Assignment 5

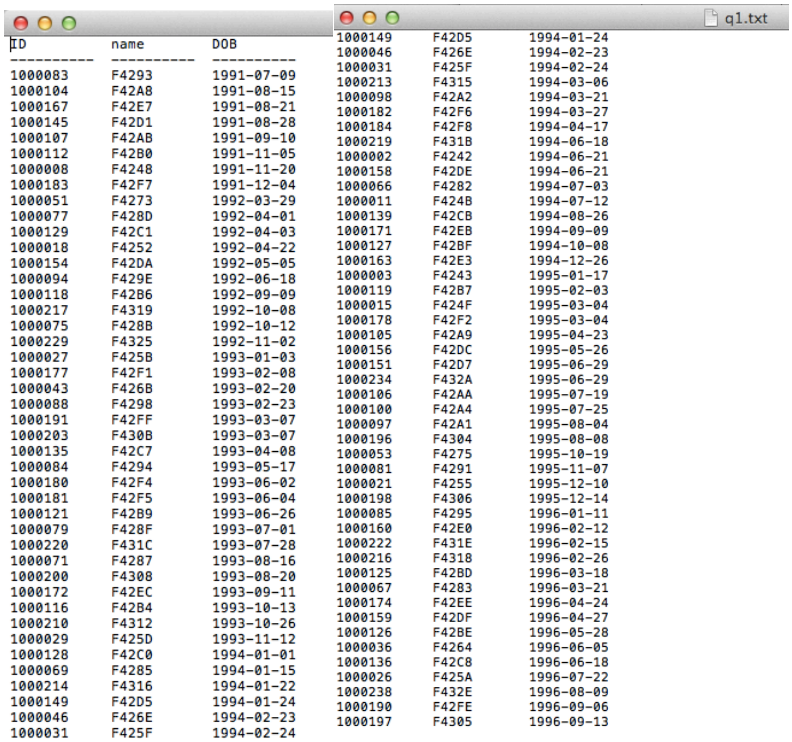
Chen Ren/1665951

Q1: Output a list of students born between June 16, 1991 and September 15, 1996

Code:

```
sqlite> .open Scores.db
sqlite> .headers on
sqlite> .mode column
sqlite> .output hw5.txt
sqlite> SELECT * FROM Students AS s WHERE DOB >='1991-06-06' AND
DOB <='1996-09-15' ORDER BY s.DOB ASC;
```

Result: (q1.txt)



ID	name	DOB
1000083	F4293	1991-07-09
1000104	F42A8	1991-08-15
1000167	F42E7	1991-08-21
1000145	F42D1	1991-08-28
1000107	F42AB	1991-09-10
1000112	F42B0	1991-11-05
1000008	F4248	1991-11-20
1000183	F42F7	1991-12-04
1000051	F4273	1992-03-29
1000077	F428D	1992-04-01
1000129	F42C1	1992-04-03
1000018	F4252	1992-04-22
1000154	F42DA	1992-05-05
1000094	F429E	1992-06-18
1000118	F42B6	1992-09-09
1000217	F4319	1992-10-08
1000075	F428B	1992-10-12
1000229	F4325	1992-11-02
1000027	F425B	1993-01-03
1000177	F42F1	1993-02-08
1000043	F426B	1993-02-20
1000088	F4298	1993-02-23
1000191	F42FF	1993-03-07
1000203	F430B	1993-03-07
1000135	F42C7	1993-04-08
1000084	F4294	1993-05-17
1000180	F42F4	1993-06-02
1000181	F42F5	1993-06-04
1000121	F42B9	1993-06-26
1000079	F428F	1993-07-01
1000220	F431C	1993-07-28
1000071	F4287	1993-08-16
1000200	F4308	1993-08-20
1000172	F42EC	1993-09-11
1000116	F42B4	1993-10-13
1000210	F4312	1993-10-26
1000029	F425D	1993-11-12
1000128	F42C0	1994-01-01
1000069	F4285	1994-01-15
1000214	F4316	1994-01-22
1000149	F42D5	1994-01-24
1000046	F426E	1994-02-23
1000031	F425F	1994-02-24

