

5. Versuch: Objektorientierung in ORACLE

Beachten Sie bei der Bearbeitung dieses Aufgabenblattes, dass die Aufgabe 5.2 auf dem Ergebnis von Aufgabe 5.1 aufbaut.

Aufgabe 5.1 (Objektorientierte Darstellung I: Basis; 30 P.)

Definieren Sie Objekttypen *City_Type*, *Country_Type* und *Province_Type*, die eine objektorientierte Modellierung dieser Entity-Klassen erlauben. Verwenden Sie dabei Referenzattribute.

Erstellen Sie mit diesen Datentypen entsprechende Objekttabellen.

Definieren und implementieren Sie für *Country* eine (*member function*) *pop_growth*, die das Bevölkerungswachstum ausgibt.

Formulieren Sie einen Pfadausdruck, der für jedes Land die Einwohnerzahl der Provinz, in der die Hauptstadt des Landes liegt, ausgibt.

Modellieren Sie die *borders*-Beziehung durch ein (symmetrisches) *relationales View* mit zwei Referenzattributen. (Warum kann man hier kein Objektview definieren?)

Aufgabe 5.2 (Objektorientierte Darstellung II: Organisationen; 25 P.)

Stellen Sie die Organisationen durch ein *Object View* dar. Verändern Sie dazu den Typ *Organization_Type* aus dem Kurs so, dass das Attribut *Members* entfällt, und die Mitgliedschaften stattdessen nur über die Methoden *set_member_type* (*hinzufügen und ändern von Mitgliedschaften*) und *get_member_type* (*Abfrage der Mitgliedschaft*) sowie die relationale Tabelle *isMember* verwaltet werden.

Weiterhin soll eine Methode *get_members* angeboten werden, die alle Mitglieds"ander (einschlie"sslich der Art der Mitgliedschaft) einer Organisation ausgibt.

Verwenden Sie dabei Referenzen auf *City/City_Type* und *Country/Country_Type* wie in Aufgabe 5.1.

Definieren Sie eine freie Funktion *get_member_type(country.code, organization.abbrev)* sowie eine freie Prozedur *make_member_type(country.code, organization.abbrev, type)* zum Testen.

- Berechnen Sie die Anzahl der Einwohner, die in einer Stadt leben, die Sitz einer Organisation ist.
- Stellen Sie eine SQL-Anfrage, die die Namen aller EU-Mitglieder, die mehr als 10.000.000 Einwohner haben, ausgibt.

Zusatzfrage:

Welche Integritätsbedingung/Einschränkung verliert man gegenüber dem relationalen Original-Mondial, wenn man mit Objektreferenzen modelliert? (d.h., was kann man dann wirklich nicht mehr ausdrücken, und ist das eher ein Vorteil oder Nachteil)

Aufgabe 5.3 (Objektorientierung: Flüsse; 15 P.)

Definieren Sie Objekttypen *River_Type* und *Lake_Type*, die Methoden anbieten, die die Länge eines Flusssystemes (d.h. eines Flusses sowie aller Flüsse, die direkt oder indirekt in ihn hineinfließen, einschließlich ihm selber) berechnen.

Erzeugen und Füllen Sie die entsprechenden Objekttabellen und fragen Sie die Längen der Flusssysteme verschiedener Flüsse ab.

Aufgabe 5.4 (Traveling in Costa Rica; 15 P.) Jemand fährt nach San Jose (Costa Rica), und sieht von der Stadt aus einen hohen Berg, auf den er hinauffährt. Wie heißt der Berg? Was kann er von dort aus alles (bei klarer Sicht) sehen?

(die Aufgabe darf schrittweise gelöst werden, indem man das Ergebnis einer Anfrage nutzt um sich den nächsten Schritt zu überlegen).

Es ist dabei sinnvoll, den bereits vorliegenden Datentyp "geocoord" um eine Funktion "distance" zu erweitern, die man häufiger braucht.

Falls bei der Lösung der Aufgabe eine "unerwartete" Fehlermeldung auftritt, analysieren Sie deren Ursache und eliminieren Sie diese sauber.

Aufgabe 5.5 (Nichttriviale ORDER-Methode; 15 P.)

Definieren Sie basierend auf einem Datentyp *Team_Type*, der die Daten eines Teams enthält, eine Tabelle *Bundesliga* (vgl. Rahmen der Aufgabe im Web). Für *Team_Type* soll eine **ORDER**-Methode definiert werden, die es erlaubt, über **ORDER BY** die Tabelle auszugeben (Definition der Ordnungsrelation vgl. Kurs/Skript).

Begründen Sie, warum die angegebene Ordnung keine Zyklen enthält.

Was müssen Sie ändern, wenn (wie bis 1968/69) der Torquotient anstelle der Tordifferenz verwendet wird? Wie ändert sich die Abschlusstabelle der Saison 1997/98?

Abgabe bis 26.6.2016