung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```



6.Beispiel - Subselect, from Klausel



Beispielbeschreibung ▾

- **Schwerpunkt:** SUBSELECT, FROM KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 81

▸ Aufgabenstellung: Subselect ▾

- GEBEN SIE AN WIEVIELE MITARBEITER, DIE ABTEILUNG MIT DEN MEISTEN MITARBEITERN HAT.
- GEBEN SIE FÜR DEN DATENSATZ DEN `DEPARTMENT_NAME` UND DIE ANZAHL DER MITARBEITER AN..
- SCHREIBEN SIE DIE QUERY OHNE DIE KOMPOSITION VON AGGREGATFUNKTIONEN.

▸ Lösung: Subselect ▾

```

1  -- -----
2  -- Lösung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```



7.Beispiel - Subselect, from Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, FROM KLAUSEL
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 81

► Aufgabenstellung: Subselect ▼

- GEBEN SIE FÜR JEDE ABTEILUNG DEN MITARBEITER MIT DEM HÖCHSTEN EINKOMMEN AN.
- GEBEN SIE FÜR DEN DATENSATZ `DEPARTMENT_NAME`, `FIRST_NAME`, `LAST_NAME` AN.
- VERWENDEN SIE DAS SUBSELECT IN DER FROM KLAUSEL.

► Lösung: Subselect ▼

```

1  -- -----
2  -- Lösung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```



8.Beispiel - Subselect, from Klausel



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, FROM KLAUSEL
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 81

► Aufgabenstellung: Subselect ▼

- GEBEN SIE FÜR JEDES LAND DIE ABTEILUNG MIT DEN MEISTEN MITARBEITERN AN.
- GEBEN SIE FÜR DEN DATENSATZ DEN `COUNTRY_NAME`, `DEPARTMENT_NAME` UND DIE ANZAHL DER MITARBEITER AN.
- VERWENDEN SIE DAS SUBSELECT IN DER FROM KLAUSEL.

► Lösung: Subselect ▼

```

1  -- -----
2  -- Lösung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32 .

```



9.Beispiel - Subselect



Beispielbeschreibung ▼

- **Schwerpunkt:** SUBSELECT, SQL ENGINE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 81

► Aufgabenstellung: Subselect ▼

- BESCHREIBEN SIE IN WELCHER REIHENFOLGE DIE KLAUSELN DER ABFRAGE IN BEISPIEL 7 AUSGEFÜHRT WERDEN.

► Lösung: Subselect ▼

```

1  -- -----
2  -- Loesung: Subselect
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 .

```



1.5. Zeilenfunktionen ▼

1.Beispiel - Datumsfunktionen



Beispielbeschreibung ▼

- **Schwerpunkt:** DATUMSFUNKTIONEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITEN 59 -64

► Aufgabenstellung: Datumsfunktionen ▼

- GEBEN SIE ALLE MITARBEITER AN DIE IM 2.QUARTAL 2005 IM UNTERNEHMEN BESCHÄFTIGT WAREN.
- GEBEN SIE **FIRST_NAME** UND **LAST_NAME** AUS.
- ARBEITEN SIE FÜRS ERSTE NUR MIT DEN DATEN AUS DER **EMPLOYEES** TABELLE.

► Lösung: Datumsfunktionen ▼

```

1  -- -----
2  -- Loesung: Datumsfunktionen
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```



2.Beispiel - Datumsfunktionen



Beispielbeschreibung ▼

- **Schwerpunkt:** DATUMSFUNKTIONEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITEN 59 -64

► Aufgabenstellung: Datumsfunktionen ▼

- GEBEN SIE FÜR ALLE MITARBEITER AN WIEVIELE TAGE SIE IM UNTERNEHMEN BESCHÄFTIGT SIND.
- GEBEN SIE **FIRST_NAME**, **LAST_NAME** UND DIE ANZAHL DER TAGE DER BESCHÄFTIGUNG AUS. GEBEN SIE ZUSÄTZLICH AUS WIEVIELE JAHRE DAS SIND.
- ARBEITEN SIE FÜRS ERSTE NUR MIT DEN DATEN AUS DER **EMPLOYEES** TABELLE.

► Lösung: Datumsfunktionen ▼

```

1  -- -----
2  -- Loesung: Datumsfunktionen
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35  .

```



3.Beispiel - Datumsfunktionen



Beispielbeschreibung ▼

- **Schwerpunkt:** DATUMSFUNKTIONEN, UNION KLAUSEL
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITEN 59 -64

► Aufgabenstellung: Datumsfunktionen ▼

- GEBEN SIE FÜR ALLE MITARBEITER AN WIEVIELE TAGE SIE IM UNTERNEHMEN BESCHÄFTIGT SIND.
- GEBEN SIE **FIRST_NAME**, **LAST_NAME** UND DIE ANZAHL DER TAGE DER BESCHÄFTIGUNG AUS.
- BERÜCKSICHTIGEN SIE EBENFALLS DIE DATEN AUS DER **JOB_HISTORY** TABELLE.

► Lösung: Datumsfunktionen ▼

```

1  -- -----
2  -- Loesung: Datumsfunktionen
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35  .

```



4.Beispiel - Textfunktionen



Beispielbeschreibung ▾

- **Schwerpunkt:** TEXTFUNKTIONEN, FETCH KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITEN 59 -64

▸ Aufgabenstellung: Textfunktionen ▾

- FINDEN SIE ALLE PROJEKTE C.PROJECTS DIE IN IHRER BESCHREIBUNG (DESCRIPTION) DEN AUSDRUCK METHOD ENTHALTEN.
- GEBEN SIE FÜR DIE PROJEKTE JEWEILS DEN TITEL UND DIE BESCHREIBUNG AUS.
- EIN PROJEKT SOLL DABEI AUSGEWÄHLT WERDEN, UNABHÄNGIG DAVON WIE DER METHOD AUSDRUCK GESCHRIEBEN WIRD. Z.B.: METHOD, METHOD, USW.
- GEBEN SIE NUR 20% DER ERGEBNISSE ZURÜCK.

▸ Lösung: Textfunktionen ▾

```

1  -- -----
2  -- Lösung: Textfunktionen
3  -- -----
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30 .

```



5.Beispiel - Textfunktionen



Beispielbeschreibung ▾

- **Schwerpunkt:** TEXTFUNKTIONEN, FETCH KLAUSEL
- **Komplexität:** EINFACH
- **Skriptum:** SEITEN 59 -64

▸ Aufgabenstellung: Textfunktionen ▾

- FÜR EINEN INTERNEN REPORT SOLLEN DIE DATEN DER ANGESTELLTEN ANGEPAST WERDEN.
- GEBEN SIE FÜR ALLE ANGESTELLTEN DEN LAST_NAME UND FIRST_NAME AUS.
- FÜR DIE EMAIL ADRESSE DES ANGESTELLTEN SOLL DER ERSTE BUCHSTABE DES VORNAMENS GETRENNT DURCH EINEN PUNKT GEFOLGT VOM NACHNAMEN ANGEZEIGT WERDEN.
- ALLE BUCHSTABEN DER EMAIL SOLLEN KLEIN GESCHRIEBEN SEIN.
- IMPLEMENTIEREN SIE EINE PAGINIERUNG FÜR DIE ABFRAGE.

▸ Lösung: Textfunktionen ▾

```

1  -- -----
2  -- Lösung: Textfunktionen
3  -- -----
4  --z.b.: Max Musterman -> m.musterman@oracle.com
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27 .

```



6.Beispiel - Numerische Funktionen



Beispielbeschreibung ▼

- **Schwerpunkt:** NUMERISCHE FUNKTIONEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITEN 59 -64

► Aufgabenstellung: Textfunktionen ▼

- GEBEN SIE ALLE ANGESTELLTEN DEN `LAST_NAME`, `FIRST_NAME` UND DAS `SALARY` AUS.
- DIE ANGABE DES GEHALTS SOLL DABEI FORMATIERT SEIN.
- IMPLEMENTIEREN SIE EINE PAGINIERUNG FÜR DIE ABFRAGE.

► Lösung: Numerische Funktionen ▼

```
1  -- -----
2  -- Loesung: Numerische Funktionen
3  -- -----
4  z.B.: 10000 -> 10,000.00
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```



2. XML Schema

01

XML Schema

01. Einfache Datentypen 24

02. Komplexe Datentypen 26

2.1. Einfache Datentypen

1. Beispiel - Einfache Datentypen



Beispielbeschreibung

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253

► Aufgabenstellung: XML Schema schreiben

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.

► Codebeispiel: XML Dokument

```

1  <!-- ----- -->
2  <!--           person.xml          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <person xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="browser.xsd">
8      Max Mustermann
9  </person>

```

► Lösung: XML Schema

```

1  <!-- ----- -->
2  <!--           person.xsd          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
6      xmlns:xs="http://www.w3.org/2001/XMLSchema">
7
8
9  </xs:schema>
10
11
12
13
14
15
16
17
18
19
20  .

```



2.Beispiel - Einfache Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML Schema, EINFACHE DATENTYPEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253

▸ Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE EINEN EINFACHEN XML Schema DATENTYPEN ZUR KLASSIFIZIERUNG VON FARBEN
- DER DATENTYP SOLL SO RESTRIKTIV WIE MÖGLICH SEIN.

