Andreas Landgrebe

Laboratory Assignment 5: Using the SQL Data Manipulation Language

10/22/2014

2. Convincing evidence to demonstrate that your create table statements ran correctly.





3. A description of how you configured SQLite to run correctly for your schema.

In order to configure SQLite, there were two important things that I needed to do. The first thing that I needed to do is to use the .read statement to be able to read the databases in to be able to complete this laboratory assignment. The second thing that I needed to do is to make sure the SQLite is configured so that it does ignore the constraints of the foreign key. In order to do so, one would to write the following command: PRAGMA foreign\_keys = ON

Once I was able to write this command, SQLite was able to look for the foreign key constrictions.

4. The final listing of your create table statements ran correctly.

CREATE TABLE School

(id varchar(5), --id of school

name varchar(20) not null, --name of the school

state varchar(2) not null, --state that the school is in

town varchar(20) , --town that the school is in

size varchar(4) –size of the school

);

5. A complete listing of the update, insert, and delete statements that you ran.

INSERT STATEMENTS RAN:

-INSERT INTO classroom (building, room\_number, capacity)

VALUES ('Alden', '101', 100);

- INSERT INTO department (dept\_name,building, budget)

VALUES ('Comp\_Sci', 'Alden', 260000.00);

-INSERT INTO course (course\_id, title, dept\_name, credits)

VALUES ('CS380', 'Database System Concepts', 'Comp\_Sci', 4);

-INSERT INTO instructor (ID, name, dept\_name)

VALUES ('1', 'Kapfhammer', 'Comp\_Sci');

-INSERT INTO section (course\_id,sec\_id,semester,year,building,room\_number)

VALUES ('CS380', '380-01', 'Fall', 2014, 'Alden' , 101);

-INSERT INTO student (ID, name, dept\_name)

VALUES ('15', 'landgrebea', 'Comp\_Sci);

-INSERT INTO teaches (ID, course\_id, sec\_id, semester)

Values ('15', 'CS380', '380-01', 'Fall');

-INSERT INTO takes (ID,course\_id,sec\_id,semester,year)

VALUES ('15', 'CS380', '380-01', 'Fall', 2014);

-INSERT INTO advisor (s\_ID)

VALUES ('1');

-INSERT INTO time\_slot (time\_slot\_id,day,start\_hr,start\_min)

VALUES ('11am', 'T', 11, 12);

-INSERT INTO prereq (course\_id)

VALUES ('CS380');

UPDATE Statement ran:

UPDATE instructor SET salary = 75000.00 WHERE ID = '1';

DELETE STATEMENT RAN:

DELETE FROM classes WHERE building = 'Alden';

6. A complete listing of the select statements that you used to query the database.

SELECT STATEMENTS RAN

SELECT \* FROM classroom;

SELECT \* FROM department;

SELECT \* FROM course;

SELECT \* FROM instructor;

SELECT \* FROM section;

SELECT \* FROM teaches;

SELECT \* FROM student;

SELECT \* FROM takes;

SELECT \* FROM advisor;

SELECT \* FROM time\_slot;

SELECT \* FROM prereq;

SELECT \* FROM School;

7. For each command, the before and after state for each modified table, highlighting all changes.

Before State of classroom:

create table classroom

(building varchar(15),

room\_number varchar(7),

capacity numeric(4,0),

primary key (building, room\_number)

);

After Insert:

This is the select statement modified where the header is on and the mode is column.



After Running Delete Statement:

DELETE FROM classroom WHERE building = 'Alden';



Before state of department:

create table department

(dept\_name varchar(20),

building varchar(15),

budget numeric(12,2) check (budget > 0),

primary key (dept\_name)

);

After Insert:

This is the select statement modified where the header is on and the mode is column.



Before state of course:

create table course

(course\_id varchar(8),

title varchar(50),

dept\_name varchar(20),

credits numeric(2,0) check (credits > 0),

primary key (course\_id),

foreign key (dept\_name) references department

on delete set null

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of insructor;

create table instructor

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

salary numeric(8,2) check (salary > 29000),

primary key (ID),

foreign key (dept\_name) references department

on delete set null

);

After insert:

This is the select statement modified where the header is on and the mode is column.



After update:



Before state of section:

create table section

(course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6)

check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),

year numeric(4,0) check (year > 1701 and year < 2100),

building varchar(15),

room\_number varchar(7),

time\_slot\_id varchar(4),

primary key (course\_id, sec\_id, semester, year),

foreign key (course\_id) references course

on delete cascade,

foreign key (building, room\_number) references classroom

on delete set null

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of teaches:

create table teaches

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id,sec\_id, semester, year) references section

on delete cascade,

foreign key (ID) references instructor

on delete cascade

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of student:

create table student

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

tot\_cred numeric(3,0) check (tot\_cred >= 0),

primary key (ID),

foreign key (dept\_name) references department

on delete set null

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of takes:

create table takes

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

grade varchar(2),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id,sec\_id, semester, year) references section

on delete cascade,

foreign key (ID) references student

on delete cascade

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of advisor:

create table advisor

(s\_ID varchar(5),

i\_ID varchar(5),

primary key (s\_ID),

foreign key (i\_ID) references instructor (ID)

on delete set null,

foreign key (s\_ID) references student (ID)

on delete cascade

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of time\_slot

create table time\_slot

(time\_slot\_id varchar(4),

day varchar(1),

start\_hr numeric(2) check (start\_hr >= 0 and start\_hr < 24),

start\_min numeric(2) check (start\_min >= 0 and start\_min < 60),

end\_hr numeric(2) check (end\_hr >= 0 and end\_hr < 24),

end\_min numeric(2) check (end\_min >= 0 and end\_min < 60),

primary key (time\_slot\_id, day, start\_hr, start\_min)

);

After insert:

This is the select statement modified where the header is on and the mode is column.



Before state of prereq:

create table prereq

(course\_id varchar(8),

prereq\_id varchar(8),

primary key (course\_id, prereq\_id),

foreign key (course\_id) references course

on delete cascade,

foreign key (prereq\_id) references course

);

After insert:

This is the select statement modified where the header is on and the mode is column.



8. A reflection of the challenges that you faced when completing this laboratory assignment.

There were a number of challenges that I faced during this laboratory assignment. One of the challenges that I faced was being able to understand the syntax of the different statements to be able to run the commands. In order to overcome this challenge, I was able to look up on the Internet and in the book and studied the syntax to be able to run these commands which I was able to overcome successfully to be able to complete this laboratory assignment in time.