CEE 505

Jean-Luc Jackson

11/20/2016

Assignment #5

Database Access Problems

Problem Statement

Use the database provided (Scores.db) and the interface sqlite3 and design a single SQL statement to answer each of the following questions:

1. *Output a list of students born between June 16, 1991 and September 15, 1996*
2. *Output the number of students born between June 16, 1991 and September 15, 1996*
3. *Output a list of students who have missed one or more labs (Score <= 0.1 to avoid numeric truncation errors)*
4. *Output the name of the student with the best score at the final*
5. *Output the name of the student closest to the average score of midterm 1*
6. *Output the accumulated homework score (sum of all assignment-type score) for the students identified in 4. and 5., respectively.*
7. *Create a VIEW named altAssignments, listing Assignment.ID, Assignment.name, Type.name, and sorted by Type.name. For reporting use*
   *sqlite3> SELECT * FROM altAssignments;*
   and provide the output of
   sqlite3> .schema altAssignment
   and explain what it tells you.
8. *Create a series of INSERT statements that create a user entry for yourself, full score on all homeworks, 80% on Midterm 1, 90% on Midterm 2, and 99% on the Final. Show all the newly added information through SELECT statements on the respective tables (make sure to design those SELECT statements to filter only those showing data for your record)*

SQL Input & Output

The SQL input and output are included in the following pages. The input .sql file was ran in sqlite3 and outputted to a .txt file using the format:

```
>sqlite3 < CEE505_HW5_Jackson.sql > HW5_Output.txt
```

```
.open Scores.db

.headers on
.mode column

/* ********** Problem 1 **********
Output a list of students born between June 16, 1991 and September 15, 1996
*/
.print 'Problem 1'
.print 'List of students born between 1991-06-16 and 1996-09-15'

SELECT * FROM Students
    WHERE DOB BETWEEN
    '1991-06-16' AND '1996-09-15';

.print ''

/* ********** Problem 2 **********
Output the number of students born between June 16, 1991 and September 15, 1996
*/
.print 'Problem 2'
.print 'Number of students born between 1991-06-16 and 1996-09-15'

SELECT COUNT(*) FROM Students
    WHERE DOB BETWEEN
    '1991-06-16' AND '1996-09-15';

.print ''

/* ********** Problem 3 **********
Output a list of students who have missed one or more labs
(Score <= 0.1 to avoid numeric truncation errors)
*/
.print 'Problem 3'
.print 'List of students who have missed one or more labs'

SELECT s.ID, s.name, a.name, c.Score
    FROM Students as s, Scores as c, Assignments as a
    WHERE a.typeID = 2
    AND c.Score <= 0.1
    AND c.StudentID = s.ID
    AND c.AssignmentID = a.ID;

.print ''

/* ********** PROBLEM 4 **********
Output the name of the student with the best score at the final
*/
.print 'Problem 4'
.print 'Name of student with best score on the final'

SELECT s.name, a.name, MAX(c.Score)
    FROM Students as s, Assignments as a, Scores as c
    WHERE a.typeID = 4
        AND a.ID = c.AssignmentID
        AND c.StudentID = s.ID;

.print ''

/* ********** PROBLEM 5 **********
Output the name of the student closest to the average score of midterm 1
*/
.print 'Problem 5'

.print 'Midterm 1 Average'
--Print midterm average for verification
```

```
SELECT AVG(c.Score)
    FROM Scores as c, Assignments as a, Students as s
    WHERE a.name LIKE '%Midterm 1%'
        AND c.AssignmentID = a.ID
        AND s.ID = c.StudentID;

.print ''
.print 'Closest Student to Class Average'
