

# Wirtschaftsinformatik 2

Prof. Dr. Dr. L. Schmidt-Thieme  
MSc. André Busche

## Übung 4

---

1. Übungsblatt 4

2. Allgemeines zu XML

# Übungsblatt 4

Allgemeiner Hinweis: in den nachfolgenden SQL-Kommandos können die Hochkommata

'  
,  
,  
”  
“

Nicht stimmen. Diese sind ggf. anzupassen!

# Aufgabe 1a (1 Punkt)

Erstellen Sie mind. zwei weitere Künstler (Actors)

# Aufgabe 1a (1 Punkt)

Erstellen Sie mind. zwei weitere Künstler (Actors)

```
insert into
  actor(name)
VALUES
  ('Donald Duck');
```

# Aufgabe 1a (1 Punkt)

Erstellen Sie mind. zwei weitere Künstler (Actors)

```
insert into  
  actor(name)  
VALUES  
  ('Donald Duck');
```

```
ERROR 1062 (23000) :  
  Duplicate entry '0' for key 'PRIMARY'
```

## Aufgabe 1a (1 Punkt)

Erstellen Sie mind. zwei weitere Künstler (Actors)

```
select max(id) from actor;
```

```
insert into  
  actor(id, name)  
VALUES  
  (max+1, 'Donald Duck');
```

Natürlich ist dies durch die reale ID zu ersetzen ..

# Aufgabe 1a (1 Punkt)

Erstellen Sie mind. zwei weitere Künstler (Actors)

```
ALTER TABLE actor MODIFY id INT  
AUTO_INCREMENT;
```

**AUTO\_INCREMENT** sorgt *in MySQL* dafür, dass automatische eindeutige IDs erzeugt werden, d.h.

```
insert into  
  actor(name)  
VALUES  
  ('Donald Duck');
```

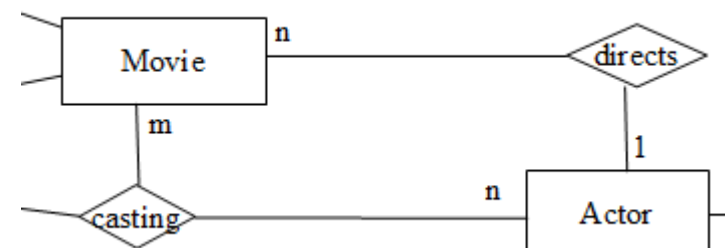
funktioniert.

In **PostgreSQL** würde man dies z.B. durch den Datentyp **serial** erreichen (Benutzung als Datentyp während des Anlegens der Tabelle)



## Aufgabe 1b (1 Punkt)

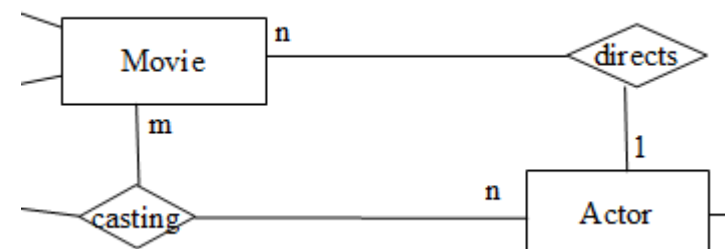
Beide Künstler / Actors haben in dem selben Film mitgewirkt, der noch nicht in der Datenbank hinterlegt ist. Diesen Film finden Sie total toll (=hohe Wertung). Legen Sie diesen Film an.



... Wenn Künstler in einem Film spielen, dann wurden sie auch gecastet ...

## Aufgabe 1b (1 Punkt)

Beide Künstler / Actors haben in dem selben Film mitgewirkt, der noch nicht in der Datenbank hinterlegt ist. Diesen Film finden Sie total toll (=hohe Wertung). Legen Sie diesen Film an.



```
insert into movie values (1846,  
"Trommelfeuer", 2010, 9, 1, 1);  
insert into casting values (1846, 5946, 1);  
insert into casting values (1846, 5947, 2);
```

## Aufgabe 1c (1 Punkt)

Der Film „Demolition Man“ wird nachträglich indexiert (vom Markt genommen). Entfernen Sie diesen.

## Aufgabe 1c (1 Punkt)

Der Film „Demolition Man“ wird nachträglich indexiert (vom Markt genommen). Entfernen Sie diesen.

```
select * from movie where title="Rocky II";
```

```
delete from movie where id=762;
```

```
select * from movie where id=762;
```

## Aufgabe 1d (1 Punkt)

Wie viele Filme mit mindestens 3000 Votes und einer durchschnittlichen Bewertung von besser als 5,5 gibt es insgesamt?

## Aufgabe 1d (1 Punkt)

Wie viele Filme mit mindestens 3000 Votes und einer durchschnittlichen Bewertung von besser als 5,5 gibt es insgesamt?

```
select
  count(*)
from
  movie
where
  votes >= 3000
and
  score > 5.5;
```

## Aufgabe 1d (1 Punkt)

Wie viele Filme mit mindestens 3000 Votes und einer durchschnittlichen Bewertung von besser als 5,5 gibt es insgesamt?

