

team	spieler	gewonnene Spiele	verlorene Spiele	unentschiedene Spiele
B	2		2	

Abbildung 3-7: Gruppierungen: Resultat der Anweisung mit HAVING-Klausel

Auch hierzu eine Java-Applikation, bei der zusätzlich über eine HAVING-Klausel das mindestens zweimalige Vorkommen eines jeden Kurstyps gefordert wird (das Beispiel ergänzt Programm 3-5 um die HAVING-Klausel).

Programm 3-6: GROUP BY- und HAVING-Klausel

```
// Programm 3-6: ./RDBMSundSQL/GroupByHaving.java
import java.sql.*;
public class GroupByHaving {
    public static void main(String[] args) throws Exception {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection c =
            DriverManager.getConnection("jdbc:odbc:Kurse");
        Statement s = c.createStatement();
        String sql =
            "SELECT typ, COUNT(*) AS count FROM Kurse GROUP BY typ";
        sql += " HAVING COUNT(typ)>=2";
        ResultSet sr = s.executeQuery(sql);
        while(sr.next()) {
            String typ = sr.getString("typ");
            System.out.println(typ+" "+sr.getString("count")+"x");
            Statement t = c.createStatement();
            ResultSet tr =
                t.executeQuery(
                    "SELECT * FROM Kurse WHERE typ LIKE '" + typ + "'");
            while(tr.next()) {
                System.out.print(" > "+ tr.getString("datum")+ ": ");
                System.out.println(tr.getString("bezeichnung"));
            }
        }
    }
} // Ende class GroupByHaving
```

Im Resultat fehlt entsprechend die Gruppe des Typs typ="V", da sie nur einfach besetzt ist:

```
P, 2x
> 1998-04-27: Objektorientierte Programmierung mit Java
> 1998-06-30: JDBC
S, 3x
> 1998-06-29: JavaScript
> 1998-07-13: HTML
> 1998-06-09: GUI-Programmierung mit Java
```