



Folien basierend auf  
Thorsten Papenbrock

Übung Datenbanksysteme I  
Besprechung

Leon Bornemann

F-2.06, Campus II  
Hasso Plattner Institut

## XPath Cheatsheet

### Expression Description

<code>nodename</code>	Select all nodes with the name "nodename"
<code>/</code>	A beginning single slash indicates a select from the root node, subsequent slashes indicate selecting a child node from current node
<code>//</code>	Select direct and indirect child nodes in the document from the current node - this gives you the ability to "skip levels"
<code>.</code>	Select the current context node
<code>..</code>	Select the parent of the context node
<code>@</code>	Select attributes
<code>text()</code>	Select the value of an element
<code> </code>	Pipe chains expressions and brings back results from either expression, think of a set union

### Operator Explanation

<code>=</code>	Equivalent comparison, can be used for numeric or text values
<code>!=</code>	Is not equivalent comparison
<code>&gt;</code> , <code>&gt;=</code>	Greater than, greater than or equal to
<code>&lt;</code> , <code>&lt;=</code>	Less than, less than or equal to
<code>or</code>	Boolean or
<code>and</code>	Boolean and
<code>not</code>	Boolean not

Note: Using '!=' is not the same as using 'not'.

### Predicates

```
[1] | Select the first node
[last()] | Select the last node
[last()-1] | Select the last but one node (also known as the second last node)
[position()<3] | Select the first two nodes, note the first position starts at 1, not = 1
[@lang] | Select nodes that have attribute 'lang'
[@lang='en'] | Select all the nodes that have a "attribute" attribute with a value of "en"
[price>15.00] | Select all nodes that have a price node with a value greater than 15.00
```

### Wildcards

XPath wildcards can be used to select unknown XML nodes.

### Wildcard Description

<code>*</code>	Matches any element node
<code>@*</code>	Matches any attribute node
<code>node()</code>	Matches any node of any kind

## In-text search

XPath can do in-text searching using functions and also supports regex with its `matches()` function. Note: in-text searching is case-sensitive!

Path Expression	Result
<code>//author[contains(., "Matt")]</code>	Matches on all author nodes, in current node contains Matt (case-sensitive)
<code>//author[starts-with(., "G")]</code>	Matches on all author nodes, in current node starts with G (case-sensitive)
<code>//author[ends-with(., "w")]</code>	Matches on all author nodes, in current node ends with w (case-sensitive)
<code>//author[matches(., "Matt.*")]</code>	regular expressions match 2.0

<https://www.freeformatter.com/xpath-tester.html>

<http://labs.timtom.ch/library-webscraping/extras/xpath-cheatsheet.md.pdf>

DBSI - Übung  
Besprechung

Leon Bornemann  
Chart 2

# Nachbesprechung Zettel 5

---

Betrachte den folgenden Schedule (entspricht  $S_1$  aus Aufgabe 1 mit ergänzten \*lock-Operationen):

$sl_1(X), r_1(X), sl_2(X), r_2(X), u_2(X), w_1(X), xl_2(Y), r_2(Y), w_2(Y), sl_1(Y), u_1(X), r_1(Y), u_1(Y), u_2(Y)$

- Gib für beide Transaktionen an, ob sie jeweils konsistent sind, und begründe kurz. **2 P**
- Gib für beide Transaktionen an, ob sie jeweils die 2PL-Bedingung erfüllen, und begründe kurz. **2 P**
- Ist der Schedule legal, wenn die Locks in der dargestellten Reihenfolge ausgeführt werden? Warum bzw. warum nicht? (Anmerkung: Wir nehmen hierbei an, dass keine Aktionen zurückgestellt oder verschoben werden.) **1 P**

Besprechung

Leon Bornemann  
Chart 3

# Nachbesprechung Zettel 5

Betrachte den folgenden Schedule (entspricht  $S_1$  aus Aufgabe 1 mit ergänzten \*lock-Operationen):

$sl_1(X), r_1(X), sl_2(X), r_2(X), u_2(X), w_1(X), xl_2(Y), r_2(Y), w_2(Y), sl_1(Y), u_1(X), r_1(Y), u_1(Y), u_2(Y)$

a) Gib für beide Transaktionen an, ob sie jeweils konsistent sind, und begründe kurz. **2 P**

TA1: nein, wegen  $sl(X)$  und  $w(X)$

TA2: ja, vor jedem Zugriff korrekter Lock, anschließend Unlock

# Nachbesprechung Zettel 5

Betrachte den folgenden Schedule (entspricht  $S_1$  aus Aufgabe 1 mit ergänzten \*lock-Operationen):