q1.txt

1000149	F42D5	1994-01-24
1000046	F426E	1994-02-23
1000031	F425F	1994-02-24
1000213	F4315	1994-03-06
1000098	F42A2	1994-03-21
1000182	F42F6	1994-03-27
1000184	F42F8	1994-04-17
1000219	F431B	1994-06-18
1000002	F4242	1994-06-21
1000158	F42DE	1994-06-21
1000066	F4282	1994-07-03
1000011	F424B	1994-07-12
1000139	F42CB	1994-08-26
1000171	F42EB	1994-09-09
1000127	F42BF	1994-10-08
1000163	F42E3	1994-12-26
1000003	F4243	1995-01-17
1000119	F42B7	1995-02-03
1000015	F424F	1995-03-04
1000178	F42F2	1995-03-04
1000105	F42A9	1995-04-23
1000156	F42DC	1995-05-26
1000151	F42D7	1995-06-29
1000234	F432A	1995-06-29
1000106	F42AA	1995-07-19
1000100	F42A4	1995-07-25
1000097	F42A1	1995-08-04
1000196	F4304	1995-08-08
1000053	F4275	1995-10-19
1000081	F4291	1995-11-07
1000021	F4255	1995-12-10
1000198	F4306	1995-12-14
1000085	F4295	1996-01-11
1000160	F42E0	1996-02-12
1000222	F431E	1996-02-15
1000216	F4318	1996-02-26
1000125	F42BD	1996-03-18
1000067	F4283	1996-03-21
1000174	F42EE	1996-04-24
1000159	F42DF	1996-04-27
1000126	F42BE	1996-05-28
1000036	F4264	1996-06-05
1000136	F42C8	1996-06-18
1000026	F425A	1996-07-22
1000238	F432E	1996-08-09
1000190	F42FE	1996-09-06
1000197	F4305	1996-09-13

Q2: Output the number of students born between June 16, 1991 and September 15, 1996

Code:

```
sqlite> .output stout
sqlite> .output stdout
sqlite> .open Scores.db
```

```
sqlite> .output q2.txt
sqlite> .headers on
sqlite> .mode column
sqlite> SELECT COUNT(name) AS NumberOfStudent FROM Students WHERE
DOB >='1991-06-16' AND DOB <='1996-09-15';
```

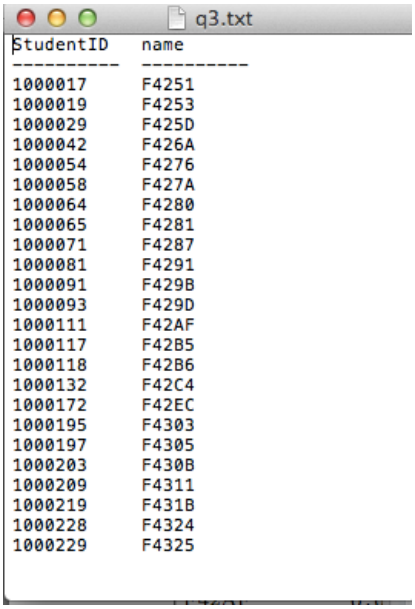
Result(q2.txt):

Q3: Output a list of students who have missed one or more labs (Score <= 0.1 to avoid numeric truncation errors)

Code:

```
.open Scores.db
sqlite> .output q3.txt
sqlite> .headers on
sqlite> .mode column
sqlite> SELECT DISTINCT s.StudentID, Stu.name FROM Students as Stu,
Scores as s, Types as t, Assignments as a WHERE s.StudentID=stu.ID
AND s.AssignmentID=a.ID AND a.typeID=t.typeID AND s.score<=0.1 AND
t.typeID=2;
```

Result(q3.txt):

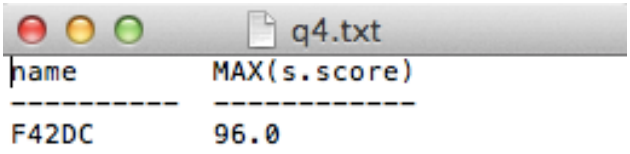


Q4: Output the name of the student with the best score at the final

Code:

```
sqlite> .open Scores.db
sqlite> .output q4.txt
sqlite> .headers on
sqlite> .mode column
sqlite> SELECT stu.name, MAX(s.score) FROM Students as stu, Scores
as s, Types as t, Assignments as a WHERE s.AssignmentID=a.ID AND
s.StudentID=stu.ID AND a.typeID=t.typeID AND t.typeID=4;
```

Result(q4.txt):



