1. GENERAL COMMANDS:

```
.open scores.db
.output results.txt
.headers on
.mode column
```

2. COMMANDS FOR QUESTIONS 1-8:

```
-- 1.) Output a list of students born between June 16, 1991 and Sep. 15, 1996

SELECT s.ID, s.name FROM Students as s
   WHERE s.dob >= "1991-06-16"
   AND s.dob <= "1996-09-15";
.print \t

--2.)Output the number of students born between June 16, 1991 and September 15, 1996

SELECT count(s.ID) FROM Students as s
   WHERE s.dob > "1991-06-16"
   AND s.dob < "1996-09-15";
.print \t

--3.)Output a list of students who have missed one or more labs (Score <= 0.1 to avoid
numeric truncation errors)

SELECT DISTINCT s.name FROM Students as s,Types as t, Assignments as a, Scores as sc
   WHERE sc.score<=0.1
   AND sc.studentID=s.ID
   AND sc.AssignmentID=a.ID
   AND a.typeID=t.typeID
   AND t.name="Labs";
.print \t

--4.)Output the name of the student with the best score at the final

SELECT s.name FROM (
      SELECT MAX(sc.score) AS maxsc FROM Types as t, Assignments as a, Scores as sc
           WHERE sc.AssignmentID=a.ID
           AND a.typeID=t.typeID
           AND t.name="Final"
         ) as ma,
Students as s,Types as t, Assignments as a, Scores as sc WHERE sc.score=ma.maxsc
AND sc.studentID=s.ID
AND sc.AssignmentID=a.ID
```

```
AND a.typeID=t.typeID
AND t.name="Final";
.print \t

--5.)Output the name of the student closest to the average score of midterm 1

--create a view for the avaerage score

DROP VIEW IF EXISTS average;

CREATE VIEW average AS
SELECT avg(sc.score) as avgscore
   FROM Assignments as a, Scores as sc
   WHERE sc.AssignmentID=a.ID AND a.name Like "%Midterm 1%";

--creat a view for the minumum difference between score and the average

DROP VIEW IF EXISTS minimum;

CREATE VIEW minimum AS
SELECT min(abs(sc.score - av.avgscore)) AS mindiff
FROM average as av, Assignments as a, Scores as sc
   WHERE sc.AssignmentID=a.ID
   AND a.name Like "%Midterm 1%";

SELECT s.name, sc.score
     FROM minimum as mn, average as av, Students as s, Assignments as a, Scores as sc
     WHERE abs(abs(sc.score-av.avgscore)-mn.mindiff)<=0.0001
        AND sc.studentID=s.ID
        AND sc.AssignmentID=a.ID
        AND a.name Like "%Midterm 1%";
.print \n

--6.)Output the accumulated homework score (sum of all assignment-type score) for the
students identified in 4. and 5., respectively.

--6.1) for 4.

DROP VIEW IF EXISTS fourthans;

CREATE VIEW fourthans AS

SELECT s.name as fname FROM (
      SELECT MAX(sc.score) AS maxsc FROM Types as t, Assignments as a, Scores as sc
```

```
            WHERE sc.AssignmentID=a.ID
            AND a.typeID=t.typeID
            AND t.name="Final"
          ) as ma,
Students as s,Types as t, Assignments as a, Scores as sc WHERE sc.score=ma.maxsc
AND sc.studentID=s.ID
AND sc.AssignmentID=a.ID
AND a.typeID=t.typeID
AND t.name="Final";


SELECT s.name, SUM(sc.score)
      FROM Students as s,Types as t, Assignments as a, Scores as sc, fourthans as four
      WHERE sc.studentID=s.ID
         AND sc.AssignmentID=a.ID
         AND a.typeID=t.typeID
         AND t.name="Assignment"
         AND s.name=four.fname;

.print \n

--6.2) for 5.

SELECT s.name, SUM(sc.score)
      FROM Students as s,Types as t, Assignments as a, Scores as sc
      WHERE sc.studentID=s.ID
         AND sc.AssignmentID=a.ID
         AND a.typeID=t.typeID
         AND t.name="Assignment"
         AND s.name="F4274"
         ;

SELECT s.name, SUM(sc.score)
      FROM Students as s,Types as t, Assignments as a, Scores as sc
      WHERE sc.studentID=s.ID
         AND sc.AssignmentID=a.ID
         AND a.typeID=t.typeID
         AND t.name="Assignment"
         AND s.name="F4296";

SELECT s.name, SUM(sc.score)
      FROM Students as s,Types as t, Assignments as a, Scores as sc
      WHERE sc.studentID=s.ID
         AND sc.AssignmentID=a.ID
```

```
        AND a.typeID=t.typeID
        AND t.name="Assignment"
        AND s.name="F42C0";
.print \t
```