$sl_1(X), r_1(X), sl_2(X), r_2(X), u_2(X), w_1(X), xl_2(Y), r_2(Y), w_2(Y), sl_1(Y), u_1(X), r_1(Y), u_1(Y), u_2(Y)$

- Gib für beide Transaktionen an, ob sie jeweils konsistent sind, und begründe kurz. **2 P**
- Gib für beide Transaktionen an, ob sie jeweils die 2PL-Bedingung erfüllen, und begründe

TA1: ja, kein lock nach erstem unlock

TA2: nein, wegen  $u(X)$  vor  $xl(Y)$

# Nachbesprechung Zettel 5

Betrachte den folgenden Schedule (entspricht  $S_1$  aus Aufgabe 1 mit ergänzten \*lock-Operationen):

$sl_1(X), r_1(X), sl_2(X), r_2(X), u_2(X), w_1(X), xl_2(Y), r_2(Y), w_2(Y), sl_1(Y), u_1(X), r_1(Y), u_1(Y), u_2(Y)$

- Gib für beide Transaktionen an, ob sie jeweils konsistent sind, und begründe kurz. **2 P**
- Gib für beide Transaktionen an, ob sie jeweils die 2PL-Bedingung erfüllen, und begründe kurz. **2 P**
- Ist der Schedule legal, wenn die Locks in der dargestellten Reihenfolge ausgeführt werden? Warum bzw. warum nicht? (Anmerkung: Wir nehmen hierbei an, dass keine Aktionen zurückgestellt oder verschoben werden.) **1 P**

Nein, wegen gleichzeitigem  $xl_2(Y)$  und  $sl_1(Y)$ .

# Nachbesprechung Zettel 5

---

## Aufgabe 3: Scheduler

Betrachte den folgenden Schedule:

$r_1(A), r_2(B), r_3(C), r_1(B), r_2(C), r_3(D), w_1(C), w_2(D), w_3(E)$

Überlege zunächst, wo in diesem Schedule \*lock-Operationen eingefügt werden müssen, damit die drei beteiligten Transaktionen konsistent und 2PL-konform sperren. Gib anschließend den tabellarischen Ablaufplan der Transaktionsausführung eines DBMS-Schedulers an. Kennzeichne darin, welche Sperranforderungen akzeptiert bzw. zunächst abgelehnt werden.

a) Nutze zur Erstellung der Ablaufpläne einfache Lock- und Unlock-Operationen.

besprechung

Nachbes

	$T_1$	$T_2$	$T_3$
	$l(A), r(A)$		
	$l(B), r(B)$		
			$l(C), r(C)$
$r_1(A), r_2(l$	$l(B) \downarrow$		
		$l(C) \downarrow$	
a) Nutze zur			$l(D), r(D)$ $l(E), w(E)$ $u(C), u(D), u(E)$
		$l(C), r(C)$ $l(D), w(D)$ $u(B), u(C), u(D)$	
	$l(B), r(B)$ $l(C), w(C)$ $u(A), u(B), u(C)$		

$w_3(E)$

Operationen.

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart 8

# Nachbespr

	$T_1$	$T_2$	$T_3$
	$sl(A), r(A)$		
		$sl(B), r(B)$	
$r_1(A), r_2(B),$			$sl(C), r(C)$
	$sl(B), r(B)$		
		$sl(C), r(C)$	
			$sl(D), r(D)$
b) Nutze zur Erstell	$xl(C) \downarrow$		
		$xl(D) \downarrow$	
			$xl(E), w(E)$
			$u(C), u(D), u(E)$
		$xl(D), w(D)$	
		$u(B), u(C), u(D)$	
	$xl(C), w(C)$		
	$u(A), u(B), u(C)$		

$v), w_3(E)$

Unlock-Operationen.

