**CSCD 327 Lab #6 (20 points) Brandon Fowler**

**Due: February 19, 2014**

**Section 1: Database Update (2 points for each question)**

**Create a new database named as “*YourUserName\_University*”, and run the script database\_university.sql posted on Canvas. Use “*YourUserName\_university*” for the following exercises.**

**Write the following queries in BOTH RA and SQL statements, respectively. Include RA expressions and SQL statements in your submission.**

**1. Increase the salary of each instructor in the Comp. Sci. department by 10%.**

SQL

update instructor

set salary = salary \* 1.10

where dept\_name = 'Comp. Sci.';

R.A.

**2. Delete all courses that have never been offered (i.e., do not occur in the *section* relation).**

SQL

delete

from course

where course\_id not in(select course\_id from section);

R.A.

**3. Insert every student whose *tot\_cred* attribute is greater than 100 as an instructor in the same department, with a salary of $10,000.**

SQL

insert into instructor

select ID, name, dept\_name, 10000

from student

where tot\_cred > 100;

R.A.

**Brandon Fowler**

**4. Enroll every student in the Comp. Sci. department into CS-001 course, Section 1 of Fall 2009.**

SQL

insert into takes

select ID, 'CS-001', 1, 'Fall', 2009, null

from student

where dept\_name = 'Comp. Sci.';

R.A.

**5. Delete enrollments in CS-001 course, Section 1 of Fall 2009 where the student’s name is Zhang.**

SQL

delete

from takes

where course\_id = 'CS-001' and sec\_id = 1 and semester = 'Fall' and year = 2009 and ID in (select ID from student where name = 'Zhang');

R.A.

**6. Delete all *takes* tuples corresponding to any section of any course with the word “database” as a part of the title; ignore case when matching the word with the title.**

SQL

delete

from takes

where course\_id in (select course\_id from course where lower(title) like '%database%');

R.A.

**Brandon Fowler**

**7. Update *Tot\_Cred* in *student* relation. When you look at the *student* relation you will find that the *tot\_cred* field provides incorrect information. Now you are going to update this field with the real total credits the students received. Note that if a student got an *F* or the grade is *null*, he/she got 0 credits for that course. Display the *student* table after the update.**

SQL

update student, (select sum(credits) as sumofcreds, ID from takes natural join course where grade <> 'F' and grade is not null group by ID) as T1

set student.tot\_cred = T1.sumofcreds

where student.ID = T1.ID;

select \*

from student;

R.A

**Section 2: DDL with Constraints (1 point for each question)**

**Create a new database named as “YourUsername\_Constraints”, and use it for the following exercises.**

**8. Modify the following SQL command so that the Rep\_ID column is the PRIMARY KEY for the table and the default value of Y is assigned to the Comm column. (The Comm column indicates whether the sales representative earns commission.)**

CREATE TABLE STORE\_REPS (

Rep\_ID INT(5),

Last VARCHAR(15),

First VARCHAR(10),

Comm CHAR(1) DEFAULT 'Y',

PRIMARY KEY (Rep\_ID));

**9. Change the STORE\_REPS table so that NULL values CANNOT be entered in the name columns (First and Last).**

ALTER TABLE STORE\_REPS CHANGE First First VARCHAR(10) NOT NULL;

ALTER TABLE STORE\_REPS CHANGE Last Last VARCHAR(15) NOT NULL;

**Brandon Fowler**

**10. Create a table named BOOK\_STORES to include the columns listed in the following chart:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Datatype** | **Constraint Comments** |
| **Store\_ID** | **INT(8)** | **PRIMARY KEY column** |
| **Name** | **VARCHAR(30)** | **Should be UNIQUE and NOT NULL** |
| **Contact** | **VARCHAR(20)** | |
| **Rep\_ID** | **INT(5)** | |

CREATE TABLE BOOK\_STORES( Store\_ID INT(8), Name VARCHAR(30) UNIQUE NOT NULL, Contact VARCHAR(20), REP\_ID INT(5), PRIMARY KEY(STORE\_ID));

**11. Add a constraint to make sure the Rep\_ID value entered in the BOOK\_STORES table is a valid value contained In the STORE\_REPS table.**

ALTER TABLE BOOK\_STORES ADD CONSTRAINT Valid\_Rep\_ID FOREIGN KEY (Rep\_ID) REFERENCES STORE\_REPS (Rep\_ID);

**12. Change the constraint created in the previous question so that associated rows of the BOOK\_STORES table are deleted automatically if a row in the STORE\_REPS table is deleted.**

ALTER TABLE BOOK\_STORES ADD CONSTRAINT Valid\_Rep\_ID FOREIGN KEY (Rep\_ID) REFERENCES STORE\_REPS (Rep\_ID) ON DELETE CASCADE;

**13. Create a table named REP\_CONTRACTS containing the columns listed in the following chart. A composite PRIMARY KEY constraint including the Rep\_ID, Store\_ID, and Quarter columns should be assigned. In addition, FOREIGN KEY constraints should be assigned to both the Rep\_ID and Store\_ID columns.**

|  |  |
| --- | --- |
| **Column Name** | **DataType** |
| **Store\_ID** | **INT(8)** |
| **Name** | **INT(5)** |
| **Quarter** | **CHAR(3)** |
| **Rep\_ID** | **INT(5)** |

CREATE TABLE REP\_CONTRACTS(Store\_ID INT(8), Name INT(5), Quarter CHAR(3), Rep\_ID INT(5), PRIMARY KEY(Rep\_ID, Store\_ID, Quarter), FOREIGN KEY (Rep\_ID) REFERENCES STORE\_REPS (Rep\_ID), FOREIGN KEY (Store\_ID) REFERENCES STORE\_REPS (Store\_ID));