Structured Query Language (Teil 1)

Tobias Bittner

2024-11-27

Übersicht

Übersicht

SQL - Testumgebung I

https://pad.stuve.fau.de/edb-ss24-sql

Übersicht

1.2 Allgemeine Fragen I

1. Nennen Sie eine gültige Reihenfolge der `INSERT`-Befehle, wenn ein Schadensfall mit drei beteiligten Fahrzeugen aufzunehmen ist. Dabei soll ein Fahrzeug zu einem eigenen Kunden“ und zwei Fahrzeuge zu Fremdkunden“ gehören; die eine fremde“ Versicherungsgesellschaft soll schon gespeichert sein, die andere nicht.

1.2 Allgemeine Fragen II

2. Liefern folgende Queries (bei beliebigem Zustand der Datenbank) unterschiedliche Ergebnisse? Warum bzw. warum nicht?

```
   SELECT *
     FROM Versicherungsvertrag vv
LEFT JOIN Versicherungsnehmer vn
       ON (vv.Versicherungsnehmer_ID = vn.ID
           AND vn.ID > 5)
```

```
   SELECT *
     FROM Versicherungsvertrag vv
LEFT JOIN Versicherungsnehmer vn
       ON (vv.Versicherungsnehmer_ID = vn.ID)
    WHERE vn.ID > 5
```

Übersicht

1.3 Fehlersuche I

1. Was ist an dieser Anfrage falsch?

```
SELECT ID, Vorname + ' ' + Name AS Kunde, Ort,
       Name AS Sachbearbeiter, Telefon
  FROM Versicherungsvertrag, Versicherungsnehmer, Mitarbei
 WHERE ID = 27
   AND Versicherungsvertrag.Versicherungsnehmer_ID
         = Versicherungsnehmer.ID
   AND Versicherungsvertrag.Mitarbeiter_ID
   = Mitarbeiter.ID;
```

Übersicht

1.4 Anfrageformulierung I

1. Erstellen Sie unter Verwendung eines Existenzquantors eine aufsteigend sortierte, wiederholungsfreie Liste aller Geburtsdaten von Versicherungsnehmern. Berücksichtigen Sie dabei nur Versicherungsnehmer, die aus Orten stammen, in denen mindestens eine Abteilung ansässig ist.
2. Erweitern Sie die vorherige Anfrage, sodass die ausgewählten Zeilen den folgenden Bedingungen entsprechen: Es geht ausschließlich um eigene Kunden (Eigener_kunde = 'J'). Vollkasko-Verträge sollen immer angezeigt werden, ebenso Fahrzeuge aus dem Kreis Recklinghausen 'RE'. Teilkasko-Verträge sollen angezeigt werden, wenn sie nach '1990-12-31' abgeschlossen wurden. Haftpflicht-Verträge sollen angezeigt werden, wenn sie nach '1985-12-31' abgeschlossen wurden. Wie viele Einträge zeigt die Ergebnismenge an?

1.4 Anfrageformulierung II

3. Suchen Sie zu jedem Mitarbeiter den Namen und Vornamen des Leiters der zugehörigen Abteilung (Ist_Leiter = 'J'). Die Abteilungsleiter in unserer einfachen Firmenhierarchie haben keinen Vorgesetzten; sie sollen in der Liste deshalb nicht aufgeführt werden.
4. Suchen Sie Einträge in der Tabelle `Versicherungsnehmer`, bei denen Name, Vorname, PLZ, Strasse übereinstimmen. Jeweils zwei dieser Adressen sollen mit ihrer ID und den übereinstimmenden Angaben aufgeführt werden.
   Hinweis: Benutzen Sie einen JOIN, der sich nicht auf übereinstimmende IDs bezieht.

Übersicht

1.5 Datenmodifikation I

1. Von der Deutschen Post AG wird eine Tabelle `plz_aenderung`(ID, PLZalt, Ortalt, PLZneu, Ortneu) geliefert:

```
CREATE TABLE PLZ_Aenderung (
  ID INT NOT NULL AUTO_INCREMENT,
  PLZalt INT NOT NULL,
  Ortalt VARCHAR(64) NOT NULL,
  PLZNeu INT NOT NULL,
  Ortneu VARCHAR(64) NOT NULL,
  PRIMARY KEY (ID));

INSERT INTO
    PLZ_Aenderung(ID, PLZalt, Ortalt, PLZNeu, Ortneu)
    VALUES ('1', '45658', 'Recklinghausen',
    '45659', 'Recklinghausen');
INSERT INTO
```

1.5 Datenmodifikation II

```
    PLZ_Aenderung(ID, PLZalt, Ortalt, PLZNeu, Ortneu)
    VALUES ('2', '45721', 'Hamm-Bossendorf',
    '45721', 'Haltern-Hamm');
INSERT INTO
    PLZ_Aenderung(ID, PLZalt, Ortalt, PLZNeu, Ortneu)
    VALUES ('3', '45772', 'Marl', '45770', 'Marl');
INSERT INTO
    PLZ_Aenderung(ID, PLZalt, Ortalt, PLZNeu, Ortneu)
    VALUES ('4', '45701', 'Herten', '45699', 'Herten');
```

Ändern Sie die Tabelle `Versicherungsnehmer` so, dass bei Adressen, bei denen `PLZ/Ort` mit `PLZalt/Ortalt` übereinstimmen, diese Angaben durch `PLZneu/Ortneu` geändert werden. Beschränken Sie sich auf die Änderung mit der `ID` 3.

Übersicht