

DB II B: DBMS-Implementierung

Übung 2: Data Dictionaries

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2021/22

<http://www.informatik.uni-halle.de/~brass/dbi21/>

Inhalt

- 1 Klausurtermin
- 2 Software
- 3 Zugriff auf eine Oracle Datenbank
- 4 Oracle Data Dictionary

Klausurtermin

- Es wurde angemerkt, dass die bisherigen Termine (31.03.2022, 29.09.2022) ungünstig sind für Studierende, die ihr Studium in diesem Semester abschließen.

5 LP Module setzen ca. 150 Stunden durchschnittliche Arbeitszeit der Studierenden voraus. Wer z.B. SQL nicht kann, wird sicher mehr arbeiten müssen. Wenn man 30 LP pro Semester schaffen will, braucht man noch Zeit in den Semesterferien (sonst hat man 60 Stunden-Wochen). Es ist ausdrücklich erwünscht, dass Sie noch ein Buch lesen, oder ggf. mit anderen Materialien die für Selbststudium geplante Zeit wirklich erbringen. Die Klausur kann daher nicht am Anfang der Semesterferien liegen.

- Es wurde der **10.03.2022** in Aussicht genommen.

Da außerdem mehrheitlich eine elektronische Klausur gewünscht wurde, muss noch festgestellt werden, ob der Prüfungsraum zu diesem Zeitpunkt frei ist. Für Durchfaller, die ihr Studium abschließen wollen, wird es Ende März/Anfang April einen Termin für mündliche Nachprüfungen geben. Der normale zweite Termin bleibt im September.

Inhalt

- 1 Klausurtermin
- 2 Software
- 3 Zugriff auf eine Oracle Datenbank
- 4 Oracle Data Dictionary

Software in dieser Vorlesung

- In der Übung: Oracle (mit DBA-Rechten).

Der Dozent ist Oracle Certified Professional (Oracle 8 DBA). Diese Vorlesung ist daher etwas Oracle-lastig. Ich begrüße es aber ausdrücklich, wenn Sie andere Systeme ausprobieren und mir über Gemeinsamkeiten und Unterschiede berichten. Letztendlich ist diese Vorlesung kein Oracle-Kurs, sondern wir wollen die allgemeinen Prinzipien und Probleme studieren. Da DBMS-Interna aber notwendigerweise sehr systemspezifisch sind, und wir das zumindest exemplarisch konkret machen wollen, tun wir das anhand von Oracle. Es ist das System, mit dem ich mich am besten auskenne.

- Geplant ist, auch PostgreSQL auszuprobieren, und (gemeinsam) etwas über die Implementierung zu lernen.

Es ist ja interessant, die Lösungen für bestimmte Probleme in beiden Systemen zu vergleichen.

DB Software: Marktanteile 2017 (IDC)

Relational Database Vendor Share 2017 [IDC]

Anbieter	Einnahmen	Marktanteil
Oracle	14,0 Mrd. \$	42%
Microsoft	8,0 Mrd. \$	24%
IBM	4,2 Mrd. \$	13%
SAP/Sybase	2,3 Mrd. \$	7%
Teradata	1,0 Mrd. \$	3%
Andere	3,7 Mrd. \$	11%

Marktgröße (2017): 33 Mrd. US-Dollar.

Zitiert nach: Database Software Market: The Long Awaited Shake-up
(Jason Adler, Billy Fitzsimmons, Sebastian Naji - William Blair
Equity Research)

[<https://blocksandfiles.com/wp-content/uploads/2019/03//>]

[Database-Software-Market-White-Paper.pdf]

DB Software: Popularität im Netz

Rang	System	Art	Punkte
1.	Oracle	Relational DBMS	1270
2.	MySQL	Relational DBMS	1220
3.	Microsoft SQL Server	Relational DBMS	971
4.	PostgreSQL	Relational DBMS	587
5.	MongoDB	Document Store	494
6.	Redis	Key-Value Store	171
7.	DB2	Relational DBMS	166
8.	Elasticsearch	Suchmaschine	158
9.	SQLite	Relational DBMS	129
10.	Cassandra	Wide Column Store	119

[<https://db-engines.com/de/ranking>] Alle großen nicht nur relational („Multi-Model“).

Es fließen u.a. ein: Anzahl Treffer in Suchmaschinen (Webseiten), Google Trends (Suchanfragen), Stack Overflow, DBA Stack Exchange, Job-Angebote auf Indeed und Simply Hired, Profile auf LinkedIn und Upwork, die das System nennen, Twitter.