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart 9

# Nachbesprechung Zettel 5

---

```
<movie>
  <mid>Ghosts of the Past (1991) (TV)</mid>
  <title>Ghosts of the Past</title>
  <year>1991</year>
  <actors>
    <actor>
      <name>Davis, Carl (IV)</name>
    </actor>
    <actor>
      <name>McCartney, Paul</name>
    </actor>
  </actors>
</movie>
```

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **10**

# Nachbesprechung Zettel 5

---

- a) Gib ID, Titel und Jahr aus für alle Filme, in denen der Schauspieler mit dem Namen Affleck, Ben mitgespielt hat. **2 P**
- b) Gib ID, Titel und Jahr aus für alle Filme, in denen ein Schauspieler mitgespielt hat, dessen Name Kramer beinhaltet. **2 P**
- c) Gib ID, Titel, Jahr und zusätzlich die Anzahl der Schauspieler aus für alle Filme, in denen mehr als 50 Schauspieler mitgespielt haben. **3 P**

```
</actor>
<actor>
  <name>McCartney, Paul</name>
</actor>
</actors>
```

```
</movie>
```

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **11**

# Nachbesprechung Zettel 5

---

- a) Gib ID, Titel und Jahr aus für alle Filme, in denen der Schauspieler mit dem Namen `Affleck`, `Ben` mitgespielt hat.

```
select mid,  
       xpath('/movie/title/text()',moviexml.content) as year,  
       xpath('/movie/year/text()',moviexml.content) as title  
FROM moviexml  
where xmlexists('/movie/actors/actor/name[text()="Affleck, Ben"]' PASSING BY REF  
moviexml.content);
```

# Nachbesprechung Zettel 5

---

- b) Gib ID, Titel und Jahr aus für alle Filme, in denen ein Schauspieler mitgespielt hat, dessen Name `Kramer` beinhaltet. **2 P**

```
select mid,  
       xpath('/movie/title/text()',moviexml.content) as title,  
       xpath('/movie/year/text()',moviexml.content) as year  
from moviexml  
where xmlexists('/movie/actors/actor/name[contains(., "Kramer")]'  
PASSING BY REF  
moviexml.content);
```

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **13**

## Nachbesprechung Zettel 5

---

- c) Gib ID, Titel, Jahr und zusätzlich die Anzahl der Schauspieler aus für alle Filme, in denen mehr als 50 Schauspieler mitgespielt haben. **3 P**

```
select mid,  
    xpath('/movie/title/text()',moviexml.content) as title,  
    xpath('/movie/year/text()',moviexml.content) as year,  
    xpath('count(/movie/actors/actor)',moviexml.content) as Num_Actors  
from moviexml  
where xmlexists('/movie/actors[count(actor)>50]' PASSING BY REF moviexml.content);
```

**DBSI - Übung**

Besprechung

Leon Bornemann

Chart **14**

# Nachbesprechung Zettel 5

---

- Product(maker, model, type)
- PC(model, speed, ram, hd, rd, price)
- Laptop(model, speed, ram, hd, screen, price)
- Printer(model, color, type, price)

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **15**

# Wiederholung Scheduling Anomalien



**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **16**

- Dirty Read (Write-Read Konflikt)
  - „Lesen eines falschen Wertes“
  - $T_1$  schreibt Objekt A, das von  $T_2$  gelesen wird bevor  $T_1$  committed wurde; das Lesen ist dirty, wenn  $T_1$  Objekt A erneut schreibt oder zurückgerollt wird.

**DBSI - Übung**

Besprechung

Leon Bornemann  
Chart **17**

- Non-Repeatable Read (Read-Write Konflikt)
  - „Lesen eines veralteten Wertes“
  - $T_1$  liest Objekt A, das von  $T_2$  geschrieben wird bevor  $T_1$  committed wurde; das Lesen ist non-repeatable, weil  $T_1$  Objekt A erneut lesen könnte und dann einen anderen Wert erhält.

- Lost Update (Write-Write Konflikt)

- „Verlorenes Schreiben eines Wertes“

- $T_1$  schreibt Objekt A, das von  $T_2$  erneut geschrieben wird bevor  $T_1$  committed wurde; das Schreiben ist lost, weil  $T_1$  Objekt A erneut lesen könnte oder mit seinem Wert von A ein anderes Objekt inkonsistent ändern könnte.

Variation von Non-Repeatable Read  
Bzw. Kann dadurch entstehen

- Phantom Read (Read-Write und Write-Read Konflikt)
  - „Lesen inkonsistenter Werte“
  - Dirty Read + Non-Repeatable Read
  - $T_1$  liest mehrere Objekte A und B, die während des Lesens von  $T_2$  beschrieben werden; das Lesen liefert Phantom Werte, wenn Lesen und Schreiben asynchron verläuft.