Was passiert?

```
SELECT
  count (*)
FROM
  movie
where
  votes >= 3000
and
  (Select avg(score) FROM movie) > 5.5
```

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?



## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme **mit den meisten Schauspielern** zeigen. Welche sind es?

```
mysql> describe casting;
```

Field	Type	Null	Key	Default	Extra
movieid	int(11)	NO	PRI	NULL	
actorid	int(11)	NO	PRI	NULL	
ord	int(11)	YES	MUL	NULL	

... mal eine dumme Frage: Was ist ein Film?

# Aufgabe 1e (2 Punkte)

Was ist ein Film?

`movieid` identisch?

`title` identisch?

# Exkurs: Gruppierung

movieid	actorid	ord
101	1193	3
101	1477	2
101	1745	5
101	1873	4
101	2276	1
101	2953	8
102	237	1
102	372	5
102	699	2
102	779	3
102	1326	14
102	3087	7
102	3528	10
102	5769	12
103	1635	4
103	1974	9
103	2915	5
103	3274	8

*mit den meisten*

**Schauspielern :=**

Zähle für jeden Film die Anzahl  
zugehöriger Schauspieler  
**movieid=101**

**movieid=102**

**movieid=103**

# Exkurs: Gruppierung

d	ord
3	3
7	2
5	5
3	4
6	1
3	8
7	1
2	5
9	2
9	3
6	14
7	7
8	10
9	12
5	4
4	9
5	5
4	8

movieid=101

count(\*)

movieid=102

count(\*)

movieid=103

count(\*)

Rank / order  
by amounts

# Exkurs: Gruppierung

d	ord
3	3
7	2
5	5
3	4
6	1
3	8
7	1
2	5
9	2
9	3
6	14
7	7
8	10
9	12
5	4
4	9
5	5
4	8

movieid=101

movieid=102

movieid=103

## Aggregatsfunktionen

müssen auf diese  
„Untergruppen“  
angewendet werden,  
z. B.

**Avg (ord) ,**

**Min (ord) ,**

**Max (ord) ,**

...

# Exkurs: Gruppierung – Achtung MySQL!

d	ord
3	3
7	2
5	5
3	4
6	1
3	8
7	1
2	5
9	2
9	3
6	14
7	7
8	10
9	12
5	4
4	9
5	5
4	8

movieid=101

movieid=102

movieid=103

Spaltenselektionen sind **nicht** möglich!

Sobald gruppiert wird, muss mit (Elementen) der Gruppe gearbeitet werden!

Also z. B.

**Select Max(ord) from movies group by movieid**

# Exkurs: Gruppierung – Achtung MySQL!

```
Select Max(ord) from movies group by movieid
```

Die Selektion von Spalten während einer Gruppierung ist **falsch**

```
Select ord from movies group by movieid
```

Warum?

# Exkurs: Gruppierung – Achtung MySQL!

```
Select ord from movies group by movieid
```

MySQL gibt dennoch ein Ergebnis zurück.

Der Wert des Ergebnisses ist **arbiträr** und bezieht sich auf eine **beliebige** Ergebniszeile der Gruppe.

Beispielsweise abhängig von:

- \* Sortierung
- \* Join? Wie?

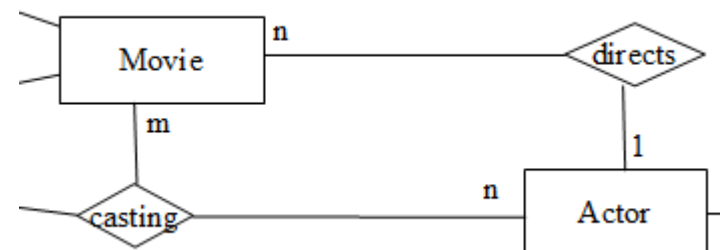


# Aufgabe 1e (2 Punkte)

Was ist ein Film?

`movieid` identisch?

`title` identisch?



# Exkurs: Join

movieid	actorid	ord
101	1193	3
101	2953	8
102	237	1
102	372	5

id	title
102	Austin Powers: The Spy Who
...	

**Casting c join movie m on (c.movieid=m.id)**

movieid	actorid	ord	title
102	237	1	Austin Powers: The Spy Who Shag
102	372	5	Austin Powers: The Spy Who Shag

```
select c.*, m.title from casting c join movie m on (c.movieid=m.id) where c.movieid in (101, 102);
```

# Exkurs: Left Join

movieid	actorid	ord
101	1193	3
101	2953	8
102	237	1
102	372	5

id	title
102	Austin Powers: The Spy Who
...	

