COMPSCI 751 S1 C – Assignment 2

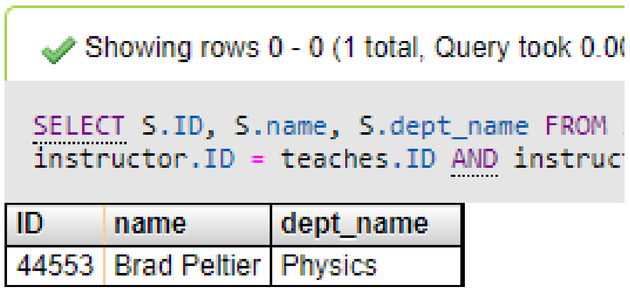
Name: Yixuan Li

UPI: yil845

Q1. The UNIVERSITY database

1)

SELECT S.ID, S.name, S.dept_name FROM student AS S WHERE S.ID IN (SELECT takes.ID FROM takes JOIN teaches ON takes.course_id=teaches.course_id AND takes.sec_id=teaches.sec_id AND takes.semester=teaches.semester AND takes.year = teaches.year JOIN instructor WHERE instructor.ID = teaches.ID AND instructor.name ="Einstein") AND S.ID in (SELECT takes.ID FROM takes JOIN advisor on takes.ID = advisor.s_ID JOIN instructor ON advisor.i_ID=instructor.ID WHERE instructor.name ="Einstein");

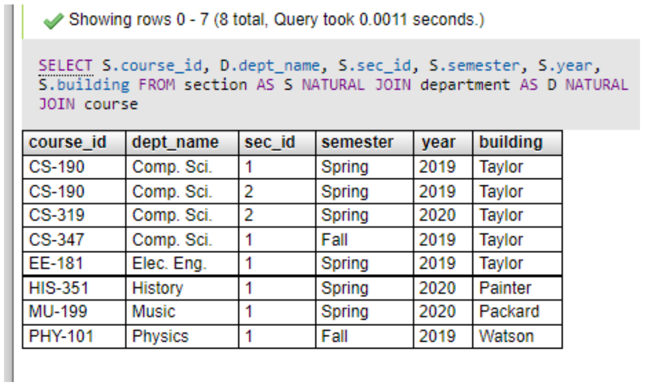


```
Showing rows 0 - 0 (1 total, Query took 0.0
SELECT S.ID, S.name, S.dept_name FROM
instructor.ID = teaches.ID AND instruc
```

ID	name	dept_name
44553	Brad Peltier	Physics

2)

SELECT S.course_id, C.dept_name, S.sec_id, S.semester, S.year, S.building FROM section S INNER JOIN course C ON S.course_id = C.course_id LEFT JOIN department D ON C.dept_name = D.dept_name WHERE S.building = D.building;

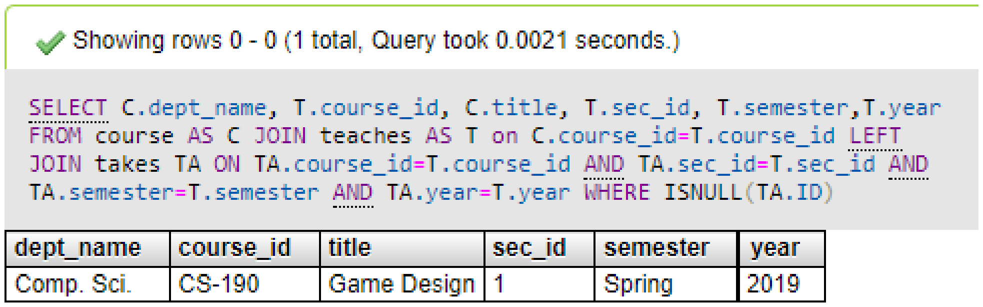


```
Showing rows 0 - 7 (8 total, Query took 0.0011 seconds.)
SELECT S.course_id, D.dept_name, S.sec_id, S.semester, S.year,
S.building FROM section AS S NATURAL JOIN department AS D NATURAL
JOIN course
```

course_id	dept_name	sec_id	semester	year	building
CS-190	Comp. Sci.	1	Spring	2019	Taylor
CS-190	Comp. Sci.	2	Spring	2019	Taylor
CS-319	Comp. Sci.	2	Spring	2020	Taylor
CS-347	Comp. Sci.	1	Fall	2019	Taylor
EE-181	Elec. Eng.	1	Spring	2019	Taylor
HIS-351	History	1	Spring	2020	Painter
MU-199	Music	1	Spring	2020	Packard
PHY-101	Physics	1	Fall	2019	Watson

