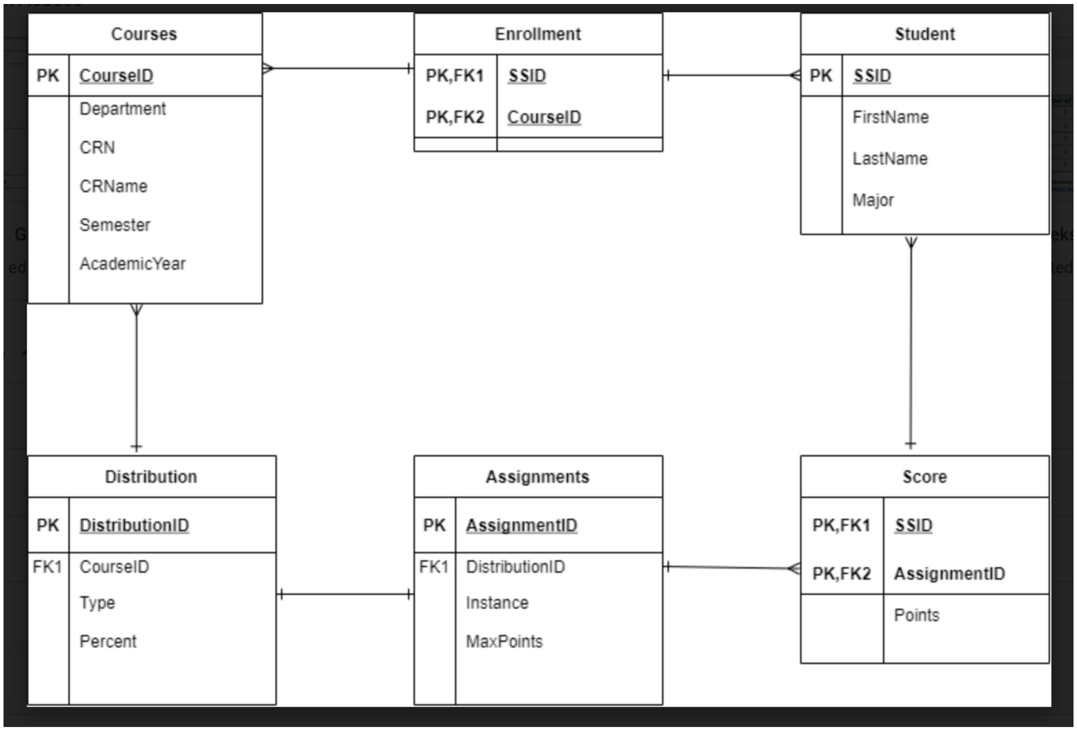
Alexis Garrett
Database Systems
4/13/2021

Gradebook Report:

ERD Diagram :

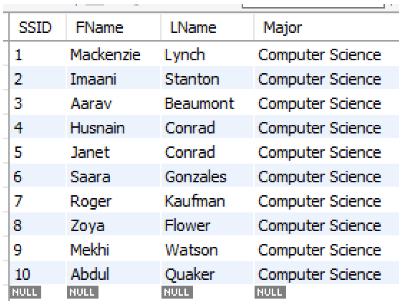


#3. Show the tables with the contents that you have inserted;

```
DROP TABLE IF EXISTS `COURSES`;
CREATE TABLE `COURSES` (
   `CourseID` int(11) NOT NULL AUTO_INCREMENT,
   `CourseName` varchar(255) DEFAULT NULL,
   `Department` varchar(255) DEFAULT NULL,
   `CRN` varchar(255) DEFAULT NULL,
   `Semester` varchar(255) DEFAULT NULL,
   `AcademicYear` int(4) DEFAULT NULL,
   PRIMARY KEY (`CourseID`)
);
```