**Casting c left join movie m on (c.movieid=m.id)**

movieid	actorid	ord	title
101	1193	3	NULL
101	2953	8	NULL
102	237	1	Austin Powers: The Spy Who Shaq
102	372	5	Austin Powers: The Spy Who Shaq

```
select c.*, m.title from casting c left join movie m on (c.movieid=m.id) where c.movieid in (101, 102);
```

## Aufgabe 1e (2 Punkte)

Was ist ein Film?

`movieid` identisch?

`title` identisch?

Hypothese: Es gibt namensgleiche Filme mit unterschiedlichen IDs.

Wie bekommen wir also diese IDs / Duplikate heraus?

# Aufgabe 1e (2 Punkte)

```
select
  m1.id,
  m2.id,
  m1.title,
  m1.yr,
  m2.yr
from
  movie m1
join
  movie m2
on (m1.title=m2.title)
where
  m1.id!=m2.id;
```

# Aufgabe 1e (2 Punkte)

```
select m1.id, m2.id, m1.title, m1.yr, m2.yr
from movie m1 join movie m2 on
(m1.title=m2.title) where m1.id!=m2.id;
```

id	id	title	yr	yr
54	746	Psycho	1960	1998
313	1047	Hamlet	1996	1990
353	1000	Haunting, The	1999	1963
527	1364	Shaft	2000	1971
553	957	Sabrina	1995	1954
746	54	Psycho	1998	1960
881	1064	Frankenstein	1994	1931
899	1396	Three Musketeers, The	1993	1973
957	553	Sabrina	1954	1995
1000	353	Haunting, The	1963	1999
1047	313	Hamlet	1990	1996
1064	881	Frankenstein	1931	1994

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
select
  movieid, count(*) as mid
from
  casting
group by
  movieid
order by
  mid DESC limit 3
```

MySQL-  
spezifisch!

... und die Filmtitel?

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
select id, title
from
  movie m
join
  (select movieid, count(*) as mid
   from casting
   group by movieid
   order by mid DESC limit 3) t
on (m.id=t.movieid);
```



## Aufgabe 1e (2 Punkte)

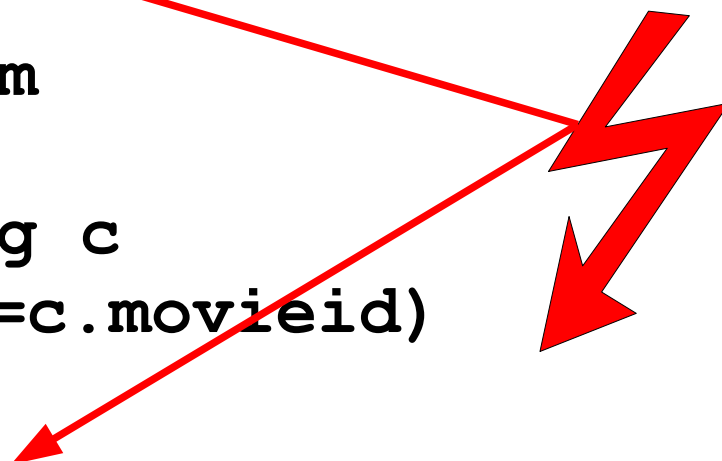
An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
Select
  m.id, count(*) as AnzahlSchauspieler
from
  movie m
join
  casting c
on (m.id=c.movieid)
group by
  m.id
order by
  AnzahlSchauspieler DESC limit 3;
```

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

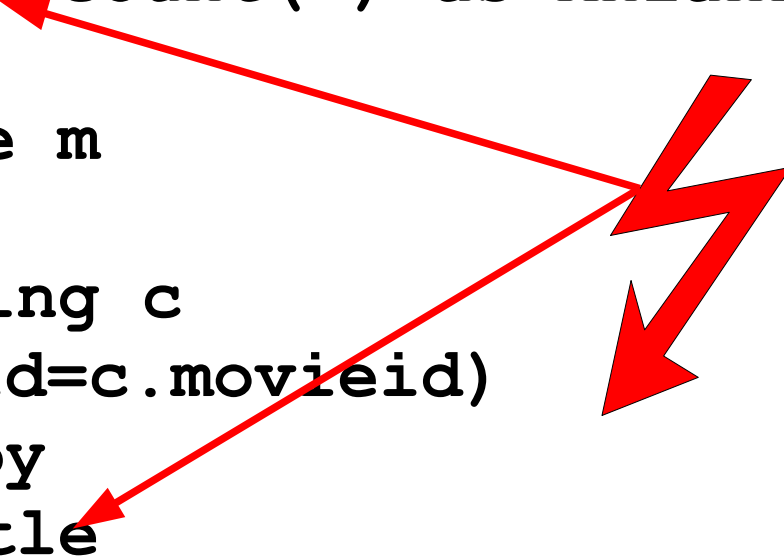
```
Select
  m.title, count(*) as AnzahlSchauspieler
from
  movie m
join
  casting c
on (m.id=c.movieid)
group by
  m.id
order by
  AnzahlSchauspieler DESC limit 3;
```

A red lightning bolt symbol is positioned over the SQL query, with two red arrows pointing to the 'count(\*)' expression and the 'm.id' in the 'group by' clause, indicating errors in the query.

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
Select
  m.id, count(*) as AnzahlSchauspieler
from
  movie m
join
  casting c
on (m.id=c.movieid)
group by
  m.title
order by
  AnzahlSchauspieler DESC limit 3;
```

A red lightning bolt symbol is positioned to the right of the SQL query. Two red arrows originate from the lightning bolt: one points to the comma after 'm.id' in the SELECT clause, and the other points to the 'm.title' in the GROUP BY clause, indicating errors in the query.