▸ Lösung: XML Schema ▾

```

1  <!-- ----- -->
2  <!--           color.xsd           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
6      xmlns:xs="http://www.w3.org/2001/XMLSchema">
7
8
9  </xs:schema>
10
11
12
13  <!-- Farben -->
14  Black, Lightslategray
15  Gray
16  Silver
17  White
18  Ivory
19  Linen
20  Beige
21  Khaki
22  Goldenrod
23  Dardred
24  Maroon
25
26
27
28
29
30
31
32
33
34
35
36
37  .

```



3.Beispiel - Einfache Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML Schema, EINFACHE DATENTYPEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253

▸ Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE EINEN EINFACHEN DATENTYPEN ZUR DARSTELLUNG VON PERSONENNAMEN
- ACHTEN SIE DARAUF DASS EIN PERSONENNAME MINIMAL 2 ZEICHEN UND MAXIMAL 50 ZEICHEN HAT.

▸ Lösung: XML Schema ▾

```

1  <!-- ----- -->
2  <!--           persondata.xsd           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
6      xmlns:xs="http://www.w3.org/2001/XMLSchema">
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37  </xs:schema>

```



2.2. Komplexe Datentypen

1. Beispiel - Komplexe Datentypen



Beispielbeschreibung

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN, SEQUENZ
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253, 254

► Aufgabenstellung: XML Schema schreiben

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.
- DIE ELEMENTE DES XML DOKUMENTS KOMMEN GENAU IN DER REIHENFOLGE VOR WIE SIE IM XML DOKUMENT AUFTRETEN.

► Codebeispiel: XML Dokument

```

1 <!-- ----- -->
2 <!--          book.xml          -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <book xmlns:xs=
6     "http://www.w3.org/2001/XMLSchema-instance"
7     xs:noNamespaceSchemaLocation="book.xsd">
8     <title>Moby Dick</title>
9     <author>Herman Melville</author>
10    <print>5-5-1927</print>
11    <pages>814</pages>
12    <description>
13        Das erzählerische Rueckgrat des Romans ist
14        die schicksalhafte Fahrt des Walfangschiffes
15        Pequod, dessen Kapitaen Ahab mit blindem Hass
16        den weissen Pottwal Moby Dick jagt, der ihm
17        ein Bein abgerissen hat.
18    </description>
19 </book>

```

► Lösung: XML Schema

```

1 <!-- ----- -->
2 <!--          book.xsd          -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <xs:schema
6     xmlns:xs="http://www.w3.org/2001/XMLSchema"
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68 </xs:schema>

```



2.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▼

- **Schwerpunkt:** XML Schema, einfache Datentypen, komplexe Datentypen, Sequenz
- **Komplexität:** einfach
- **Skriptum:** Seite 248-253, 254

► Aufgabenstellung: XML Schema schreiben ▼

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML Schema
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.
- DIE ELEMENTE DES XML DOKUMENTS KOMMEN GENAU IN DER REIHENFOLGE VOR WIE SIE IM XML DOKUMENT AUFTRETEN.

► Codebeispiel: XML Dokument ▼

```

1  <!-- ----- -->
2  <!--          book.xml          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <book xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="book.xsd">
8      <title>Moby Dick</title>
9      <author>
10         <given-name>Herman</given-name>
11         <middle-name>J. L.</middle-name>
12         <last-name>Melville</last-name>
13     </author>
14     <print>5-5-1927</print>
15     <pages>814</pages>
16     <description>
17         Das erzählerische Rückgrat des Romans ist
18         die schicksalhafte Fahrt des Walfangschiffes
19         Pequod, dessen Kapitän Ahab mit blindem Hass
20         den weissen Pottwal Moby Dick jagt, der ihm
21         ein Bein abgerissen hat.
22     </description>
23     <literature>
24         <symbol>
25             Ahabs Jagd auf den Weissen Wal steht im
26             Widerspruch zu den materiellen Interessen
27             von Mannschaft und Eignern. Dem Ersten
28             Maat Starbuck erscheint Ahab's Rachsucht
29             gegen das unvernünftige Tier Moby Dick
30             gotteslästerlich.
31         </symbol>
32     </literature>
33 </book>

```

► Lösung: XML Schema ▼

```

1  <!-- ----- -->
2  <!--          book.xsd          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
6      xmlns:xs="http://www.w3.org/2001/XMLSchema">
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57 </xs:schema>

```



3.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN, SEQUENZ
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253, 254

▸ Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- DIE ELEMENTE DES XML DOKUMENTS KÖNNEN IN BELIEBIGER REIHENFOLGE AUFTRETEN.

▸ Codebeispiel: XML Dokument ▾

```

1 <!-- ----- -->
2 <!--           person.xml           -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <person>
6   <first-name>J.</first-name>
7   <middle-name>R. R.</middle-name>
8   <last-name>Tolkin</last-name>
9 </person>

```

▸ Lösung: XML Schema ▾

```

1 <!-- ----- -->
2 <!--           person.xsd           -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <xs:schema
6   xmlns:xs="http://www.w3.org/2001/XMLSchema"
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 </xs:schema>

```

4.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 248-253, 254

▸ Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.

▸ Codebeispiel: XML Dokument ▾

```

1 <!-- ----- -->
2 <!--           euro.xml           -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <euro xmlns:xs=
6   "http://www.w3.org/2001/XMLSchema-instance"
7   xs:noNamespaceSchemaLocation="euro.xsd">
8   <teams>
9     <team>Schwitzer Team</team>
10    <team>France</team>
11  </teams>
12  <player>Gareth Bale</player>
13  <player>Jmai Vardi</player>
14  <qualifying>
15    <group>
16      <team-play>FRA</team-play>
17      <team-play>SUI</team-play>
18      <news>
19        The last match in Group A ended
20      </news>
21    </group>
22    <group>
23      <team-play>RUS</team-play>
24      <team-play>GER</team-play>
25      <team-play>JAP</team-play>
26    </group>
27  </qualifying>
28 </euro>

```

▸ Lösung: XML Schema ▾

```

1 <!-- ----- -->
2 <!--           euro.xsd           -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <xs:schema
6   xmlns:xs="http://www.w3.org/2001/XMLSchema"

```



6

5.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 248-253, 254

► Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.

► Codebeispiel: XML Dokument ▾

</xs:schema>

```

1  <!-- ----- -->
2  <!--          euro.xml          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <euro year="2016" xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="euro.xsd">
8      <teams>
9          <team shortname="SUI" name="Switzerland">
10             Schwitzer National Team
11          </team>
12          <team shortname="FRA" name="France">
13             France
14          </team>
15      </teams>
16      <player id="0210" name="Gareth Bale">
17          Gareth Bale
18      </player>
19      <player id="0202" name="Jami Vardy">
20          Jmai Vardi
21      </player>
22      <qualifying>
23          <group name="A">
24              <team-play>FRA</team-play>
25              <team-play>SUI</team-play>
26              <news>
27                  The last match in Group A ended
28              </news>
29          </group>
30          <group name="B">
31              <team-play>RUS</team-play>
32              <team-play>GER</team-play>
33              <team-play>JAP</team-play>
34          </group>
35      </qualifying>
36  </euro>

```



► Lösung: XML Schema ▼

```

1  <!-- ----->
2  <!--          euro.xsd          -->
3  <!-- ----->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57  </xs:schema>

```

6.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▼

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 248-253, 254

► Aufgabenstellung: XML Schema schreiben ▼

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.

► Codebeispiel: XML Dokument ▼

```

1  <!-- ----->
2  <!--          report.xml          -->
3  <!-- ----->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <report xmlns:xs=
      "http://www.w3.org/2001/XMLSchema-instance"
      xs:noNamespaceSchemaLocation="report.xsd">
6
7
8      <authors>
9          <author id="author1">
10             <first-name>Arthur</first-name>
11             <middle-name>D.</middle-name>
12             <last-name>Dent</last-name>
13          </author>
14          <author id="author2">
15             <name>Ford Prefect</name>
16             <is-alive>>false</is-alive>
17          </author>
18      </authors>
19      <content>
20          This report is co-authored <authorref
21             id="author1"/> and <authorref
22             id="author2"/> and organized in sections.
23          <section>
24              This section is based on the data
25          </section>
26      </content>
27      <appendix id="app1" title="A">
28          some data
29      </appendix>
30      <appendix id="app2" title="B">
31          even more data
32      </appendix>
33  </report>

```



► Lösung: XML Schema ▼

```

1  <!-- ----->
2  <!--          report.xsd          -->
3  <!-- ----->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57  </xs:schema>

```

7.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▼

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 248-253, 254

► Aufgabenstellung: XML Schema schreiben ▼

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.
- JEDES DER ELEMENTE **OR** UND **AND** ENTHALTEN IMMER 2 ELEMENTE. DAS KÖNNEN BELIEBIGE KOMBINATIONEN DER FOLGENDEN ELEMENTE SEIN **AND**, **OR**, **NOT** UND **TERM**.
- **NOT** ENTHÄLT EIN ELEMENT. FOLGENDE ELEMENTE KÖNNEN IN NOT AUFTRETEN: **OR**, **AND**, **NOT**, **TERM**.