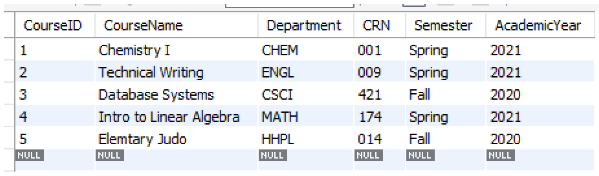
```
/* inserting COURSE info */
INSERT INTO `COURSES` (CourseID, CourseName, Department, CRN, Semester, AcademicYear)
VALUES (CourseID, 'Chemistry I', 'CHEM', '001', 'Spring', 2021),
(CourseID, 'Technical Writing', 'ENGL', '009', 'Spring', 2021),
(CourseID, 'Database Systems', 'CSCI', '421', 'Fall', 2020),
(CourseID, 'Intro to Linear Algebra', 'MATH', '174', 'Spring', 2021),
(CourseID, 'Elemtary Judo', 'HHPL', '014', 'Fall', 2020);
```

STUDENTS



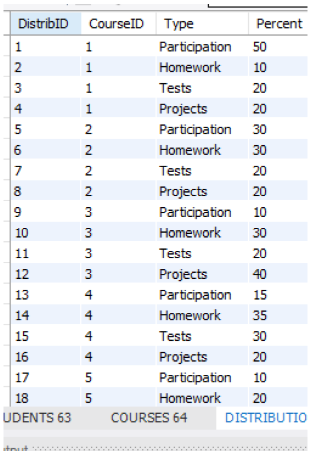
SSID	FName	LName	Major
1	Mackenzie	Lynch	Computer Science
2	Imaani	Stanton	Computer Science
3	Aarav	Beaumont	Computer Science
4	Husnain	Conrad	Computer Science
5	Janet	Conrad	Computer Science
6	Saara	Gonzales	Computer Science
7	Roger	Kaufman	Computer Science
8	Zoya	Flower	Computer Science
9	Mekhi	Watson	Computer Science
10	Abdul	Quaker	Computer Science
NULL	NULL	NULL	NULL

COURSES



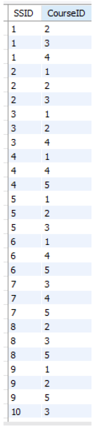
CourseID	CourseName	Department	CRN	Semester	AcademicYear
1	Chemistry I	CHEM	001	Spring	2021
2	Technical Writing	ENGL	009	Spring	2021
3	Database Systems	CSCI	421	Fall	2020
4	Intro to Linear Algebra	MATH	174	Spring	2021
5	Elemtary Judo	HHPL	014	Fall	2020
NULL	NULL	NULL	NULL	NULL	NULL

DISTRIBUTION



DistribID	CourseID	Type	Percent
1	1	Participation	50
2	1	Homework	10
3	1	Tests	20
4	1	Projects	20
5	2	Participation	30
6	2	Homework	30
7	2	Tests	20
8	2	Projects	20
9	3	Participation	10
10	3	Homework	30
11	3	Tests	20
12	3	Projects	40
13	4	Participation	15
14	4	Homework	35
15	4	Tests	30
16	4	Projects	20
17	5	Participation	10
18	5	Homework	20

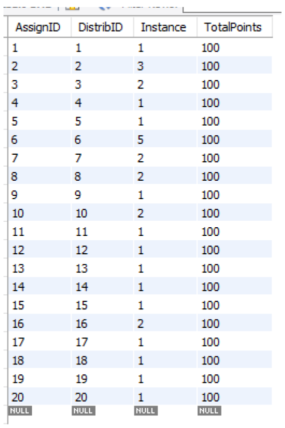
ENROLLMENT



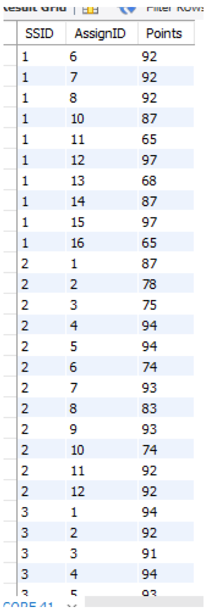
SSID	CourseID
1	2
1	3
1	4
2	1
2	2
2	3
3	1
3	2
3	4
4	1
4	4
4	5
5	1
5	2
5	3
6	1
6	4
6	5
7	3
7	4
7	5
8	2
8	3
8	5
9	1
9	2
9	5
10	3