--Make list sorted by the absolute different b/n student's score and average,
--Take the 1st value in that sorted table
SELECT s.name, a.name, c.Score
    FROM Students as s, Assignments as a, Scores as c,
        (SELECT (
            SELECT AVG(c.Score)
                FROM Scores as c, Assignments as a, Students as s
                WHERE a.name LIKE '%Midterm 1%'
                    AND c.AssignmentID = a.ID
                    AND s.ID = c.StudentID
                ) as Midterm1Ave) as m
    WHERE a.name LIKE '%Midterm 1%'
        AND c.AssignmentID = a.ID
        AND s.ID = c.StudentID
    GROUP BY ABS(c.Score - m.Midterm1Ave)
    LIMIT 1;

.print ''

/* ********** PROBLEM 6 **********
Output the accumulated homework score (sum of all assignment-type score)
for the students identified in 4. and 5., respectively.
*/
.print 'Problem 6'
.print 'Accumulated homework score for students from Problems 4 and 5'

--Best final score by student:              F42DC
--Closest to midterm1average by student:    F42C0

SELECT stuAss.name as 'Student Name',
       stuAss.'Assignment Sum' as 'Assignment Sum',
       stuLab.'Lab Sum' as 'Lab Sum',
       stuMid.'Midterm Sum' as 'Midterm Sum',
       stuFin.'Final Sum' as 'Final Sum'

    -- get all assignment scores
FROM (SELECT stu.name, SUM(stu.Score) as 'Assignment Sum'
        FROM (SELECT s.name, s.ID, a.name, c.Score, a.typeID
                FROM Scores as c, Students as s, Assignments as a, Types as t
                WHERE s.ID = c.StudentID
                AND c.AssignmentID = a.ID
                AND a.typeID = t.typeID
                AND (s.name = 'F42DC'
                     OR s.name ='F42C0')
              ) as stu
        WHERE stu.typeID = 1
        GROUP BY stu.name) as stuAss,

    -- get all lab scores
    (SELECT stu.name, SUM(stu.Score) as 'Lab Sum'
        FROM (SELECT s.name, s.ID, a.name, c.Score, a.typeID
                FROM Scores as c, Students as s, Assignments as a, Types as t
                WHERE s.ID = c.StudentID
                AND c.AssignmentID = a.ID
                AND a.typeID = t.typeID
                AND (s.name = 'F42DC'
                     OR s.name ='F42C0')
             ) as stu
        WHERE stu.typeID = 2
        GROUP BY stu.name) as stuLab,

    -- get all midterm scores
    (SELECT stu.name, SUM(stu.Score) as 'Midterm Sum'
        FROM (SELECT s.name, s.ID, a.name, c.Score, a.typeID
                FROM Scores as c, Students as s, Assignments as a, Types as t
                WHERE s.ID = c.StudentID
                AND c.AssignmentID = a.ID
                AND a.typeID = t.typeID
                AND (s.name = 'F42DC'
                     OR s.name ='F42C0')
             ) as stu
        WHERE stu.typeID = 3
        GROUP BY stu.name) as stuMid,

    -- get all final scores
    (SELECT stu.name, SUM(stu.Score) as 'Final Sum'
        FROM (SELECT s.name, s.ID, a.name, c.Score, a.typeID
                FROM Scores as c, Students as s, Assignments as a, Types as t
                WHERE s.ID = c.StudentID
                AND c.AssignmentID = a.ID
                AND a.typeID = t.typeID
                AND (s.name = 'F42DC'
                     OR s.name ='F42C0')
             ) as stu
        WHERE stu.typeID = 4
        GROUP BY stu.name) as stuFin

WHERE stuAss.name = stuLab.name
AND stuLab.name = stuMid.name
AND stuMid.name = stuFin.name;


.print ''

/* ********** PROBLEM 7 **********
Create a VIEW named altAssignments, listing Assignment.ID, Assignment.name,
Type.name, and sorted by Type.name.  For reporting use
    sqlite3> SELECT * FROM altAssignments;