3)

SELECT C.dept_name, T.course_id, C.title, T.sec_id, T.semester,T.year FROM course AS C JOIN teaches AS T on C.course_id=T.course_id LEFT JOIN takes TA ON TA.course_id=T.course_id AND TA.sec_id=T.sec_id AND TA.semester=T.semester AND TA.year=T.year WHERE ISNULL(TA.ID)



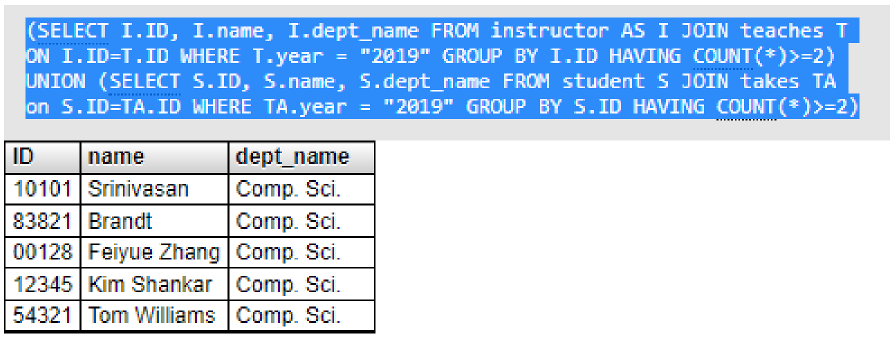
Showing rows 0 - 0 (1 total, Query took 0.0021 seconds.)

```
SELECT C.dept_name, T.course_id, C.title, T.sec_id, T.semester,T.year
FROM course AS C JOIN teaches AS T on C.course_id=T.course_id LEFT
JOIN takes TA ON TA.course_id=T.course_id AND TA.sec_id=T.sec_id AND
TA.semester=T.semester AND TA.year=T.year WHERE ISNULL(TA.ID)
```

dept_name	course_id	title	sec_id	semester	year
Comp. Sci.	CS-190	Game Design	1	Spring	2019

4)

SELECT I.ID, I.name, I.dept_name FROM instructor AS I JOIN teaches T ON I.ID=T.ID WHERE T.year = "2019" GROUP BY I.ID HAVING COUNT(*)>=2) UNION (SELECT S.ID, S.name, S.dept_name FROM student S JOIN takes TA on S.ID=TA.ID WHERE TA.year = "2019" GROUP BY S.ID HAVING COUNT(*)>=2

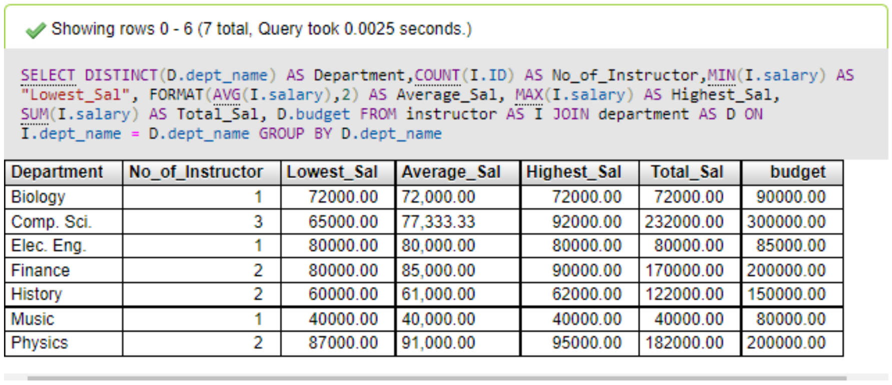


```
(SELECT I.ID, I.name, I.dept_name FROM instructor AS I JOIN teaches T
ON I.ID=T.ID WHERE T.year = "2019" GROUP BY I.ID HAVING COUNT(*)>=2)
UNION (SELECT S.ID, S.name, S.dept_name FROM student S JOIN takes TA
on S.ID=TA.ID WHERE TA.year = "2019" GROUP BY S.ID HAVING COUNT(*)>=2)
```

ID	name	dept_name
10101	Srinivasan	Comp. Sci.
83821	Brandt	Comp. Sci.
00128	Feiyue Zhang	Comp. Sci.
12345	Kim Shankar	Comp. Sci.
54321	Tom Williams	Comp. Sci.