ASSIGNMENT

SCORES



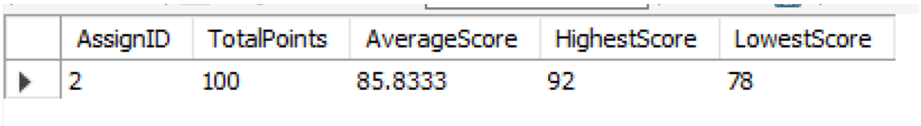
AssignID	DistribID	Instance	TotalPoints
1	1	1	100
2	2	3	100
3	3	2	100
4	4	1	100
5	5	1	100
6	6	5	100
7	7	2	100
8	8	2	100
9	9	1	100
10	10	2	100
11	11	1	100
12	12	1	100
13	13	1	100
14	14	1	100
15	15	1	100
16	16	2	100
17	17	1	100
18	18	1	100
19	19	1	100
20	20	1	100
NULL	NULL	NULL	NULL



SSID	AssignID	Points
1	6	92
1	7	92
1	8	92
1	10	87
1	11	65
1	12	97
1	13	68
1	14	87
1	15	97
1	16	65
2	1	87
2	2	78
2	3	75
2	4	94
2	5	94
2	6	74
2	7	93
2	8	83
2	9	93
2	10	74
2	11	92
2	12	92
3	1	94
3	2	92
3	3	91
3	4	94
3	5	93

#4. Compute the average/highest/lowest score of an assignment

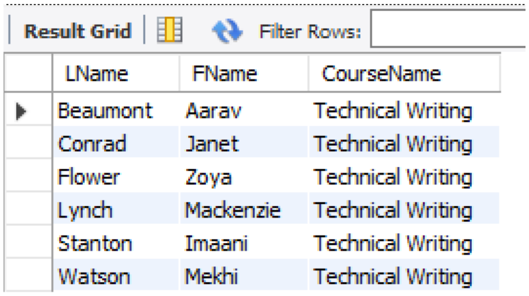
```
select a.AssignID, a.TotalPoints, avg(s.POINTS) AS AverageScore, max(s.POINTS) AS HighestScore,
min(s.POINTS) AS LowestScore
from ASSIGNMENTS a, SCORE s
where a.AssignID=2 AND s.AssignID=a.AssignID;
```



AssignID	TotalPoints	AverageScore	HighestScore	LowestScore
2	100	85.8333	92	78

#5. List all of the students in a given course

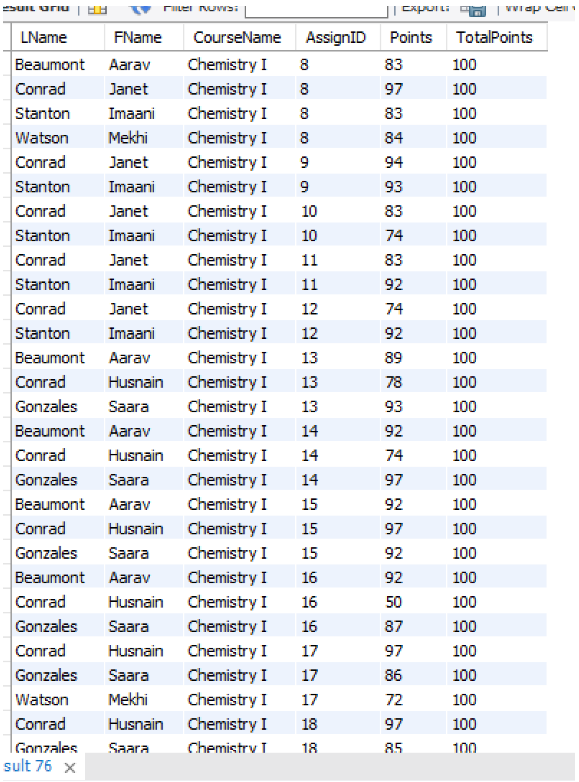
```
select LName, FName, CourseName from `ENROLLMENT`
inner join `STUDENTS` on (STUDENTS.SSID = ENROLLMENT.SSID)
inner join `COURSES` on (COURSES.CourseID = ENROLLMENT.CourseID) where (COURSES.CourseID = 2)
order by LName asc;
```