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
Select
  m.title, count(*) as AnzahlSchauspieler
from
  movie m
join
  casting c
on (m.id=c.movieid)
group by
  m.title
order by
  AnzahlSchauspieler DESC limit 3;
```

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
select title, id
from movie m
join
  (select movieid, count(*) as mid
   from casting
   group by movieid
   having mid>14) t
on (m.id=t.movieid);
```

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
select
  movieid, count(*) as mid
from casting c
group by movieid
having mid >=15
```

## Aufgabe 1e (2 Punkte)

An einem speziellen Filmnachmittag wollen Sie die 3 Filme mit den meisten Schauspielern zeigen. Welche sind es?

```
select
  movieid, count(*) as mid
from casting c
group by movieid
having mid >=
  (SELECT max(cnt)
   FROM (
     select count(*) cnt
     from casting c
     group by movieid)
   t);
```

## Aufgabe 1f (2 Punkte)

Erstellen Sie eine Statistik über die Anzahl Filme pro Jahr.  
(Das Ergebnis der Anfrage soll genau 2 Informationen/Spalten  
beinhalten: das Jahr sowie die Anzahl der Filme in diesem  
Jahr)

Dies ist ein Standard-Problem der Datenanalyse  
„Wie viele des Typs X gibt es?“



## Aufgabe 1f (2 Punkte)

Erstellen Sie eine Statistik über die Anzahl Filme pro Jahr. (Das Ergebnis der Anfrage soll genau 2 Informationen/Spalten beinhalten: das Jahr sowie die Anzahl der Filme in diesem Jahr)

```
select
  yr, count(yr) as anzahl
from
  movie
group by
  yr
order by
  yr desc;
```

## Aufgabe 1g (3 Punkte)

Filtern Sie die Ergebnisse der Statistik aus Aufgabe 1f, indem Sie:

- nur solche Filme berücksichtigen, die mindestens 10000 Votes haben, SOWIE
- nur solche Filme berücksichtigen, in denen mindestens 7 Schauspieler gespielt haben.

# Aufgabe 1g (3 Punkte)

- nur solche Filme berücksichtigen, die mindestens 10000 Votes haben,

Bedingung in der gleichen Tabelle  
→ WHERE-Klausel

## SOWIE

- nur solche Filme berücksichtigen, in denen mindestens 7 Schauspieler gespielt haben.

Kardinalitätseinschränkung einer referenzierten Tabelle  
→ WHERE-Klausel

(... aber wie?)

# Aufgabe 1g (3 Punkte)

- nur solche Filme berücksichtigen, die mindestens 10000 Votes haben, SOWIE
- nur solche Filme berücksichtigen, in denen mindestens 7 Schauspieler gespielt haben.

```
select m.yr, count(m.yr) as Anzahl
```

```
from
```

```
    movie m
```

```
join
```

```
    casting c
```

```
on (m.id=c.movieid)
```

```
where
```

```
    m.votes >= 10000
```

```
group by
```

```
    m.yr
```

```
having
```

```
    count(c.actorid) > 7
```

Selektiert „Alle Jahre, in denen es mehr als 7 Filme gab“



# Aufgabe 1g (3 Punkte)

- nur solche Filme berücksichtigen, die mindestens 10000 Votes haben, SOWIE
- nur solche Filme berücksichtigen, in denen mindestens 7 Schauspieler gespielt haben.

```
select yr, count(*)
from (
  select *
  from
    movie m
  left join
    casting c
  on (c.movieid=m.id)
  where votes >= 10000
  group by c.movieid
  having count(*)>=7 ) t
group by yr
```

# Aufgabe 1g (3 Punkte)

Filtern Sie die Ergebnisse der Statistik aus Aufgabe 1f ...