5)

SELECT DISTINCT(D.dept_name) AS Department,COUNT(I.ID) AS No_of_Instructor,MIN(I.salary) AS "Lowest_Sal", FORMAT(AVG(I.salary),2) AS Average_Sal, MAX(I.salary) AS Highest_Sal, SUM(I.salary) AS Total_Sal, D.budget FROM instructor AS I JOIN department AS D ON I.dept_name = D.dept_name GROUP BY D.dept_name;



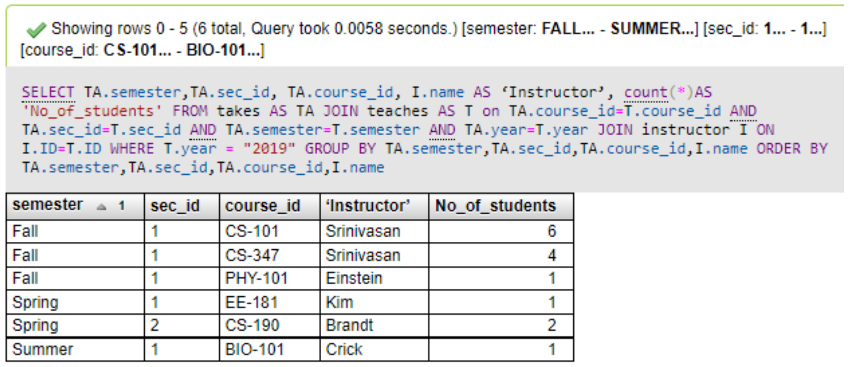
Showing rows 0 - 6 (7 total, Query took 0.0025 seconds.)

```
SELECT DISTINCT(D.dept_name) AS Department,COUNT(I.ID) AS No_of_Instructor,MIN(I.salary) AS
"Lowest_Sal", FORMAT(AVG(I.salary),2) AS Average_Sal, MAX(I.salary) AS Highest_Sal,
SUM(I.salary) AS Total_Sal, D.budget FROM instructor AS I JOIN department AS D ON
I.dept_name = D.dept_name GROUP BY D.dept_name
```

Department	No_of_Instructor	Lowest_Sal	Average_Sal	Highest_Sal	Total_Sal	budget
Biology	1	72000.00	72,000.00	72000.00	72000.00	90000.00
Comp. Sci.	3	65000.00	77,333.33	92000.00	232000.00	300000.00
Elec. Eng.	1	80000.00	80,000.00	80000.00	80000.00	85000.00
Finance	2	80000.00	85,000.00	90000.00	170000.00	200000.00
History	2	60000.00	61,000.00	62000.00	122000.00	150000.00
Music	1	40000.00	40,000.00	40000.00	40000.00	80000.00
Physics	2	87000.00	91,000.00	95000.00	182000.00	200000.00

6)

SELECT TA.semester,TA.sec_id, TA.course_id, I.name AS ‘Instructor’, count(*)AS 'No_of_students' FROM takes AS TA JOIN teaches AS T on TA.course_id=T.course_id AND TA.sec_id=T.sec_id AND TA.semester=T.semester AND TA.year=T.year JOIN instructor I ON I.ID=T.ID WHERE T.year = "2019" GROUP BY TA.semester,TA.sec_id,TA.course_id,I.name ORDER BY TA.semester, TA.sec_id, TA.course_id, I.name;