DB Software: Verwendung durch Entwickler

Rang	System	Entwickler
1.	MySQL	48.2%
2.	PostgreSQL	44.0%
3.	SQLite	30.9%
4.	Microsoft SQL Server	29.4%
5.	MongoDB	28.0%
6.	Redis	24.5%
7.	MariaDB	17.1%
8.	Firebase	15.9%
9.	Elasticsearch	15.7%
10.	Oracle	12.9%

Quelle: Stack Overflow Developer Survey 2021 (jeweils „Professional Developers“)

[<https://insights.stackoverflow.com/survey/2021>]

Unter „Programming, Scripting and Markup Languages“ steht SQL (50.7%) an dritter Stelle hinter JavaScript (68.6%) und HTML/CSS (56.0%).

Software: Oracle

- Verbreitetes kommerzielles System. Vermutlich Marktführer.

Ich persönlich habe mit Oracle gearbeitet seit meinem Studium.

Es ist das System, was ich am besten kenne. Recht gut standard-kompatibel.

Oracle war die erste kommerzielle SQL-Datenbank überhaupt (1979).

- Die “Express Edition” kostet nichts:

[<https://www.oracle.com/database/technologies/appdev/xe.html>]

Gibt es für Windows und Linux. Bei Linux funktioniert jedenfalls die Installation unter CentOS problemlos (ähnlich zu Red Hat Linux).

Maximal 4 GB/11 GB Datenbank, 1 Kern, 1 GB RAM, u.a. Einschränkungen.

[<https://blogs.oracle.com/oraclemagazine/post/deliver-oracle-database-18c-express-edition-in-containers>]: Installation in Docker.

- Oracle ist auch Anbieter von betriebswirtschaftlicher Standard-Software (Konkurrent zu SAP) und Cloud-Anbieter.

Software: IBM DB2

- Weiteres wichtiges kommerzielles DBMS.

Edgar F. Codd, der Erfinder des relationalen Datenbankmodells, arbeite im IBM Forschungslabor in San Jose, als der diese Ideen entwickelt hat. Sein Artikel "A Relational Model of Data for Large Shared Data Banks" erschien 1970 (in den Communications der ACM). 1981 bekam er den Turing Preis. SEQUEL, eine frühere Version von SQL, wurde von Chamberlin, Boyce et al. 1974 im gleichen Forschungslabor entwickelt. System/R, einer der beiden ersten Forschungs-Prototypen einer relationalen Datenbank, wurde dort 1976/77 entwickelt. Der andere war Ingres von Michael Stonebraker (Berkeley).

- Kostenlose Express-C Version (läuft aus?)

[<https://www-01.ibm.com/marketing/iwm/iwm/web/pickUrxNew.do?source=swg-db2expressc>]

Für Entwickler gibt es diese neueren Versionen, die zum Test kostenlos sind (?):

[<https://www.ibm.com/us-en/marketplace/ibm-db2-direct-and-developer-editions/purchase>]

PostgreSQL (1)

- POSTGRES wurde von Michael Stonebraker (+ Team) an der University of California at Berkeley ab 1985 entwickelt.
Michael Stonebraker hatte zuvor das INGRES-System entwickelt (ebenfalls in Berkeley), eines der beiden ersten relationalen Datenbanksysteme (das andere war System R, Forschungsprototyp von IBM). Michael Stonebraker bekam 2014 den Turing Award, eine Art Nobelpreis für Informatik [<https://amturing.acm.org/byyear.cfm>].
- POSTGRES war Vorreiter der objektrelationalen Datenbanken.
Michael Stonebraker gründete zusammen mit anderen die Firma Illustra Information Technologies, die eine kommerzielle Version von POSTGRES herausbrachte. Sie wurde 1997 an Informix verkauft (Informix 2001 von IBM).
- 1994 entwickelten Andrew Yu und Jolly Chen (Studenten in Berkeley) eine Version mit SQL-Unterstützung (statt der Datenbank-Sprache QUEL/POSTQUEL).

PostgreSQL (2)

- Im Ranking von DB-Engines.com belegt PostgreSQL den vierten Platz (hinter Oracle, MySQL, Microsoft SQL Server).
[<https://db-engines.com/de/ranking>]
Siehe auch [<https://www.datanyze.com/market-share/databases>].
- Folgende Plattformen werden mit fertigen Installations-Paketen unterstützt [<https://www.postgresql.org/download/>]:
 - Linux (Red Hat/CentOS/Fedora, Debian, Ubuntu, SuSE)
 - BSD (FreeBSD, OpenBSD)
 - Windows
 - macOS
 - Solaris
- Der Quellcode ist auch frei verfügbar.