```
select
  count(*) , yr
from
  movie
where
  (select count(*) from casting where movieid=id) >= 7
and
  votes >= 10000
group by yr
order by count(*) ;
```

# XML

# Allgemeines zu XML

Gute XML Editoren...

- validieren nach DTD/XML Schema
- Erlauben „point-and-click“ XSLT Transformationen und/oder XQuery anfragen
- Unterstützen verschiedene XML Parser/Prozessoren

Eine kleine Auswahl:

- \* Oxygen XML Editor (<http://www.oxygenxml.com>) - für alle Betriebssysteme
- \* Altova XML Spy (<http://www.altova.com/>) - Windows
- \* Saxon auf der Kommandozeile



# XML von der Kommandozeile

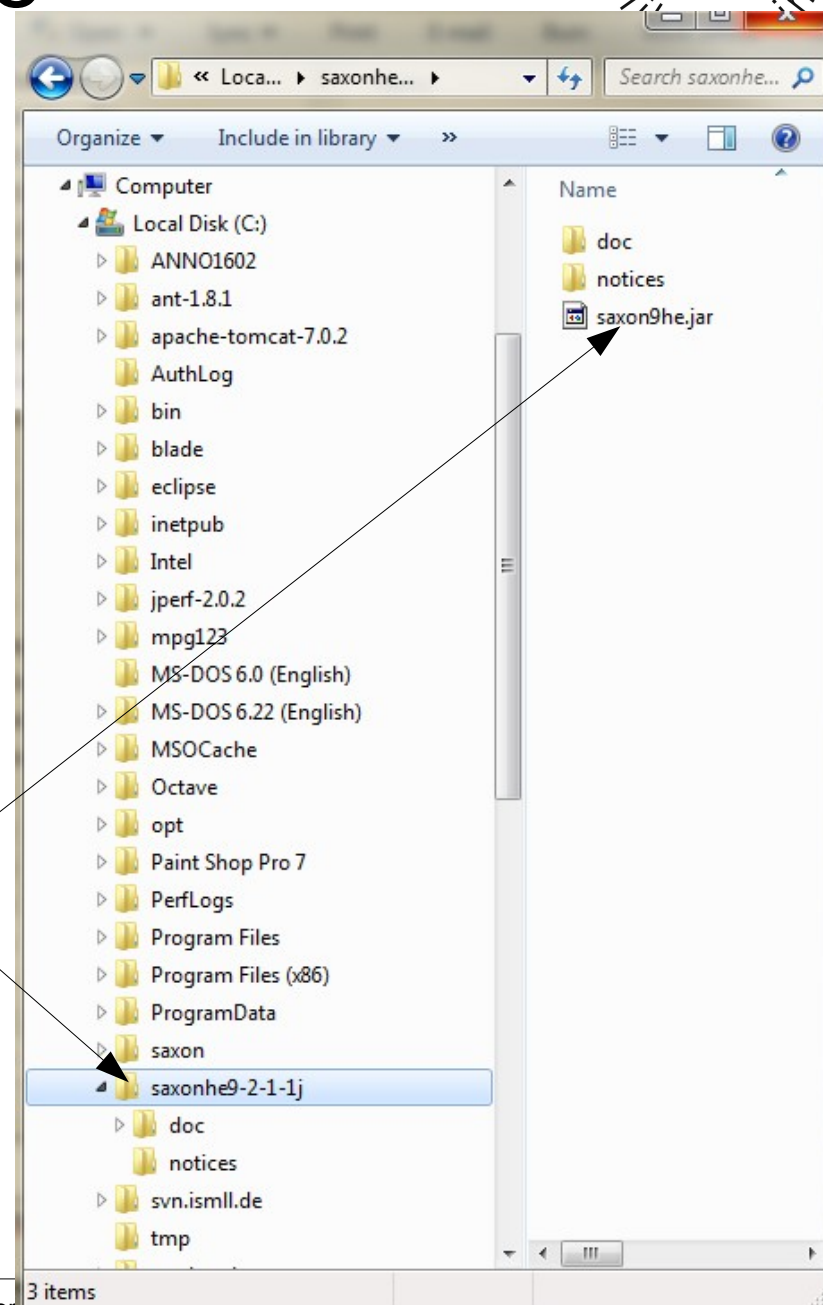
... am Beispiel Saxon  
(<http://saxon.sourceforge.net/>)

... download und unzip ...

Hier: nach `c:\saxonhe9-2-1-1j`

(unter Linux auch nur entpacken  
und z. B. nach `/home/<user>/saxon`)

Viel wichtiger: die `.jar-Datei`

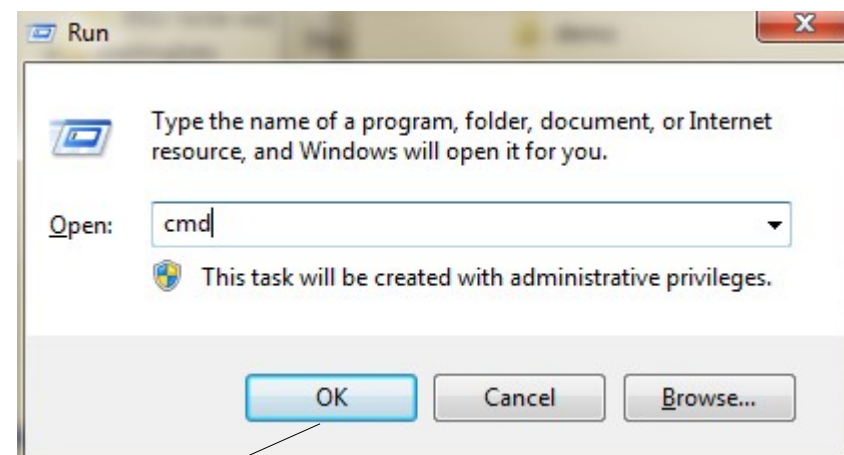


# XML von der Kommandozeile

Kommandozeile öffnen

Windows: (Win-R → „cmd“ → Enter)

Linux/Mac: sollte bekannt sein...



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\John>java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Oracle JRockit(R) (build R28.1.0-123-138454-1.6.0_20-20101014-1351-windows-x86_64, compiled mode)

C:\Users\John>
```

