- Rick F. van der Lans:

Das SQLLehrbuch

AddisonWesley (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 0211476537 8467

44 Bäcker E Dr. 1963 M 1980 Lichtstraße 23 4030 Ratingen 0210236875 1124

83 Hofmann PK ? 1956 M 1982 Marienufer 16A 4000 Düsseldorf 0211353548 1608

2 Elfers R ? 1948 M 1975 Stadtring 43 4000 Düsseldorf 0211237893 2411

27 Kohl DD ? 1964 W 1983 Luisenpfad 804 4005 Meerbusch 0210523485 2513

104 Maurer D ? 1970 W 1984 Stutenallee 65 4005 Meerbusch 0210598757 7060

7 Wiegand GWS ? 1963 M 1981 Erasmusweg 39 4000 Düsseldorf 0211347689 ?

57 Böhmen M von 1971 M 1985 Erasmusweg 16 4000 Düsseldorf 0211473458 6409

39 Bischof D ? 1956 M 1980 Erikaplatz 78 4000 Düsseldorf 0211393435 ?

112 Bauer IP von 1963 W 1984 Fuchsweg 8 4150 Krefeld 0215154874 1319

8 Neuhaus B ? 1962 W 1980 Sporenallee 4 4030 Ratingen 0210245845 2983

100 Peters P ? 1963 M 1979 Hafenallee 80 4000 Düsseldorf 0211494593 6524

28 Kohl C ? 1963 W 1983 Domplatz 10 4040 Neuss 0210165959 ?

95 Müller P ? 1934 M 1972 Hauptweg 33A 4010 Hilden 0210386756 ?

**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!)**

• ~~Starten Sie Ihren Datenbank-Client und anschließend ISQL.~~

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

• ~~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.)~~

• ~~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:~~

• ~~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,~~

• ~~eine Folge von~~ **~~insert~~**~~-Anweisungen, um die definierten Tabellen mit Daten zu füllen.)~~

• ~~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!)~~

• ~~Lassen Sie das Skript ausführen.~~

• ~~Lassen Sie sich jede Tabelle vollständig anzeigen und vergleichen Sie mit den Angaben auf der vorigen Seite!~~

• 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

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

• Lassen Sie sich STADIEN vollständig anzeigen!

• ~~Geben Sie den Namen und das Geburtsjahr von jedem Spieler an, der in Düsseldorf wohnt!~~

• ~~Wie vorhergehende Aufgabe, jedoch sollen die Ergebniszeilen nach Namen aufsteigend sortiert ausgegeben werden!~~

• ~~Geben Sie die Nummer von jedem Spieler an, dem am 8. Dezember 1980 eine Geldbuße von mehr als DM 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!)~~

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

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

• ~~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!)~~

• Wie heißt der Spieler mit der höchsten Strafensumme?

Select top 1 sp.name, sp.Vornamen, sp.SPIELERNR, sum(Betrag)Strafe

from Strafen st, SPIELER sp

where st.SPIELERNR=sp.SPIELERNR

group by sp.NAME, sp.VORNAMEN, sp.SPIELERNR

order by Strafe desc

• 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 sp.name, sp.Vornamen, sp.SPIELERNR, sum(Betrag)Strafe

from Strafen st, SPIELER sp

where st.SPIELERNR=sp.SPIELERNR

group by sp.NAME, sp.VORNAMEN, sp.SPIELERNR

order by Strafe desc

• Welcher Kapitän hat keine Strafe erhalten?

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

select sum (Betrag) Strafe, str.spielernr

from Strafen str, spieler sp

where str.SPIELERNR = sp.SPIELERNR and (str.SpielerNr = '6' or str.Spielernr ='27')

group by str.SPIELERNR

select sp.spielernr, name, sum(Betrag)Strafe

from (Spieler sp inner join TEAMS t on sp.SPIELERNR = t.SPIELERNR) left outer join Strafen st on sp.SPIELERNR = st.SPIELERNR

where Betrag != 0

group by sp.SPIELERNR, sp.NAME, st.Betrag

select sp.spielernr, name, st.Betrag

from (Spieler sp inner join TEAMS t on sp.SPIELERNR = t.SPIELERNR) left outer join Strafen st on sp.SPIELERNR = st.SPIELERNR

where st.BETRAG IS NULL

• 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**

**Wert**

Spieler

SpielerNr

77

Strafen

ZahlungsNr

27

28

Name

Klein

Betrag

40

30

Ort

Nauen

INSERT INTO SPIELER (SPIELERNR, NAME, ORT)

VALUES (77, 'Klein','Nauen');

insert Strafen

VALUES(28,77,NULL,30)

• Bei dem Spieler mit der Nummer 104 hat sich die Hausnummer geändert. Er wohnt jetzt in Nummer 33.

Führen Sie die Änderung aus!

update SPIELER

SET HAUSNR = 33

where SPIELERNR = 104

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

delete strafen

where Spielernr = 77 and betrag < 40

• Lassen Sie alle Daten von jeder Geldbuße anzeigen!

select \* from spieler sp inner JOIN strafen st on sp.spielernr = st.SPIELERNR

where st.Betrag IS NOT NULL

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

• Löschen Sie jede Geldbuße, deren Betrag DM 80,- übersteigt! Lassen Sie sich die restlichen Strafen anzeigen!

• 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?)

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

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

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

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

• 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?)

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

• /\* Aufgabe entällt! \*/

• /\* Aufgabe entällt! \*/

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

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

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

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

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

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

• Lassen Sie sich für jeden Spieler, für den eine oder mehrere Strafen eingetragen sind, die Gesamtsumme seiner Strafen anzeigen!

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

• Lassen Sie sich die Gesamtstrafe für jeden Mannschaftskapitän anzeigen!

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

**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] ]... );

- DatenbankElemente 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.)