

5. Versuch: View Updates, Integrität, Indexe, PL/SQL

Aufgabe 5.1 (View Updates; 20 P.)

Führen Sie diese Aufgabe ohne referentielle Integritätsbedingungen aus, damit eventuelle Updates nicht daran scheitern.

- a) Erzeugen Sie eine Sicht, die für alle Länder einige der wichtigen Attribute aus *Politics*, *Economy* und *Population* sowie die Einwohnerdichte enthält. Formulieren Sie dazu eine Anfrage, die zu jedem Land das Bruttoinlandsprodukt pro Person, die Kindersterblichkeit und das Bevölkerungswachstum geordnet nach BIP pro Person ausgibt.

Stellen Sie fest, auf welchen Spalten der Sicht Updates erlaubt sind, und begründen Sie dies. Aktualisieren Sie die Datenbasis so, dass sie die erwartete Bevölkerungszahl im nächsten Jahr angibt.

- b) Erzeugen Sie eine zweite Sicht, die zusätzlich zu jedem Land den Kontinent angibt, auf dem das Land hauptsächlich liegt.

Stellen Sie auch hier fest, auf welchen Spalten der Sicht Updates erlaubt sind und begründen Sie dies. Versuchen Sie, durch Veränderungen an der Sicht die Datenbank so zu ändern, dass Russland in Afrika liegt.

Aufgabe 5.2 (Integrität; 15 P.)

Im Repository finden Sie ein ER-Diagramm (*diagramm_4.pdf*) und ein Skript (*diagramm_4.sql*), das ein dazugehöriges Datenbankschema erzeugt. Erweitern Sie das Schema um die folgenden Integritätsbedingungen, jeweils so einfach wie möglich (z.B. lieber mit einer CHECK-Klausel als per Trigger). Verändern Sie, wenn möglich, nur die vorhandenen Tabellendefinitionen.

- Nur Teilnehmer aus demselben Wettbewerbszeitraum (Saison) können Spiele austragen.
- Die Spielrunde muss ebenfalls zum Wettbewerbszeitraum der Teilnehmer gehören.
- Vereine können nicht gegen sich selbst spielen.
- Jeder Halbzeitstand muss geringer als der Endstand sein.

Hinweis: Verändern Sie in dieser und den folgenden Aufgaben das ursprüngliche Skript. Achten Sie darauf, dass es immer ausführbar bleibt. Passen Sie gegebenenfalls auch die Insert-Anweisungen an.

Aufgabe 5.3 (Indexe; 5 P.)

Die vorhandenen Indexe im Schema aus Aufgabe 2 sind nicht immer gut gewählt (für Primärschlüssel und UNIQUE-Bedingungen wird immer ein Index angelegt). Verbessern Sie die Indexe, um den Verbund zwischen Parent- und Child-Tabellen effizienter zu machen. Fügen Sie auch neue Indexe dazu, wenn nötig.

Aufgabe 5.4 (Universelle Relation für Obertyp; 15 P.)

Erweitern Sie die Tabelle *Spiel* zu einer universellen Relation für den Obertyp (vgl. *Spielrunde*). Entfernen Sie die Tabellen für die Untertypen von *Spiel* und passen Sie die Sichten an. Stellen Sie sicher, dass Tupel in *Verlängerung* trotzdem nur Spiele des Typs *Entscheidungsspiel* referenzieren können, obwohl der Fremdschlüssel nun direkt auf *Spiel* verweist. **Hinweis:** Sie müssen dazu nur die Tabellendefinitionen von *Verlängerung* und *Spiel* bearbeiten.

Aufgabe 5.5 (Disjunktheit Untertypen; 15 P.)

Die Tabelle *Wettbewerb* lässt sich wegen der verschiedenen Attribute der Untertypen nicht zu einer universellen Relation erweitern, ohne für die verschiedenen Untertypen jeweils einige Nullwerte für die unzutreffenden Attribute zu erhalten. Behalten Sie deshalb die vorhandene Tabellenstruktur bei. Erweitern Sie diese aber so, dass gewährleistet wird, dass die Extensionen der Untertypen paarweise disjunkt sind.

Aufgabe 5.6 (Funktionsbasierte Indexe; 10 P.)

Bisher sind die Bedingungen im Schema so formuliert, dass pro Saison ein Hinspiel und ein Rückspiel zwischen zwei Teilnehmern ausgetragen werden kann. Im DFB-Pokal gibt es jedoch immer nur ein Spiel. Schreiben Sie einen funktionsbasierten Index, der diese Eindeutigkeit sicherstellt.

Aufgabe 5.7 (Cursor; 10 P.)

Schreiben Sie eine PL/SQL-Prozedur, um zu berechnen, welcher Anteil der Weltbevölkerung mindestens notwendig ist, um 50% des weltweiten Bruttoinlandsproduktes zu erzeugen. Nehmen Sie dabei an, dass sich das

Bruttoinlandsprodukt eines Landes gleichmäßig auf alle Einwohner verteilt. Geben Sie an, wieviele Einwohner aus welchen Ländern beteiligt sind. **Hinweis:** Verwenden Sie einen Cursor über ein geeignet gewähltes `SELECT`-Statement. Sie können das Ergebnis in eine Tabelle einfügen oder per `DBMS_OUTPUT.PUT_LINE()` ausgeben.

Abgabe: 7.7.2010, 11h