Rick F. van der Lans:

Das SQL‑Lehrbuch

Addison‑Wesley (Deutschland) GmbH, Bonn, München u.a., 1990

Kapitel 2.1

**SQL-Übung:**

**Die Beispieldatenbank eines Sportvereins**

Die Datenbank besteht aus vier Relationen:

‑ SPIELER

‑ TEAMS

‑ WETTKÄMPFE

‑ STRAFEN

Die Relation SPIELER enthält Daten über die Spieler, die dem Verein als Mitglieder angehören. Die Relation enthält keine "historischen" Daten. Falls ein Spieler oder eine Spielerin seine oder ihre Mitgliedschaft aufkündigt, werden die entsprechen­den Daten aus der Relation gelöscht. Auch bei einem eventuellen Umzug wird die alte Adresse durch die neue ersetzt, die alte Adresse wird also nirgendwo gesondert festgehalten.

Der Sportverein unterscheidet zwischen zwei Mitgliedergruppen:

ERHOLUNGSSPIELER und WETTKAMPFSPIELER. Die erste Gruppe trägt nur vereinsinterne Wettkämpfe aus, nimmt also nicht an Wettkämpfen gegen Spieler anderer Vereine teil. Die Ergebnisse dieser internen Wettkämpfe werden nicht registriert. Wettkampfspieler dagegen spielen im Mannschaftsverband gegen die Spieler anderer Vereine. Jeder Spieler verfügt über eine eindeutige Nummer, gleichgültig, ob es sich bei ihm oder ihr um eine(n) Wettkampf­spieler(in) handelt oder nicht. Diese Spielernummer wird vom Verein vergeben. Jeder Wettkampf­spieler muss beim Verein registriert sein. Der Verband teilt außerdem jedem Wettkampf­spieler eine eindeutige Verbandsnummer zu. Falls ein Wett­kampf­spieler nicht mehr an Verbands­wettkämpfen teilnimmt, somit also zum Erholungsspieler wird, verfällt seine Verbandsnummer. Erholungsspieler haben wohlgemerkt keine Verbandsnummer, wohl aber eine Spielernummer.

Der Sportverein hat mehrere Mannschaften oder Teams, die an Wett­kampf­serien teilnehmen. Für jedes Team wird die Liga registriert, in der es sich momentan befindet. Auch diese Relation enthält keine veralteten Daten. Falls ein Team in eine andere Liga auf‑ oder absteigt, wird die frühere Liga einfach überschrieben. Das gleiche gilt für den Kapitän eines Teams: Ist er neu ein­gesetzt, wird der Name des früheren Kapitäns überschrieben.

Ein Team besteht aus vier Spielern. Während eines Wettkampfes spielt jeder Spieler gegen einen Spieler aus dem Team eines anderen Vereins. Ein Team setzt sich nicht immer aus den gleichen Spielern zusammen. Falls ein Spieler krank oder verreist ist, muss gelegentlich ein Ersatzspieler für ihn einspringen. Ein(e) Spieler(in) kann seinen oder ihren Wettkampf gewinnen oder verlieren; ein Unentschieden ist nicht möglich. In der Relation WETTKÄMPFE wird aufgezeigt, wie viele Wettkämpfe ein bestimmter Spieler für ein bestimmtes Team gewonnen oder verloren hat, somit enthält diese Relation auch historische Daten. Zu Beginn einer neuen Spielzeit wird die Anzahl gewonnener und verlorener Wett­kämpfe nämlich nicht auf Null gesetzt, sondern durchgezählt. Ein Spieler kann also mehrere Male in der Relation WETTKÄMPFE erschei­nen, jedes Mal jedoch für ein anderes Team. Aus welchen Spielern sich ein bestimmtes Team zusammen­setzt, lässt sich dementsprechend aus einer solchen Relation nicht ersehen.

Bei regelwidrigem Betragen eines Spielers (Zuspätkommen, Nicht­erscheinen oder aggressives Verhalten) verhängt der Verband Bußgelder. Diese Bußgelder entrichtet der Verein. Solange ein Spieler an Wettkämpfen teilnimmt, bleiben alle Bußgelder aufge­zeichnet, mit denen er belegt worden ist.