and provide the output of
    sqlite3> .schema altAssignment
and explain what it tells you.
*/
.print 'Problem 7'

.print 'Create a VIEW'
DROP VIEW IF EXISTS altAssignments;

CREATE VIEW altAssignments AS
    SELECT a.ID, a.name, t.name
    FROM Assignments as a, Types as t
    WHERE t.typeID = a.typeID
    ORDER BY t.name;

.print 'Reporting VIEW:'
SELECT * FROM altAssignments;

.print ''

.print '.schema altAssignments'
.schema altAssignments

.print ''

/* ********** PROBLEM 8 **********
```

```
Create a series of INSERT statements that create a user entry for yourself,
full score on all homeworks, 80% on Midterm 1, 90% on Midterm 2,
and 99% on the Final.  Show all the newly added information through
SELECT statements on the respective tables (make sure to design those
SELECT statements to filter only those showing data for your record)
*/
.print 'Problem 8'

.print 'Create a user entry for yourself in Students'
INSERT INTO Students ( ID , name , DOB )
VALUES ( 5052392 , 'JLJ23', '1992-03-23' );

.print 'Input assignment scores in Scores'
INSERT INTO Scores --( itemID , AssignmentID , StudentID , Score )
       -- Assignments
VALUES ( 5498 , 1 , 5052392 , 60.0 ),
       ( 5499 , 3 , 5052392 , 60.0 ),
       ( 5500 , 5 , 5052392 , 70.0 ),
       ( 5501 , 7 , 5052392 , 80.0 ),
       ( 5502 , 9 , 5052392 , 21.0 ),
       ( 5503 , 10 , 5052392 , 50.0 ),
       ( 5504 , 13 , 5052392 , 70.0 ),
       ( 5505 , 15 , 5052392 , 60.0 ),
       ( 5506 , 17 , 5052392 , 60.0 ),
       ( 5507 , 23 , 5052392 , 20.0 ),
       -- Labs
       ( 5508 , 2 , 5052392 , 10.0 ),
       ( 5509 , 4 , 5052392 , 10.0 ),
       ( 5510 , 6 , 5052392 , 10.0 ),
       ( 5511 , 8 , 5052392 , 10.0 ),
       ( 5512 , 11 , 5052392 , 10.0 ),
       ( 5513 , 14 , 5052392 , 10.0 ),
       ( 5514 , 16 , 5052392 , 10.0 ),
       ( 5515 , 19 , 5052392 , 10.0 ),
       ( 5516 , 20 , 5052392 , 10.0 ),
       ( 5517 , 21 , 5052392 , 10.0 ),
       -- Midterms
       ( 5518 , 12 , 5052392 , 80.0 ),
       ( 5519 , 18 , 5052392 , 90.0 ),
       -- Final
       ( 5520 , 22 , 5052392 , 99.0 );

.print 'Displaying updated Students'
SELECT * FROM Students
WHERE name = 'JLJ23';

.print ''

.print 'Displaying updated Scores'
SELECT * FROM Scores
WHERE StudentID = 5052392;
```

```
Problem 1
List of students born between 1991-06-16 and 1996-09-15
ID          name        DOB
----------  ----------  ----------
1000002     F4242       1994-06-21
1000003     F4243       1995-01-17
1000008     F4248       1991-11-20
1000011     F424B       1994-07-12
1000015     F424F       1995-03-04
1000018     F4252       1992-04-22
1000021     F4255       1995-12-10
1000026     F425A       1996-07-22
1000027     F425B       1993-01-03
1000029     F425D       1993-11-12
1000031     F425F       1994-02-24
1000036     F4264       1996-06-05
1000043     F426B       1993-02-20
1000046     F426E       1994-02-23
1000051     F4273       1992-03-29
1000053     F4275       1995-10-19
1000066     F4282       1994-07-03
1000067     F4283       1996-03-21
1000069     F4285       1994-01-15
1000071     F4287       1993-08-16
1000075     F428B       1992-10-12
1000077     F428D       1992-04-01
1000079     F428F       1993-07-01
1000081     F4291       1995-11-07
1000083     F4293       1991-07-09
1000084     F4294       1993-05-17
1000085     F4295       1996-01-11
1000088     F4298       1993-02-23
1000094     F429E       1992-06-18
1000097     F42A1       1995-08-04
1000098     F42A2       1994-03-21
1000100     F42A4       1995-07-25
1000104     F42A8       1991-08-15