LName	FName	CourseName
Beaumont	Aarav	Technical Writing
Conrad	Janet	Technical Writing
Flower	Zoya	Technical Writing
Lynch	Mackenzie	Technical Writing
Stanton	Imaani	Technical Writing
Watson	Mekhi	Technical Writing

#6. List all of the students in a course and all of their scores on every assignment

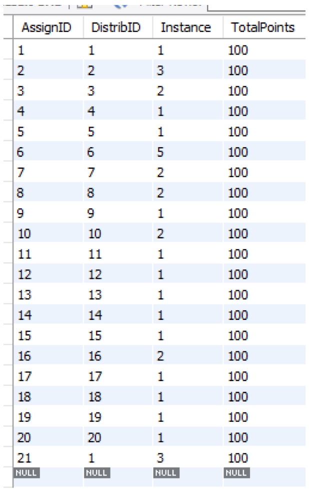
```
select LName, FName, CourseName, ASSIGNMENTS.AssignID, Points, TotalPoints from `ENROLLMENT`
inner join `STUDENTS` on (STUDENTS.SSID = ENROLLMENT.SSID)
inner join `SCORE` on (SCORE.SSID = ENROLLMENT.SSID)
inner join `ASSIGNMENTS` on (ASSIGNMENTS.AssignID = SCORE.AssignID)
inner join `COURSES` on (COURSES.CourseID = ENROLLMENT.CourseID) where (COURSES.CourseID = 1)
order by AssignID, LName asc;
```



LName	FName	CourseName	AssignID	Points	TotalPoints
Beaumont	Aarav	Chemistry I	8	83	100
Conrad	Janet	Chemistry I	8	97	100
Stanton	Imaani	Chemistry I	8	83	100
Watson	Mekhi	Chemistry I	8	84	100
Conrad	Janet	Chemistry I	9	94	100
Stanton	Imaani	Chemistry I	9	93	100
Conrad	Janet	Chemistry I	10	83	100
Stanton	Imaani	Chemistry I	10	74	100
Conrad	Janet	Chemistry I	11	83	100
Stanton	Imaani	Chemistry I	11	92	100
Conrad	Janet	Chemistry I	12	74	100
Stanton	Imaani	Chemistry I	12	92	100
Beaumont	Aarav	Chemistry I	13	89	100
Conrad	Husnain	Chemistry I	13	78	100
Gonzales	Saara	Chemistry I	13	93	100
Beaumont	Aarav	Chemistry I	14	92	100
Conrad	Husnain	Chemistry I	14	74	100
Gonzales	Saara	Chemistry I	14	97	100
Beaumont	Aarav	Chemistry I	15	92	100
Conrad	Husnain	Chemistry I	15	97	100
Gonzales	Saara	Chemistry I	15	92	100
Beaumont	Aarav	Chemistry I	16	92	100
Conrad	Husnain	Chemistry I	16	50	100
Gonzales	Saara	Chemistry I	16	87	100
Conrad	Husnain	Chemistry I	17	97	100
Gonzales	Saara	Chemistry I	17	86	100
Watson	Mekhi	Chemistry I	17	72	100
Conrad	Husnain	Chemistry I	18	97	100
Gonzales	Saara	Chemistry I	18	85	100

#7. Add an assignment to a course

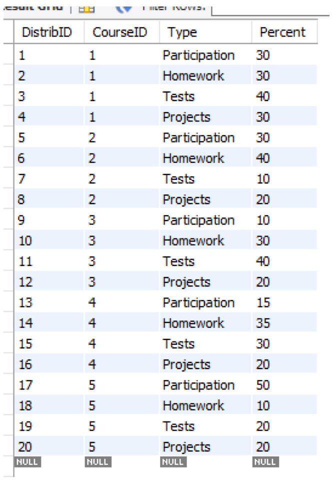
```
insert into ASSIGNMENTS(DistID, Instance, TotalPoints)
VALUES (1, 3, 100);
SELECT * FROM ASSIGNMENTS;
```



