

Übung 6 zur Vorlesung Datenbanksysteme

Durch Votieren für eine Teilaufgabe erklären Sie sich bereit, die Lösung im Rahmen der Besprechung zu zeigen. (Bitte keine syntaktisch fehlerhaften Statements. Sie haben die Möglichkeit die Lösungen vorab auf IN_HORUS zu testen und weiterzuentwickeln.)

1. Entwickeln Sie SQL-Statements für die folgenden Aufgaben. Verwenden Sie bitte meine¹ Relationen bzgl. des in den vorigen Übungen entwickelten Datenmodells zum Kopierservice.
 - a) Liste Benutzer (Name, ID) mit dem Datum ihrer letzten Bestellung (sofern vorhanden), der Zahl bestellter Artikel und der Zahl der Bestellungen. Sortierung absteigend nach Datum der letzten Bestellung und Benutzername.

```
SELECT VName, Benutzer_Nr,  
      Max(Bestelldatum) AS Letzte_Bestellung,  
      count(*) AS Gesamtzahl,  
      count (Bestelldatum) AS Bestellte_Artikel,  
      count(DISTINCT Bestelldatum) AS Anzahl_Bestellungen  
FROM   Ggr_kop_Benutzer LEFT JOIN Ggr_kop_bestellt using (Benutzer_Nr)  
GROUP BY Benutzer_Nr  
ORDER BY Max(Bestelldatum) DESC, VName
```

- b) Nennen Sie Benutzer, die am Arbeitsort wohnen; d.h. Ort v. Privat- und Geschäftsadresse sind gleich). *Die Aufgabe zählt doppelt.*

```
SELECT Benutzer_Nr  
FROM   Ggr_kop_Adresse AS ap JOIN  
      Ggr_kop_Adresse AS ag USING (Benutzer_Nr)  
WHERE  ap.Adr_Typ='P' AND ag.Adr_Typ='G' AND  
      ap.ort = ag.ort
```

- c) Erzeuge eine Liste der Bibliotheksbenutzer mit Name und Privatadressen. Ist keine Privatadresse bekannt, soll stattdessen die Geschäftsadresse ausgegeben werden. Falls gar keine Adresse bekannt ist, sollen die Adress-Attribute leer bleiben. Verwenden Sie für die Lösung *keine* Mengenoperationen. *Die Aufgabe zählt doppelt.*

```
SELECT b.Benutzer_Nr, vname, nname,  
      COALESCE (padr.plz, gadr.plz) AS plz,  
      COALESCE (padr.ort, gadr.ort) AS ort,  
      COALESCE (padr.str, gadr.str) AS str,  
      COALESCE (padr.hausnr, gadr.hausnr) AS hausnr,  
      COALESCE (padr.adr_typ, gadr.adr_typ) AS adr_typ  
FROM   Ggr_kop_Benutzer as b LEFT JOIN Ggr_kop_Adresse as padr  
      on b.benutzer_nr=padr.benutzer_nr and padr.adr_typ='P'  
LEFT JOIN Ggr_kop_Adresse as gadr  
      on b.benutzer_nr=gadr.benutzer_nr and gadr.adr_typ='G' and padr.adr_typ is null
```

¹Relationen, deren Name mit ggr_ beginnt.

- d) Welche Benutzer haben die meisten *verschiedenen* Artikel bestellt?
Die Aufgabe zählt doppelt.

```
SELECT Benutzer_Nr, count(distinct aufsatz_id) FROM ggr_kop_bestellt
GROUP BY Benutzer_Nr HAVING count(distinct aufsatz_id) >= ALL
(SELECT count(distinct aufsatz_id) FROM ggr_kop_bestellt GROUP BY Benutzer_Nr)
```

- e) Von welchen Autoren wurde 2009 kein Artikel bestellt? Geben Sie auch die Namen der Autoren an. *Die Aufgabe zählt doppelt.*

```
SELECT autor_id, name
FROM ggr_kop_autor LEFT JOIN ggr_kop_aufsatz USING (autor_id) LEFT JOIN
ggr_kop_bestellt
ON (ggr_kop_aufsatz.aufsatz_id=ggr_kop_bestellt.aufsatz_id AND
EXTRACT (YEAR FROM bestelldatum) = 2009)
GROUP BY autor_id HAVING (count(bestelldatum)=0)
ORDER BY autor_id -- Sortierung nicht gefordert
```

- f) Erzeugen Sie eine Liste von Autoren, mit der Gesamtzahl der Bestellungen der von ihnen verfassten Artikel. Berücksichtigen Sie nur Autoren, wenn diese Gesamtzahl mindestens 2 ist und sortieren Sie nach dieser Zahl. *Die Aufgabe zählt doppelt.*

```
SELECT autor_id, name, count(bestelldatum) AS Gesamtzahl_Bestellungen
FROM ggr_kop_autor NATURAL JOIN ggr_kop_aufsatz NATURAL JOIN ggr_kop_bestellt
GROUP BY autor_id HAVING (count(bestelldatum)>=2)
ORDER BY Gesamtzahl_Bestellungen, autor_id -- Sortierung nach autor_id nicht gefordert
```

- g) Bestimmen Sie die Gesamtzahl der bestellten Kopien und der insgesamt kopierten Seiten von Artikeln, die etwas mit Datenbanken zu tun haben. Das sind Artikel, die mindestens eines der (Teil-)Wörter "Datenbank" oder "Database" im Titel haben, sowie Artikel aus Zeitschriften, die eines der beiden (Teil-) Wörter im Zeitschriftennamen haben.
Die Aufgabe zählt doppelt.

```
SELECT count(*) AS bestellte_Kopien, sum(bis_seite-von_seite+1) as Gesamtseitenzahl
FROM Ggr_kop_bestellt NATURAL JOIN Ggr_kop_Aufsatz NATURAL JOIN
Ggr_kop_Zeitschrift
WHERE
  zname LIKE '%Datenbank%' OR zname LIKE '%Database%' OR
  titel LIKE '%Datenbank%' OR titel LIKE '%Database%'
```

-- Hinweis: wenn Groß-/Kleinschreibung unberücksichtigt bleiben soll, muss die Bedingung lauten:

```
WHERE
  lower (zname) LIKE '%datenbank%' OR
  lower (zname) LIKE '%database%' OR
  lower (titel) LIKE '%datenbank%' OR lower (titel) LIKE '%database%'
```