

# Datenbanken II A: DB-Entwurf

---

## Kapitel 0: Informationen zur Vorlesung

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2020/21

<http://www.informatik.uni-halle.de/~brass/dd20/>

# Inhalt

- 1 Inhalte
- 2 Organisatorisches
- 3 Prüfung, Hausaufgaben
- 4 Literatur
- 5 Software
- 6 Schlussbemerkung

# Wichtigste Lernziele

Nach dieser Vorlesung sollten Sie ...

- ein Datenbank-Schema auch für (etwas) größere Anwendungen erstellen können.
- Korrektheit und Qualität von DB-Schemata bewerten und alternative Lösungen vergleichen können.
- mehrere ER-Notation benutzen können.
- mit mindestens einem DB-Entwurfswerkzeug praktisch umgehen können (Oracle SQL Developer Data Modeler).
- Trigger und objektrelationale Features beherrschen.

# Themen

- Datenbank-Projekte: Übersicht
- Fortgeschrittener Entity-Relationship-Entwurf
  - Weitere ER-Konstrukte: Subklassen, mehrstellige Relationships, ...
- Oracle SQL Developer Data Modeler (ER-Entwurf)
  - Praktische Erfahrung mit einer etwas größeren Aufgabe ("Projekt").
- Übersetzung von ER-Modell ins Relationenmodell
- Relationale Normalformen (vertieft)
- UML Klassendiagramme
- Trigger, Stored Procedures, Objektrelationales SQL
  - Ggf. Reverse Engineering, Ontologien

# Voraussetzungen

Dies ist eine fortgeschrittene DB-Vorlesung.

Vorausgesetzt werden:

- Relationales Modell
- SQL, insbesondere `CREATE TABLE`
- Nullwerte, Schlüssel, Fremdschlüssel
- Erster Eindruck vom ER-Modell inklusive Klassifizierung von Relationships (z.B. "one-to-many").
- Programmierkenntnisse
- (mehr oder weniger:) Englische Sprachkenntnisse

# Inhalt

- 1 Inhalte
- 2 Organisatorisches**
- 3 Prüfung, Hausaufgaben
- 4 Literatur
- 5 Software
- 6 Schlussbemerkung

# Ansprechpartner (1)

## Vorlesung: Prof. Dr. Stefan Brass

- Email: [brass@informatik.uni-halle.de](mailto:brass@informatik.uni-halle.de)

Betreff-Zeile sollte mit [dd20] beginnen, möglichst aussagefähig.

- Büro: Von-Seckendorff-Platz 1, Raum 313

Ich mache viel „Home Office“, bin dort also eher selten zufällig anzutreffen.

- Sprechstunde (Büro, 0345/55-24740): Montags, 12<sup>15</sup>–13<sup>00</sup>

- Online-Sprechstunde: Montags, 16<sup>30</sup>–16<sup>45</sup> (und nach Bedarf),  
[<https://mluconf.uni-halle.de/b/ste-p6j-fmd-i21>]

- Frühere Unis: Braunschweig, Dortmund, Hannover, Hildesheim, Pittsburgh, Gießen, Clausthal.

- Oracle8 Certified Database Administrator (aktuell: 12C→18C,19C)  
IBM Certified Advanced DBA (DB2 UDB 8.1) (aktuell: 12)

## Ansprechpartner (2)

### Übungsleiter: M. Sc. Mario Wenzel

- Büro: Von-Seckendorff-Platz 1, Raum 315
- Telefon: 0345/55-24776
- Email: [mario.wenzel@informatik.uni-halle.de](mailto:mario.wenzel@informatik.uni-halle.de)

### Sekretärin: Ramona Vahrenhold

- Büro: Von-Seckendorff-Platz 1, Raum 324
- Telefon: 0345/55-24750, Fax: 0345/55-27333
- Email: [vahrenho@informatik.uni-halle.de](mailto:vahrenho@informatik.uni-halle.de)

# Zeit und Ort (1)

## Vorlesung (2 SWS):

- Mittwochs, 10<sup>30</sup>–12<sup>00</sup>, Raum 3.04 (18 Plätze).  
Und online asynchron als Aufzeichnung.
- Wegen der realen Gefahr durch Corona (COVID-19) ist möglichst großer Abstand zu halten und die Plätze im Hörsaal reichen nicht für alle Teilnehmer.