# XML von der Kommandozeile

*Windows-user:*

... und wer an dieser Stelle einen Fehler bekommt,  
dass java nicht gefunden wird ...

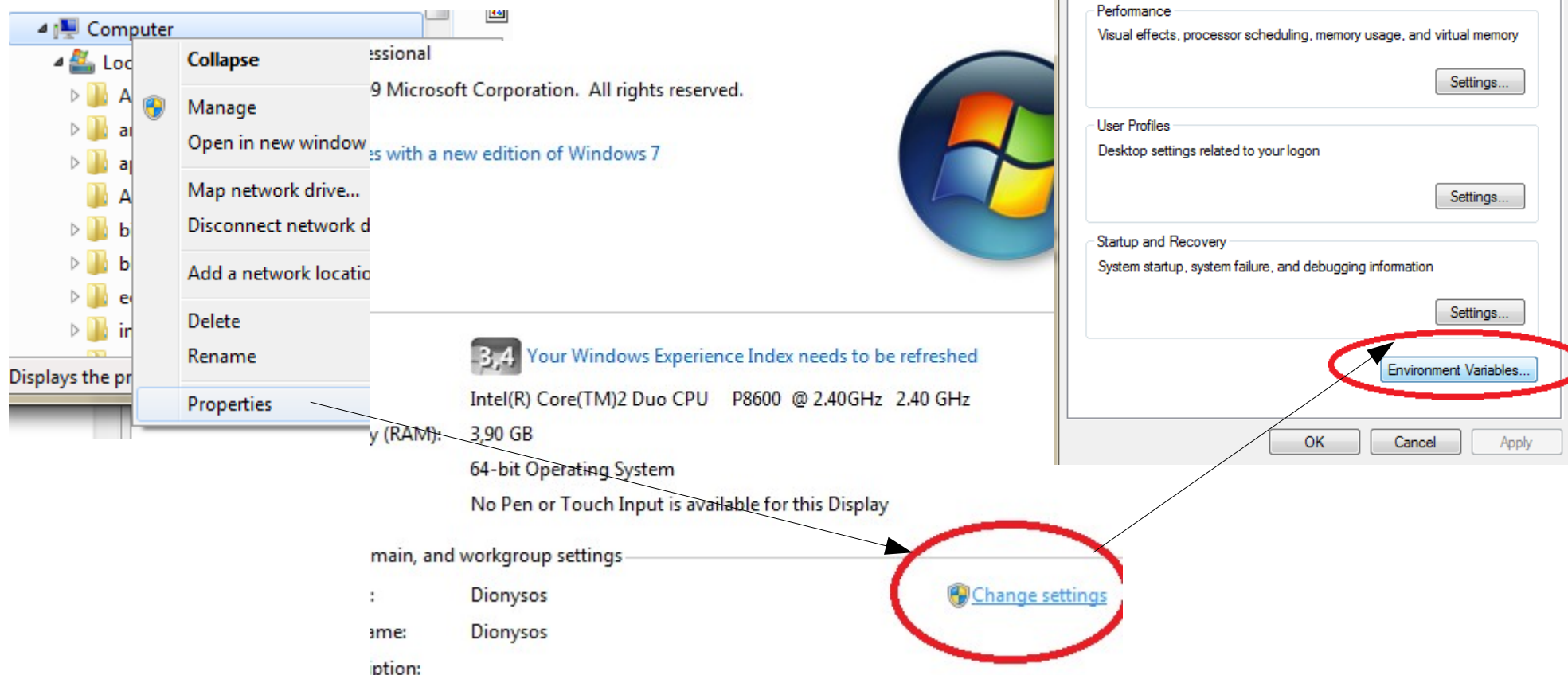
*Linux-user:*

... java nachinstallieren (verschiedene Wege,  
z. B. Über den Paketmanager)

# XML von der Kommandozeile

... einfach die .jar-Datei von der Kommandozeile ausführen ...

(evtl. vorher ein JAVA\_HOME/PATH zu der Java-Runtime setzen ...)