1000105     F42A9       1995-04-23
1000106     F42AA       1995-07-19
1000107     F42AB       1991-09-10
1000112     F42B0       1991-11-05
1000116     F42B4       1993-10-13
1000118     F42B6       1992-09-09
1000119     F42B7       1995-02-03
1000121     F42B9       1993-06-26
1000125     F42BD       1996-03-18
1000126     F42BE       1996-05-28
1000127     F42BF       1994-10-08
```

```
1000128     F42C0       1994-01-01
1000129     F42C1       1992-04-03
1000135     F42C7       1993-04-08
1000136     F42C8       1996-06-18
1000139     F42CB       1994-08-26
1000145     F42D1       1991-08-28
1000149     F42D5       1994-01-24
1000151     F42D7       1995-06-29
1000154     F42DA       1992-05-05
1000156     F42DC       1995-05-26
1000158     F42DE       1994-06-21
1000159     F42DF       1996-04-27
1000160     F42E0       1996-02-12
1000163     F42E3       1994-12-26
1000167     F42E7       1991-08-21
1000171     F42EB       1994-09-09
1000172     F42EC       1993-09-11
1000174     F42EE       1996-04-24
1000177     F42F1       1993-02-08
1000178     F42F2       1995-03-04
1000180     F42F4       1993-06-02
1000181     F42F5       1993-06-04
1000182     F42F6       1994-03-27
1000183     F42F7       1991-12-04
1000184     F42F8       1994-04-17
1000190     F42FE       1996-09-06
1000191     F42FF       1993-03-07
1000196     F4304       1995-08-08
1000197     F4305       1996-09-13
1000198     F4306       1995-12-14
1000200     F4308       1993-08-20
1000203     F430B       1993-03-07
1000210     F4312       1993-10-26
1000213     F4315       1994-03-06
1000214     F4316       1994-01-22
1000216     F4318       1996-02-26
1000217     F4319       1992-10-08
1000219     F431B       1994-06-18
1000220     F431C       1993-07-28
1000222     F431E       1996-02-15
1000229     F4325       1992-11-02
1000234     F432A       1995-06-29
1000238     F432E       1996-08-09

Problem 2
Number of students born between 1991-06-16 and 1996-09-15
COUNT(*)
----------
```

```
87

Problem 3
List of students who have missed one or more labs
ID          name        name              Score
----------  ----------  ----------------  ----------
1000017     F4251       Lab #3 (2786810)  0.0
1000019     F4253       Lab #1 (2829219)  0.0
1000029     F425D       Lab #2 (2786809)  0.0
1000029     F425D       Lab #7 (2786813)  0.0
1000042     F426A       Lab #1 (2829219)  0.0
1000054     F4276       Lab #2 (2786809)  0.0
1000058     F427A       Lab #6 (2786812)  0.0
1000064     F4280       Lab #10 (2786815  0.0
1000065     F4281       Lab #10 (2786815  0.0
1000071     F4287       Lab #6 (2786812)  0.0
1000081     F4291       Lab #10 (2786815  0.0
1000091     F429B       Lab #10 (2786815  0.0
1000093     F429D       Lab #3 (2786810)  0.0
1000111     F42AF       Lab #1 (2829219)  0.0
1000111     F42AF       Lab #10 (2786815  0.0
1000117     F42B5       Lab #1 (2829219)  0.0
1000118     F42B6       Lab #3 (2786810)  0.0
1000132     F42C4       Lab #1 (2829219)  0.0
1000172     F42EC       Lab #1 (2829219)  0.0
1000172     F42EC       Lab #2 (2786809)  0.0
1000195     F4303       Lab #1 (2829219)  0.0
1000197     F4305       Lab #1 (2829219)  0.0
1000203     F430B       Lab #8 (2870743)  0.0
1000209     F4311       Lab #2 (2786809)  0.0
1000209     F4311       Lab #7 (2786813)  0.0
1000219     F431B       Lab #8 (2870743)  0.0
1000228     F4324       Lab #8 (2870743)  0.0
1000229     F4325       Lab #1 (2829219)  0.0

Problem 4
Name of student with best score on the final
name        name                  MAX(c.Score)
