

Die Datenbank-Verbindungsklassen

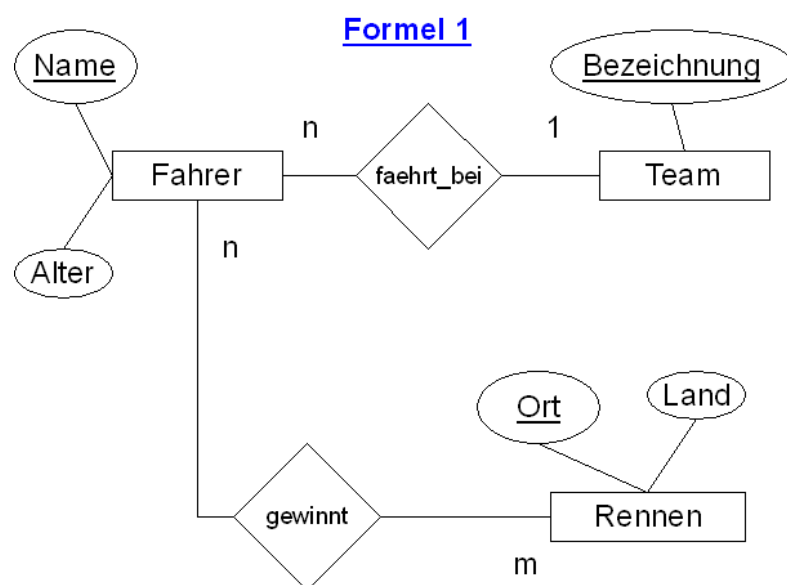
Die Toolbox hat zwei Klassen, die den Verbindungsaufbau zu einer MySQL-Datenbank bzw. zu einer Access-Datenbank kapseln.

In der Anwendung, die einen Datenbank-Zugriff braucht, sind dann nur folgende Schritte zu unternehmen:

- Eine Instanz der benötigten Datenbank-Verbindungsklasse anlegen.
- Über den Aufruf **conOeffnen** die Verbindung aufbauen.
- Mit **conExecute** SQL-Anweisungen an die Datenbank absetzen.
- Mit **conAbfrage** Abfragen an die Datenbank stellen. Das Ergebnis wird als **ResultSet** zurückgeliefert.

DB_Access	
db: C	DB_MySQL
conC	db: Connection
conC	conOeffnen(String, String, String): Connection
Auto	conClose(): boolean
Com	AutoCommit(boolean): void
Rollt	Commit(): void
conA	Rollback(): void
anza	conAbfrage(String): ResultSet
neue	anzahlDatensaetze(ResultSet): int
warh	neuerDatensatz(ResultSet): boolean
absC	warNull(ResultSet): boolean
getP	absolute(ResultSet, int): boolean
befo	getRow(ResultSet): int
istVc	beforeFirst(ResultSet): void
istNe	istVorErstemDatensatz(ResultSet): boolean
isClc	istNachLetztemDatensatz(ResultSet): boolean
getB	isClosed(ResultSet): boolean
getB	getBoolean(ResultSet, int): boolean
getD	getBoolean(ResultSet, String): boolean
getD	getDouble(ResultSet, int): double
getF	getDouble(ResultSet, String): double
getF	getFloat(ResultSet, int): float
getIn	getFloat(ResultSet, String): float
getIn	getInt(ResultSet, int): int
getL	getInt(ResultSet, String): int
getL	getLong(ResultSet, int): long
getS	getLong(ResultSet, String): long
getS	getString(ResultSet, int): String
getD	getString(ResultSet, String): String
getD	getDate(ResultSet, int): String
getT	getDate(ResultSet, String): String
getT	getTime(ResultSet, int): String
getT	getTime(ResultSet, String): String
getT	getTimestamp(ResultSet, int): String
getT	getTimestamp(ResultSet, String): String
conE	conExecute(String): boolean
conli	conInsertAutoIncrement(String): int