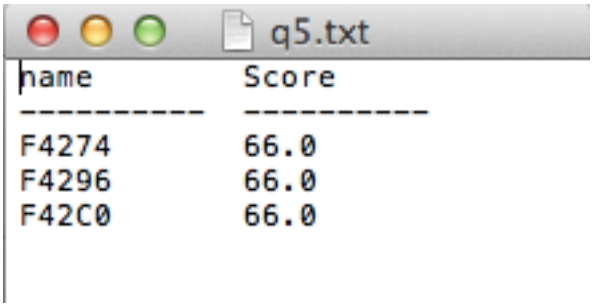
Q5: Output the name of the student closest to the average score of midterm 1

Code:

```
sqlite> .open Scores.db
sqlite> .output q5.txt
sqlite> .headers on
sqlite> .mode column
sqlite> DROP VIEW IF EXISTS AVG_Midterm;
sqlite> CREATE VIEW AVG_Midterm AS
   ...> SELECT avg(s.score) as avgscore FROM Assignments as a,
Scores as s
   ...> WHERE s.AssignmentID=a.ID AND a.name Like "%Midterm 1%";
sqlite> CREATE VIEW min_diff AS
   ...> SELECT min(abs(s.score - avg.avgscore)) AS mindiff FROM
AVG_Midterm as avg, Assignments as a, Scores as s
   ...> WHERE s.AssignmentID=a.ID AND a.name Like "%Midterm 1%";
sqlite> SELECT stu.name, s.score
   ...> FROM min_diff as min, AVG_Midterm as avg, Students as stu,
Assignments as a, Scores as s
   ...> WHERE abs(abs(s.score-avg.avgscore)-min.mindiff)<=0.0001
```

```
 AND s.studentID=stu.ID AND s.AssignmentID=a.ID AND a.name Like
"%Midterm 1%";
```

Result(q5.txt):

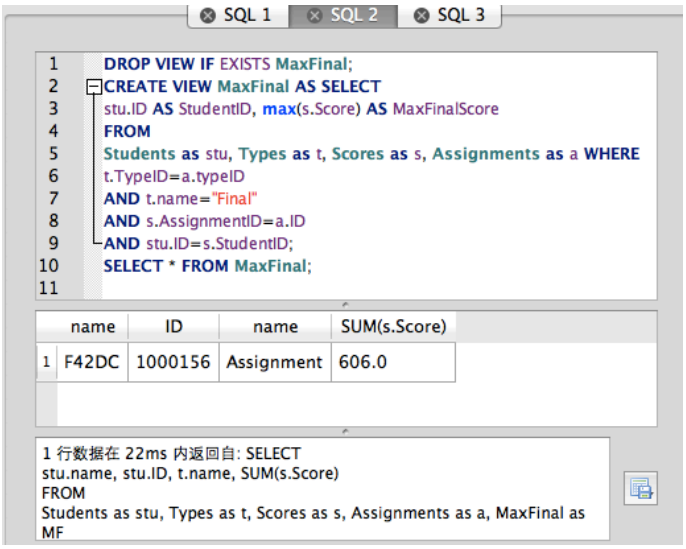


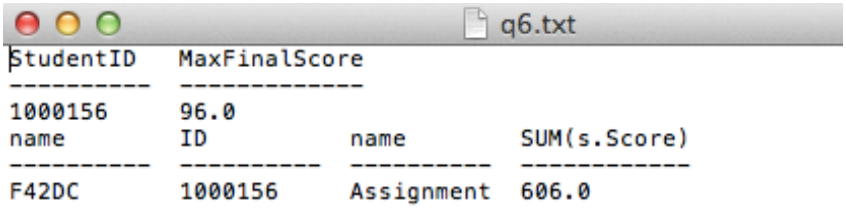
Q6: Output the accumulated homework score (sum of all assignment-type score) for the students identified in 4. and 5., respectively.

Code:

a)

```
sqlite> .open Scores.db
sqlite> .output q6.txt
sqlite> .headers on
sqlite> .mode column
sqlite> DROP VIEW IF EXISTS MaxFinal;
sqlite> CREATE VIEW MaxFinal AS SELECT
   ...> stu.ID AS StudentID, max(s.Score) AS MaxFinalScore
   ...> FROM Students as stu, Types as t, Scores as s, Assignments
as a
   ...> WHERE t.TypeID=a.typeID AND t.name="Final" AND
s.AssignmentID=a.ID AND stu.ID=s.StudentID;
sqlite> SELECT * FROM MaxFinal;
sqlite> SELECT stu.name, stu.ID, t.name, SUM(s.Score) FROM
Students as stu, Types as t, Scores as s, Assignments as a, MaxFinal
as MF
   ...> WHERE stu.ID = MF.StudentID AND s.StudentID = MF.StudentID
AND t.typeID = a.typeID AND t.typeID = 1 AND s.AssignmentID = a.ID;
```