► Codebeispiel: XML Dokument ▼

```

1  <!-- ----->
2  <!--          search.xml          -->
3  <!-- ----->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <search xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="search.xsd">
8      <or>
9          <and>
10             <term>x + 2</term>
11             <or>
12                 <term>y - 1</term>
13                 <not>
14                     <term>k + 3</term>
15                 </not>
16             </or>
17          </and>
18          <term>x + 2</term>
19      </or>
20  </search>

```



3. XML - XPath

03

XPath

01. XPath Grundlagen	32
02. Lokalisierungspfade	34
03. Standardschreibweise	38

3.1. XPath - Grundlagen

1. Beispiel - Knotenbaum



Beispiel ▾

- **Schwerpunkt:** KNOTENBAUMREPRÄSENTATION, DOKUMENTREIHENFOLGE
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 262-264

▸ Aufgabenstellung: Knotenbaumrepräsentation ▾

- ERSTELLEN SIE FÜR DAS FOLGENDEN XML DOKUMENT EINE GRAPHISCHE **Knotenbaumrepräsentation**.
- KENNZEICHNEN SIE ANSCHLIESSEND IN WELCHER **Reihenfolge** DIE KNOTEN VERARBEITET WERDEN.

▸ Codebeispiel: Knotenbaumrepräsentation ▾

```

1  <!-- ----- -->
2  <!--          euro.xml          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <euro year="2016" xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="euro.xsd">
8      <teams>
9          <team shortname="SUI" name="Switzerland">
10             Schwitter National Team
11          </team>
12          <team shortname="FRA" name="France">
13             France
14          </team>
15      </teams>
16      <player id="0210" name="Gareth Bale">
17          Gareth Bale
18      </player>
19      <player id="0202" name="Jami Vardy">
20          Jmai Vardi
21      </player>
22      <qualifying>
23          <group name="A">
24              <team-play>FRA</team-play>
25              <team-play>SUI</team-play>
26              <news>
27                  The last match in Group A ended
28              </news>
29          </group>
30          <group name="B">
31              <team-play>RUS</team-play>
32              <team-play>GER</team-play>
33          </group>
34      </qualifying>
35  </euro>

```



2.Beispiel - Knotenbaum



Beispielbeschreibung ▾

- **Schwerpunkt:** KNOTENBAUMREPRÄSENTATION, DOKUMENTREIHENFOLGE
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 262-264

▸ Aufgabenstellung: Knotenbaumrepräsentation ▾

- ERSTELLEN SIE FÜR DAS FOLGENDEN XML DOKUMENT EINE GRAPHISCHE **Knotenbaumrepräsentation**.
- KENNZEICHNEN SIE ANSCHLIESSEND IN WELCHER **Reihenfolge** DIE KNOTEN VERARBEITET WERDEN.

▸ Codebeispiel: Knotenbaumrepräsentation ▾

```

1 <!-- ----- -->
2 <!--          report.xml          -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <report xmlns:xs=
6     "http://www.w3.org/2001/XMLSchema-instance"
7     xs:noNamespaceSchemaLocation="report.xsd">
8     <authors>
9         <author id="author1">
10             <first-name>Arthur</first-name>
11             <middle-name>D.</middle-name>
12             <last-name>Dent</last-name>
13         </author>
14         <author id="author2">
15             <name>Ford Prefect</name>
16             <is-alive>false</is-alive>
17         </author>
18     </authors>
19     <content>
20         This report is co-authored <authorref
21             id="author1"/> and <authorref
22             id="author2"/> and organized in sections.
23         <section>
24             This section is based on the data
25         </section>
26     </content>
27     <appendix id="app1" title="A">
28         some data
29     </appendix>
30     <appendix id="app2" title="B">
31         even more data
32     </appendix>
33 </report>
34
35
36
37
38 .

```

3.Beispiel - Stringdarstellung



Beispiel ▾

- **Schwerpunkt:** STRINGDARSTELLUNG
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 264

▸ Aufgabenstellung: Stringdarstellung ▾

- GEBEN SIE FÜR DIE FOLGENDEN **Knoten** EINE **Stringdarstellung** AN.

▸ Lösung: Stringdarstellung ▾

```

1 <!-- ----- -->
2 <!--          Stringdarstellung: text()          -->
3 <!-- ----- -->
4 1. Geben Sie eine Stringdarstellung des
5     <first-name> Elements an.
6
7 -- xpath: first-name/text()
8 <first-name>Arthur</first-name>
9
10
11 2. Geben Sie eine Stringdarstellung des
12     <author> Elements an.
13
14 -- xpath: author/text()
15 <author id="author1">
16     <first-name>Arthur</first-name>
17     <middle-name>D.</middle-name>
18     <last-name>Dent</last-name>
19 </author>
20
21
22 3. Geben Sie eine Stringdarstellung des id
23     Attributes an.
24
25 -- xpath: author/@id
26 <author id="author1">
27     <first-name>Arthur</first-name>
28     <middle-name>D.</middle-name>
29     <last-name>Dent</last-name>
30 </author>
31
32
33
34
35
36
37
38
39
40
41
42 .

```

3.2. Lokalisierungspfade

1. Beispiel - Lokalisierungspfade



Beispiel

- **Schwerpunkt:** LOKALISIERUNGSPFADE, RELATIVE UND ABSOLUTE PFADANGABEN
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 265-266

► Aufgabenstellung: Lokalisierungspfade

- DEFINIEREN SIE DIE FOLGENDEN Lokalisierungspfade.

► Codebeispiel: project.xml

```

1 <!-- ----- -->
2 <!--      project.xml      -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <project id="343225">
6   <name>Simulation</name>
7   <description>
8     AI is the Buzz Word in modern application
9   </description>
10  <partners>
11    <partner>
12      <name>MIT</name>
13      <partner-role>
14        research-partner
15      </partner-role>
16      <project-manager>
17        Dr. Aron Gupta
18      </project-manager>
19    </partner>
20  </partners>
21  <subprojects>
22    <subproject id="342566" code="simulation">
23      <name>Computersimulation</name>
24      <started-at>21.12.2014</started-at>
25      <institute>
26        Angewandte Mathematik
27      </institute>
28      <stuff>
29        <project-manager>
30          <first-name>
31            Christian Karl
32          </first-name>
33          <last-name>
34            Lauer
35          </last-name>
36        </project-manager>
37        <member>
38          <first-name>
39            Mika
40          </first-name>
41          <last-name>

```

```

42      Harkonnen
43    </last-name>
44  </member>
45 </member>
46   <first-name>
47     Stefan
48   </first-name>
49   <last-name>
50     Jell
51   </last-name>
52 </member>
53 </stuff>
54 </subproject>
55 </subprojects>
56 </project>

```

► Lösung: XPath Ausdrücke

```

1 <!-- ----- -->
2 <!--      Lokalisierungspfade      -->
3 <!-- ----- -->
4 1. Finden Sie alle <subproject> Elemente
5   Kontext: Wurzelknoten
6
7
8
9 2. Finden Sie alle <subproject> Elemente
10  Kontext: subprojects
11
12
13
14 3. Finden Sie alle id Attribute der <subproject>
15  Elemente.
16  Kontext: Wurzelknoten
17
18
19
20 4. Finden Sie alle Mitarbeiter der Subprojekte.
21  Kontext: Wurzelknoten
22
23
24
25 5. Finden Sie die Namen aller Mitarbeiter der
26  Subprojekte.
27  Kontext: project
28
29
30
31 6. Finden Sie die Namen aller Mitarbeiter der
32  Subprojekte.
33  Kontext: partners
34
35
36
37 7. Finden Sie alle Attribute der Subprojekte
38  Kontext: Wurzelknoten
39
40
41 .

```



2. Beispiel - Lokalisierungspfade



Beispielbeschreibung ▾

- **Schwerpunkt:** LOKALISIERUNGSPFADE
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 265 - 266

▸ Aufgabenstellung: Lokalisierungspfade ▾

- DEFINIEREN SIE DIE FOLGENDEN **Lokalisierungspfade**.
- DIE EINGABEDATEI IST DIE **EURO.XML** DATEI.

▸ Lösung: XPath Ausdrücke ▾

```

1  <!-- ----- -->
2  <!--          euro.xml          -->
3  <!-- ----- -->
4  1. Alle <nation> Elementknoten
5     Kontext: Wurzelknoten
6
7
8
9  2. Alle <nation> Elementknoten
10    Kontext: <winner> Element
11
12
13
14  3. Alle <last-name> Elementknoten der <player>
15     Knoten
16     Kontext: Wurzelknoten
17
18
19
20  4. Alle <last-name> Elementknoten der <player>
21     Knoten
22     Kontext: <winner> Element
23
24
25
26  5. Alle <short-name> Elementknoten der <team>
27     Knoten
28     Kontext: Wurzelknoten
29
30
31
32  6. Alle <short-name> Elementknoten der <team>
33     Knoten
34     Kontext: <winner> Element
35
36
37
38  7. Alle id Attribute der Gruppen
39     Kontext: Wurzelknoten
40
41  .

```



3. Beispiel - Lösungsobjekte



Beispiel ▾

- **Schwerpunkt:** LÖSUNGSOBJEKTE
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 267

▸ Aufgabenstellung: Lösungsobjekte ▾

- GEBEN SIE FÜR JEDEN DER FOLGENDEN XPATH DEN TYP DES LÖSUNGSOBJEKTS AN.

