



6a. SQL-Abfragen über mehrere Tabellen mit Joins – Infotext

Immer wieder kommt es vor, dass Kunden die Ausleihfrist überziehen. Um das Erstellen von Mahnungen zu erleichtern, wünscht sich die Geschäftsführung des VideoCenters eine Liste mit den Namen aller Kunden und den von ihnen aktuell ausgeliehenen Filmen.

Ist ein Film noch nicht zurückgegeben, so hat das Attribut *leirueck* den vom System beim Ausleihen eingetragenen Wert '0000-00-00'. Doch allein mit dieser Information kommen Sie mit Ihren bisherigen Kenntnissen nicht ans Ziel: Die Kundennamen stehen in der Tabelle *kunden*, die Ausleihinformationen jedoch in der Tabelle *ausleihe*:

kunr	kuname	kuvorna	vinr	kunr	leiausda	leirueck
3270	Unfried	Nico	2344	3852	2013-04-12	2013-05-04
3852	Becker	Lester	2344	3270	2013-04-27	0000-00-00
2944	Flantz	Jaromir	2342	3852	2013-04-29	0000-00-00

Anfragen, die Einträge mehrerer Tabellen miteinander kombinieren, bezeichnet man als **join**.

In einem ersten Schritt wird die Anfrage auf mehrere Tabellen bezogen:

```
SELECT kuvorna, kuname, kunden.kunr, ausleihe.kunr, vinr, leiausda
FROM kunden, ausleihe
WHERE leirueck LIKE '0000-00-00'
```

Diese Anfrage liefert das **vollständige Kreuzprodukt** aller Einträge in beiden Tabellen, d. h. jeder Eintrag der einen Tabelle wird mit jedem Eintrag der anderen Tabelle kombiniert:

kuname	Kuvorna	kunden.kunr	ausleihe.kunr	vinr	leiausda	sinnvoll?
Unfried	Nico	3270	3270	2344	2013-04-27	✓
Unfried	Nico	3270	3852	2342	2013-04-29	✗
Becker	Lester	3852	3270	2344	2013-04-27	✗
Becker	Lester	3852	3852	2342	2013-04-29	✓
Flantz	Jaromir	2944	3270	2344	2013-04-27	✗
Flantz	Jaromir	2944	3852	2342	2013-04-29	✗

Das vollständige Kreuzprodukt macht allerdings gar **keinen Sinn**: Durch die vollständige Kombination aller Einträge entstehen auch unsinnige Kombinationen, Jaromir Flantz mit der Kundennummer 2944 hat z. B. nie einen Film ausgeliehen! In einem zweiten Schritt muss daher nun die Ergebnismenge auf diejenigen Einträge **eingeschränkt werden**, in denen die Kundennummer des Kunden der in der Ausleihe registrierten Kundennummer entspricht:

```
SELECT kuvorna, kuname, kunden.kunr, ausleihe.kunr, vinr, leiausda
FROM kunden, ausleihe
WHERE leirueck LIKE '0000-00-00' AND kunden.kunr = ausleihe.kunr
```

kuname	kuvorna	kunden.kunr	ausleihe.kunr	vinr	leiausda	sinnvoll?
Unfried	Nico	3270	3270	2344	2013-04-27	✓
Becker	Lester	3852	3852	2342	2013-04-29	✓

Da die Zusammenstellung des vollständigen Kreuzprodukts, d. h. eine Verknüpfung mehrerer Tabellen ohne weitere Einschränkung viel Rechenzeit verbraucht und manchmal zu einem Speicherüberlauf führt, bitten wir Sie auf solche Abfragen zu verzichten.



6b. SQL-Abfragen über mehrere Tabellen mit Joins – Aufgaben

Die Umsatzzahlen des VideoCenters lassen in letzter Zeit zu wünschen übrig. Das Sortiment soll verändert werden. Der Chef hält Sie für besonders pfiffig und beauftragt Sie einige Informationen zu besorgen.

1. Formulieren und notieren Sie SQL-Anfragen über mehrere Tabellen um die **Aufträge 1 bis 3** der [Auftragsliste 2](#) zu bearbeiten. Lassen Sie die Anfragen ausführen und notieren Sie sich die wichtigsten Ergebnisse.
Überprüfen Sie jeweils, ob das Ergebnis der Aufgabenstellung entspricht.
2. Formulieren und notieren Sie SQL-Anfragen über mehrere Tabellen um **drei weitere Aufträge Ihrer Wahl** aus der [Auftragsliste 2](#) zu bearbeiten.
Überprüfen Sie jeweils, ob das Ergebnis der Aufgabenstellung entspricht.
3. Sollten Sie noch Zeit haben, so bearbeiten Sie weitere Aufträge der [Auftragsliste 2](#).
4. Anspruchsvolle Aufträge finden Sie in der [Auftragsliste 3](#).