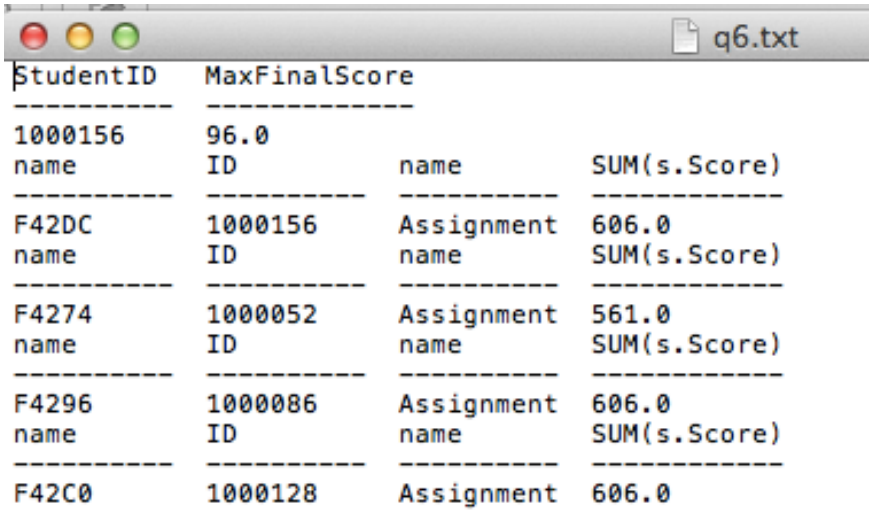




b)

```
sqlite> SELECT stu.name, stu.ID, t.name, SUM(s.Score)
   ...> FROM Students as stu, Types as t, Scores as s, Assignments
as a WHERE stu.name = "F4274"
   ...> AND stu.ID = s.StudentID AND t.typeID = a.typeID AND t.typeID
= 1 AND s.AssignmentID = a.ID;
sqlite> SELECT stu.name, stu.ID, t.name, SUM(s.Score)
   ...> FROM Students as stu, Types as t, Scores as s, Assignments
as a
   ...> WHERE stu.name = "F4296" AND stu.ID = s.StudentID
AND t.typeID = a.typeID AND t.typeID = 1 AND s.AssignmentID = a.ID;
sqlite> SELECT stu.name, stu.ID, t.name, SUM(s.Score)
   ...> FROM Students as stu, Types as t, Scores as s, Assignments
as a WHERE stu.name = "F42C0"
   ...> AND stu.ID = s.StudentID AND t.typeID = a.typeID AND t.typeID
= 1 AND s.AssignmentID = a.ID;
```

Result(q6.txt):



Q7: Create a VIEW named altAssignments, listing Assignment.ID, Assignment.name, Type.name, and sorted by Type.name.

Code:

```
sqlite> .open Scores.db
sqlite> .output q7.txt
sqlite> .headers on
sqlite> .mode column
sqlite> DROP VIEW IF EXISTS altAssignments;
sqlite> CREATE VIEW altAssignments AS
   ...> SELECT a.ID as ID, a.name, t.name AS typeName FROM Types
as t, Assignments as a WHERE a.typeID=t.typeID ORDER BY t.name;
sqlite> SELECT a.ID as ID, a.name, t.name AS typeName FROM Types
as t, Assignments as a WHERE a.typeID=t.typeID ORDER BY ID;
sqlite> SELECT * FROM altAssignments;
sqlite> .schema altAssignments
sqlite> CREATE VIEW altAssignments AS
   ...> SELECT a.ID, a.name, t.name FROM Assignments as a, Types
as t WHERE a.typeID=t.typeID;
Error: table altAssignments already exists
```

View is not a table. But we can use View altAssignment as a table when we make SELECT statement.

