Introduction to Databases (winter term 2004/2005)

Assignment 5

RWTH Aachen: hand in on November 29, 2004 during tutorial
B-IT: hand in on December 2, 2004 during tutorial

You can use the web front end to our Oracle test database to solve the tasks of this assignment. To access this database, you need an Oracle Access Ticket. If you do not have one yet, you can still get one at our chair’s office. The URL of the web front end is http://robinie.informatik.rwth-aachen.de:7783/isqlplus. Use testbas2 as the connection string.

On our homepage you find some links to a set of slides with a more extensive introduction to SQL. This might be helpful if you have questions concerning the concrete syntax.

Task 5.1 (SQL DDL) (3 Points)

Given is the following ER model:

It denotes a simplified hospital scenario describing doctors who work in hospitals and patients that are treated by doctors for a certain time period and stay in a hospital.

Assume the following domains and value constraints:

- Name, Specialization Area, ZIP, City and Street are represented as strings.
- SSN is stored as an integer.
- From and To store the start and end dates of the doctor’s treatment.
- Salary is an integer between 30,000 and 100,000.
- Type is a string. Hospitals can be either ‘private’ or ‘public’, but nothing else.

Design for the above ER model a relational database schema by formulating SQL statements that create the tables with appropriate constraints (e.g. foreign key constraints). Test your SQL statements by using the web interface to our Oracle test database.

Task 5.2 (SQL Queries I) (3 Points)

Given are the following relations:

- Visits with attributes guest and pub, which contains for every guest the pubs visited by him/her (Visits = ({guest, pub}, {{guest, pub} → ALL})).
- Sells with attributes pub and beer, which contains for every pub the beer brands sold by this pub (Sells = ({pub, beer}, {{pub, beer} → ALL})).
- Drinks with attributes guest and beer, which contains the preferred beer brands of every guest (Drinks = ({guest, beer}, {{guest, beer} → ALL})).

All attributes are string attributes. Every relation represents an n:m-relationship, i.e. the key is formed (as you can see from the schema definition) by all attributes of the relation.

Define the following queries as expressions in SQL:

a) Find all guests that drink ‘Hannen’.

b) Find all pubs which sell beer, that is preferred by John.

c) Find all guests who visit a pub and the pub sells a beer brand which is preferred by the guest.
d) Find all beers which are not preferred by any guest, but are sold by at least one pub.
e) Find all guests who drink beer, but that do not visit a pub which sells a beer preferred by them.
f) Find all guests who drink beer of at least two different brands.

Test your SQL queries in our Oracle test database. You will find three tables with the above names directly accessible from your user account.

Task 5.3 (SQL Queries II)  
Consider the following simplified ER model that is the basis for the luding database http://sunsite.informatik.rwth-aachen.de/luding/:

This model is mapped to the following relational database schema (all fields are text fields except the ids that are integers):

- game = ({gameid, gameName, type, ISBN, price}, {gameid})
- author = ({authorid, authorLastName, authorGivName}, {authorid})
- game_author = ({gameid, authorid}, {{gameid, authorid}})
- nIndReview = ({indReviewid, URL}, {indReviewid})
- Game_IndReview = ({indReviewid, gameid}, {{indReviewid, gameid}})
- publisher = ({publisherid, publishername}, {publisherid})
- game_publisher = ({gameid, publisherid}, {{gameid, publisherid}})

Define the following queries against this database in SQL:

a) Find the names of all games with the number of reviews, sorted by the number of reviews.
b) Find all authors who have designed more than two role playing games (‘RPG’).
c) Find all authors who have designed more than two games, which were published by the same publisher.
d) Find all authors who have designed a game of the type ‘BOARD’ and a game of the type ‘RPG’.

Check your queries with the Oracle test database. An (slightly outdated) excerpt of the luding database was copied to the Oracle schema edb2004. You can access its data e.g. by SELECT * FROM EDB2004.game.

Task 5.4 (SQL Updates)  
Given is the following relational database schema:

- Employee = ({ENumber, EName, Salary, DNumber}, {ENumber} )
- Department = ( {DNumber, DName, Location, Performance}, {DNumber} )
- Employee[DNumber] ⊆ Department[DNumber] (i.e. foreign key constraint)

Define the following operations in SQL:

a) All employees get a 10% raise in their salary.
b) All employees in Berlin and Munich get a raise depending on the performance of their department, i.e. the performance value equals the salary raise in percent. (E.g.: A performance value of 55 causes a raise of 55% of all employees who work in that department).

Make use of nested SELECT-statements if necessary. Test your statements on the tables employee and department in your private Oracle schema.

Some useful SQL commands for the Oracle database server:

- SELECT TABLE_NAME FROM USER_TABLES: Lists the names of all tables that are accessible by the user.
- DESCRIBE employee: Displays the schema of the table employee.