# Wiederholung Themen der Übung

---

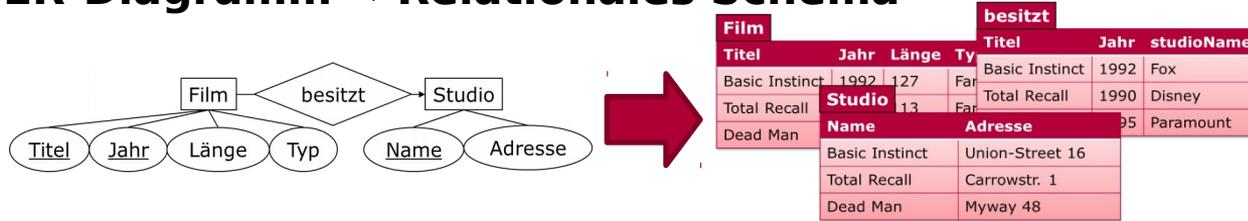


**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **21**

# Wiederholung Themen Übung 1

## 1. ER-Diagramm → Relationales Schema



## 2. Funktionale Abhängigkeiten und Schlüssel

Film			
Titel	Jahr	Länge	Typ
Basic Instinct	1992	127	Farbe
Total Recall	1990	113	Farbe
Dead Man	1995	121	s/w

Studio	
Name	Adresse
Basic Instinct	Union-Street 16
Total Recall	Carrowstr. 1
Dead Man	Myway 48

besitzt		
Titel	Jahr	studioName
Basic Instinct	1992	Fox
Total Recall	1990	Disney
Dead Man	1995	Paramount

## 3. BCNF und Dekomposition

Film					
Titel	Jahr	Länge	Typ	StudioName	SchauspName
Total Recall	1990	113	Farbe	Fox	Sharon Stone
Basic Instinct	1992	127	Farbe	Disney	Sharon Stone
Total Recall	1990	113	Farbe	Fox	Arnold
Dead Man	1995	121	s/w	Paramount	Johnny Depp

Film1					
Titel	Jahr	Länge	Typ	StudioName	
Total Recall	1990	113	Farbe	Fox	
Basic Instinct	1992	127	Farbe	Disney	
Dead Man	1995	121	s/w	Paramount	

Film2		
Titel	Jahr	SchauspName
Total Recall	1990	Sharon Stone
Basic Instinct	1992	Sharon Stone
Total Recall	1990	Arnold
Dead Man	1995	Johnny Depp

DBSI - Übung  
Besprechung

Leon Bornemann  
Chart 22

# Wiederholung Themen Übung 2

## 4. Relationale Algebra

### Unäre Operatoren

Operator	Beschreibung
$\pi$ (pi)	Projektion
$\sigma$ (sigma)	Selektion
$\delta$ (delta)	Duplikateliminierung
$\rho$ (rho)	Umbenennung
$\tau$ (tau)	Sortierung
$\gamma$ (gamma)	Gruppierung

### Binäre Operatoren

Operator	Beschreibung
$\cap$	Schnittmenge
$\cup$	Vereinigung
$-$	Differenz (auch „\“)
$\times$	Kreuzprodukt
$\bowtie$	Natürlicher Join
$\bowtie_{\theta}$	Theta-Join
$ \bowtie$	Left outer Join
$\bowtie $	Right outer Join
$ \bowtie $	Full outer Join
$\ltimes$	Semijoin

$$\pi_{\text{model}}(\sigma_{\text{speed} \geq 1000}(\text{PC}))$$

$$\pi_{\text{maker}}(\sigma_{\text{hd} \geq 10}(\text{Product} \bowtie \text{Laptop}))$$

$$\delta(\pi_{\text{maker}}(\text{Product} \bowtie \text{Laptop})) - \delta(\pi_{\text{maker}}(\text{Product} \bowtie \text{PC}))$$

$$\pi_{\text{hd}}(\sigma_{\text{Anzahl} > 2}(\gamma_{\text{hd}, \text{count}(\text{model}) \rightarrow \text{Anzahl}}(\text{PC})))$$

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **23**

# Wiederholung Themen Übung 3

## 5. SQL

**SELECT** <Attributliste>  
**FROM** <Relationenliste>  
**WHERE** <Bedingungen>  
**GROUP BY** <Gruppierungsattribute>  
**HAVING** <Bedingungen auf Gruppierungsattribute>  
**ORDER BY** <Attributliste>;

$\pi_{\text{model,price}}(\sigma_{\text{maker}='Apple'}(\text{Product} \bowtie (\pi_{\text{model,price}}(\text{PC}) \cup \pi_{\text{model,price}}(\text{Laptop}) \cup \pi_{\text{model,price}}(\text{Printer}))))$



Weitere Schlüsselwörter:

**DISTINCT, AS, JOIN**

**AND, OR**

**MIN, MAX, AVG, SUM, COUNT**

**NOT, IN, LIKE, ANY, ALL, EXISTS**

**UNION, EXCEPT, INTERSECT**

...