Falls ein Spieler den Verein verlässt, werden alle ihn betreffen­den Daten in den vier Relationen gelöscht. Falls der Verein ein Team zurückzieht, werden in den Relationen TEAMS und WETTKÄMPFE alle Daten zu diesem Team gelöscht.

Es folgt eine Beschreibung der einzelnen Attribute aller vier Relationen:

**SPIELER**

SPIELERNR eindeutige Nummer des Spielers

NAME Nachname des Spielers (ohne Vornamenkürzel, Namenszusätze oder Titel)

VORNAMEN Anfangsbuchstaben des (der) Vornamen des Spielers. Es werden keine Punkte gesetzt

TITEL Namenszusätze und Titel wie "von" oder "Dr."

(Diese Spalte bleibt häufig leer!)

GEB\_JAHR Geburtsjahr des Spielers

GESCHLECHT Geschlecht des Spielers: M(ännlich) oder

W(eiblich)

JAHRBEI das Jahr, in dem der Spieler dem Verein beitrat

STRASSE Name der Straße, in der der Spieler wohnt

HAUSNR Hausnummer

PLZ Postleitzahl (noch 4-stellig!)

ORT Wohnort des Spielers

TELEFON Vorwahlnummer, Bindestrich, Teilnehmernummer

VERB\_NR vom Verband zugeteilte Verbandsnummer (nur für Wettkampfspieler)

**TEAMS**

TEAMNR eindeutige Nummer des Teams

SPIELERNR Spielernummer des Mannschaftskapitäns

LIGA Die Liga, in die der Verband das TEAM eingeteilt hat

**WETTKAEMPFE** *(diese Tabelle entspricht einer n:m-Beziehungsmenge zwischen TEAMS und SPIELER)*

TEAMNR Nummer des TEAMS

SPIELERNR Nummer des Spielers

GEWONNEN Anzahl der Wettkämpfe, die der Spieler für das Team gewonnen hat

VERLOREN Anzahl der Wettkämpfe, die der Spieler für das Team verloren hat

**STRAFEN**

ZAHLUNGSNR Jede Geldbuße, die der Verein bezahlen muss, erhält eine eindeutige Nummer

SPIELERNR Nummer des Spielers, der bestraft worden ist

DATUM Tag, an dem die Strafe bezahlt worden ist

BETRAG Geldbetrag, der als Strafe bezahlt worden ist

Der Inhalt der einzelnen Relationen ist unten wiedergegeben. Wenn nicht anders angegeben, beziehen sich alle folgenden Beispiele und Aufgaben auf diese Relationeninhalte. Es wurden absichtlich kleine Relationen angelegt. Der Grund für diese Entscheidung wird in den Beispielen deutlich werden. In der Realität benötigt ein Sportverein natürlich wesentlich größere Relationen.