Im Folgenden wird eine Datenbank zum folgenden ER-Diagramm aufgebaut, die Datenbankl gefüllt und Abfragen auf die Datenbank durchgeführt.
Siehe Kommentar im Kopf der Datenbank-Komponente.



```
import java.sql.ResultSet;

/**
 * @author Witt
 *
 */
public class DBFormell_MySQL {

    DB_MySQL db;

    public DBFormell_MySQL() {
        db = new DB_MySQL();
        db.openConnection("localhost", "root", "");
    }

    public void dbAufgabe() {
        db.conExecute("CREATE Database IF NOT EXISTS Formell;");
        db.conExecute("USE Formell");

        db.conExecute("DROP TABLE IF EXISTS faehrt_bei;");
        db.conExecute("DROP TABLE IF EXISTS gewinnt ;");
        db.conExecute("DROP TABLE IF EXISTS fahrer ;");
        db.conExecute("DROP TABLE IF EXISTS team ;");
        db.conExecute("DROP TABLE IF EXISTS rennen ;");

        db.conExecute("CREATE TABLE fahrer ("
            + "Name VARCHAR(50) NOT NULL,"
            + "Geburtsjahr Integer ,"
            + "PRIMARY KEY (Name)"
            + " );");

        db.conExecute("INSERT INTO fahrer VALUES('Sebastian Vettel',1987);");
        db.conExecute("INSERT INTO fahrer VALUES('Mark Webber',1976 );");
        db.conExecute("INSERT INTO fahrer VALUES('Jenson Button',1980);");
        db.conExecute("INSERT INTO fahrer VALUES('Rubens Barrichello',1972);");
        db.conExecute("INSERT INTO fahrer VALUES('Lewis Hamilton',1985);");
        db.conExecute("INSERT INTO fahrer VALUES('Kimi Räikkönen',1979);");
        db.conExecute("INSERT INTO fahrer VALUES('Nick Heidfeld',1977);");

        db.conExecute("CREATE TABLE team ("
            + "Bezeichnung VARCHAR(50) NOT NULL,"
            + "PRIMARY KEY (Bezeichnung)"
            + " );");

        db.conExecute("INSERT INTO team VALUES('Red Bull Racing');");
        db.conExecute("INSERT INTO team VALUES('Brawn GP');");
        db.conExecute("INSERT INTO team VALUES('McLaren-Mercedes');");
        db.conExecute("INSERT INTO team VALUES('Ferrari');");
        db.conExecute("INSERT INTO team VALUES('BMW Sauber F1 Team');");

        db.conExecute("CREATE TABLE faehrt_bei ("
            + "Name VARCHAR(50) NOT NULL,"
            + "Bezeichnung VARCHAR(50) NOT NULL,"
            + "PRIMARY KEY (Name),"
            + "FOREIGN KEY (Name) REFERENCES fahrer (Name),"
            + "FOREIGN KEY (Bezeichnung) REFERENCES team (Bezeichnung)"
            + " );");

        db.conExecute("INSERT INTO faehrt_bei VALUES('Sebastian Vettel', 'Red Bull Racing');");
        db.conExecute("INSERT INTO faehrt_bei VALUES('Mark Webber', 'Red Bull Racing');");
        db.conExecute("INSERT INTO faehrt_bei VALUES('Jenson Button', 'Brawn GP');");
        db.conExecute("INSERT INTO faehrt_bei VALUES('Rubens Barrichello', 'Brawn GP');");
        db.conExecute("INSERT INTO faehrt_bei VALUES('Lewis Hamilton', 'McLaren-Mercedes');");
        db.conExecute("INSERT INTO faehrt_bei VALUES('Kimi Räikkönen', 'Ferrari');");
        db.conExecute("INSERT INTO faehrt_bei VALUES('Nick Heidfeld', 'BMW Sauber F1 Team');");

        db.conExecute("CREATE TABLE rennen ("
            + "Ort VARCHAR(50) NOT NULL,"
            + "Land VARCHAR(50),"
            + "Datum Date,"
            + "PRIMARY KEY (Ort)"
            + " );");

        db.conExecute("INSERT INTO rennen VALUES( 'Melbourne', 'Australien','2009-03-29');");
        db.conExecute("INSERT INTO rennen VALUES( 'Sepang', 'Malaysia','2009-04-05');");
        db.conExecute("INSERT INTO rennen VALUES( 'Shanghai', 'China','2009-04-19');");