----------  --------------------  ------------
F42DC       Final Exam (2786798)  96.0

Problem 5
Midterm 1 Average
AVG(c.Score)
----------------
66.2238493723849

Closest Student to Class Average
```

```
name        name                 Score
----------  -------------------  ----------
F42C0       Midterm 1 (2786796)  66.0

Problem 6
Accumulated homework score for students from Problems 4 and 5
Student Name  Assignment Sum  Lab Sum     Midterm Sum  Final Sum
------------  --------------  ----------  -----------  ----------
F42C0         606.0           101.0       142.5        71.5
F42DC         606.0           103.0       179.0        96.0

Problem 7
Create a VIEW
Reporting VIEW:
ID          name                              name:1
----------  --------------------------------  ----------
1           Homework Assignment #1 (2786783)  Assignment
3           Homework Assignment #2 (2786789)  Assignment
5           Homework Assignment #3 (2786785)  Assignment
7           Homework Assignment #4 (2786784)  Assignment
9           Homework Assignment #5 - Quick A  Assignment
10          Homework Assignment #5 - Problem  Assignment
13          Homework Assignment #6 (2786791)  Assignment
15          Homework Assignment #7 (2786790)  Assignment
17          Homework Assignment #8 (2786787)  Assignment
23          Bonus Assignment #9 (2786795)     Assignment
22          Final Exam (2786798)              Final
2           Lab #1 (2829219)                  Labs
4           Lab #2 (2786809)                  Labs
6           Lab #3 (2786810)                  Labs
8           Lab #4 (2786811)                  Labs
11          Lab #5 (2856765)                  Labs
14          Lab #6 (2786812)                  Labs
16          Lab #7 (2786813)                  Labs
19          Lab #8 (2870743)                  Labs
20          Lab #9 - Beam Lab (2786814)       Labs
21          Lab #10 (2786815)                 Labs
12          Midterm 1 (2786796)               Midterm
18          Midterm 2 (2786797)               Midterm

.schema altAssignments
CREATE VIEW altAssignments AS
        SELECT a.ID, a.name, t.name
        FROM Assignments as a, Types as t
        WHERE t.typeID = a.typeID
        ORDER BY t.name;

Problem 8
```

```
Create a user entry for yourself in Students
Input assignment scores in Scores
Displaying updated Students
ID          name        DOB
----------  ----------  ----------
5052392     JLJ23       1992-03-23

Displaying updated Scores
itemID      AssignmentID  StudentID   Score
----------  ------------  ----------  ----------
5498        1             5052392     60.0
5499        3             5052392     60.0
5500        5             5052392     70.0
5501        7             5052392     80.0
5502        9             5052392     21.0
5503        10            5052392     50.0
5504        13            5052392     70.0
5505        15            5052392     60.0
5506        17            5052392     60.0
5507        23            5052392     20.0
5508        2             5052392     10.0
5509        4             5052392     10.0
5510        6             5052392     10.0
5511        8             5052392     10.0
5512        11            5052392     10.0
5513        14            5052392     10.0
5514        16            5052392     10.0
5515        19            5052392     10.0
5516        20            5052392     10.0
5517        21            5052392     10.0
5518        12            5052392     80.0
5519        18            5052392     90.0
5520        22            5052392     99.0
```