AssignID	DistribID	Instance	TotalPoints
1	1	1	100
2	2	3	100
3	3	2	100
4	4	1	100
5	5	1	100
6	6	5	100
7	7	2	100
8	8	2	100
9	9	1	100
10	10	2	100
11	11	1	100
12	12	1	100
13	13	1	100
14	14	1	100
15	15	1	100
16	16	2	100
17	17	1	100
18	18	1	100
19	19	1	100
20	20	1	100
21	1	3	100
NULL	NULL	NULL	NULL

#8. Change the percentages of the categories for a course

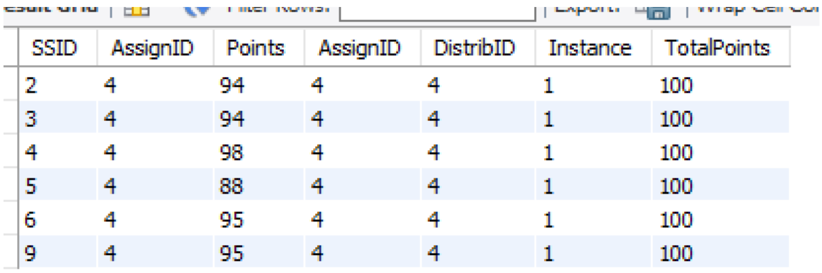
```
update DISTRIBUTION set Percent = 30
where DistID = 1;
update DISTRIBUTION set Percent = 40
where DistID = 3;
-- updated #8 table
SELECT * FROM DISTRIBUTION;
```



DistribID	CourseID	Type	Percent
1	1	Participation	30
2	1	Homework	30
3	1	Tests	40
4	1	Projects	30
5	2	Participation	30
6	2	Homework	40
7	2	Tests	10
8	2	Projects	20
9	3	Participation	10
10	3	Homework	30
11	3	Tests	40
12	3	Projects	20
13	4	Participation	15
14	4	Homework	35
15	4	Tests	30
16	4	Projects	20
17	5	Participation	50
18	5	Homework	10
19	5	Tests	20
20	5	Projects	20
NULL	NULL	NULL	NULL

#9. Add 2 points to the score of each student on an assignment

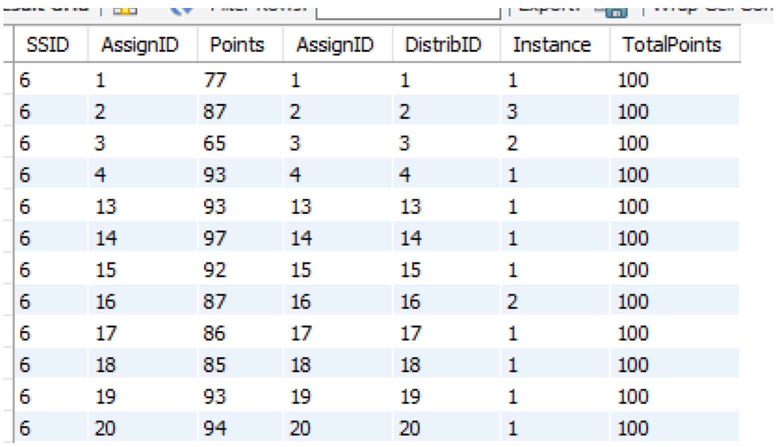
```
UPDATE SCORE SET Points = Points + 2
WHERE AssignID = 4;
SELECT * FROM `SCORE`
LEFT JOIN `ASSIGNMENTS` ON (SCORE.AssignID = ASSIGNMENTS.AssignID) WHERE (SCORE.AssignID = 4)
order by SSID;
```



SSID	AssignID	Points	AssignID	DistribID	Instance	TotalPoints
2	4	94	4	4	1	100
3	4	94	4	4	1	100
4	4	98	4	4	1	100
5	4	88	4	4	1	100
6	4	95	4	4	1	100
9	4	95	4	4	1	100