        db.conExecute("INSERT INTO rennen VALUES( 'Manama', 'Bahrain','2009-04-26');");
        db.conExecute("INSERT INTO rennen VALUES( 'Barcelona', 'Spanien','2009-05-10');");
        db.conExecute("INSERT INTO rennen VALUES( 'Monte Carlo', 'Monaco','2009-05-24');");
        db.conExecute("INSERT INTO rennen VALUES( 'Istanbul', 'Türkei','2009-06-07');");
        db.conExecute("INSERT INTO rennen VALUES( 'Silverstone', 'Großbritannien','2009-06-21');");
        db.conExecute("INSERT INTO rennen VALUES( 'Nürburgring', 'Deutschland','2009-07-12');");
        db.conExecute("INSERT INTO rennen VALUES( 'Budapest', 'Ungarn','2009-07-26');");
```

```
db.conExecute("INSERT INTO rennen VALUES( 'Valencia', 'Spanien','2009-08-23');");
db.conExecute("INSERT INTO rennen VALUES( 'Spa-Francorchamps', 'Belgien','2009-08-30');");
db.conExecute("INSERT INTO rennen VALUES( 'Monza', 'Italien','2009-09-13');");
db.conExecute("INSERT INTO rennen VALUES( 'Singapur', 'Singapur','2009-09-27');");
db.conExecute("INSERT INTO rennen VALUES( 'Suzuka', 'Japan','2009-10-04');");
db.conExecute("INSERT INTO rennen VALUES( 'Sao Paulo', 'Brasilien','2009-10-18');");
db.conExecute("INSERT INTO rennen VALUES( 'Abu Dhabi', 'Abu Dhabi','2009-11-01');");

db.conExecute("CREATE TABLE gewinnt ( "
+ "Name VARCHAR(50) NOT NULL, "
+ "Ort VARCHAR(50) NOT NULL, "
+ "PRIMARY KEY (Name,Ort), "
+ "FOREIGN KEY (Name) REFERENCES fahrer (Name), "
+ "FOREIGN KEY (Ort) REFERENCES rennen (Ort) "
+ ")");

db.conExecute("INSERT INTO gewinnt VALUES('Jenson Button', 'Melbourne');");
db.conExecute("INSERT INTO gewinnt VALUES('Jenson Button', 'Sepang');");
db.conExecute("INSERT INTO gewinnt VALUES('Sebastian Vettel', 'Abu Dhabi');");
db.conExecute("INSERT INTO gewinnt VALUES('Jenson Button', 'Manama');");
db.conExecute("INSERT INTO gewinnt VALUES('Jenson Button', 'Barcelona');");
db.conExecute("INSERT INTO gewinnt VALUES('Jenson Button', 'Monte Carlo');");
db.conExecute("INSERT INTO gewinnt VALUES('Jenson Button', 'Istanbul');");

// Abfrage
ResultSet rs1 = db
.conAbfrage("SELECT f.Name , t.Bezeichnung FROM fahrer f , faehrt_bei a , team t"
+ " where (f.Name = a.Name) AND (a.Bezeichnung =t.Bezeichnung)");

int anzahl = db.anzahlDatensaetze(rs1);
System.out.println("Anzahl der Datensätze Fahrer - Team: " + anzahl);

while (db.neuerDatensatz(rs1)) {
    System.out.println("      "+db.getString(rs1, "Name") + " --- "
+ db.getString(rs1, 2));
}

ResultSet rs2 = db
.conAbfrage("SELECT f.Name , r.Ort FROM fahrer f , gewinnt g , rennen r"
+ " where (f.Name = g.Name) AND (g.Ort = r.Ort)");

anzahl = db.anzahlDatensaetze(rs2);
System.out.println("Anzahl der Datensätze Fahrer - Rennstrecke: " + anzahl);

while (db.neuerDatensatz(rs2)) {
    System.out.println("      "+db.getString(rs2, "Name") + " --- "
+ db.getString(rs2, 2));
}

db.conExecute("UPDATE gewinnt SET Ort='Shanghai' WHERE Ort = 'Abu Dhabi');");

ResultSet rs3 = db
.conAbfrage("SELECT f.Name , r.Ort , r.Datum FROM fahrer f , gewinnt g , rennen r"
+ " where (f.Name = g.Name) AND (g.Ort = r.Ort) "
+ " ORDER BY r.Datum");

anzahl = db.anzahlDatensaetze(rs3);
System.out.println("Anzahl der Datensätze Fahrer - Rennstrecke - aktualisiert: " + anzahl);

while (db.neuerDatensatz(rs3)) {
    System.out.println("      "+db.getString(rs3, "Name") + " --- "
+ db.getString(rs3, 2) + " : "
+ db.getDate(rs3, 3));
}

db.conClose();
}

/**
 * @param args
 */
public static void main(String[] args) {
    DBFormell_MySQL t = new DBFormell_MySQL();
    t.dbAufgabe();
}
}
```