Momentan werden die beschränkten Plätze über eine Umfrage in [StudIP](#) vergeben. Nur wer dort „ja“ angekreuzt hat, darf kommen. Im Gebäude und im Hörsaal gilt Maskenpflicht (inzwischen selbst dann, wenn der Mindestabstand eingehalten werden kann). Es wird häufig gelüftet (ziehen Sie sich warm an). Nutzen Sie die [Corona-Warn-App](#).

- Die Aufzeichnung wird (mit etwas Verzögerung) über StudIP zugänglich gemacht (ggf. zusätzlich vorher anderes Video).

## Zeit und Ort (2)

### Übung (2 SWS, online):

- Zwei Gruppen (Anmeldung: StudIP):

Nr	Tag	Zeit	Raum	StudIP	Beginn
1	Mittwoch	13 <sup>00</sup> –14 <sup>30</sup>	(online)	[Link]	4.11.
2	Mittwoch	15 <sup>30</sup> –17 <sup>00</sup>	(online)	[Link]	4.11.

Gruppe 1: [<https://conf.informatik.uni-halle.de/mar-yi5-mbq-6td>]

Gruppe 2: [<https://conf.informatik.uni-halle.de/mar-h3q-oxw-w7x>]

Sie brauchen einen Zugangscode, den Sie im Forum im StudIP-Eintrag der Vorlesung finden: „/ Allgemein / Ankündigungen / Gruppenräume f. Übungen“.

Beide Gruppen sind eigentlich voll.

- “Hands-on Experience” ist wichtig!

Sie brauchen Software auf Ihrem Rechner. Ggf. Bildschirm sharen bei Problemen.

# Webseite

<http://www.informatik.uni-halle.de/~brass/dd20/>

- Folien
- Hausaufgaben
- Alte Klausuren
- Verweise auf CASE-Tool Hersteller etc.
- Verweise auf Literatur im WWW

Hinweise auf weitere interessante Quellen in Web (Links) sind willkommen!

# Inhalt

- 1 Inhalte
- 2 Organisatorisches
- 3 Prüfung, Hausaufgaben**
- 4 Literatur
- 5 Software
- 6 Schlussbemerkung

# Projekt statt Klausur (1)

- Bisher gab es in dieser Vorlesung ein Projekt als Studienleistung (plus nicht korrigierte Hausaufgaben) und eine Klausur als Prüfung.
- Wegen Corona werden Präsenzprüfungen problematisch.
- Ich möchte daher das Projekt als Prüfungsleistung verwenden. Genauer gibt es folgende Bestandteile:
  - Bericht (ca. 15–20 Seiten),
  - Skripte zum Anlegen der Datenbank,
  - Kurzvortrag zum Projekt (15 min),

Hier würden Ihre Kollegen/Kolleginnen zuhören. Ggf. kurze Diskussion.
  - mündliche Kurzprüfung als Verteidigung (ca. 15min).

Dies wäre online als Videokonferenz, nur mit Dozent und Übungsleiter.

## Projekt statt Klausur (2)

- Für eine Vorlesung über Datenbankentwurf ist ein praktisches Projekt ohnehin sehr wichtig.
  - Sie können die Anwendung selbst wählen.  
Etwas, was Sie interessant finden.
- Durch die Verwendung als Prüfungsleistung wird die Bedeutung des Projektes aufgewertet.
  - Die investierte Arbeit zahlt sicher eher in der Note aus.
- Projekte müssen einzeln bearbeitet werden.  
Kein Recyclen von Aufgabenlösungen aus anderen Vorlesungen, auch keine bezahlte Aufgabe.
- Da es eine Abweichung von der Modulbeschreibung ist, würden wir bei Einsprüchen der Teilnehmer noch nachdenken.