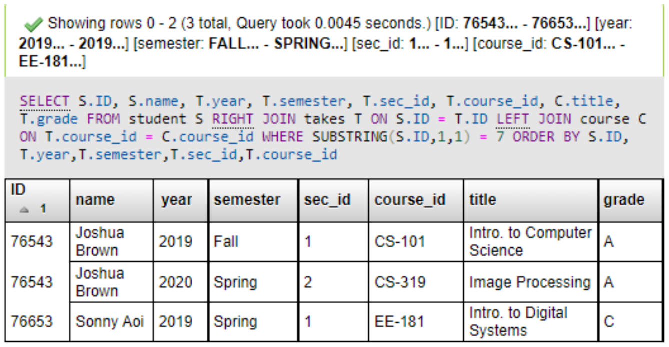

✓ Showing rows 0 - 5 (6 total, Query took 0.0058 seconds.) [semester: **FALL...** - **SUMMER...**] [sec_id: **1...** - **1...**] [course_id: **CS-101...** - **BIO-101...**]

```
SELECT TA.semester,TA.sec_id, TA.course_id, I.name AS ‘Instructor’, count(*)AS
'No_of_students' FROM takes AS TA JOIN teaches AS T on TA.course_id=T.course_id AND
TA.sec_id=T.sec_id AND TA.semester=T.semester AND TA.year=T.year JOIN instructor I ON
I.ID=T.ID WHERE T.year = "2019" GROUP BY TA.semester,TA.sec_id,TA.course_id,I.name ORDER BY
TA.semester,TA.sec_id,TA.course_id,I.name
```

semester ▲ 1	sec_id	course_id	‘Instructor’	No_of_students
Fall	1	CS-101	Srinivasan	6
Fall	1	CS-347	Srinivasan	4
Fall	1	PHY-101	Einstein	1
Spring	1	EE-181	Kim	1
Spring	2	CS-190	Brandt	2
Summer	1	BIO-101	Crick	1

7)

SELECT S.ID, S.name, T.year, T.semester, T.sec_id, T.course_id, C.title, T.grade FROM student S RIGHT JOIN takes T ON S.ID = T.ID LEFT JOIN course C ON T.course_id = C.course_id WHERE SUBSTRING(S.ID,1,1) = 7 ORDER BY S.ID, T.year,T.semester,T.sec_id,T.course_i

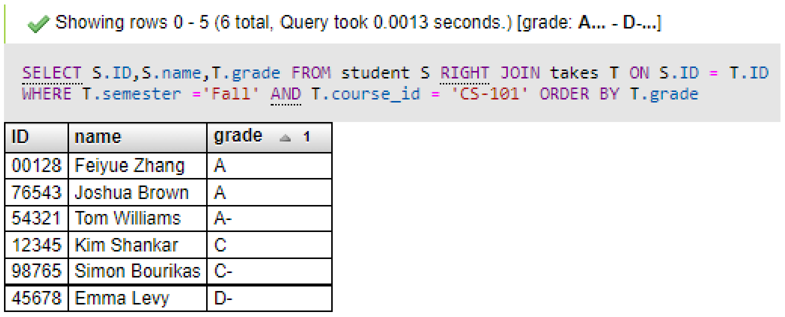

✓ Showing rows 0 - 2 (3 total, Query took 0.0045 seconds.) [ID: **76543...** - **76653...**] [year: **2019...** - **2019...**] [semester: **FALL...** - **SPRING...**] [sec_id: **1...** - **1...**] [course_id: **CS-101...** - **EE-181...**]

```
SELECT S.ID, S.name, T.year, T.semester, T.sec_id, T.course_id, C.title,
T.grade FROM student S RIGHT JOIN takes T ON S.ID = T.ID LEFT JOIN course C
ON T.course_id = C.course_id WHERE SUBSTRING(S.ID,1,1) = 7 ORDER BY S.ID,
T.year,T.semester,T.sec_id,T.course_id
```

ID ▲ 1	name	year	semester	sec_id	course_id	title	grade
76543	Joshua Brown	2019	Fall	1	CS-101	Intro. to Computer Science	A
76543	Joshua Brown	2020	Spring	2	CS-319	Image Processing	A
76653	Sonny Aoi	2019	Spring	1	EE-181	Intro. to Digital Systems	C

8)

SELECT S.ID,S.name,T.grade FROM student S RIGHT JOIN takes T ON S.ID = T.ID WHERE T.semester =’Fall’ AND T.course_id = ‘CS-101’ ORDER BY (T.grade);


✓ Showing rows 0 - 5 (6 total, Query took 0.0013 seconds.) [grade: **A...** - **D-...**]

```
SELECT S.ID,S.name,T.grade FROM student S RIGHT JOIN takes T ON S.ID = T.ID
WHERE T.semester ='Fall' AND T.course_id = 'CS-101' ORDER BY T.grade
```