Open Source DBMS (1)

- Es gibt inzwischen recht viele Open Source DBMS, bei denen man also den Quellcode einsehen kann:

- PostgreSQL

[<https://www.postgresql.org/ftp/source/>]

[https://wiki.postgresql.org/wiki/Developer_FAQ]

[<http://www.postgresql.org/developer/coding/>]

[<http://doxygen.postgresql.org/index.html>]

- MySQL (gehört seit 2010 Oracle)

[<https://dev.mysql.com/doc/internals/en/>]

- MariaDB (Abspaltung von MySQL)

[<https://mariadb.com/kb/en/mariadb/>

[source-getting-the-mariadb-source-code/](https://mariadb.com/kb/en/mariadb-internal-source-getting-the-mariadb-source-code/)]

[<https://mariadb.com/kb/en/library/mariadb-internal-documentation/>]

Open Source DBMS (2)

- Weitere Open Source DBMS:

- SQLite

[<https://www.sqlite.org/src/doc/trunk/README.md>]

- HSQLDB

[<http://hsqldb.org/web/hsqldbDevelopment.html>]

- H2 Database

[<https://www.h2database.com/html/main.html>]

- Firebird

[<http://www.firebirdsql.org/en/code-repository/>]

- Apache Derby

[<https://db.apache.org/derby/>]

Vormals Cloudscape, dann IBM Cloudscape, IBM DB2 SQL Syntax.

War auch mal als “Java DB” im JDK enthalten.

Installations-Erfahrungen (2)

- Haben Sie eine andere Datenbank auf Ihrem Rechner?
 - A. MySQL oder MariaDB
 - B. Microsoft Access
 - C. Microsoft SQL Server
 - D. Anderes DBMS
 - E. Nein.
- Weiteres Web-Interface (sogar mit verschiedenen DBMS):

[\[http://sqlfiddle.com/\]](http://sqlfiddle.com/)

Links geben Sie CREATE TABLE-Befehle und INSERT-Anweisungen ein,
z.B. [\[https://users.informatik.uni-halle.de/~brass/db21/sql/db0.sql\]](https://users.informatik.uni-halle.de/~brass/db21/sql/db0.sql).

Dann klicken Sie auf „Build Schema“. Anschließend geben Sie rechts Ihre
Anfrage ein und klicken auf „Run Query“.

Inhalt

- 1 Klausurtermin
- 2 Software
- 3 Zugriff auf eine Oracle Datenbank**
- 4 Oracle Data Dictionary

Oracle über Adminer Webschnittstelle (1)

- Auf dem Rechner `dbs2.informatik.uni-halle.de` läuft eine Oracle 18c Express Edition in einem Docker Container.
- Man kann darauf mit der Web-Schnittstelle Adminer zugreifen:

`https://dbs.informatik.uni-halle.de/db2b/adminer?
oracle=oracle-18.4-xe-db2b%2FXEPDB1&
username=scott&db=USERS`

- Die Eingaben im Login-Formular des Adminers sind:
 - System: Oracle (beta) auswählen
 - Server: `oracle-18.4-xe-db2b/XEPDB1`
 - Username: `scott`
 - Password: (Wird in Übung bekannt gegeben, nicht `Tiger`)
 - Database: `USERS`

Oracle über Adminer Webschnittstelle (2)

- Der Adminer ist nicht speziell für Oracle gemacht, sondern funktioniert mit vielen Datenbanken.

Die Anpassung an Oracle ist auch nicht besonders gut.

- Was im Adminer als „Database“ angezeigt wird (**USERS**), ist tatsächlich ein „Tablespace“.

Ein Tablespace ist ein physischer Container für Tabellen, also eine Datei oder ggf. auch mehrere auf der Platte. Darin können Tabellen gespeichert werden. Es gibt auch Tablespaces für spezielle Aufgaben, in denen man keine Tabellen speichern kann (z.B. Undo-Tablespaces).

- Oracle hat eine 1:1 Beziehung zwischen Schemata und Nutzern. Deswegen kann man normalerweise kein Schema auswählen.

Wenn man Nutzer mit Administrator-Rechten ist, kann man schon ein Schema/Nutzer wählen, dessen Tabellen aufgelistet werden.