# Umfang des Projekts (1)

- Beginnen Sie den Bericht mit einer Einführung in die Anwendung und mindestens exemplarisch den Anforderungen (welche Fragen aus der Datenbank zu beantworten sind).
- Das ER-Schema sollte ca. 7–15 Entitytypen haben, möglichst auch fortgeschrittene Konstrukte (z.B. Subklassen) verwenden.

Im Bericht soll es dazu eine gute Dokumentation geben, nicht nur das reine Diagramm. Mindestens für den Vortrag wird es nötig sein, das Diagramm in mehrere Teile zu zerlegen. Erfahrungsgemäß sind die Diagramme nicht mehr lesbar, wenn sie als Ganzes auf eine Folie kommen. Selbstverständlich wäre die Zerlegung in mehrere Teildiagramme auch für den Bericht sinnvoll.

- Beim ER-Entwurf sollen auch Integritätsbedingungen dokumentiert werden.

## Umfang des Projekts (2)

- Die Übersetzung ins relationale Modell ist durchgeführt (bis zu den `CREATE TABLE`-Anweisungen), eventuelle Einschränkungen sind zu diskutieren.
- Es sind mindestens einige Beispieldaten zu laden.

Dafür könnte auch etwas Programmierung nötig sein, wenn Sie eine größere Datensammlung im Internet finden (in irgendeinem Format), und diese als Tabellenzeilen in die Datenbank laden wollen.
- Einige typische Anfragen sollen gezeigt werden.

Alternativ können Sie auch einige Sichten anlegen.

Um zu zeigen, dass das DB-Schema den Projektanforderungen gerecht wird.
- Diskutieren Sie Abhängigkeiten und Normalformen für mindestens eine Relation.

Die, für die es am interessantesten/kompliziertesten scheint.

## Umfang des Projekts (3)

- Es sind Trigger oder serverseitige Prozeduren und ggf. objektrelationale Konstrukte zu verwenden.

Mindestens hier ist Programmierung unbedingt nötig. Stellen Sie die Einhaltung der Integritätsbedingungen sicher, entweder mit Triggern, oder indem Sie die Änderung der Daten nur über Stored Procedures ("Methoden") erlauben.

- Ein Entwurfswerkzeug muss verwendet werden.

Falls Sie nicht Oracle SQL Developer Data Modeler verwenden wollen, sollten wir vorher darüber reden. Das Werkzeug muss eine Trennung von ER-Entwurf und relationalem Schema erlauben (kein Entwurf direkt im relationalen Modell, oder mit einer ER-Notation, die schon Fremdschlüssel anzeigt).

- Selbstverständlich können Sie Übungsleiter und Dozent bei Problemen um Rat fragen.

# Zeitplan

- Sie können fast das ganze Semester an Ihrem Projekt arbeiten.

Ein Teil der Übungszeit ist dafür vorgesehen.

- Abgabetermin für Projektberichte: 15. März 2021, 12<sup>00</sup>.

- Vorträge in der Woche: 22.–26. März 2021.

- Sie müssen mindestens zwei Tage vormittags alle Vorträge anhören.

Ca. 10 Vorträge pro Tag. Man lernt ja auch aus den anderen Vorträgen. Es ist explizit erwünscht, dass in den Vorträgen auch auf allgemeine Erfahrungen aus dem Projekt eingegangen wird, und nicht unbedingt das ganze Schema in allen Details besprochen wird.

- An dem Tag, an dem Ihr Vortrag liegt, machen wir nachmittags die Verteidigung.

# Studienleistung: Übung (1)

- Es gibt außerdem Hausaufgaben (in relativ kleinem Umfang).
- Sie müssen in Gruppen von 1–3 Teilnehmern mindestens die Hälfte abgeben (mit ernsthaftem Bemühen).
- Die Aufgaben werden nicht korrigiert, aber in der Übung besprochen.
- In der Übung müssen Sie Ihre Lösung vorführen und erklären können (inkl. Fragen zum Umfeld).
  - Ohne Unterstützung von ggf. anderen Gruppenmitgliedern.
- Verlangt wird außerdem regelmäßige und aktive Teilnahme an der Übung.

