Datenbanksysteme

WS 05/ 06

Gruppe 12

Martin Tintel
Tatjana Triebl

Inhaltsverzeichnis

1. Einleitung

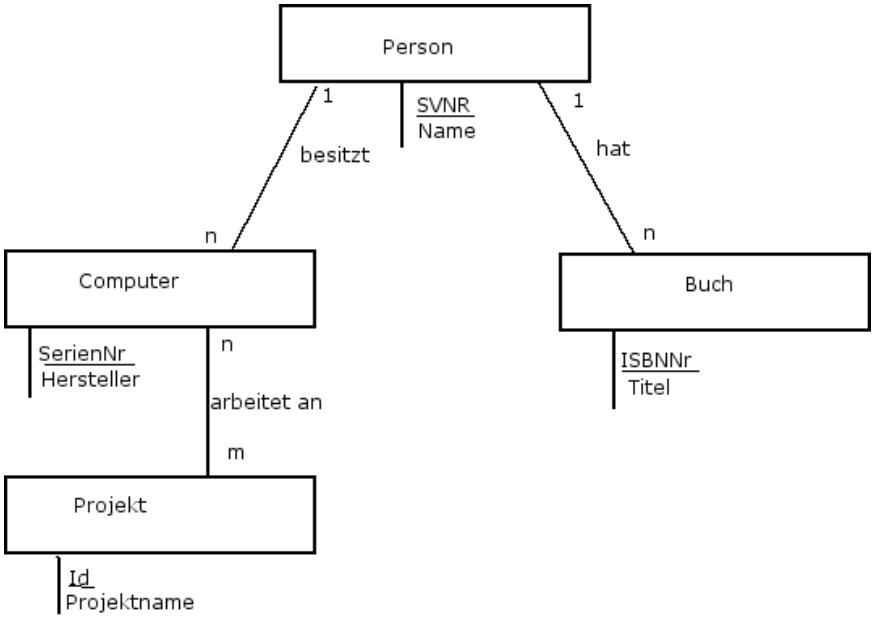
Die Arbeit an diesem Projekt und an der Laborübung war sehr interessant und natürlich auch sehr lehrreich.
Unser Projekt befasst sich mit dem Vergleich von 3 Datenbanken Oracle, MySQL und MS SQL. Wir haben die Performance beim Einfügen, Updaten und Auswählen von Daten in die/ von der Datenbanken getestet

Wir haben dabei viel über das Erstellen und das Arbeiten mit Datenbanken gelernt.
Was uns sehr überrascht hat, war, dass wir auch viele Elemente von anderen Lehrveranstaltungen einfliesen lassen konnten. Zum Beispiel Java Programmierung von EPROG oder Verteilte Systeme und wir konnten uns mit PHP beschäftigen, was wir sehr interessant gefunden haben.

In den nachfolgenden Seiten beschreiben wir unsere Erfahrungen bei der Arbeit mit verschiedenen Datenbanken, mit JAVA und mit PHP.
Wir werden aufzeigen, welche Probleme auftreten können, wenn man mit Datenbanken und PHP arbeitet.

2. Datenbanken

Der Grund warum wir uns für MS SQL und nicht für PostgresSQL entschieden haben, war, dass mein Kollege gerne wissen wollte, wie MS SQL im Vergleich zu den anderen Datenbanken abschneiden würde.
Wir werden hier die Datenbanken kurz beschreiben und die Vor und Nachteile der einzelnen Datenbanken aufzeigen.
Das EER Modell zeigt die Struktur, die allen Datenbanken zu Grunde liegt. Wie man sehen kann, gibt es zwei 1 : n Beziehungen und eine n : m Beziehung.
Kurze Beschreibung der Datenbank:
Eine Person besitzt viele Computer, die an mehreren Projekten arbeiten und eine Person hat auch mehrere Bücher.



2.1. Oracle

Man sollte meinen, dass wir mit Oracle die wenigsten Probleme haben sollten, da wir es schon während der Übung kennen gelernt haben.
Dem war auch so, wäre da nicht PHP und JAVA gewesen. Aber zu PHP und JAVA kommen wir in einem späteren Kapitel.