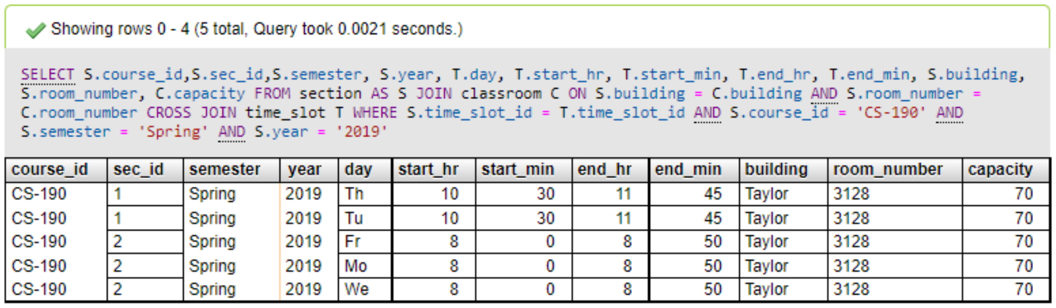
ID	name	grade ▲ 1
00128	Feiyue Zhang	A
76543	Joshua Brown	A
54321	Tom Williams	A-
12345	Kim Shankar	C
98765	Simon Bourikas	C-
45678	Emma Levy	D-

--- order grade by (A+,A,A-,B+,B,B-…)

SELECT S.ID,S.name,T.grade FROM student S RIGHT JOIN takes T ON S.ID = T.ID WHERE T.semester ='Fall' AND T.course_id = 'CS-101' ORDER BY LEFT(T.grade,1), CASE(RIGHT(T.grade, 1)) WHEN '+' THEN RIGHT(T.grade, 1) WHEN '-' THEN RIGHT(T.grade, 3) ELSE RIGHT(T.grade, 2) END;

9)

SELECT S.course_id,S.sec_id,S.semester, S.year, T.day, T.start_hr, T.start_min, T.end_hr, T.end_min, S.building, S.room_number, C.capacity FROM section AS S JOIN classroom C ON S.building = C.building AND S.room_number = C.room_number CROSS JOIN time_slot T WHERE S.time_slot_id = T.time_slot_id AND S.course_id = 'CS-190' AND S.semester = 'Spring' AND S.year = '2019';



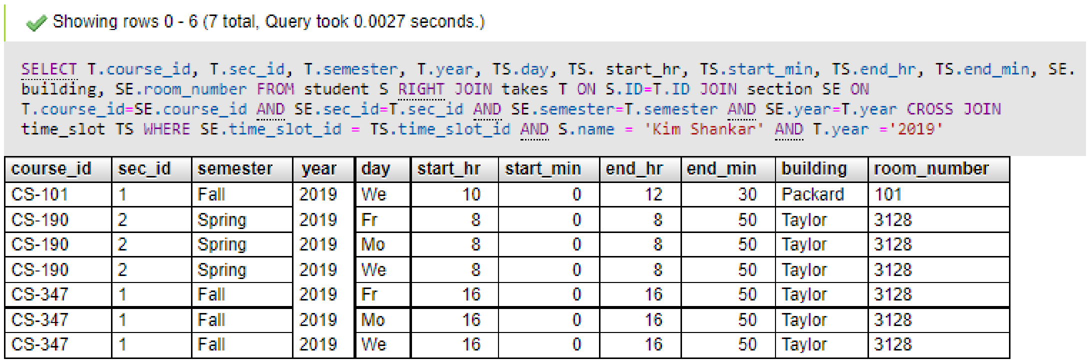
```
Showing rows 0 - 4 (5 total, Query took 0.0021 seconds.)

SELECT S.course_id,S.sec_id,S.semester, S.year, T.day, T.start_hr, T.start_min, T.end_hr, T.end_min, S.building,
S.room_number, C.capacity FROM section AS S JOIN classroom C ON S.building = C.building AND S.room_number =
C.room_number CROSS JOIN time_slot T WHERE S.time_slot_id = T.time_slot_id AND S.course_id = 'CS-190' AND
S.semester = 'Spring' AND S.year = '2019'
```

course_id	sec_id	semester	year	day	start_hr	start_min	end_hr	end_min	building	room_number	capacity
CS-190	1	Spring	2019	Th	10	30	11	45	Taylor	3128	70
CS-190	1	Spring	2019	Tu	10	30	11	45	Taylor	3128	70
CS-190	2	Spring	2019	Fr	8	0	8	50	Taylor	3128	70
CS-190	2	Spring	2019	Mo	8	0	8	50	Taylor	3128	70
CS-190	2	Spring	2019	We	8	0	8	50	Taylor	3128	70