Oracle über Adminer Webschnittstelle (3)

- Klassisch hat jeder Oracle Server genau eine Datenbank verwaltet (mit vielen Nutzern/Schemata).
- Seit Oracle 12c gibt es die „Multitenant Architecture“ mit
 - einer „Container Database (CDB)“ und
 - vielen „Pluggable Databases (PDBs)“.
- Damit kann ein Oracle Server jetzt doch mehrere Datenbanken verwalten.

Wie der Name schon sagt, kann man „pluggable databases“ recht einfach zwischen verschiedenen Servern verschieben. Wahrscheinlich wurde dieses Konzept auch eingeführt, um als Cloud-Anbieter vielen Kunden effizient „eigene Datenbanken“ zur Verfügung stellen zu können.

- „**XEPDB1**“ ist eine „Pluggable Database“, die bei der Installation der Express Edition (XE) angelegt wird.

Datenbank-Nutzer (1)

- Ich habe Nutzer für alle in StudIP eingetragenen Teilnehmer eingerichtet, wobei der Nutzernamen dem Nachnamen entspricht:
 - Umlaute habe ich wie üblich durch ASCII-Zeichen ersetzt (ae, oe, ue, ss).
 - Bei Studenten mit gleichem Nachnamen habe ich den Vornamen davor gehängt (wie in **StefanBrass**).
 - Da Nutzernamen SQL-Bezeichner sind, ist die Groß-/Kleinschreibung nicht relevant.

Vermutlich gilt das bei Oracle auch für Passworte.
 - „Van der ...“ habe ich durch „**Vd...**“ ersetzt.
- Passwort wird in der Übung bekannt gegeben.

Datenbank-Nutzer (2)

- Ich habe folgenden Befehl ausgeführt:

```
CREATE USER Brass IDENTIFIED BY "...";
```

Dabei ist „...“ das Passwort.

- Der so angelegte Nutzer hat zunächst keine Rechte.

Er kann sich nicht einmal einloggen.

- Ich habe daher noch folgenden Befehl ausgeführt:

```
GRANT CONNECT, RESOURCE, CREATE VIEW TO Brass;
```

- Sie können Ihr Passwort ändern mit folgendem Befehl:

```
ALTER USER Brass IDENTIFIED BY "...";
```

Nutzung des Adminers (1)

- Wenn Sie auf einen Tabellennamen klicken, bekommen Sie die Schema-Information zur Tabelle.
- Wenn Sie auf „zeigen“ links neben dem Tabellennamen klicken, werden die Tabellendaten (der Zustand) angezeigt.
- Wenn Sie auf „SQL-Kommando“ im Menu links klicken, bekommen Sie eine Dialogbox für SQL-Befehle.
- Es ist egal, ob Sie SQL-Anweisungen mit Semikolon abschließen oder nicht.

Nach dem SQL-Standard gehört das Semikolon nicht zur Anfrage.

In einigen Schnittstellen ist es aber nötig, um die Anfrage zu beenden, bzw. um mehrere Anfragen von einander zu trennen.

Nutzung des Adminers (2)

- Unter „Import“ können Sie auch eine Datei mit SQL-Kommandos hochladen und ausführen.

Kurze Skripte können Sie auch mit Copy&Paste in der SQL-Eingabebox kopieren. Sie erlaubt mehrere Befehle hintereinander.

- [<https://www.adminer.org/de/>]

[<https://en.wikipedia.org/wiki/Adminer>]

[<https://sourceforge.net/p/adminer/discussion/>]

- Hinweis: In der Liste der Tabellen werden nicht nur Ihre eigenen Tabellen angezeigt, sondern auch andere Tabellen, auf die Sie Zugriff haben (aber nicht das Data Dictionary).

Z.B. sehen Sie da die Tabelle EMP. Wenn Sie dann „**SELECT * FROM EMP**“ eingeben, bekommen Sie eine Fehlermeldung, dass die Tabelle nicht existiert. Sie haben Zugriff auf die Tabelle EMP des Nutzers SCOTT. Sie müssen daher „**SELECT * FROM SCOTT.EMP**“ eingeben.