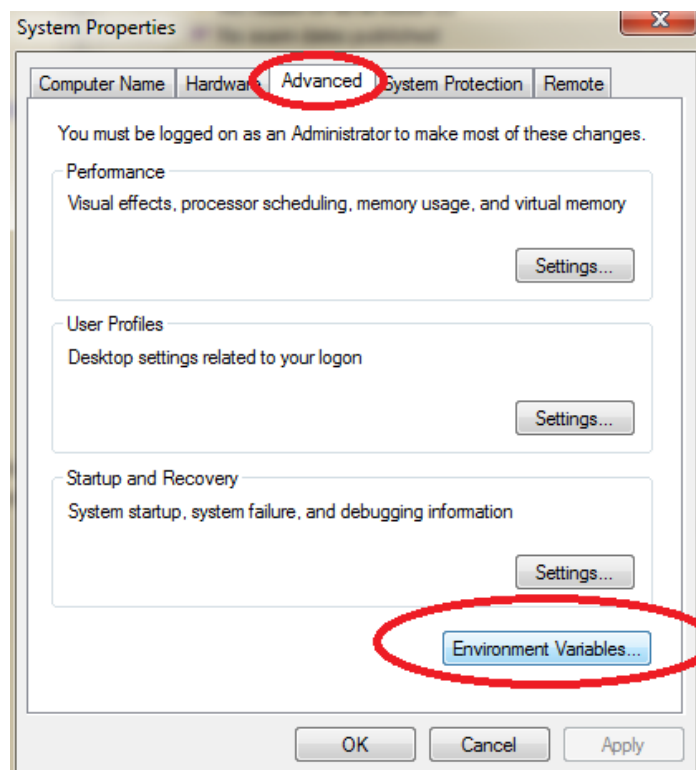


The image shows a Windows 7 desktop environment. On the left, a File Explorer window is open to 'Computer', with the 'Properties' menu option highlighted. In the center, the 'System' properties window is open, showing the 'Advanced' tab. The 'Advanced' tab is circled in red. At the bottom of this window, the 'Environment Variables...' button is also circled in red. An arrow points from this button to the 'Change settings' link in the Windows Experience Index (WEI) notification area, which is also circled in red. The WEI notification area shows a score of 3.4 and a message: 'Your Windows Experience Index needs to be refreshed'. Below this, system specifications are listed: Intel(R) Core(TM)2 Duo CPU P8600 @ 2.40GHz 2.40 GHz, 3.90 GB RAM, and 64-bit Operating System.