Sie dürfen nicht zu häufig fehlen (drei Mal wären akzeptabel, es gibt einen Ermessensspielraum des Übungsleiters, ggf. Zusatzaufgabe).

# Zeitliche Belastung

- Diese Vorlesung hat 5 Leistungspunkte.

Auch "credit points" genannt.

- Entspricht 150 Stunden studentischer Arbeitszeit:

Lernform	SWS	Stunden
Vorlesung	2	30
Selbststudium (, Projekt)	0	30
Übung	1	15
Praktische Übung, Projekt	1	15
Hausaufgaben	0	15
Projekt-Bericht	0	15
Vorbereitung Vortrag/Verteidigung	0	15
Seminar (Projekt-Vorstellung)	0	15

# Inhalt

- 1 Inhalte
- 2 Organisatorisches
- 3 Prüfung, Hausaufgaben
- 4 Literatur**
- 5 Software
- 6 Schlussbemerkung

# Lehrbücher (1)

- Toby J. Teorey et al.:  
Database Modeling and Design. Logical Design.  
Weitere Autoren: Sam S. Lightstone, Tom Nadeau, H. V. Jagadish. Morgan Kaufmann, 2011, 5th Ed., ISBN 0123820200, 368 pages, 50.24 Euro.
- Sam Lightstone, Toby Teorey, Tom Nadeau:  
Physical Database Design.  
Morgan Kaufmann, 2007, ISBN 0123693896, 448 pages, 43.82.  
→ mehr für Datenbanken II B.
- Database Design: Know it All (Know It All)  
Viele Autoren, u.a. Teorey (jeder 1–3 Kapitel).  
Morgan Kaufmann, 2008, ISBN 0123746302, 349 Seiten, 32.27 Euro.

## Lehrbücher (2)

- Graeme C. Simsion, Graham C. Witt:  
Data Modeling Essentials, 3rd Edition.

Morgan Kaufmann, 2004, ISBN 0126445516, 550 pages, 48.45 Euro.

- Graeme Simsion:  
Data Modeling: Theory and Practice.

Technics Publications, 2007, ISBN 0977140016, 400 pages, 47.99.

Aus Dissertation entstanden, nicht neuere Auflage obigen Buches.

- Bernhard Thalheim: Entity-Relationship Modeling.  
Foundations of Database Technology.

Springer Verlag, 2000, ISBN 3540654704, 627 pages, 64.15 Euro.

## Lehrbücher (3)

- Antoni Olivé:  
Conceptual Modeling of Information Systems.  
Springer, 2007, ISBN 3540393897, 455 pages, 58.80 Euro.
- Otto Rauh, Eberhard Stickel:  
Konzeptuelle Datenmodellierung.  
Teubner, 1997, ISBN 3-8154-2601-4.
- Carlo Batini, Stefano Ceri, Shamkant B. Navathe:  
Conceptual Database Design.  
Benjamin/Cummings, 1992, ISBN 0-8053-0244-1, 470 pages.

## Lehrbücher (4)

- Richard Barker:  
CASE\*Method: Entity Relationship Modelling.  
Addison-Wesley, 1990, ISBN 0-201-41696-4, nur noch gebraucht.
- Heli Helskyaho: Oracle SQL Developer Data Modeler for Database Design Mastery.  
McGraw Hill Education / Oracle Press, 2015, ISBN 0071850090,  
336 Seiten.
- Peter Koletzke, Paul Dorsey:  
Oracle Designer Handbook, 2nd Edition.  
ORACLE Press, 1998, ISBN 0-07-882417-6, 1075 pages, nur noch  
gebraucht.

## Lehrbücher (5)

- Grady Booch, James Rumbaugh, Ivar Jacobson:  
The Unified Modeling Language User Guide.