**Inhalt der Relation SPIELER:**

SPIELERNR TITEL JAHRBEI PLZ VERB\_NR

NAME GEB\_JAHR STRASSE ORT

VORNAMEN GESCHLECHT HAUSNR TELEFON

6 Peters R ? 1964 M 1977 Hafenallee 80 4000 Düsseldorf 0211‑476537 8467

44 Bäcker E Dr. 1963 M 1980 Lichtstraße 23 4030 Ratingen 02102‑36875 1124

83 Hofmann PK ? 1956 M 1982 Marienufer 16A 4000 Düsseldorf 0211‑353548 1608

2 Elfers R ? 1948 M 1975 Stadtring 43 4000 Düsseldorf 0211‑237893 2411

27 Kohl DD ? 1964 W 1983 Luisenpfad 804 4005 Meerbusch 02105‑23485 2513

104 Maurer D ? 1970 W 1984 Stutenallee 65 4005 Meerbusch 02105‑98757 7060

7 Wiegand GWS ? 1963 M 1981 Erasmusweg 39 4000 Düsseldorf 0211‑347689 ?

57 Böhmen M von 1971 M 1985 Erasmusweg 16 4000 Düsseldorf 0211‑473458 6409

39 Bischof D ? 1956 M 1980 Erikaplatz 78 4000 Düsseldorf 0211‑393435 ?

112 Bauer IP von 1963 W 1984 Fuchsweg 8 4150 Krefeld 02151‑54874 1319

8 Neuhaus B ? 1962 W 1980 Sporenallee 4 4030 Ratingen 02102‑45845 2983

100 Peters P ? 1963 M 1979 Hafenallee 80 4000 Düsseldorf 0211‑494593 6524

28 Kohl C ? 1963 W 1983 Domplatz 10 4040 Neuss 02101‑65959 ?

95 Müller P ? 1934 M 1972 Hauptweg 33A 4010 Hilden 02103‑86756 ?

**Inhalt der Relation STRAFEN:** **Inhalt der Relation WETTKÄMPFE:**

ZAHLUNGSNR SPIELERNR DATUM BETRAG TEAMNR SPIELERNR GEWONNEN VERLOREN

1 6 801208 100.00 1 44 7 5

2 44 810505 75.00 1 83 3 3

3 27 830910 100.00 1 2 4 8

4 104 841208 50.00 1 57 5 0

5 44 801208 25.00 1 8 0 1

6 8 801208 25.00 2 27 11 2

7 44 821230 30.00 2 104 8 4

8 27 841112 75.00 2 112 4 8

2 8 4 4

1 6 9 1

**Inhalt der Relation TEAMS:**

TEAMNR SPIELERNR LIGA

1 6 erste

2 27 zweite

**Aufgaben:**

***(Beachten Sie auch den Anhang mit den Syntax-Definitionen einiger SQL-Anweisungen!)***

1. Starten Sie Ihren Datenbank-Client und anschließend ISQL.

*(Beachten Sie bitte die Ihnen dazu gegebenen aktuellen Hinweise!)*

1. Legen Sie sich eine eigene Datenbank an: Als Datenbanknamen verwenden Sie bitte Ihren **login-Namen!**

*(Falls Sie Ihre Datenbank später einmal weitergeben wollen oder eine fremde Datenbank auf Ihr home-Verzeichnis kopieren wollen, können Sie so Konflikte wegen gleichlautender Datenbanknamen von vornherein vermeiden.)*

1. Laden Sie das vorbereitete SQL-Generierungsskript „SPORT.SQL“ und speichern Sie sich dieses Skript sofort in Ihren eigenen Speicherbereich, damit Sie es auch später jederzeit zur Verfügung haben.

*(Das Skript wurde mit* ***PowerDesigner*** *erstellt und um zwei Teile erweitert:*

1. *die Definition von zwei Sichten,* ***meine\_tab*** *und* ***meine\_col****, mit denen Sie sich später die in der Datenbank enthaltenen Nutzertabellen bzw. die in diesen Tabellen enthaltenen Spalten anzeigen lassen können,*
2. *eine Folge von* ***insert****-Anweisungen, um die definierten Tabellen mit Daten zu füllen.)*
3. Sehen Sie sich den im Skript enthaltenen Quelltext sehr genau an!!!

*(Denken Sie daran, dass Sie in der Lage sein müssen, derartige Folgen von SQL-Anweisungen auch* ***selbständig*** *zu schreiben!)*

1. Lassen Sie das Skript ausführen.
2. Lassen Sie sich jede Tabelle vollständig anzeigen und vergleichen Sie mit den Angaben auf der vorigen Seite!
3. Geben Sie den Namen und das Geburtsjahr von jedem Spieler an, der in Düsseldorf wohnt!
4. Legen Sie eine Tabelle STADIEN an wie unten angegeben. Geeignete Datentypen wählen Sie bitte selbst. Achten Sie aber darauf, nicht zu lange Texte zu vereinbaren, damit Ihre Bildschirm-Recherchen später einigermaßen übersichtlich bleiben.

STADNR eineindeutige Nummer des Stadions (Primärschlüssel)

NAME Bezeichnung des Stadions

VEREIN Trägerverein

PLAETZE Anzahl der Plätze im Stadion

ORT Ort

PLATZWART Name des Platzwarts

TELEFON Telefonnummer des Platzwarts

create table STADIEN

(STADNR int primary key,

NAME varchar(255),

VEREIN varchar(255),

PLAETZE int,

ORT varchar(255),

PLATZWART varchar(255),

TELEFON varchar(255));

1. Geben Sie in die Tabelle STADIEN fünf Tupel nach eigenem Ermessen ein!

INSERT INTO STADIEN VALUES (1, 'American CC', 'Eaglesclub', 10000, 'Berlin', 'Mr Donner','030 58439090');

INSERT INTO STADIEN VALUES (2, 'Indian Fhilak CE', 'Laksmi', 11000, 'Bremen', 'Fr Glock','03030 52223490');

1. Lassen Sie sich STADIEN vollständig anzeigen!
2. Wie vorhergehende Aufgabe, jedoch sollen die Ergebniszeilen nach Namen aufsteigend sortiert ausge­geben werden!
3. Geben Sie die Nummer von jedem Spieler an, dem am 8. Dezember 1980 eine Geldbuße von mehr als € 25,- auferlegt wurde. Sortieren Sie das Ergebnis nach der Spielernummer!