10)

SELECT T.course_id, T.sec_id, T.semester, T.year, TS.day, TS. start_hr, TS.start_min, TS.end_hr, TS.end_min, SE. building, SE.room_number FROM student S RIGHT JOIN takes T ON S.ID=T.ID JOIN section SE ON T.course_id=SE.course_id AND SE.sec_id=T.sec_id AND SE.semester=T.semester AND SE.year=T.year CROSS JOIN time_slot TS WHERE SE.time_slot_id = TS.time_slot_id AND S.name = 'Kim Shankar' AND T.year ='2019';



```
Showing rows 0 - 6 (7 total, Query took 0.0027 seconds.)

SELECT T.course_id, T.sec_id, T.semester, T.year, TS.day, TS. start_hr, TS.start_min, TS.end_hr, TS.end_min, SE.
building, SE.room_number FROM student S RIGHT JOIN takes T ON S.ID=T.ID JOIN section SE ON
T.course_id=SE.course_id AND SE.sec_id=T.sec_id AND SE.semester=T.semester AND SE.year=T.year CROSS JOIN
time_slot TS WHERE SE.time_slot_id = TS.time_slot_id AND S.name = 'Kim Shankar' AND T.year ='2019'
```

course_id	sec_id	semester	year	day	start_hr	start_min	end_hr	end_min	building	room_number
CS-101	1	Fall	2019	We	10	0	12	30	Packard	101
CS-190	2	Spring	2019	Fr	8	0	8	50	Taylor	3128
CS-190	2	Spring	2019	Mo	8	0	8	50	Taylor	3128
CS-190	2	Spring	2019	We	8	0	8	50	Taylor	3128
CS-347	1	Fall	2019	Fr	16	0	16	50	Taylor	3128
CS-347	1	Fall	2019	Mo	16	0	16	50	Taylor	3128
CS-347	1	Fall	2019	We	16	0	16	50	Taylor	3128

11)

(a) Assertion definition

CREATE ASSERTION enrolment_restriction

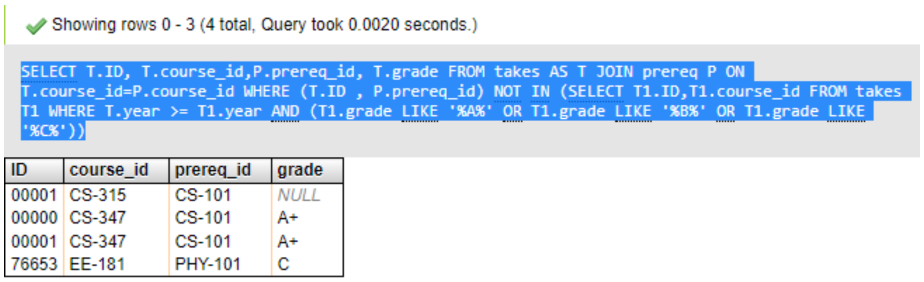
CHECK(NOT EXISTS (SELECT T.ID, T.course_id,P.prereq_id FROM takes AS T JOIN prereq P ON T.course_id=P.course_id WHERE (T.ID , P.prereq_id) NOT IN (SELECT T1.ID,T1.course_id FROM takes T1 WHERE T.year >= T1.year AND T1.grade IN ('A+','A','A-','B+','B','B-','C+','C','C-')));

Or

```
SELECT T.ID, T.course_id,P.prereq_id, T.grade FROM takes AS T JOIN prereq P ON T.course_i
d=P.course_id WHERE (T.ID , P.prereq_id) NOT IN (SELECT T1.ID,T1.course_id FROM takes T1
WHERE T.year >= T1.year AND (T1.grade LIKE '%A%' OR T1.grade LIKE '%B%' OR T1.grade LIKE
'%C%'))
```