Mit dem Erstellen der Datenbank und dem Einfügen und Auslesen von Daten hatten wir keine Probleme. Auch funktionierte das Erstellen und Löschen von Tabellen gleich einwandfrei.

Wobei zu sagen ist, dass wir diese Funktionen zuerst am Server getestet haben, bevor wir alles mit PHP ausprobiert haben.

2.2. MySQL

Dies ist eine, vom Aufbau und von der Komplexität her, wesentlich einfachere Datenbank als Oracle.

Diese Datenbank hat auch die beste Online Dokumentation, da sie für jeden kostenlos zugänglich ist und daher von sehr vielen genutzt wird.

Man kann das Open- source Programm MySQL Center benutzen, um eine MySQL Datenbank zu erstellen. Bei diesem Programm kann man alle Tabellen graphisch erstellen und sich dann ein Generierungsscipt ausgeben lassen.
Wir haben dies getan, dann dieses Script am Server ausgeführt und alles verlief vollkommen problemlos.
Wie auch bei Oracle haben wir danach die Datenbank und die Tabellen mittels PHP erzeugt.

2.3 MS SQL

MS SQL ist MySQL sehr ähnlich, sowohl was das Erstellen der Datenbank, als auch die Syntax der SQL Befehle betrifft.
Es gibt allerdings einen großen Unterschied zwischen diesen beiden Datenbanken. MS SQL ist wie auch Oracle eine kommerzielle Datenbank und somit eher für große Unternehmen geeignet.
Für den kleinen Benutzer ist weder MS SQL noch Oracle wirklich empfehlenswert.

Das wohl größte Problem mit MS SQL war, dass es kaum Hilfe in Netz gab.
Wir erhielten aber zum Glück Unterstützung von einer Tutorin und fanden ein gutes Buch über MS SQL, was uns die Arbeit sehr erleichtert hat.

3. PHP

3.1 PHP allgemeines

Wir fanden es sehr toll, dass wir mit PHP arbeiten konnten. Die Gründe, warum wir PHP empfehlen würden, sind:

- Der wohl beste Grund war, dass PHP kostenlos für jeden verfügbar ist.
- Es gibt eine sehr umfassende Dokumentation im Netz.(gute Hilfestellung auf der PHP Homepage)
- PHP unterstützt fast alle Datenbanken.
- Es ist leicht sich schnell in PHP einzuarbeiten.

3. 2 PHP und Datenbanken

Wir haben mit Hilfe eines JAVA. Programms, das im nächsten Kapitel erklärt wird, PHP Dateien erzeugt, die wir dann auf den Server hochgeladen und ausgeführt haben.

Diese PHP Dateien enthalten die Befehle für SQL:

- Create table
- Drop table
- Insert
- Select
- Update

Hier einige Beispiele der MySQL Version zur Verdeutlichung:

```
mysql_query("DROP TABLE computer");
mysql_query("DROP TABLE buch");
mysql_query("DROP TABLE person");

mysql_query("CREATE TABLE person(SVNR char(11) NOT NULL,Name varchar(100) NOT NULL,PRIMARY KEY (SVNR))");
mysql_query("CREATE TABLE buch(SVNR char(11) NOT NULL, ISBNNR char(11) NOT NULL,Titel varchar(100) NOT NULL,PRIMARY KEY (ISBNNR), Foreign Key (SVNR) references person (SVNR))");
mysql_query("CREATE TABLE computer(SVNR char(11) NOT NULL, SerienNr char(11) NOT NULL,Hersteller varchar(100) NOT NULL,PRIMARY KEY(SerienNr), Foreign Key (SVNR) references person (SVNR))");
```