(Beachten Sie den Datentyp des Felds DATUM: Ein Datum wird in SQLAnywhere wie eine Zeichenkette im Format JJJJ-MM-TT kodiert!)

select Spielernr

from Strafen

where Datum = '1980-12-08' and Betrag>25

order by Spielernr;

1. Es wurde vergessen, einen Spieler einzutragen. Dieser Spieler hat bereits zwei Strafen erhalten. Fügen Sie die folgenden Daten in die Datenbank ein:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabelle** | **Attribut** | **Wert** |  | **Tabelle** | **Attribut** | **Wert**  (Strafe1) | **Wert**  (Strafe2) |
| Spieler | SpielerNr | 77 |  | Strafen | ZahlungsNr | 27 | 28 |
|  | Name | Klein |  |  | Betrag | 40 | 30 |
|  | Ort | Nauen |  |  |  |  |  |

INSERT INTO Spieler (Spieler.spielernr, Name, Ort) VALUES (77, 'Klein', 'Nauen');

INSERT INTO Strafen (SpielerNr, ZahlungsNr, Betrag) VALUES (77, 27, 40);

INSERT INTO Strafen (SpielerNr, ZahlungsNr, Betrag) VALUES (77, 28, 30);

1. Ermitteln Sie die Nummern aller Spieler, die nach 1960 geboren wurden!

select \* from spieler where Geb\_Jahr >1960

1. Geben Sie den Namen, das Geschlecht und das Geburtsjahr von jedem männlichen Spieler an, der nach 1970 geboren wurde!

select Spieler.name, Ort, Geschlecht from spieler where geschlecht='M' and Geb\_Jahr>1970

1. Geben Sie die Namen und Wohnorte der Spieler aus Krefeld und Meerbusch an!

select Spieler.name, Geschlecht, Ort from spieler where Ort='Krefeld' or Ort='Meerbusch'

1. Geben Sie die Namen und Wohnorte der Spieler an, die nicht in Düsseldorf wohnen!

select Spieler.name, Geschlecht, Ort from spieler where Ort!='Duesseldorf'

1. Geben Sie den Namen und den Wohnort von jeder Spielerin(!) an, die nicht in Düsseldorf wohnt!

select Spieler.name, Geschlecht, Ort from spieler where Ort!='Duesseldorf' and Geschlecht='W'

1. Ermitteln Sie Gesamtanzahl, Gesamtsumme, und Gesamtdurchschnitt der Strafen aller Spieler.

select count (\*) as Gesamtanzahl, sum(betrag) as Gesamtsumme, avg(betrag) as Gesamtdurchschnitt from Strafen

1. Ermitteln Sie die Anzahl, die Summe, und den Durchschnitt der Strafen jedes einzelnen Spielers.

select spielernr,count(\*) Summe, avg(betrag) Durchschnitt from Strafen group by Spielernr order by Summe

1. Ermitteln Sie die Differenzen zwischen dem Durchschnitt der Strafen aller Spieler und dem jeweiligen Durchschnitt der Strafen jedes einzelnen Spielers.

(Schreiben Sie dazu genau eine SQL-Anweisung, ohne die aus den vorhergehenden Aufgaben bekannten Werte direkt zu benutzen!)

select spielernr, avg(betrag)-(select avg(betrag) from Strafen) from Strafen group by spielernr

1. Wie heißt der Spieler mit der höchsten Strafensumme?

select top 1 sum(Strafen.Betrag) Summe, Strafen.SpielerNr, Spieler.Name

from Strafen, Spieler

where Strafen.SpielerNr = Spieler.SpielerNr

group by Strafen.SpielerNr, Spieler.Name

order by Summe desc

where → keine Verdichtungsfunktionen

Bsp. „select top 1 start at 2“ → nur mit order by

bei verdichtung → nur verdichtung oder das was in group by steht

1. Wie heißt der Spieler mit der zweithöchsten Strafensumme?

(Schreiben Sie dazu genau eine SQL-Anweisung, ohne den in der vorhergehenden Aufgaben ermittelten Wert direkt zu benutzen!)