Addison Wesley, 1999, ISBN 0-201-57168-4, 482 pages, 50 Euro.

- Martin Fowler, Kendall Scott:  
UML Distilled, Second Edition.

Addison-Wesley, 2000, ISBN 0-201-65783-X, 185 pages, 35 Euro.

- Robert J. Muller: Database Design for Smarties — Using  
UML for Data Modeling.

Morgan Kaufmann, 1999, ISBN 1-55860-515-0, 442 pages, 47 Euro.

# Inhalt

- 1 Inhalte
- 2 Organisatorisches
- 3 Prüfung, Hausaufgaben
- 4 Literatur
- 5 Software**
- 6 Schlussbemerkung

# Oracle SQL Developer Data Modeler

- In der Übung: Oracle SQL Developer Data Modeler.

Installiert im PC-Pool 3.32 und im Thin Client Pool 3.34. In Corona-Zeiten kommen Sie praktisch nicht drumrum, es auch auf Ihrem Rechner zu Hause zu installieren. Bei großen Problemen bitte melden.

- Er ist in Java geschrieben, kostenlos erhältlich und läuft auch unter Linux.

[\[https://www.oracle.com/technetwork/developer-tools/datamodeler/\]](https://www.oracle.com/technetwork/developer-tools/datamodeler/)

[\[https://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html\]](https://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html)

[\[https://docs.oracle.com/en/database/oracle/sql-developer-data-modeler/\]](https://docs.oracle.com/en/database/oracle/sql-developer-data-modeler/)

- Es wird Java/Fx verwendet.

Dies ist im original Oracle JDK enthalten. Mit OpenJDK 8 gibt es Schwierigkeiten. OpenJDK 11 scheint zu funktionieren, obwohl sich der Data Modeler beschwert, dass er maximal mit Java-Version 9 getestet ist.

- Die Entwurfsdaten werden in XML Dateien gespeichert.

## Weitere Oracle Entwurfs-Software

- Der “Oracle Designer” wird nicht mehr weiterentwickelt, verwendet aber die gleiche ER-Notation und ist teils technisch noch interessant.

Der Oracle Designer speicherte die Entwurfs-Daten in einer relationalen Datenbank. Das musste unbedingt eine Oracle-Datenbank sein, da die Möglichkeiten zur serverseitigen Programmierung intensiv genutzt wurden. Teils war sogar die „Enterprise Edition“ der Datenbank verlangt.

- Es gibt auch den Oracle JDeveloper als IDE.

[<http://www.oracle.com/technetwork/developer-tools/jdev/>]

Unterstützt u.a. Java, UML, HTML/JSP/JSF, Web Services, SQL, Oracle ADF (Application Development Framework, erlaubt visuelle, deklarative Programmierung von Web-Datenbank Anwendungen).

# Andere DB-Entwicklungswerkzeuge (1)

- Sybase Powerdesigner

[<http://powerdesigner.de/>]

Im Pool installiert.

- ERwin

[<http://erwin.com/products/data-modeler>]

Wird im Pool installiert.

- Rational Rose

[<http://www.ibm.com/software/products/en/datamodeler>]

Im Pool installiert (von Softwaretechnik-Gruppe).

- Oracle JDeveloper

[<http://www.oracle.com/technology/products/jdev/index.html>]

## Andere DB-Entwicklungswerkzeuge (2)

- DB-MAIN

[<http://www.rever.eu/en/content/db-main-homepage>]

- ER Studio

[<https://www.idera.com/er-studio-data-architect-software>]

- Borland Together

[<https://www.borland.com/en-GB/Products/Requirements-Management/Together>]

- Microsoft Visio

[<http://www.microsoft.com/office/visio/prodinfo/default.mspx>]

## Andere DB-Entwicklungswerkzeuge (3)

- Datanamic DeZign

[<http://www.datanamic.com/>]

- CASE Studio: Toad Data Modeler

[<http://www.casestudio.com/>]