--7.)Create a VIEW named altAssignments, listing Assignment.ID, Assignment.name, Type.name, and sorted by Type.name.

```
DROP VIEW IF EXISTS altAssignments;
CREATE VIEW altAssignments AS SELECT a.ID, a.name, t.name AS typeName FROM
Types as t, Assignments as a WHERE a.typeID=t.typeID ORDER BY t.name;
SELECT * FROM altAssignments;
.print \t
.schema altAssignments
.print \t
```

--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)

--insert into the table students my student information

```
INSERT INTO Students(ID, Name, DOB)
   VALUES(1568037, "Changming Feng", "1990-10-07");
```

--insert into table scores scores

```
 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%"),
       1568037,
       (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #1%"));


 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%"),
       1568037,
```

```
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #2%"));


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%"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #3%"));


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%"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #4%"));


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 %"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #5 - Quick %"));


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%"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #5 - Problem%"));


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%"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #6%"));


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%"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #7%"));


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%"),
        1568037,
        (SELECT a.targetScore FROM Assignments as a WHERE a.name LIKE
"%Assignment #8%"));

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%"),
        1568037,
        (SELECT (0.8)*(a.targetScore) FROM Assignments as a WHERE a.name LIKE
"%Midterm 1%"));

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%"),
        1568037,
        (SELECT (0.9)*(a.targetScore) FROM Assignments as a WHERE a.name LIKE
"%Midterm 2%"));

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%"),
```

```
    1568037,
    (SELECT (0.99)*(a.targetScore) FROM Assignments as a WHERE a.name LIKE
"%Final%"));


SELECT * FROM Students WHERE ID="1568037";
.print \n

SELECT * FROM Scores WHERE studentID="1568037";
.print \t
```

3. OUTPUTS & EXPLANATION FOR QUESTIONS 1-8:

1).

ID	name
1000002	F4242
1000003	F4243
1000008	F4248
1000011	F424B
1000015	F424F
1000018	F4252
1000021	F4255
1000026	F425A
1000027	F425B
1000029	F425D
1000031	F425F
1000036	F4264
1000043	F426B
1000046	F426E
1000051	F4273
1000053	F4275
1000066	F4282
1000067	F4283
1000069	F4285
1000071	F4287
1000075	F428B
1000077	F428D
1000079	F428F
1000081	F4291
1000083	F4293
1000084	F4294
1000085	F4295
1000088	F4298

1000094	F429E
1000097	F42A1
1000098	F42A2
1000100	F42A4
1000104	F42A8
1000105	F42A9
1000106	F42AA
1000107	F42AB
1000112	F42B0
1000116	F42B4
1000118	F42B6
1000119	F42B7
1000121	F42B9
1000125	F42BD
1000126	F42BE
1000127	F42BF
1000128	F42C0
1000129	F42C1
1000135	F42C7
1000136	F42C8
1000139	F42CB
1000145	F42D1
1000149	F42D5
1000151	F42D7
1000154	F42DA
1000156	F42DC
1000158	F42DE
1000159	F42DF
1000160	F42E0
1000163	F42E3
1000167	F42E7
1000171	F42EB
1000172	F42EC
1000174	F42EE
1000177	F42F1
1000178	F42F2
1000180	F42F4
1000181	F42F5
1000182	F42F6
1000183	F42F7
1000184	F42F8
1000190	F42FE
1000191	F42FF
1000196	F4304

1000197	F4305
1000198	F4306
1000200	F4308
1000203	F430B
1000210	F4312
1000213	F4315
1000214	F4316
1000216	F4318
1000217	F4319
1000219	F431B
1000220	F431C
1000222	F431E
1000229	F4325
1000234	F432A
1000238	F432E

2).

count(s.ID)
87

3).

name
F4251
F4253
F425D
F426A
F4276
F427A
F4280
F4281
F4287
F4291
F429B
F429D
F42AF
F42B5
F42B6
F42C4
F42EC
F4303
F4305

F430B
F4311
F431B
F4324
F4325

4).
name

F42DC

5).
name Score
--- ---
F4274 66.0
F4296 66.0
F42C0 66.0

6).
s.name SUM(sc.score)
--- ---
F42DC 606.0

name SUM(sc.score)
--- ---
F4274 561.0
name SUM(sc.score)
--- ---
F4296 606.0
name SUM(sc.score)
--- ---
F42C0 606.0

7).

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, a.name, t.name AS typeName FROM Types as t, Assignments as a WHERE a.typeID=t.typeID ORDER BY t.name;

Explanation: The output shows the contents in the VIEW altAssignment as a table and ".schema altAssignment" shows the CREATE statement of the view altAssignments and I can just use this view as a table in "SELECT" and ".schema" command.

8).

ID	name	DOB
1568037	Changming Feng	1990-10-07

itemID	AssignmentID	StudentID	Score
5498	1	1568037	60.0
5499	3	1568037	60.0
5500	5	1568037	70.0
5501	7	1568037	80.0
5502	9	1568037	21.0
5503	10	1568037	50.0
5504	13	1568037	70.0
5505	15	1568037	60.0
5506	17	1568037	60.0
5507	12	1568037	80.0
5508	18	1568037	90.0

5509	22	1568037	99.0