▸ Lösung: Lösungsobjekte ▾

```

1  <!-- ----- -->
2  <!--          project.xml          -->
3  <!-- ----- -->
4  1. Finden Sie alle <subproject> Elemente
5     Kontext: Wurzelknoten
6
7
8
9  2. Finden Sie alle <subproject> Elemente
10    Kontext: subprojects
11
12
13
14  3. Finden Sie alle id Attribute der <subproject>
15     Elemente.
16     Kontext: Wurzelknoten
17
18
19
20  4. Finden Sie alle Mitarbeiter der Subprojekte.
21     Kontext: Wurzelknoten
22
23
24
25  5. Finden Sie die Namen aller Mitarbeiter der
26     Subprojekte.
27     Kontext: project
28
29
30
31  6. Finden Sie die Namen aller Mitarbeiter der
32     Subprojekte.
33     Kontext: partners
34
35
36
37  7. Finden Sie alle Attribute der Subprojekte
38     Kontext: Wurzelknoten
39
40
41  .
42  .

```



4.Beispiel - Prädikate



Beispiel ▾

- **Schwerpunkt:** PRÄDIKATE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 267

▸ Aufgabenstellung: Prädikate ▾

- DEFINIEREN SIE DIE FOLGENDEN **Lokalisierungspfade**.

▸ Lösung: Prädikate ▾

```

1 <!-- ----- -->
2 <!--           project.xml           -->
3 <!-- ----- -->
4 1. Finden Sie alle <project> Elemente mit dem
5    <name> 'Simulation'.
6    Kontext: Wurzelknoten
7
8
9 2. Finden Sie alle <subproject> Elemente mit der
10   id 342566.
11   Kontext: Wurzelknoten
12
13
14 3. Finden Sie alle <subproject> Elemente die
15   nicht die id 342566 haben.
16   Kontext: Wurzelknoten
17
18
19 4. Finden Sie alle <partner> die die Partnerrolle
20   'research-partner' haben.
21   Kontext: Wurzelknoten
22
23
24 5. Finden Sie alle <subproject> Elemente die ein
25   <description> Element haben.
26   Kontext: Wurzelknoten
27
28
29 6. Finden Sie das erste <subproject> mit dem code
30   'simulation'.
31   Kontext: Wurzelknoten
32
33
34 7. Finden den letzten Projektmitarbeiter für das
35   <subproject> mit der id 342566
36   Kontext: Wurzelknoten
37
38
39 8. Finden Sie die <description> Beschreibung aller
40   <project> Projekte die ein <subproject>
41   Subprojekt mit dem namen Computersimulation
42   enthalten.
43 .

```



5.Beispiel - Prädikate



Beispielbeschreibung ▾

- **Schwerpunkt:** LOKALISIERUNGSPFADE, PRÄDIKATE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 262-267

▸ Aufgabenstellung: Prädikate ▾

- DEFINIEREN SIE DIE FOLGENDEN **Lokalisierungspfade**.

▸ Codebeispiel: XPath Ausdrücke ▾

```

1 <!-- ----- -->
2 <!--           euro.xml           -->
3 <!-- ----- -->
4 1. Finden Sie die Nachnamen aller französischen
5   Abwehrspieler.
6   Kontext: Wurzelknoten
7
8
9
10 2. Finden Sie die Namen der Klubs in denen die
11   spanischen Stürmer spielen.
12   Kontext: Wurzelknoten
13
14
15
16
17 3. Finden Sie die Namen aller Spieler die im Sturm
18   spielen
19   Kontext: Wurzelknoten
20
21
22
23 4. Finden Sie die <name> Elemente aller <team>
24   Teams die unter den ersten 8 sind.
25   Kontext: Wurzelknoten
26
27
28
29 5. Geben Sie die Anzahl aller Spieler an, die an
30   der Euro teilnehmen.
31
32
33
34 6. Geben Sie die Anzahl aller französischen
35   Abwehrspieler an.
36
37
38
39 7. Geben Sie für jede Position der spanischen
40   Mannschaft den ersten Spieler aus.
41
42 .

```



6.Beispiel - Lokalisierungspfade



Beispielbeschreibung ▾

- **Schwerpunkt:** LOKALISIERUNGSPFADE
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 262-267

▸ Aufgabenstellung: Lokalisierungspfade ▾

- DEFINIEREN SIE DIE FOLGENDEN **Lokalisierungspfade**.

▸ Codebeispiel: XML Dokument ▾

```

1  <!-- ----- -->
2  <!--      travelAgency.xml      -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <travelAgency>
6    <sportcamps>
7      <summercamp id="c1">
8        <name>Fun Diving</name>
9        <venue id="v1">
10         Krk, <country>Croatia</country>
11       </venue>
12       <sport>Scuba Diving</sport>
13       <duration>7</duration>
14       <date>5.7.2018</date>
15       <date>5.8.2018</date>
16     </summercamp>
17     <summercamp id="c2">
18       <name>Biking with kids</name>
19       <venue id="v2">
20         Hinterglemm, Austria
21       </venue>
22       <sport>Mountain Biking</sport>
23       <sport>Hiking</sport>
24       <duration>6</duration>
25       <date>15.7.2018</date>
26     </summercamp>
27     <summercamp id="c3">
28       <name>Big Waves Tour</name>
29       <venue id="v3">
30         Honolulu, Hawaii
31       </venue>
32       <sport>Surfing</sport>
33       <sport>Sailing</sport>
34       <duration>14</duration>
35       <date>1.8.2018</date>
36     </summercamp>
37     <wintercamp id="c4">
38       <name>Advanced Skiing</name>
39       <venue id="v2">
40         Hinterglemm, Austria
41       </venue>
42       <sport>Skiing</sport>
43       <sport>Snow Board</sport>
44       <duration>7</duration>
45     </wintercamp>

```

```

46     <summercamp id="c5">
47       <name>Sailing for beginners</name>
48       <venue id="v4">
49         Rust, <country>Austria</country>
50       </venue>
51       <sport>Sailing</sport>
52       <duration>5</duration>
53       <date>15.8.2018</date>
54     </summercamp>
55   </sportcamps>
56   <customers>
57     <customer id="p1">
58       Sophie Haas
59     </customer>
60     <customer id="p2">
61       Samuel Mumm
62     </customer>
63     <customer id="p3">
64       Arthur Dent
65     </customer>
66     <customer id="p4">
67       Harvey Dent
68     </customer>
69     <customer id="p5">
70       Miles Brien
71     </customer>
72   </customers>
73   <bookings>
74     <booking customerid="p2" campid="c2"
75       date="15.7.2018"/>
76     <booking customerid="p3" campid="c4"
77       date="6.1.2017"/>
78     <booking customerid="p1" campid="c1"
79       date="5.7.2018"/>
80     <booking customerid="p1" campid="c3"
81       date="5.8.2016"/>
82     <booking customerid="p4" campid="c5" />
83   </bookings>
84 </travelAgency>

```

▸ Codebeispiel: XPath Ausdrücke ▾

```

1  <!-- ----- -->
2  <!--      Lokalisierungspfade      -->
3  <!-- ----- -->
4  1. Geben Sie die Namen aller Sportcamps aus die
5     bereits gebucht worden sind.
6
7
8
9
10
11 2. Geben Sie das erste sportcamp aus fuer das kein
12   Buchungsdatum bekannt ist.
13
14
15
16
17
18 .

```



3.3. Standardpfadangaben

1.Beispiel - Standardschreibweise



Beispiel ▾

- **Schwerpunkt:** LOKALISIERUNGSPFADE, STANDARDSCHREIBWEISE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 268-270

▸ Aufgabenstellung: Lokalisierungspfade ▾

- SCHREIBEN SIE DIE FOLGENDEN XPATH AUSDRÜCKE IN DER STANDARDSCHREIBWEISE.

▸ Lösung: Standardschreibweise ▾

```

1 <!-- ----- -->
2 <!--           project.xml           -->
3 <!-- ----- -->
4 1. Finden Sie alle <project> Elemente mit dem
5    <name> 'Simulation'.
6    Kontext: Wurzelknoten
7
8
9 2. Finden Sie alle <subproject> Elemente mit der
10   id 342566.
11   Kontext: Wurzelknoten
12
13
14 3. Finden Sie alle <subproject> Elemente die
15   nicht die id 342566 haben.
16   Kontext: Wurzelknoten
17
18
19 4. Finden Sie alle <partner> die die Partnerrolle
20   'research-partner' haben.
21   Kontext: Wurzelknoten
22
23
24 5. Finden Sie alle <subproject> Elemente die ein
25   <description> Element haben.
26   Kontext: Wurzelknoten
27
28
29 6. Finden Sie das erste <subproject> mit dem code
30   'simulation'.
31   Kontext: Wurzelknoten
32
33
34 7. Finden den letzten Projektmitarbeiter fr das
35   <subproject> mit der id 342566
36   Kontext: Wurzelknoten

```



2.Beispiel - Standardschreibweise



Beispiel ▾

- **Schwerpunkt:** LOKALISIERUNGSPFADE, STANDARDSCHREIBWEISE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 268-270

▸ Aufgabenstellung: Lokalisierungspfade ▾

- DEFINIEREN SIE DIE FOLGENDEN Lokalisierungspfade.

▸ Codebeispiel: Standardschreibweise ▾

```

1 <!-- ----- -->
2 <!--           family.xml           -->
3 <!-- ----- -->
4 1. Geben Sie alle maenlichen Nachkommen von John
5   Washington aus.
6
7
8
9
10
11
12
13
14 2. Geben Sie alle Nachkommen von John Washington
15   aus.
16
17
18
19
20
21
22
23
24 3. Geben Sie alle maenliche Vorfahren von George
25   Washington aus.
26
27
28
29
30
31
32
33
34 4. Geben Sie alle Grosseltern von George Washington
35   aus.
36
37
38
39
40
41 .