- Dia: A Drawing Program

[<https://wiki.gnome.org/Apps/Dia>] [<http://dia-installer.de/>]

- Schemaspy: Visualisierung relationaler Schemata

[<http://schemaspy.sourceforge.net/>]

## Andere DB-Entwicklungswerkzeuge (4)

- Visual Paradigm: Database Visual Architect

[<http://www.visual-paradigm.com/product/dbva/>]

- IBM Rational Architect (Integration)

[<http://www-306.ibm.com/software/data/integration/rda/>]

- Toad Data Modeler

[<http://www.quest.com/Toad-Data-Modeler/>]

- Xcase DB Design Software / Data Modeling Tool

[<http://www.xcase.com/>]

- Open Modelsphere (data, process, UML modeling)

[<http://www.modelsphere.org/>] (open source, früher kommerziell)

## Andere DB-Entwicklungswerkzeuge (5)

- DBDesigner 4

[<http://www.fabforce.net/dbdesigner4/>] (open source)

- In der Wikipedia gibt es einen Vergleich verschiedener “Data Modeling Tools”.

[[http://en.wikipedia.org/wiki/Comparison\\_of\\_data\\_modeling\\_tools](http://en.wikipedia.org/wiki/Comparison_of_data_modeling_tools)]

Kriterien sind u.a.: License (Commercial, Free, etc.), Supported Database Platforms, Supported Operating Systems (Windows, Linux, etc.), Supported Data Models (Conceptual, logical, physical), Supported Notations (IDEF1X, Crows feet, etc.), Forward Engineering, Reverse Engineering, Model/database synchronization, Support of Teamwork.

# Software: Bemerkung

- Mit gewissen Einschränkungen kommt es für diese Vorlesung nicht sehr auf das Werkzeug an.

Mir ist eine klare Trennung zwischen ER-Ebene und relationaler Ebene wichtig, manche Werkzeuge vermischen das. Auch sonst ist der Funktionsumfang der Werkzeuge natürlich unterschiedlich.

- Probieren Sie verschiedene Werkzeuge aus!

Sie sollen einen Eindruck davon gewinnen, was solche Werkzeuge für Sie tun können.

- Es ist auch ein Lernziel der Vorlesung, dass Sie mit verschiedenen ER-Notationen arbeiten können.

In Klausur gegebene ER-Schemata voraussichtlich in Oracle Designer Notation ("Barker Notation") oder als UML Klassendiagramm.

# Software: DBMS

- Sie müssen den erstellten SQL-Code auch ausprobieren!

Syntaxfehler darin würden Ihre Note deutlich verschlechtern.

- In dieser Vorlesung sollen auch Trigger und serverseitige Programmierung sowie objektrelationale Konstrukte unterrichtet werden.
- Diese Dinge sind sehr systemabhängig.
- Die Vorlesung bezieht sich dabei auf PostgreSQL.

Bei Bedarf wird Ihnen ein Account zur Verfügung gestellt, mit dem Sie per Remote Login auf einem unserer Rechner mit PostgreSQL arbeiten können.

Es empfiehlt sich aber, das selbst zu installieren: [<https://www.postgresql.org/>]

# Inhalt

- 1 Inhalte
- 2 Organisatorisches
- 3 Prüfung, Hausaufgaben
- 4 Literatur
- 5 Software
- 6 Schlussbemerkung**

# Verbesserung der Lehre

- Gute Lehre ist für mich wichtig.  
Ich möchte ein Datenbank-Lehrbuch schreiben.
- Vorschläge zur Verbesserung der Vorlesung sind sehr willkommen. Fragen sind sehr willkommen.

Für Hinweise auf Fehlern in den Vorlesungs-Materialien, sowie nützliche Links für die WWW-Seite wäre ich besonders dankbar.

- Eine Vorlesung, gerade auch auf diesem Gebiet, lebt ganz wesentlich auch von der Diskussion.

Es gibt öfters nicht die eine perfekte Lösung. Leider sind auch nicht alle Qualitätskriterien 100% objektiv.