Result(q7.txt):

q7.txt

```
ID          name                            typeName
----------  ------------------------------  ----------
1           Homework Assignment #1 (2786783)  Assignment
2           Lab #1 (2829219)                Labs
3           Homework Assignment #2 (2786789)  Assignment
4           Lab #2 (2786809)                Labs
5           Homework Assignment #3 (2786785)  Assignment
6           Lab #3 (2786810)                Labs
7           Homework Assignment #4 (2786784)  Assignment
8           Lab #4 (2786811)                Labs
9           Homework Assignment #5 - Quick A  Assignment
10          Homework Assignment #5 - Problem  Assignment
11          Lab #5 (2856765)                Labs
12          Midterm 1 (2786796)             Midterm
13          Homework Assignment #6 (2786791)  Assignment
14          Lab #6 (2786812)                Labs
15          Homework Assignment #7 (2786790)  Assignment
16          Lab #7 (2786813)                Labs
17          Homework Assignment #8 (2786787)  Assignment
18          Midterm 2 (2786797)             Midterm
19          Lab #8 (2870743)                Labs
20          Lab #9 - Beam Lab (2786814)     Labs
21          Lab #10 (2786815)               Labs
22          Final Exam (2786798)            Final
23          Bonus Assignment #9 (2786795)   Assignment
ID          name                            typeName
----------  ------------------------------  ----------
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
CREATE VIEW altAssignments AS
SELECT a.ID as ID, a.name, t.name AS typeName FROM Types as t,
Assignments as a WHERE a.typeID=t.typeID ORDER BY t.name;
```

Q8: 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)

Code:

#1000240

```
sqlite> .open Scores.db
sqlite> .output q8.txt
sqlite> .headers on
sqlite> .mode column
```

```
sqlite> DELETE FROM Students WHERE ID = 1000240;
sqlite> DELETE FROM Scores WHERE StudentID = 1000240;
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),
   ...> (SELECT a.ID FROM Assignments as a WHERE a.name LIKE
"%Assignment #1%"),1000240,(SELECT a.targetScore FROM Assignments
as a WHERE a.name LIKE "%Assignment #1%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #2%"),1000240,
(SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #2%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #3%"),1000240,
(SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #3%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #4%"),1000240,
(SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #4%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #5 - Quick %"),
1000240,(SELECT a.targetScore FROM Assignments as a WHERE a.name
LIKE "%Assignment #5 - Quick %"));
```

```
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #5 - Problem%"),
1000240,(SELECT a.targetScore FROM Assignments as a WHERE a.name
LIKE "%Assignment #5 - Problem%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #6%"),1000240,
(SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #6%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #7%"),1000240,
(SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #7%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores), (SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Assignment #8%"),1000240,
(SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #8%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),(SELECT a.ID FROM
Assignments as a WHERE a.name LIKE "%Midterm 1%"),1000240,(SELECT
(0.8)*(a.targetScore) FROM Assignments as a WHERE a.name LIKE
"%Midterm 1%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),
```

```
   ...> (SELECT a.ID FROM Assignments as a WHERE a.name LIKE
"%Midterm 2%"),1000240,(SELECT (0.9)*(a.targetScore) FROM
Assignments as a WHERE a.name LIKE "%Midterm 2%"));
sqlite> INSERT INTO Scores( itemID, AssignmentID, StudentID, Score)
VALUES(
   ...> (SELECT MAX(itemID)+1 FROM Scores),
   ...> (SELECT a.ID FROM Assignments as a WHERE a.name LIKE
"%Final%"),1000240,(SELECT (0.99)*(a.targetScore) FROM
Assignments as a WHERE a.name LIKE "%Final%"));
sqlite> SELECT * FROM Students WHERE ID="1000240";
sqlite> SELECT * FROM Scores WHERE studentID="1000240";
sqlite> DELETE FROM Students WHERE ID = 1000240;
sqlite> DELETE FROM Scores WHERE StudentID = 1000240;
```

Result(q8.txt):

q8.txt

```
ID          name        DOB
----------  ----------  ----------
1000240     Chen Ren    1993-11-09
itemID      AssignmentID  StudentID   Score
----------  ------------  ----------  ----------
5498        1             1000240     60.0
5499        3             1000240     60.0
5500        5             1000240     70.0
5501        7             1000240     80.0
5502        7             1000240     80.0
5503        9             1000240     21.0
5504        10            1000240     50.0
5505        13            1000240     70.0
5506        15            1000240     60.0
5507        17            1000240     60.0
5508        12            1000240     80.0
5509        18            1000240     90.0
5510        22            1000240     99.0
5511        1             1000240     60.0
5512        3             1000240     60.0
5513        5             1000240     70.0
5514        7             1000240     80.0
5515        9             1000240     21.0
5516        10            1000240     50.0
5517        13            1000240     70.0
5518        15            1000240     60.0
5519        17            1000240     60.0
5520        12            1000240     80.0
5521        18            1000240     90.0
5522        22            1000240     99.0
```