Beispiel-Datenbank von Oracle (1)

- Die klassische Beispiel-Datenbank von Oracle enthält eine Tabelle **EMP** mit Daten von Angestellten („employees“):
 - **EMPNO**: Angestellten-Nummer (Primärschlüssel)
 - **ENAME**: Angestellten-Name
 - **JOB**: Berufsbezeichnung
 - **MGR**: Angestellten-Nummer des direkten Vorgesetzten
„Manager“. Dies ist ein Fremdschlüssel auf die Tabelle selbst.
Nullwerte sind erlaubt: Der Präsident der Firma hat keinen Vorgesetzten.
 - **HIREDATE**: Datum der Anstellung
 - **SAL**: Gehalt
 - **COMM**: Provision (nur für Verkäufer, d.h. JOB='SALESMAN')
 - **DEPTNO**: Abteilungs-Nummer (Fremdschlüssel zu DEPT)

Beispiel-Datenbank von Oracle (2)

EMP					
EMPNO	ENAME	JOB	MGR	SAL	DEPTNO
7369	SMITH	CLERK	7902	800	20
7499	ALLEN	SALESMAN	7698	1600	30
7521	WARD	SALESMAN	7698	1250	30
7566	JONES	MANAGER	7839	2975	20
7654	MARTIN	SALESMAN	7698	1250	30
7698	BLAKE	MANAGER	7839	2850	30
7782	CLARK	MANAGER	7839	2450	10
7788	SCOTT	ANALYST	7566	3000	20
7839	KING	PRESIDENT		5000	10
7844	TURNER	SALESMAN	7698	1500	30
7876	ADAMS	CLERK	7788	1100	20
7900	JAMES	CLERK	7698	950	30
7902	FORD	ANALYST	7566	3000	20
7934	MILLER	CLERK	7782	1300	10

Beispiel-Datenbank von Oracle (3)

- Die Tabelle **DEPT** enthält Informationen über Abteilungen:
 - **DEPTNO**: Abteilungs-Nummer (Primärschlüssel)
 - **DNAME**: Abteilungs-Name
 - **LOC**: Sitz der Abteilung (Name einer Stadt)

DEPT		
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Beispiel-Datenbank von Oracle (4)

- Schema in Kurznotation:

- **DEPT**(DEPTNO, DNAME°, LOC°)

Tatsächlich enthält die Tabelle keine Nullwerte.

- **EMP**(EMPNO, ENAME°, JOB°, MGR° → EMP, HIREDATE°, SAL°, COMM°, DEPTNO° → DEPT)

Tatsächlich Nullwerte enthalten sind in nur in MGR (Mr. King hat keinen Vorgesetzten), COMM (nur Angestellte mit JOB = 'SALESMAN' bekommen eine Provision/„commission“) und DEPTNO (Mr. Smith ist keiner Abteilung zugeordnet). Es ist ein schlechter Entwurf, dass Nullwerte z.B. auch im Namen erlaubt sind.

- **SALGRADE**(GRADE, LOSAL°, HISAL°)

Diese Tabelle enthält Gehaltsstufen mit jeweils einem Intervall von möglichen Gehältern.

- [<https://users.informatik.uni-halle.de/~brass/dbi21/sql/empdept.sql>]

Beispiel-Datenbank von Oracle (5)

- Die Beispiel-Datenbank wurde im Laufe der Zeit erweitert:
 - DUMMY(DUMMY)
 - CUSTOMER(CUSTID, NAME,
ADDRESS, CITY, STATE, ZIP,
AREA, PHONE,
REPID, CREDITLIMIT, COMMENTS)
 - ORD(ORDID, ORDERDATE, COMMPLAN,
CUSTID→CUSTOMER, SHIPDATE, TOTAL)
 - ITEM(ORDID→ORDER, ITEMID, PRODID→PRODUCT,
ACTUALPRICE, QTY, ITEMTOT)
 - PRODUCT(PRODID, DESCRIP)
 - PRICE(PRODID, STDPRICE, MINPRICE,
STARTDATE, ENDDATE)

Inhalt

- 1 Klausurtermin
- 2 Software
- 3 Zugriff auf eine Oracle Datenbank
- 4 Oracle Data Dictionary**

Erste Anfragen an das Oracle Data Dictionary

- Lassen Sie sich alle Nutzer der Datenbank anzeigen.

Tabelle **ALL_USERS**

- Welche Daten hat das System über Ihren Account?

Tabelle **USER_USERS**

- Wie viele Tabellen sind für Sie zugreifbar?

Tabelle **ALL_CATALOG**

- Wer besitzt eine Tabelle mit Namen **EMP**?

- Welche Tabellen des Nutzers **SCOTT** sind für Sie zugreifbar?