#10. Add 2 points just to those students whose last name contains a ‘Q’.

```
SELECT * FROM `SCORE`
LEFT JOIN `ASSIGNMENTS` ON (SCORE.AssignID = ASSIGNMENTS.AssignID)
WHERE (SCORE.SSID = 6)
order by SSID;

update `SCORE`
left join `STUDENTS`
on (SCORE.SSID = STUDENTS.SSID)
set Points = Points + 2
where (STUDENTS.LName LIKE '%q%')
or (STUDENTS.LName LIKE '%Q%')
and (STUDENTS.SSID = 6);

SELECT * FROM `SCORE`
LEFT JOIN `ASSIGNMENTS` ON (SCORE.AssignID = ASSIGNMENTS.AssignID)
WHERE (SCORE.SSID = 6)
order by SSID;
```

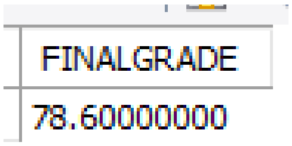


SSID	AssignID	Points	AssignID	DistribID	Instance	TotalPoints
6	1	77	1	1	1	100
6	2	87	2	2	3	100
6	3	65	3	3	2	100
6	4	93	4	4	1	100
6	13	93	13	13	1	100
6	14	97	14	14	1	100
6	15	92	15	15	1	100
6	16	87	16	16	2	100
6	17	86	17	17	1	100
6	18	85	18	18	1	100
6	19	93	19	19	1	100
6	20	94	20	20	1	100

#11. Compute the grade for a student

```
Select SUM(((sg.Points * 100) / a.TotalPoints) * (Percent / 100)) AS FINALGRADE
FROM DISTRIBUTION d, ASSIGNMENTS a, SCORE sg
```

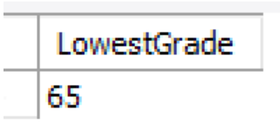
```
WHERE d.DistID = a.DistID
AND sg.AssignID = a.AssignID
AND d.CourseID = 3
AND SSID = 1;

Select SUM(((sg.Points * 100) / a.TotalPoints) * (Percent / 100)) AS FINALGRADE
FROM DISTRIBUTION d, ASSIGNMENTS a, SCORE sg
WHERE d.DistID = a.DistID
AND sg.AssignID = a.AssignID
AND d.CourseID = 3
AND SSID = 1;
```



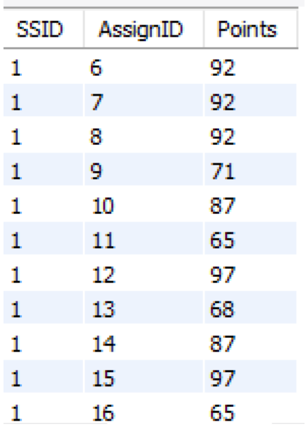
FINALGRADE
78.60000000

#12. Compute the grade for a student, where the lowest score for a given category is dropped.

```
SELECT min(s.Points) as LowestGrade FROM SCORE s
left join `ASSIGNMENTS` a on (s.AssignID = a.AssignID)
left join `DISTRIBUTION` d on (d.DistID = a.DistID)
left join `COURSES` c on (c.CourseID = d.courseID)
WHERE  s.SSID = 1
and d.CourseID = 3
order by s.SSID asc;
SELECT * FROM SCORE where SSID = 1;

Delete from `SCORE`
where SSID = 1 and AssignID = 9;

Select SUM(((sg.Points * 100) / a.TotalPoints) * (Percent / 100)) AS FINALGRADE
FROM DISTRIBUTION d, ASSIGNMENTS a, SCORE sg
WHERE d.DistID = a.DistID
AND sg.AssignID = a.AssignID
AND d.CourseID = 3
AND SSID = 1;

SELECT s.AssignID, s.SSID,d.CourseID, s.Points  FROM SCORE s
left join `ASSIGNMENTS` a on (s.AssignID = a.AssignID)
left join `DISTRIBUTION` d on (d.DistID = a.DistID)
left join `COURSES` c on (c.CourseID = d.courseID)
WHERE  s.SSID = 1
and d.CourseID = 3
order by s.SSID asc;
```



LowestGrade
65



SSID	AssignID	Points
1	6	92
1	7	92
1	8	92
1	9	71
1	10	87
1	11	65
1	12	97
1	13	68
1	14	87
1	15	97
1	16	65