```



3.Beispiel - Standardschreibweise



Beispielbeschreibung ▼

- **Schwerpunkt:** LOKALISIERUNGSPFADE
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 268 - 270

► Aufgabenstellung: Lokalisierungspfade ▼

- DEFINIEREN SIE FOLGENDEN LOKALISIERUNGSPFADE.

► Codebeispiel: XML Dokument ▼

```
1  <!-- ----- -->
2  <!--          euro.xml          -->
3  <!-- ----- -->
4  1. Geben Sie fr das deutsche Team die Anzahl der
5     Spiele nach der Gruppenphase an.
6
7
8
9  2. Geben Sie alle Gegner des portugisichen Teams
10     in der Gruppenphase aus.
11
12
13
14  3. Geben Sie alle Gegner des portugisichen Teams
15     nach der Gruppenphase aus.
16
17
18  4. Geben Sie die hoechste Gruppen id aus. Sie
19     koenen nicht davon ausgehen das die Gruppe mit
20     der hoechsten Gruppe die letzte Gruppe ist.
21  .
```



4. XML - XSLT

04

XSLT

01. Deklarative Programmierung 40

02. Deklarative Programmierung 44

4.1. Deklarative Templates

1.Beispiel - Transformationsprozesse



Beispiel ▾

- **Schwerpunkt:** XSLT, `<XSL:APPLY-TEMPLATES>`, `<XSL:VALUE-OF>`, `<XSL:ATTRIBUTE>`
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 278, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER `EURO.XML` EINGABE-DATEI ZU ERZEUGEN.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           teams.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <teams>
6      <team code="SPA">
7          <player>Jordi Alba</player>
8          <player>Dani Carvajal</player>
9          <player>...</player>
10         <player>Aritz Aduriz</player>
11     </team>
12     ...
13
14     <team code="GER">
15         <player>Jerome Botang</player>
16         <player>...</player>
17         <player>Toni Kroos</player>
18     </team>
19 </teams>
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36 .

```



2.Beispiel - Transformationsprozesse



Beispiel ▾

- **Schwerpunkt:** XSLT, <XSL:APPLY-TEMPLATES>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 278, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABE-DATEI ZU ERZEUGEN.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           finals.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <finals>
6
7      <team code="POR">
8          <player>Raphael Varane</player>
9          ...
10     </team>
11
12     <team code="FRA">
13         <player>Luis Neto</player>
14         ...
15     </team>
16
17 </finals>
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39 .

```



3.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:APPLY-TEMPLATES>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:PARAM>, <XSL:WITH-PARAM>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 278, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABE-DATEI ZU ERZEUGEN.
- PROGRAMMIEREN SIE EIN EINZELNES TEMPLATE ZUM VERARBEITEN DER <TEAM> ELEMENTE.
- PROGRAMMIEREN SIE EIN EINZELNES TEMPLATE ZUM VERARBEITEN DER <PLAYER> ELEMENTE.
- **Hinweis:** DEFINIEREN SIE AM 3 PARAMETER ATTACK, DEFENSE, PLAYGROUND AUF STYLESHEET EBENE.

▸ Codebeispiel: Stylesheet ▾

```

1  <!-- ----- -->
2  <!--           positions.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xsl:stylesheet
6      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
7      version="2.0">
8
9      <xsl:param name="defense" select="'defense'"/>
10     <xsl:param name="playground"
11         select="'playground'"/>
12     <xsl:param name="attack" select="'attack'"/>
13
14     ...
15
16 </xsl:stylesheet>

```

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           positions.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <positions>
6      <position type="defense">
7          <team code="FRA">
8              <player>Laurent Koscielny</player>
9              ...
10         </team>
11         ...
12     <team code="NIR">

```

```

13     <player>Pat Jennings</player>
14     ...
15 </team>
16 </position>
17
18 <position type="playground">
19     <team code="FRA">
20         <player>Paul Pogba</player>
21         ...
22     </team>
23     ...
24 </position>
25
26 <position type="attack">
27     <team code="FRA">
28         <player>Antoine Griezmann</player>
29         ...
30     </team>
31     <team code="NIR">
32         <player>David Healy</player>
33         ...
34     </team>
35 </position>
36 </positions>

```

4. Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:APPLY-TEMPLATES>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:PARAM>, <XSL:WITH-PARAM>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 278, 284

► Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER EURO.XML EINGABE-DATEI ZU ERZEUGEN.
- FÜR JEDE MANSCHAFT SOLL ANGEGEBEN WERDEN BIS ZU WELCHER RUNDE SIE GEKOMMEN IST.
- DIE GRUPPENPHASE WIRD ALS 1 RUNDE GERECHNET.
- **Hinweise** VERWENDEN SIE DIE COUNT FUNKTION UM DIE ANZAHL VON ELEMENTEN IN EINEM **Nodeset** ZU ZÄHLEN.



► Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           result.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <result>
6
7      <group id="1">
8          <team code="FRA" round="5"/>
9          <team code="SPA" round="2"/>
10         <team code="GER" round="4"/>
11         <team code="GBS" round="2"/>
12         <team code="POR" round="5"/>
13         <team code="BEL" round="3"/>
14     </group>
15     ...
16
17     <group id="4">
18         <team code="TUR" round="1"/>
19         <team code="IRL" round="2"/>
20         <team code="ISL" round="3"/>
21         <team code="WAL" round="4"/>
22         <team code="ALB" round=""/>
23         <team code="NIR" round="2"/>
24     </group>
25
26 </result>
27
28
29
30 .

```



5. Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:APPLY-TEMPLATES>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:PARAM>, <XSL:WITH-PARAM>, <XSL:ELEMENT>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 278, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABE-DATEI ZU ERZEUGEN.
- PROGRAMMIEREN SIE SO WENIGE TEMPLATES WIE MÖGLICH. 2 TEMPLATES SIND AUSREICHEND.
- **Hinweis:** VERWENDEN SIE DAS **XSL:ELEMENT** UM DYNAMISCH ELEMENTE ZU ERZEUGEN.

▸ Codebeispiel: Ausgabedatei ▾

```

1 <!-- ----- -->
2 <!--          report.xml          -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <report>
6   <last-sixteen>
7     <sixteen code="SUI">
8       <name>Switzerland</name>
9     </sixteen>
10    ...
11  </last-sixteen>
12  <last-eight>
13    <eight code='PLA'>
14      <name>.. </name>
15    </eight>
16  </last-eight>
17  <last-four>
18    <four code="POR">
19      <name>... </name>
20    </four>
21  </last-four>
22  <last-two>
23    <two code="POR">
24      <name> ... </name>
25    </two>
26  </last-two>
27 </report>
28
29
30
31 .

```



6. Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:APPLY-TEMPLATES>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:PARAM>, <XSL:WITH-PARAM>, <XSL:ELEMENT>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 278, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABE-DATEI ZU ERZEUGEN.
- **Hinweis:** VERWENDEN SIE DAS **MODE** ATTRIBUT.

▸ Codebeispiel: Ausgabedatei ▾

```

1 <!-- ----- -->
2 <!--          euro-finals.xml          -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <euro-finals>
6   <team group="1" code="FRA">
7     <name>France</name>
8     <opponents>
9       <team code="SPA"/>
10      <team code="POR"/>
11      <team code="BEL"/>
12      <team code="IRL"/>
13      ...
14      <team code="GER"/>
15    </opponents>
16  </team>
17  <team group="1" code="SPA">
18    ...
19    <team group="1" code="GBS"/>
20    <team group="1" code="POR">
21      <name>Portugal</name>
22      <opponents>
23        <team code="FRA"/>
24        <team code="GBS"/>
25        <team code="BEL"/>
26        <team code="CRO"/>
27        ...
28        <team code="WAL"/>
29      </opponents>
30    </team>
31    <team group="1" code="BEL">
32      ...
33    <team group="4" code="NIR"/>
34
35  </euro-finals>
36
37 .

```



7.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:APPLY-TEMPLATES>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:PARAM>, <XSL:WITH-PARAM>, <XSL:ELEMENT>
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 278, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER FAMILY.XML EINGABEDATEI ZU ERZEUGEN.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           children.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <children>
6      <person>
7          <name>John Washington</name>
8          <child>
9              <person>
10                 <name>Lawrence Washington</name>
11                 <child>
12                     <name>August Washington</name>
13                     ...
14                 </child>
15             </person>
16         </child>
17     </person>
18     <person>
19         <name>Anne Pope</name>
20         <child>
21             ...
22         </person>
23 </children>
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 .

```

4.2. Prozedurale Programmierung ▾

1.Beispiel - Transformationsprozesse



Beispiel ▾

- **Schwerpunkt:** XSLT, <XSL:FOR-EACH>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 279 - 281, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER EURO.XML EINGABEDATEI ZU ERZEUGEN.
- VERWENDEN SIE PROZEDURALE UND DEKLARATIVE ELEMENTE FÜR EINE SINNVOLLE VERARBEITUNG.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           teams.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <teams>
6      <team code="SPA">
7          <player>Jordi Alba</player>
8          <player>Dani Carvajal</player>
9          <player>...</player>
10         <player>Aritz Aduriz</player>
11     </team>
12     ...
13
14     <team code="GER">
15         <player>Jerome Botang</player>
16         <player>...</player>
17         <player>Toni Kroos</player>
18     </team>
19 </teams>
20
21
22
23
24
25
26
27
28
29
30
31
32
33 .