Hier einige Beispiele der Oracle Version zur Verdeutlichung:

```
$s = OCIParse($conn, "drop table computer");
OCIExecute($s, OCI_DEFAULT);
$s = OCIParse($conn, "drop table buch");
OCIExecute($s, OCI_DEFAULT);
$s = OCIParse($conn, "drop table person");
OCIExecute($s, OCI_DEFAULT);

$s = OCIParse($conn, "create table person (SVNR varchar2(30), Name varchar2(30),
PRIMARY KEY (SVNR))");
OCIExecute($s, OCI_DEFAULT);
$s = OCIParse($conn, "create table buch (SVNR varchar2(30), ISBNNR varchar2(30), Titel
varchar2(30), PRIMARY KEY (ISBNNR), Foreign Key (SVNR) references person
(SVNR))");
OCIExecute($s, OCI_DEFAULT);
$s = OCIParse($conn, "create table computer (SVNR varchar2(30), SerienNr varchar2(30),
Hersteller varchar2(30), PRIMARY KEY (SerienNr), Foreign Key (SVNR) references person
(SVNR))");
```

Wie man deutlich erkennen kann, gibt es deutliche Unterschiede zwischen der MySQL und der Oracle Version.
Die MS SQL Version ist ähnlich wie die Version von MySQL. Es gibt nur geringe Abweichungen bei der Syntax. Zum Beispiel statt `mysql_query` schreibt man `mssql_query`.

Bei MS SQL und bei MySQL gab es nicht sehr viele Probleme und die, die aufgetreten sind, konnten mit Hilfe der Homepage von PHP gelöst werden.
Allerdings taten wir uns bei Oracle etwas schwerer. Zum Glück fanden wir auf der Oracle Homepage eine gute Dokumentation über die Verwendung von PHP mit Oracle.

4. JAVA

4.1 Gründe für JAVA

Wie bereits oben beschrieben, werden bei unserem Projekt die PHP Dateien aus dem JAVA Programm erzeugt.
Aus diesem Grund ist das JAVA Programm die Basis für unser Projekt.

Die Gründe für ein solches Vorgehen liegen auf der Hand.
Da wir mehrere 100.000 Einträge zu machen hatten, entschieden wir uns für ein JAVA Programm, das automatisch Einträge erzeugt. Die Anzahl der Einträge kann man frei wählen.

4.2 Aufbau JAVA Programm

Unser Programm setzt sich aus mehreren Teilen zusammen, wobei wir anfangs nur ein File hatten, das alle Funktionen macht.
Aufgrund der Performance Tests mussten wir das File aufteilen, so dass nun jeder Teil eine Funktion ausführt.
Diese Teile/ Funktionen werden wir jetzt beschreiben:

4.2.1 Drop table

Hier werden die Tabellen gelöscht.

4.2.2 Create table

In diesem Programm werden die Tabellen erzeugt.

4.2.2 Create inserts

Dieser Teilbereich erzeugt die Einträge in die Datenbank, wobei man frei wählen kann, wie viele Einträge man machen möchte.
Das Programm ist so angelegt, dass es den Dateinamen anpasst, je nachdem wie viele Einträge gemacht werden und wie oft man die Einträge machen möchte. Wenn man zum Beispiel 10 Einträge und 3 Durchläufe machen will, dann erkennt man dies am Dateinamen der erzeugten PHP Dateien.

Die Durchläufe sind wichtig, da es, während man eine PHP Datei öfters auf dem Server ausführt, zu Schwankungen bei der Zeitmessung kommt.
Wir geben die Zeit für jeden Durchlauf und für alle Durchläufe aus.
Um aussagekräftige Benchmarktests zu erhalten, bilden wir das arithmetische Mittel aus den Durchläufen.
Dieser Durchschnitt wird dann mit denen, der anderen Datenbanken verglichen.

4.2.3 Update

Hier werden die Updates gemacht, wobei in jeder Tabelle jeder Eintrag von einem Feld verändert wird.
Wie oben beim Einfügen bereits beschrieben, gibt es auch hier Durchläufe.
Die Gründe warum wir dies so machen, haben wir bereits im vorangegangenen Absatz beschrieben.