(b) current violation

SELECT A.ID, A.course_id, B.course_id AS prereq_id FROM (SELECT T.ID, T.course_id, T.year, T.grade FROM takes T, prereq P WHERE T.course_id = P.course_id) AS A JOIN (SELECT DISTINCT T.ID,T.course_id,T.year,T.grade FROM takes T, prereq P WHERE T.course_id = P.prereq_id) AS B ON A.ID = B.ID WHERE B.year > A.year OR (B.grade NOT IN ('A+','A','A-','B+','B','B-','C+','C','C-'));


✓ Showing rows 0 - 3 (4 total, Query took 0.0020 seconds.)

```
SELECT T.ID, T.course_id,P.prereq_id, T.grade FROM takes AS T JOIN prereq P ON
T.course_id=P.course_id WHERE (T.ID , P.prereq_id) NOT IN (SELECT T1.ID,T1.course_id FROM takes
T1 WHERE T.year >= T1.year AND (T1.grade LIKE '%A%' OR T1.grade LIKE '%B%' OR T1.grade LIKE
'%C%'))
```

ID	course_id	prereq_id	grade
00001	CS-315	CS-101	NULL
00000	CS-347	CS-101	A+
00001	CS-347	CS-101	A+
76653	EE-181	PHY-101	C

12)

a)

DELIMITER $$

CREATE TRIGGER `PREREQUISITE_VIOLATION`

BEFORE INSERT ON `takes`

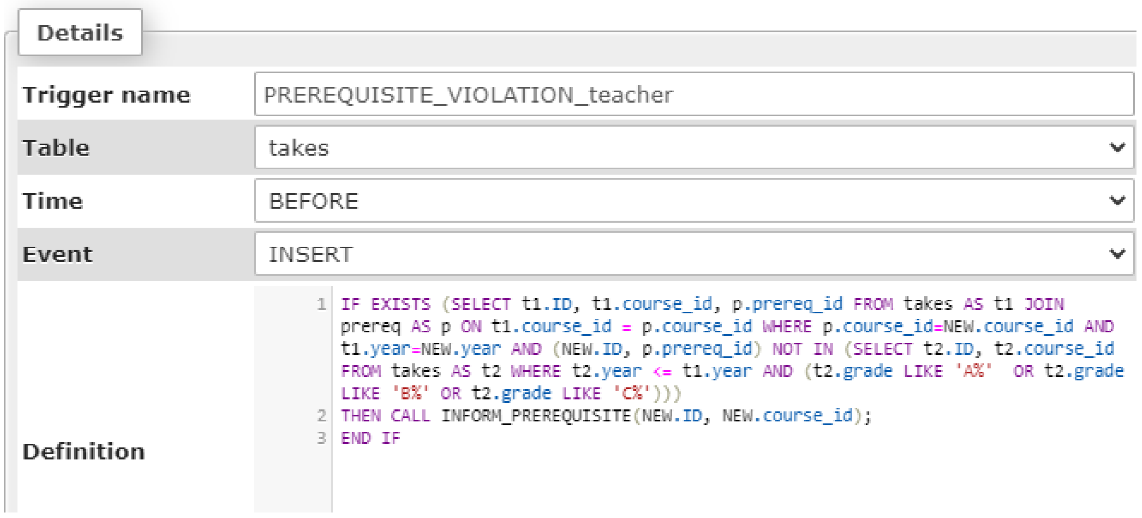
FOR EACH ROW

 IF EXISTS (SELECT t1.ID, t1.course_id, p.prereq_id FROM takes AS t1 JOIN prereq AS p ON t1.course_id = p.course_id WHERE p.course_id=NEW.course_id AND t1.year=NEW.year AND (NEW.ID, p.prereq_id) NOT IN (SELECT t2.ID, t2.course_id FROM takes AS t2 WHERE t2.year <= t1.year AND (t2.grade LIKE 'A%' OR t2.grade LIKE 'B%' OR t2.grade LIKE 'C%')))

THEN CALL INFORM_PREREQUISITE(NEW.ID, NEW.course_id);