```

2.Beispiel - Transformationsprozesse



Beispiel ▾

- **Schwerpunkt:** XSLT, `<XSL:FOR-EACH>`, `<XSL:VALUE-OF>`, `<XSL:ATTRIBUTE>`, `<XSL:VARIABLE>`
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 279 - 281, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABEDATEI ZU ERZEUGEN.
- VERWENDEN SIE PROZEDURALE UND DEKLARATIVE ELEMENTE FÜR EINE SINNVOLLE VERARBEITUNG.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           finals.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <finals>
6
7      <team code="POR">
8          <player>Raphael Varane</player>
9          ...
10     </team>
11
12     <team code="FRA">
13         <player>Luis Neto</player>
14         ...
15     </team>
16
17 </finals>
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35 .

```



3.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, `<XSL:FOR-EACH>`, `<XSL:VALUE-OF>`, `<XSL:ATTRIBUTE>`, `<XSL:VARIABLE>`
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 279 - 281, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABEDATEI ZU ERZEUGEN.
- VERWENDEN SIE PROZEDURALE UND DEKLARATIVE ELEMENTE FÜR EINE SINNVOLLE VERARBEITUNG.
- **Hinweis:** DEFINIEREN SIE AM 3 VARIABLEN **ATTACK**, **DEFENSE**, **PLAYGROUND** AUF STYLESHEET EBENE.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           positions.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <positions>
6
7      <position type="defense">
8          <team code="FRA">
9              <player>Laurent Koscielny</player>
10             ...
11         </team>
12         ...
13         <team code="NIR">
14             <player>Pat Jennings</player>
15             ...
16         </team>
17     </position>
18
19     <position type="playground">
20         ...
21     </position>
22
23     <position type="attack">
24         <team code="FRA">
25             <player>Antoine Griezmann</player>
26             ...
27         </team>
28         <team code="NIR">
29             <player>David Healy</player>
30             ...
31         </team>
32     </position>
33 </positions>

```



4.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:FOR-EACH>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:VARIABLE>, <XSL:IF>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 279 - 281, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABEDATEI ZU ERZEUGEN.
- FÜR JEDE MANSCHAFT SOLL ANGEGEBEN WERDEN BIS ZU WELCHER RUNDE SIE GEKOMMEN IST.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           result.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <result>
6      <group id="1">
7          <team code="FRA" round="5"/>
8          <team code="SPA" round="2"/>
9          <team code="GER" round="4"/>
10         <team code="GBS" round="2"/>
11         <team code="POR" round="5"/>
12         <team code="BEL" round="3"/>
13     </group>
14     <group id="2">
15         <team code="ITA" round="3"/>
16         <team code="RUS" round="1"/>
17         <team code="SUI" round="2"/>
18         <team code="AUT" round="1"/>
19         <team code="CRO" round="2"/>
20         <team code="UKR" round="1"/>
21     </group>
22     <group id="3">
23         <team code="CZE" round="1"/>
24         <team code="SWE" round="1"/>
25         <team code="PLA" round="3"/>
26         <team code="ROU" round="1"/>
27         <team code="SVK" round="2"/>
28         <team code="HUN" round="2"/>
29     </group>
30     <group id="4">
31         <team code="TUR" round="1"/>
32         <team code="IRL" round="2"/>
33         <team code="ISL" round="3"/>
34         <team code="WAL" round="4"/>
35         <team code="ALB" round="1"/>
36         <team code="NIR" round="2"/>
37     </group>
38 </result>

```

5.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:FOR-EACH>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:VARIABLE>, <XSL:IF>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 279 - 281, 284

▸ Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABEDATEI ZU ERZEUGEN.
- VERWENDEN SIE PROZEDURALE UND DEKLARATIVE ELEMENTE FÜR EINE SINNVOLLE VERARBEITUNG.

▸ Codebeispiel: Ausgabedatei ▾

```

1  <!-- ----- -->
2  <!--           report.xml           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <report>
6      <last-sixteen>
7          <sixteen code="SUI">
8              <name>Switzerland</name>
9          </sixteen>
10         ...
11     </last-sixteen>
12     <last-eight>
13         <eight code="PLA">
14             <name>.. </name>
15         </eight>
16     </last-eight>
17     <last-four>
18         <four code="POR">
19             <name>... </name>
20         </four>
21     </last-four>
22     <last-two>
23         <two code="POR">
24             <name> ... </name>
25         </two>
26     </last-two>
27 </report>
28
29
30
31
32
33
34
35
36 .

```



6.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:FOR-EACH>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:VARIABLE>, <XSL:IF>
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 279 - 281, 284

► Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **EURO.XML** EINGABEDATEI ZU ERZEUGEN.
- VERWENDEN SIE PROZEDURALE UND DEKLARATIVE ELEMENTE FÜR EINE SINNVOLLE VERARBEITUNG.
- **Hinweis:** VERWENDEN SIE DAS **MODE** ATTRIBUT.

► Codebeispiel: Ausgabedatei ▾

```

1 <!-- ----- -->
2 <!--      euro-finals.xml      -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <euro-finals>
6   <team group="1" code="FRA">
7     <name>France</name>
8     <opponents>
9       <team code="SPA"/>
10      <team code="POR"/>
11      <team code="BEL"/>
12      <team code="IRL"/>
13      ...
14      <team code="GER"/>
15    </opponents>
16  </team>
17  <team group="1" code="SPA"/>
18  ...
19  <team group="1" code="GBS"/>
20  <team group="1" code="POR">
21    <name>Portugal</name>
22    <opponents>
23      <team code="FRA"/>
24      <team code="GBS"/>
25      <team code="BEL"/>
26      <team code="CRO"/>
27      ...
28      <team code="WAL"/>
29    </opponents>
30  </team>
31  <team group="1" code="BEL"/>
32  ...
33  <team group="4" code="NIR"/>
34 </euro-finals>

```



7.Beispiel - Transformationsprozesse



Beispielbeschreibung ▾

- **Schwerpunkt:** XSLT, <XSL:FOR-EACH>, <XSL:VALUE-OF>, <XSL:ATTRIBUTE>, <XSL:VARIABLE>, <XSL:IF>
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 279 - 281, 284

► Aufgabenstellung: XSLT Stylesheet ▾

- SCHREIBEN SIE EIN **XSLT Stylesheet** UM FOLGENDE XML DATEI ALS OUTPUT AUS DER **FAMILY.XML** EINGABEDATEI ZU ERZEUGEN.
- VERWENDEN SIE PROZEDURALE UND DEKLARATIVE ELEMENTE FÜR EINE SINNVOLLE VERARBEITUNG.

► Codebeispiel: Ausgabedatei ▾