(**SELECT** Product.model, price  
**FROM** Product **JOIN** PC **ON** Product.model = PC.model  
**WHERE** maker = 'Apple')

**UNION**

(**SELECT** Product.model, price  
**FROM** Product **JOIN** Laptop **ON** Product.model = Laptop.model  
**WHERE** maker = 'Apple')

**UNION**

(**SELECT** Product.model, price  
**FROM** Product **JOIN** Printer **ON** Product.model = Printer.model  
**WHERE** maker = 'Apple');

## 6. Impedence Mismatch

- Generisches Modell (Programmiersprachen)
  - Pointer, verschachtelte Strukturen und Objekte
  - Schleifen und Verzweigungen
  - i.d.R. Imperativ
- Relationales Modell (DBMS)
  - Relationen und Attribute
  - Nebenbedingungen
  - Deklarativ

## 7. Embedded SQL, Stored Procedures, JDBC und Hibernate

Programmiersprache &  
Embedded SQL

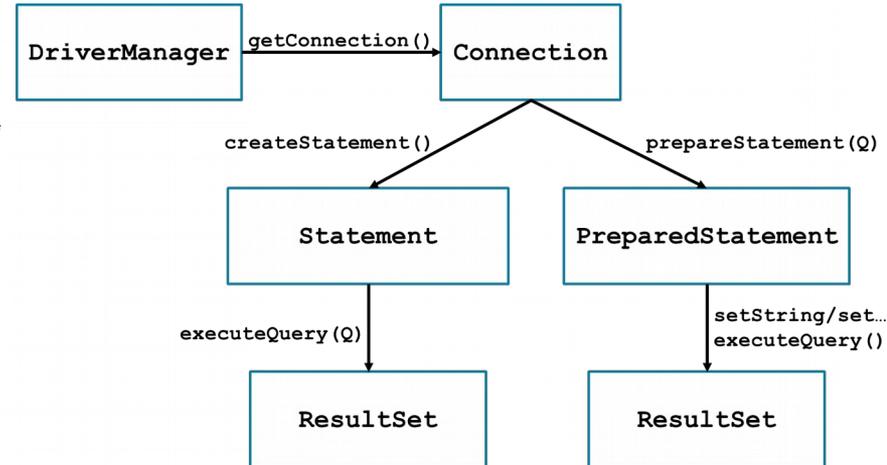


Programmiersprache &  
Funktionsaufrufe

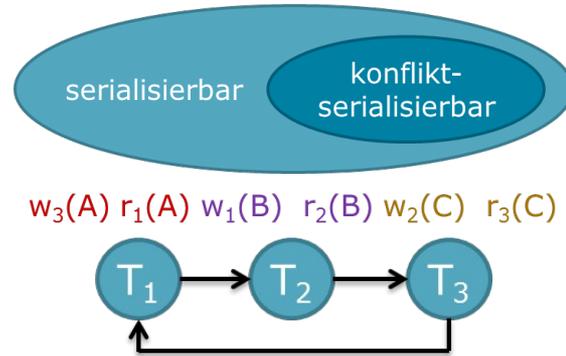


Ausführbares Programm

```
CREATE PROCEDURE MeanVar (IN studioName CHAR[15],  
                          OUT mittelwert REAL,  
                          OUT varianz REAL)  
  
DECLARE Not_Found CONDITION FOR SQLSTATE '02000';  
DECLARE FilmCursor CURSOR FOR  
    SELECT Laenge FROM Filme WHERE StudioName = studioName;  
DECLARE neueLaenge INTEGER;  
DECLARE filmAnzahl INTEGER;  
  
BEGIN  
    SET mittelwert = 0.0;  
    SET varianz = 0.0;  
    SET filmAnzahl = 0;  
    OPEN FilmCursor;  
    FilmLoop: LOOP  
        FETCH FilmCursor INTO neueLaenge;  
        IF Not_Found THEN LEAVE FilmLoop END IF;  
        SET filmAnzahl = filmAnzahl + 1;  
        SET mittelwert = mittelwert + neueLaenge ;  
        SET varianz = varianz + neueLaenge * neueLaenge;  
    END LOOP;  
    CLOSE FilmCursor;  
    SET mittelwert = mittelwert / filmAnzahl;  
    SET varianz = varianz / filmAnzahl - mittelwert * mittelwert;  
END;
```