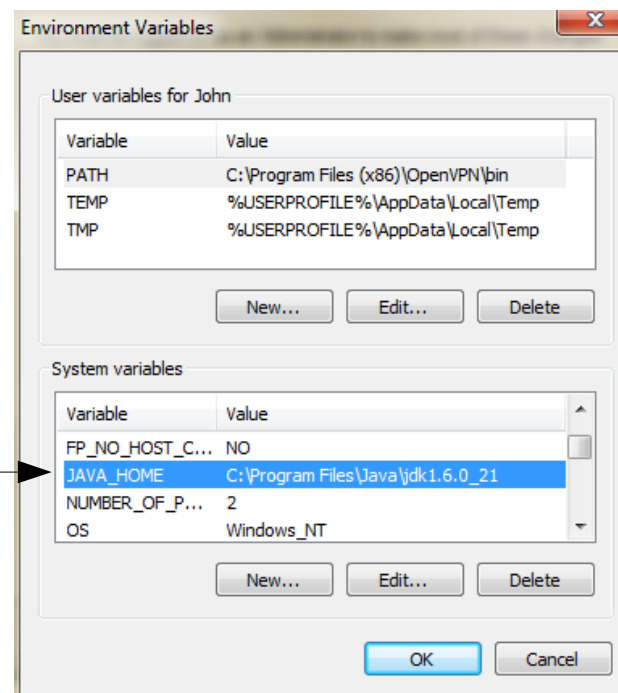
# XML von der Kommandozeile

Variable: JAVA\_HOME

Value: C:\Program Files\Java\jdk1.6.0\_21

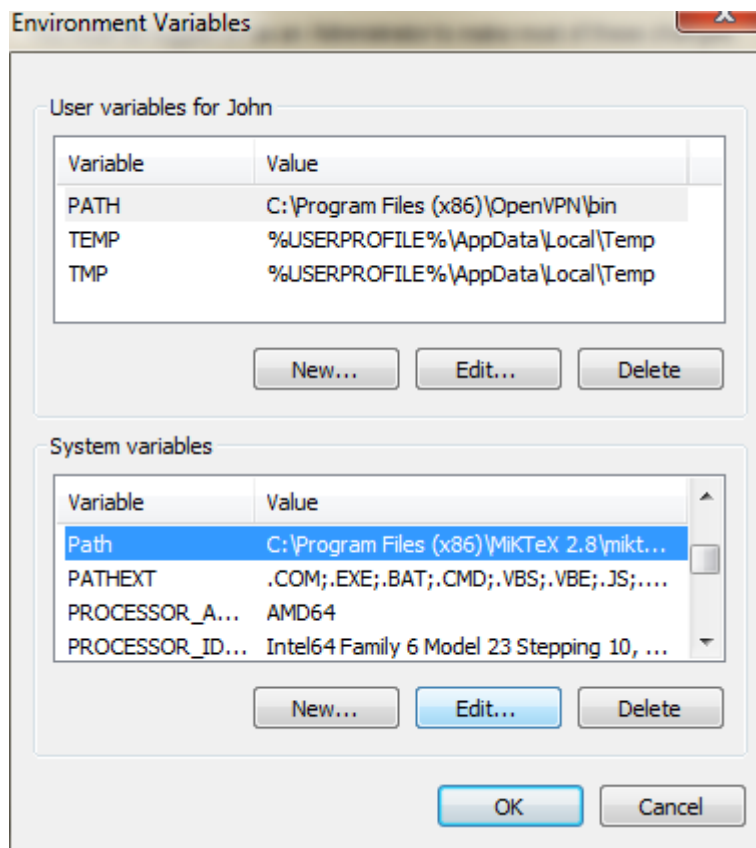
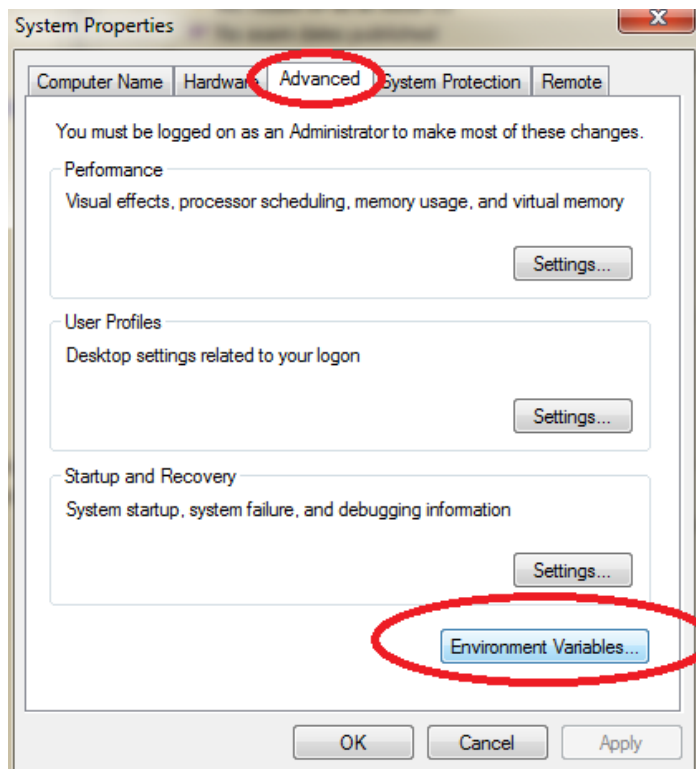


1)

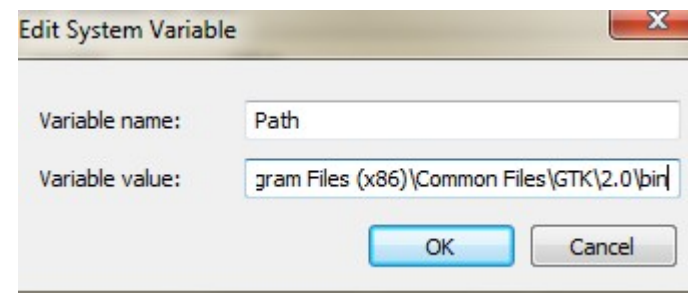


# XML von der Kommandozeile

2)



Path → Edit



Ganz ans Ende:

`;C:\Program Files\Java\jdk1.6.0_21\bin`

# XML von der Kommandozeile

... von hier ab (fast) unisex für Windows, Linux, Mac...

```
C:\Users\John>cd \saxonhe9-2-1-1j
C:\saxonhe9-2-1-1j>java -cp .;saxon9he.jar net.sf.saxon.Query
No query file name
Saxon-HE 9.2.1.1J from Saxonica
Usage: java net.sf.saxon.Query [options] query {param=value}...
Options:
  -backup:on|off      Save updated documents before overwriting
  -config:filename    Use configuration file
  -cr:classname       Use specified CollectionURIResolver class
  -dtd:on|off         Validate using DTD
  -expand:on|off      Expand defaults defined in schema/DTD
  -explain[:filename] Display compiled expression tree
  -ext[:on|off]       Allow/Disallow external Java functions
```

Aufruf Saxon  
(ohne Parameter/Eingabedatei)

(Linux: Doppelpunkt, anstatt Semikolon)

# XML von der Kommandozeile

... so viel für heute.

Nächste Woche: Einfache Xpath Abfragen im  
Hinblick auf die Übung