**L2\_2.3 Datenabfrage über zwei Tabellen – Information**

**Equi – Join**

Die gewünschte Fahrschülerliste soll die Attribute *vorname, nachname, strasse, hausnr, plz und ort* enthalten.   
Die Syntax des SELECT-Befehls lässt zunächst auf eine einfache Lösung schließen.

SELECT attribut\_1, attribut\_2, ….

FROM tabelle\_1, tabelle\_2…  
WHERE ….

Da die gewünschten Daten sowohl aus der Tabelle *fahrlehrer* als auch aus der Tabelle *orte* stammen, müssen in der FROM-Klausel beide Tabellen genannt werden.  
Demnach müsste der Befehl zur Ausgabe der Fahrlehrerliste folgendermaßen lauten:

SELECT vorname, nachname, strasse, hausnr, plz, ort

FROM fahrschueler, orte;

Das Ergebnis zeigt, dass der eingegebene Befehl nicht korrekt war.

Die Liste enthält zu jedem Fahrschüler alle Orte. Insgesamt werden 65 Zeilen aufgelistet.

Es wird also das sogenannte „kartesische Produkt“ aus den 13 Datensätzen der Tabelle *fahrschueler* und den 5 Datensätzen der Tabelle *orte* gebildet (13 \* 5 = 65).

Da in der Datenbank nur 13 Fahrschüler erfasst sind, darf eine korrekte Liste auch nur 13 Zeilen ausgeben.



Es dürfen nur die Zeilen angezeigt werden,  
bei denen der Wert des Attributs *ortnr*   
aus der Tabelle *fahrschueler* gleich dem   
Wert des Attributs *ortnr* aus der Tabelle   
*orte* ist.



Diese Selektionsbedingung muss in die WHERE-Klausel der SELECT-Anweisung aufgenommen werden.

Der vollständige SQL-Befehl muss also lauten:

SELECT nachname, vorname, strasse, hausnr, plz, ort

FROM fahrschueler, orte

WHERE fahrschueler.ortnr = orte.ortnr;

Bei Namensgleichheit von Attributen müssen in SQL-Anweisungen zusätzlich auch die Tabellennamen angegeben werden.

Da in beiden Tabellen der Name des **Attributs** *ortnr* identisch ist, müssen jeweils die **Tabellennamen** vorangestellt werden.

Diese Form der paarweisen Verknüpfung nennt man „**Equi – Join**“.