```

1 <!-- ----- -->
2 <!--      children.xml      -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <children>
6   <person>
7     <name>John Washington</name>
8     <child>
9       <person>
10        <name>Lawrence Washington</name>
11        <child>
12          <name>August Washington</name>
13          ...
14          <child>
15            </person>
16          </child>
17        </person>
18      <person>
19        <name>Anne Pope</name>
20        <child>
21          ...
22          </person>
23        </child>
24      </person>
25      ...
26      ...
27      ...
28      ...
29      ...
30      ...
31      ...
32      ...
33      ...
34      ...
35      ...

```



5. NoSQL - Informationssysteme

01

SQL Grundlagen

01. NoSQL Informationssysteme 48

02. Konsistenzmodelle 49

5.1. NoSQL Informationssysteme

1. Beispiel - Sql vs. NoSql System



Beispielbeschreibung

- **Schwerpunkt:** SQL INFORMATIONSSYSTEME, NoSQL INFORMATIONSSYSTEME
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 166 - 169

► Aufgabenstellung: Thorieklausur

- GEBEN SIE 3 KONKRETE **Beispielanwendungen** FÜR SQL DATENBANKEN AN. ARGUMENTIEREN SIE WARUM SIE FÜR DIE ANGEgebenEN SZENARIEN EIN SQL SYSTEM EINSETZEN WÜRDEN.
- GEBEN SIE NUN 3 KONKRETE **Beispielanwendungen** FÜR NoSQL DATENBANKEN AN. ARGUMENTIEREN SIE WARUM SIE FÜR DIE ANGEgebenEN SZENARIEN EIN NoSQL SYSTEM EINSETZEN WÜRDEN.



2. Beispiel - CAP Theorem



Beispielbeschreibung

- **Schwerpunkt:** CAP THEOREM
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 170 - 173

► Aufgabenstellung: Thorieklausur

- GEBEN SIE JEWEILS EIN BEISPIEL FÜR EIN CP, AP UND CA SYSTEM. ERKLÄREN SIE WARUM DIE JEWEILIGEN INFORMATIONSSYSTEM DER ENTSPRECHENDEN KATEGORIE ZUGEORDNET WERDEN.
- VERWENDEN SIE KEINE BEISPIELE AUS DEM VORLESUNGSSKRIPTUM.



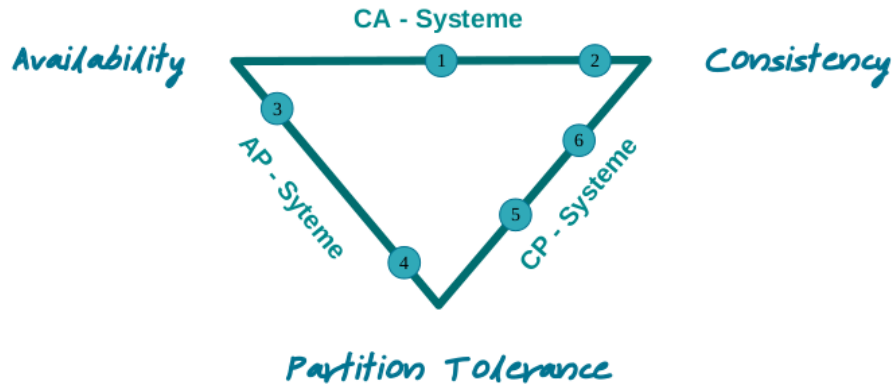


Abbildung 1. Informationssysteme

3. Beispiel - CAP Theorem



Beispielbeschreibung

- **Schwerpunkt:** CAP THEOREM
- **Komplexität:** KOMPLEX
- **Skriptum:** SEITE 170 - 173

► Aufgabenstellung: Thorioklausur

- BESCHREIBEN SIE DIE EIGENSCHAFTEN JEDES DER IN DER CAP THEOREM GRAFIK EINGEZEICHNETEN INFORMATIONSSYSTEMS.
- FINDEN SIE FÜR JEDES EINGEZEICHNETE INFORMATIONSSYSTEM EINE KONKRETE AUSPRÄGUNG.

Zur einfacheren Recherche wird eine Liste von einschlägigen Informationssystemen angegeben.

► Auflistung: Informationssysteme

- MYSQL, POSTGRE SQL, ORACLE DATABASE
- APACHE CASSANDRA, GOOGLE BIG TABLE, MONGODB
- REDIS, APACHE SPARK, APACHE HADOOP
- APACHE HBASE, APACHE PIG, COUCHBASE SERVER
- EXISTDB

5.2. Konsistenzmodelle

1. Beispiel - Transaktion

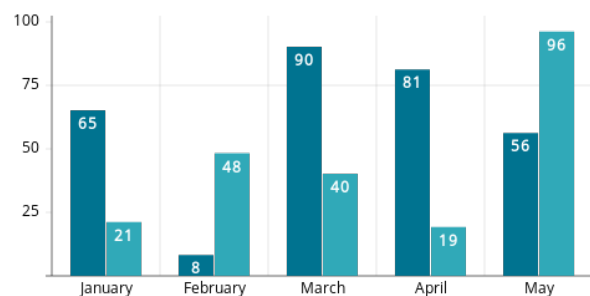


Beispielbeschreibung

- **Schwerpunkt:** TRANSAKTION
- **Komplexität:** EINFACHE
- **Skriptum:** SEITE 174 - 175

► Aufgabenstellung: Thorioklausur

- GEBEN SIE EIN BEISPIEL AUS DEM ALLTAG BEI DEM IN INFORMATIONSSYSTEMEN TRANSAKTIONEN VERWENDET WERDEN.
- ERKLÄREN SIE WARUM IN DEM VON IHNEN BESCHRIEBENEM ANWENDUNGSFALL DAS FEHLEN EINES TRANSAKTIONSMODELS ZU EINEM FEHLER FÜHREN WÜRD.
- ZEICHNEN SIE EIN ANWENDUNGSFALLDIAGRAMM FÜR IHRE SZENARIO.



2.Beispiel - Konsistenzmodelle



Beispielbeschreibung ▾

- **Schwerpunkt:** KONSISTENZMODELLE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 174

▸ Aufgabenstellung: Thorieklausur ▾

- FINDEN SIE JEWEILS 3 ANWENDUNGSFÄLLE IN DENEN **Eventual Consistency** FÜR DIE VERARBEITUNG VON DATEN IN EINER ANWENDUNG AUSREICHEN IST.
- FINDEN SIE JEWEILS 3 ANWENDUNGSFÄLLE IN DENEN FÜR DIE VERARBEITUNG VON DATEN IN EINER ANWENDUNG **Strict Consistency** ZWINGEND NOTWENDIG IST.



3.Beispiel - Sperren



Beispielbeschreibung ▾

- **Schwerpunkt:** KONSISTENZMODELLE
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 174

▸ Aufgabenstellung: Thorieklausur ▾

- FÜR DIE IMPLEMENTIERUNG DES **Isolationskriteriums** WERDEN IN INFORMATIONSSYSTEMEN SPERREN EINGESETZT. ÜBERLEGEN SIE DEN EFFIZIENTEN EINSATZ VON SPERREN BEI SCHREIB- BZW. LESEZUGRIFFEN AUF DATEN IN DER DATENBANK.
- GEBEN SIE EINE GRAPHISCHE DARSTELLUNG IHRER ÜBERLEGUNGEN.



4.Beispiel - BASE Transaktionen



Beispielbeschreibung ▾

- **Schwerpunkt:** KONSISTENZMODELLE
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 176

▸ Aufgabenstellung: Thorieklausur ▾

- ERKLÄREN SIE DEN UNTERSCHIED ZWISCHEN DER EIGENSCHAFT DES **Soft States** UND DES **Eventually Consistent**.



5.Beispiel - MVCC Verfahren



Beispielbeschreibung ▾

- **Schwerpunkt:** KONSISTENZMODELLE
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 176 - 177

▸ Aufgabenstellung: Thorieklausur ▾

- GEBEN SIE EINE GRAPHISCHE DARSTELLUNG DES **MVCC Verfahrens**.
- STELLEN SIE DIE INTERAKTION ZWISCHEN FOLGENDEN ENTITÄTEN DAR:
 - DER ZU VERÄNDERNDE DATENSATZ IM SPEICHER.
 - DER ZU VERÄNDERNDE DATENSATZ IN DER DATENBANK.
 - DIE ANWENDUNG DIE DEN DATENSATZ ÄNDERT.
 - DIE DATENBANK DIE DEN DATENSATZ SPEICHERT.



6. MongoDB

01

Mongo

01. Modellierung

52

02. Mongo Abfragen

54

6.1. Modellierung

1.Beispiel) Modellierung



Beispielbeschreibung

- **Schwerpunkt:** DOKUMENTORIENTIERTE MODELLIERUNG, MONGODB
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 182 - 192

► Aufgabenstellung: Projektmodellierung

- ENTWICKELN SIE FÜR DIE FOLGENDEN ENTITÄTEN DIE COLLECTIONS FÜR MONGODB.
- DEFINIEREN SIE FÜR DIE **Collections** JEWEILS EIN **Schema**.
- BEACHTEN SIE BEI DER **Modellierung** DIE PRINZIPIEN DER DOKUMENTORIENTIERTEN MODELLIERUNG.

► Codebeispiel: Entitätenbeschreibung

```

1  //-----
2  // Project
3  //-----
4  @Data
5  @Table(name="projects")
6  public class Project implements Serializable{
7
8      @Size(min=3, max = 50)
9      private String title;
10
11     private EProjectType type;
12
13     private EProjectState state;
14
15     @Size(min=0, max=4000)
16     private String description;
17
18     private Date projectBegin;
19
20     private Boolean isFWFSponsored;
21
22     private Boolean isFFGSponsored;
23
24     private Boolean isEUSponsored;
25
26     private Boolean isSmallProject;
27
28 }
29
30 public enum EProjectType{
31     REQUEST_FUNDING_PROJECT,
32     RESEARCH_FUNDING_PROJECT,
33     MANAGEMENT_PROJECT
34 }
```

```

35 //-----
36 // Subproject
37 //-----
38 @Data
39 @Table(name="subprojects")
40 public class Subproject implements Serializable{
41
42     @Size(min=3, max=100)
43     private String title;
44
45     @Size(min=0, max=4000)
46     private String description;
47
48     @Min(0)
49     @Max(100)
50     private Integer appliedResearch;
51
52     @Min(0)
53     @Max(100)
54     private Integer theoreticalResearch;
55
56     @Min(0)
57     @Max(100)
58     private Integer focusResearch;
59 }
60
61 //-----
62 // Facility
63 //-----
64 @Data
65 @Table(name="facilities")
66 public class Facility implements Serializable{
67
68     @Size(min=3, max=100)
69     private String name;
70
71     private String code;
72 }
73
74 //-----
75 // Debitor
76 //-----
77 @Data
78 @Table(name="debtors")
79 public class Debitor implements Serializable{
80
81     @Size(min=5, max=100)
82     private String name;
83
84     private String description;
85 }
86
87 //-----
88 // EProjectState
89 //-----
90 public enum EProjectState {
91     CREATED,
92     IN_APPROVEMENT,
93     APPROVED
94 }
95
96 }
97

```