select top 1 start at 2 sum(Strafen.Betrag) Summe, Strafen.SpielerNr, Spieler.Name

from Strafen, Spieler

where Strafen.SpielerNr = Spieler.SpielerNr

group by Strafen.SpielerNr, Spieler.Name

order by Summe desc

1. Welcher Kapitän hat keine Strafe erhalten?

(Mannschaftskapitäne sind diejenigen Spieler, deren Nummer als SPIELERNR in der Tabelle TEAMS enthalten ist.)

select Teams.SpielerNr, Spieler.Name

from Strafen right outer join Teams

on Teams.SpielerNr = Strafen.SpielerNr inner join Spieler on Spieler.SpielerNr = Teams.SpielerNr

group by Spieler.Name, Teams.SpielerNr having sum(Strafen.Betrag) is null

1. Bei dem Spieler mit der Nummer 104 hat sich die Hausnummer geändert. Er wohnt jetzt in Nummer 33.
2. Führen Sie die Änderung aus!

update spieler

set Hausnr = 33

where SpielerNr = 104;

1. Dem Spieler mit der Nummer 77 werden alle Strafen unter 40€ erlassen. Löschen Sie diese Strafe!

delete from strafen

where SpielerNr = 77 and Betrag<40;

1. Lassen Sie alle Daten von jeder Geldbuße anzeigen!

select \*

from Strafen

1. Ändern Sie für jede Strafe, die am 8.12.1980 verhängt wurde, den Betrag in € 200,- um! Lassen Sie sich anschließend alle Strafen erneut anzeigen!

update Strafen

set Betrag = 200

where datum='1980-12-08'

1. Löschen Sie jede Geldbuße, deren Betrag € 80,- übersteigt! Lassen Sie sich die restlichen Strafen anzeigen!

delete from strafen

where betrag > 80;

1. Legen Sie zum Tabellennamen SPIELER ein Synonym MITGLIEDER an!

(Synonyme werden von vielen DBBS nicht unterstützt. Was könnte man tun, damit ein und dieselbe Tabelle trotzdem unter verschiedenen Namen angesprochen werden kann?)

select \*

from Spieler as Mitglieder

where Mitglieder.SpielerNr=77

1. Geben Sie für jeden Mannschaftskapitän die Teamnummer und den Namen an!

select Teams.SpielerNr, Spieler.Name

from teams inner join spieler

on Teams.SpielerNr = Spieler.SpielerNr

1. Geben Sie die Nummern der Spieler an, deren Verbandsnummer gleich 7060 ist!

select Spieler.SpielerNr

from Spieler

where Spieler.Verb\_Nr = 7060;

1. Geben Sie die Nummer, das Geburtsjahr und das Beitrittsjahr aller Spieler an, die 17 Jahre nach ihrer Geburt dem Verein beigetreten sind!

select Spieler.SpielerNr, Geb\_Jahr, jahrbei

from Spieler

where jahrbei - geb\_jahr = 17;

1. Ermitteln Sie die Anzahl aller Spieler, die nach 1960 geboren wurden!

(Frage: Hätte man diese letzten beiden Aufgaben auch zu einer einzigen Aufgabe zusammenfassen können?)

select count(Spielernr) as Anzahl

from Spieler

where geb\_jahr > 1960;

→ nicht kombinierbar, da entweder eine Anzahl oder die Auflistung gefragt ist

1. Geben Sie die Nummer von jedem Team an, das einen anderen Mannschaftskapitän als den Spieler mit der Nummer 27 hat!

select teamnr

from teams

where spielernr=27

1. Geben Sie die Nummern der Spieler an, die in der Zeit von 1970 bis 1980 dem Verein beigetreten sind!

select spielernr

from spieler

where jahrbei between 1970 and 1980;