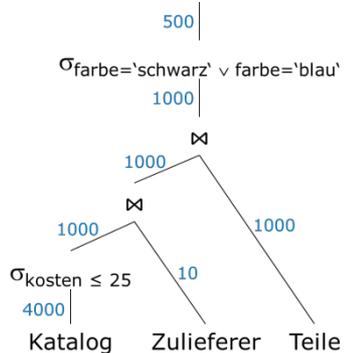


## 8. Transaktionen



$T_1$	$T_2$	$T_3$
$I_1(A) r_1(A)$	$I_2(B) r_2(B)$	$I_3(C) r_3(C)$
$I_1(B)$ ⚡	$I_2(C)$ ⚡	
	$I_2(C) r_2(C)$ $u_2(B) u_2(C)$	$I_3(D) w_3(D)$ $u_3(C) u_3(D)$
$I_1(B) r_1(B)$ $u_1(A) u_1(B)$		

## 9. Selektivität



## 3. XML

```
INSERT INTO CUSTOMER (cid, info)
VALUES (1000,
'<customerinfo cid="1000">
<name>Kathy Smith</name>
<addr country="Canada">
<street>5 Rosewood</street>
<city>Toronto</city>
<prov-state>Ontario</prov-state>
<pcode-zip>M6W 1E6</pcode-zip>
</addr>
<phone type="work">416-555-1358</phone>
</customerinfo>'
);
```

**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart 26

## Aufgabe 0: Matrikelnummern

Tragen Sie auf allen Blättern der Klausur Ihre Matrikelnummer ein. “Alle Blätter” umfasst das Deckblatt, die Aufgabenblätter und die Zusatzblätter. Tragen Sie die Matrikelnummer auch ein, falls Sie ein (Zusatz-)Blatt nicht verwenden.

*Hinweis:* Lösen Sie diese Aufgabe sofort.

**1 Punkt**

- Jede Aufgabe mind. 2x lesen!
- Bei Unklarheiten Fragen stellen
- Lösungen begründen
  - Das ist auch hilfreich wenn keine Begründung gefordert ist
- Für eine Lösung entscheiden
  - Bei mehr als einer Lösung nehmen wir die schlechtere!

**DBSI - Übung**

Besprechung

Leon Bornemann  
Chart **28**

Wiederholung  
Fragen zu Übungsinhalten



**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **29**

- **Übungsaufgaben:**
  - openHPI nutzen
  - Aufgaben ausdenken
  - Sonder- und Spezialfälle finden!
- **Klausurinhalte:**
  - Aufgaben der Übung
  - Inhalte der Vorlesung



**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart **30**

# Wiederholung Tipps

**LEARN DBMS**  
database management system

DBMS Tutorial

- DBMS - Home
- DBMS - Overview
- DBMS - Architecture
- DBMS - Data Models
- DBMS - Data Schemas
- DBMS - Data Independence

Entity Relationship Model

- DBMS - ER Model Basic Concepts
- DBMS - ER Diagram Representation
- DBMS - Generalization, Aggregation

Relational Model

- DBMS - Codd's Rules
- DBMS - Relational Data Model
- DBMS - Relational Algebra
- DBMS - ER to Relational Model
- DBMS - SQL Overview

LEARN DBMS  
simply easy learning

Previous Page Next Page

## Database Management System Tutorial

PDF Version Quick Guide Resources Job Search Discussion

Database Management System or DBMS in short refers to the technology of storing and retrieving users' data with utmost efficiency along with appropriate security measures. This tutorial explains the basics of DBMS such as its architecture, data models, data schemas, data independence, E-R model, relation model, relational database design, and storage and file structure and much more.

### Audience

This tutorial will especially help computer science graduates in understanding the basic-to-advanced concepts related to Database Management Systems.

### Prerequisites

Before you start proceeding with this tutorial, it is recommended that you have a good understanding of basic computer concepts such as primary memory, secondary memory, and data structures and algorithms.

Previous Page Print Next Page

Advertisements



**DBSI - Übung**  
Besprechung

Leon Bornemann  
Chart 31

<http://www.tutorialspoint.com/dbms/index.htm>



## Übung Datenbanksysteme I Besprechung

Leon Bornemann  
F-2.06, Campus II  
Hasso Plattner Institut