4.2.4 Select

Bei diesem Programmteil werden aus allen Tabellen alle Inhalte ausgelesen:

```
SELECT * FROM person;
SELECT * FROM buch;
.
.
.
```

Dann wird eine Abfrage über mehrere Tabellen ausgeführt:

```
SELECT person.SVNR AS PERSON, projekt.projektname AS PROJEKTNAME
FROM person, computer, projekt,arbeitetan
WHERE person.svnr = computer.svnr
AND computer.seriennr = arbeitetan.seriennr
AND arbeitetan.id= projekt.id;
```

Diese Abfrage gibt aus, welche Person an welchem Projekt arbeitet.

5. Projektergebnisse

In diesem Kapitel werden wir die Ergebnisse unserer Tests aufzeigen.
Wir werden sehen welche der drei Datenbanken am schnellsten Daten einfügt, Daten ausliest und sie updatet.
Natürlich benötigen wir zuvor noch die Rahmenbedingungen für die Benchmarktests:

A	1.000	Inserts	Updates	Alles wird wie oben beschrieben selected
B	5.000	Inserts	Updates	Alles wird wie oben beschrieben selected
C	10.000	Inserts	Updates	Alles wird wie oben beschrieben selected
D	100.000	Inserts	Updates	Alles wird wie oben beschrieben selected
E	500.000	Inserts	Updates	Alles wird wie oben beschrieben selected

Die Tests A bis C werden mit jeweils 5 Durchläufen durchgeführt, beim vorletzten Test haben wir uns aus Performance Gründen auf 3 Durchläufe und beim letzten Test auf 1 Durchlauf beschränkt.

Als Testergebnis wird jeweils das arithmetische Mittel der Zeiten der jeweiligen Durchläufe eingetragen, da es so am aussagekräftigsten ist.
Die Ergebnisse werden immer in Sekunden gemessen.

5.1 Insert

	Oracle	**MySQL**	**MS SQL**
A	6.1455432	**0.1651564**	7,667577
B	31.980074	**0.8147388**	36,22912
C	**64.68695**	**1.5791248**	68,050112
D	N.A.	**14.530926666667**	626,9919
E	N.A.	**60.19505**	N.A.

5.2 Update

	Oracle	**MySQL**	**MS SQL**
A	6.3515102	**0.1969738**	6,118038
B	**21.39147**	**0.9509842**	28,186604s
C	**41.533998**	**1.8604272**	55,887256
D	N.A.	**18.63624**	**310.75513333333**
E	N.A.	**60.58826**	N.A

5.3 Select

	Oracle	MySQL	MS SQL
A	**0.00599745**	**0.020685082**	0,018556699
B	**0.00972646**	**0.020456868**	0,06291534
C	**0.01013748**	**0.020594739**	0,10361306
D	N.A.	**0.020380947**	0.26015534
E	N.A.	**0.020354944**	N.A.

5.4 Zusammenfassung der Testsergebnisse

Überraschender Weise verliefen die Tests bei MySQL am problemlosesten und am schnellsten. Wir konnten es anfangs nicht glauben und wiederholten die Tests, aber es blieb dabei: **MySQL ist am schnellsten.**
Oracle ist MS SQL, was die Schnelligkeit betrifft, überlegen aber leider funktionierten bei Oracle die Tests aus unerfindlichen Gründen mit sehr vielen Daten (ab 300.000 Einträge) nicht. Teilweise benötigten wir sogar schon beim zweiten Test 3 Versuche, um ein Ergebnis zu erhalten. Am schlechtesten verliefen die Tests bei den Updates.
MS SQL konnte den Test mit 300.000 Einträgen auch nicht problemlos durchführen; wir benötigten 3 Versuche, bis ein Ergebnis messbar war.
Wohingegen bei MySQL alle Tests fehlerfrei durchführbar waren.
Ein möglicher Grund für das schlechte Abschneiden von Oracle liegt an der Zusammenarbeit mit PHP.

Das Ranking (Performance):

1. MySQL (mit Abstand)
2. Oracle
3. MS SQL

Das Ranking (Zuverlässigkeit):

1. My SQL
2. MS SQL
3. Oracle