1. Geben Sie die Namen aller Spieler an, für die keine Strafe eingetragen wurde!

select spieler.name

from spieler left outer join strafen

on Spieler.Spielernr = Strafen.Spielernr

except

select spieler.name

from spieler inner join strafen

on Spieler.Spielernr = Strafen.Spielernr

1. Lassen Sie sich alle die Spieler anzeigen, für die Strafen von insgesamt mindestens € 100.- eingetragen sind!

select Strafen.spielernr

from strafen

group by Strafen.spielernr having sum(betrag)>=100

1. Lassen Sie sich die Gesamtstrafe für jeden Mannschaftskapitän anzeigen!

(Mannschaftskapitäne sind diejenigen Spieler, deren Nummer als SPIELERNR in der Tabelle TEAMS enthalten ist.)

select teams.spielernr, sum(betrag) Gesamtstrafe

from strafen, teams

where Strafen.Spielernr=Teams.Spielernr

group by Teams.spielernr

**Anhang: Syntax-Definitionen einiger SQL-Anweisungen**

(stark gekürzt)

**!!! (siehe auch: ISQL-Hilfe  SQL-Anweisungen) !!!**

- Relation ( = Tabelle) anlegen:

CREATE TABLE <tabelle>

( <spalte> <datentyp> [ NOT NULL ] [ PRIMARY KEY ]

[ ,<spalte> <datentyp> [ NOT NULL ] ...] );

- Einfügen zusätzlicher Attributspalten:

ALTER TABLE <tabelle>

{ ADD <spalte> <datentyp>  [ <Spalten-Integritätsregel> ... ]

| ADD <Tabellen-Integritätsregel>

| MODIFY <spalte> <datentyp>

| DROP  Spaltenname

| RENAME <Neuer\_Tabellenname>  
 | RENAME <spalte> TO <neuer\_spaltenname> };

- Sicht ( = View) erstellen:

CREATE VIEW <viewname>

[ ( <spalte> [ ,<spalte> ...] ) ]

AS <select\_anweisung> ;

- Daten eingeben:

INSERT INTO <tabelle>

[ ( <spalte1> [,<spalte2>... ] ) ]

{ VALUES ( <wert1> [,<wert2>... ] )

| <select\_anweisung> } ;

- Daten ändern:

UPDATE <tabelle>

SET <spalte> = { <ausdruck> | NULL }

[ ,<spalte> = { <ausdruck> | NULL } ... ]

[WHERE <bedingung>];

- Daten ausgeben:

SELECT [DISTINCT | ALL] { \* | [<spaltenausdruck>]... }

FROM <tabelle> [,<tabelle>... ]

[WHERE <bedingung>]

[GROUP BY <spalte> [,<spalte>... ] [HAVING <bedingung>] ]

[ORDER BY <spalte> [,<spalte>... ] ];

- Index anlegen:

CREATE [UNIQUE] INDEX <indexname>

ON <tabelle> (<spalte> [ASC | DESC]

[,<spalte> [ASC | DESC] ]... );

- Datenbank‑Elemente löschen:

DROP { INDEX | TABLE | VIEW | PROCEDURE | TRIGGER } <name>;

- Tupel löschen:

DELETE FROM <tabelle>

[WHERE <bedingung>];

- Transaktion zurücksetzen:

{ ROLLBACK | ABORT } [ TRANSACTION ];

- Transaktion erfolgreich beenden:

COMMIT [TRANSACTION ];

- Neuen Benutzer einführen:

GRANT CONNECT

TO <benutzer>

IDENTIFIED BY <kennwort>;

- Befugnisse zur Tabellenbearbeitung erteilen:

GRANT { SELECT

| INSERT

| DELETE

| UPDATE [ <spalte> [ ,<spalte> ... ] ]

| INDEX

| ALTER

| ALL }

ON <tabelle>

TO { <benutzer> [ ,<benutzer> ... ] | PUBLIC }

[ WITH GRANT OPTION ] ;

oder:

GRANT { DBA | RESOURCE }

TO <benutzer> [ ,<benutzer> ... ] ;

- Befugnisse erteilen zum Anlegen einer Datenbank:

GRANT CREATE DATABASE

TO <benutzer> [ ,<benutzer> ... ] ;

*(Dieses Recht ist nur erforderlich, wenn <benutzer> nicht DBA ist. Es muss innerhalb der MASTER-Datenbank gegeben werden. Ebenso ist das Anlegen einer neuen Datenbank nur aus der MASTER-Datenbank heraus möglich.)*