```

98 //-----
99 // Bedingungen
100 //-----
101 1.) Ein Project Dokument soll auch
102 Informationen ueber die Subprojekte enthalten
103 aus denen es besteht.
104
105 2.) In einem Project sollen gleichzeitig die
106 Daten der Geldgeber zusammen mit dem Geldbetrag
107 vermerkt werden den der entsprechende Geldgeber
108 ueberwiesen hat.
109
110 3.) In einem Subproject Dokument sollen auch
111 Daten bezueglich des umgebenden Projects enthalten
112 sein.
113
114 4.) Gleichzeitig sollen im Subproject Dokument
115 Daten zu dem Institut (Facility) gespeichert
116 werden an dem das Subprojekt durchgefuehrt
117 wird.
118
119 5.) Fuer ein Institut muessen auch die Projekte
120 gespeichert werden an denen das Institut arbeitet.
121
122 6.) Fuer Geldgeber wird gespeichert welche Projekte
123 Sie finanzieren zusammen mit dem Betrag und dem
124 Tag der Ueberweisung.

```

□

2. Beispiel) Dokumente einfügen



Beispielbeschreibung ▾

- **Schwerpunkt:** INSERTONE, INSERTMANY
- **Komplexität:** EINFACH
- **Skriptum:** 195

► Aufgabenstellung: Dokumente einfügen ▾

- FÜGEN SIE MINDESTENS 5 PROJECT DOKUMENTE IN DIE PROJECTS COLLECTION.
- FÜGEN SIE MINDESTENS 5 SUBPROJECT DOKUMENTE IN DIE SUBPROJECTS COLLECTION.
- FÜGEN SIE MINDESTENS 5 FACILITY DOKUMENTE IN DIE FACILITIES COLLECTION.
- FÜGEN SIE MINDESTENS 5 DEBITOR DOKUMENTE IN DIE DEBITORS COLLECTION.

□

3.Beispiel) Dokumente bearbeiten



Beispielbeschreibung ▾

- **Schwerpunkt:** UPDATEONE, UPDATEMANY
- **Komplexität:** EINFACH
- **Skriptum:** 195 - 199

▸ Aufgabenstellung: Dokumente bearbeiten ▾

- BEARBEITEN SIE DIE IN DER DATENBANK VORHANDENEN DOKUMENTE.
- STELLEN SIE SICHER DASS DIE DATEN NACH DER BEARBEITUNG KONSISTENT SIND.

▸ Codebeispiel: Dokumente bearbeiten ▾

```

1  //-----
2  // 1.Aufgabe
3  //-----
4  1.) Fuegen Sie allen project Dokumenten die
5  folgenden Felder hinzu.
6
7  *)projectEnd date   defaultValue: Date('Jan 01,
8      2020')
9  *)rating   int   defaultValue: 5
10 *)partners  array item: {name: "TU Wien"}
11
12 //-----
13 // 2.Aufgabe
14 //-----
15 2.) Jedes REQUEST_FUNDING_PROJECT muss nun als
16 von der EU gefoerdert markiert werden.
17
18 //-----
19 // 3.Aufgabe
20 //-----
21 3.) Beim Anlegen der project Collection ist ein
22 Fehler unterlaufen. Benennen Sie das rating Feld
23 um in projectRating.
24
25 //-----
26 // 4.Aufgabe
27 //-----
28 4.) Fuegen Sie dem partners die folgenden Elemente
29 hinzu.
30 {name : "HTL Krems"},
31 {name : "TU Graz"}
32
33 Entfernen Sie anschliessend den HTL Krems Eintrag.
```



6.2. Abfragen

1.Beispiel) Query Kriterien



Beispielbeschreibung ▾

- **Schwerpunkt:** QUERY ABFRAGEN, CURSOR METHODEN
- **Komplexität:** MITTEL
- **Skriptum:** 264 - 270

▸ Aufgabenstellung: MongoDB DQL ▾

- SCHREIBEN SIE DIE FOLGENDEN MONGODB QUERIES FÜR DIE PROJECT COLLECTIONS.

▸ Codebeispiel: MongoDB Queries ▾

```

1  //-----
2  // 1.Aufgabe
3  //-----
4  1.) Finden Sie alle project Dokumente in der
5  projects collection.
6
7  *) Geben Sie nur die ersten 5 Project Dokumente
8  aus.
9  *) Sortieren Sie die projecte nach dem titel
10 aufsteigend
11 *) Geben Sie fuer die project Dokumente jeweils
12 den Titel, den Type und den Projektzustand aus.
13
14 //-----sor
15 // 2.Aufgabe
16 //-----
17 2.) Finden Sie alle REQUEST_FUNDING_PROJECTE die
18 sich im Zustand APPROVED befinden.
19
20 *) Geben Sie nur die ersten 5 Project Dokumente
21 aus.
22 *) Sortieren Sie die projecte nach dem titel
23 aufsteigend
24 *) Geben Sie fuer die project Dokumente jeweils
25 den Titel, den Type und den Projektzustand aus.
26
27 //-----
28 // 3.Aufgabe
29 //-----
30 3.) Finden alle Subprojecte deren appliedResearch
31 oder theoreticalResearch oder focusResearch einen
32 Wert ueber 60 vorweisen.
33
34 *) Geben Sie nur die ersten 3 Subproject Dokumente
35 aus.
36 *) Sortieren Sie die subprojecte nach dem titel
37 aufsteigend
38 *) Geben Sie fuer die subproject Dokumente jeweils
39 den Titel und die Forschungsschwerpunkte aus.
```



2.Beispiel) Query Kriterien



Beispielbeschreibung ▾

- **Schwerpunkt:** QUERY ABFRAGEN, CURSOR METHODEN
- **Komplexität:** MITTEL
- **Skriptum:** 264 - 270

▸ Aufgabenstellung: MongoDB DQL ▾

- SCHREIBEN SIE DIE FOLGENDEN MONGODB QUERIES FÜR DIE PROJECT COLLECTIONS.



▸ Codebeispiel: MongoDB Queries ▾

```

1 //-----
2 // 1.Aufgabe
3 //-----
4 1.) Finden Sie alle subproject Dokumente aus die
5 vom Institut fuer Softwareentwicklung, durchgeführt
6 werden.
7
8 *) Sortieren Sie die subprojecte nach dem titel
9
10 //-----
11 // 2.Aufgabe
12 //-----
13 2.) Finden Sie alle subproject Dokumente deren
14 Forschungsschwerpunkte in Summe 100 ergeben.
15
16 *) Sortieren Sie die subprojecte nach dem titel
17 aufsteigend.
18
19 //-----
20 // 3.Aufgabe
21 //-----
22 3.) Finden Sie alle debitoren die mehr als 2
23 Projekte finanzieren
24
25 *) Sortieren Sie die debitoren nach dem Namen
26
27 //-----
28 // 4.Aufgabe
29 //-----
30 4.) Finden Sie alle projecte die mit mehr als
31 700000 finanziert sind.
32
33 *) Sortieren Sie die projecte nach dem titel.
```



3.Beispiel) Query Kriterien



Beispielbeschreibung ▾

- **Schwerpunkt:** QUERY ABFRAGEN, CURSOR METHODEN
- **Komplexität:** MITTEL
- **Skriptum:** 195 - 270

▸ Aufgabenstellung: MongoDB DQL ▾

- SCHREIBEN SIE DIE FOLGENDEN MONGODB QUERIES.

▸ Codebeispiel: MongoDB Queries ▾

```

1 //-----
2 // 1.Aufgabe
3 //-----
4 1.)Finden Sie alle Projekte die weder
5 REQUEST_FUNDING_PROJECTS noch
6 RESEARCH_FUNDING_PROJECTS
7 sind.
8
9 *) Fuer die Projekte soll nur der Titel und der
10 Projekttyp angegeben werden.
11 *) Sortieren Sie das Ergebnis nach dem titel
12 absteigend
13
14 //-----
15 // 2.Aufgabe
16 //-----
17 2.) Finden Sie alle Projekte die weder
18 REQUEST_FUNDING_PROJECTS noch
19 RESEARCH_FUNDING_PROJECTS
20 sind. Die Projekte muessen 1 Subprojekt haben.
21 Zustzlich muessen die Projekt die TU Wien als
22 Partner haben.
23
24 *) Fuer die Projekte soll nur der Titel und der
25 Projekttyp angegeben werden.
26 *) Sortieren Sie das Ergebnis nach dem titel
27 absteigend
28
29 //-----
30 // 3.Aufgabe
31 //-----
32 3.) Finden Sie alle Projekte die ein Rating
33 zwischen 3 und 5 haben. Die Projekte müssen
34 als Partner sowohl die TU Wien als auch die
35 TU Graz haben.
```



4.Beispiel) Query Kriterien



Beispielbeschreibung ▾

- **Schwerpunkt:** QUERY ABFRAGEN, CURSOR METHODEN
- **Komplexität:** KOMPLEX
- **Skriptum:** 195 - 270

▸ Aufgabenstellung: MongoDB DQL ▾

- SCHREIBEN SIE DIE FOLGENDEN MONGODB QUERIES.

▸ Codebeispiel: MongoDB Queries ▾

```

1  //-----
2  // 1.Aufgabe
3  //-----
4  Fuegen Sie allen Subprojekten die Forschungspunkt
5  ueber 80 haben die Felder - marked : true - und
6  das Objekt - funding : {amount : NumberLong(10000)}
7  hinzu. Folgende Felder sollen folgende Felder haben
8
9  theoreticalResearch : 100
10 focusResearch : 0
11 appliedResearch : 0
12 //-----
13 // 2.Aufgabe
14 //-----
15 Fuer jedes Subprojekt soll ein project Dokument
16 angelegt werden, dass die _id des zugehoerigen
17 Projekts und seinen Titel enthaelt.
18
19
20 //-----
21 // 3.Aufgabe
22 //-----
23 Fuegen Sie jedem Subproject ein Objekt funding
24 hinzu. Das Objekt enthaelt ein Feld amount. Der
25 Wert von amount berechnet sich folgendermassen:
26
27 Berechnen Sie die Projektfoerderung in dem Sie
28 die einzelnen Foerderungen aufsummieren.
29
30 Verteilen Sie die Projektfoerderung nun
31 gleichermassen auf alle Subprojekte.
```

