

```
.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 midtermlaverage 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;
```