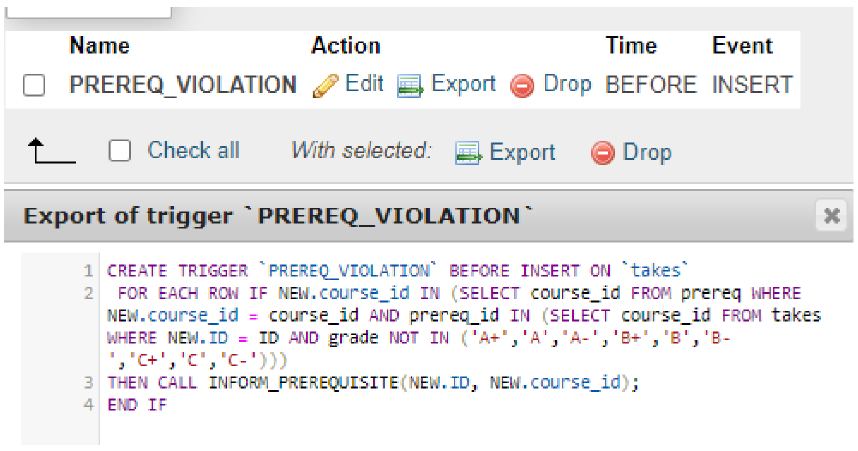
END IF

$$ DELIMITER;



Trigger name	PREREQUISITE_VIOLATION_teacher
Table	takes
Time	BEFORE
Event	INSERT
Definition	

```
IF EXISTS (SELECT t1.ID, t1.course_id, p.prereq_id FROM takes AS t1 JOIN
prereq AS p ON t1.course_id = p.course_id WHERE p.course_id=NEW.course_id AND
t1.year=NEW.year AND (NEW.ID, p.prereq_id) NOT IN (SELECT t2.ID, t2.course_id
FROM takes AS t2 WHERE t2.year <= t1.year AND (t2.grade LIKE 'A%'  OR t2.grade
LIKE 'B%' OR t2.grade LIKE 'C%')))
THEN CALL INFORM_PREREQUISITE(NEW.ID, NEW.course_id);
END IF
```



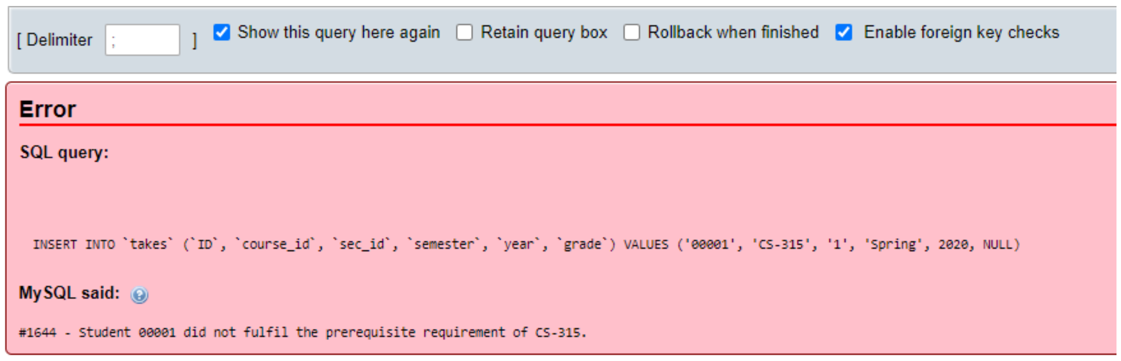
Name	Action	Time	Event
PREREQ_VIOLATION	Edit Export Drop	BEFORE	INSERT

Export of trigger `PREREQ_VIOLATION`

```
CREATE TRIGGER `PREREQ_VIOLATION` BEFORE INSERT ON `takes`
 FOR EACH ROW IF NEW.course_id IN (SELECT course_id FROM prereq WHERE
NEW.course_id = course_id AND prereq_id IN (SELECT course_id FROM takes
WHERE NEW.ID = ID AND grade NOT IN ('A+','A','A-','B+','B','B-
','C+','C','C-')))
THEN CALL INFORM_PREREQUISITE(NEW.ID, NEW.course_id);
END IF
```

b)

INSERT INTO `takes` (`ID`, `course_id`, `sec_id`, `semester`, `year`, `grade`) VALUES ('00001', 'CS-315', '1', 'Spring', 2020, NULL)



[Delimiter ;] [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks]

Error

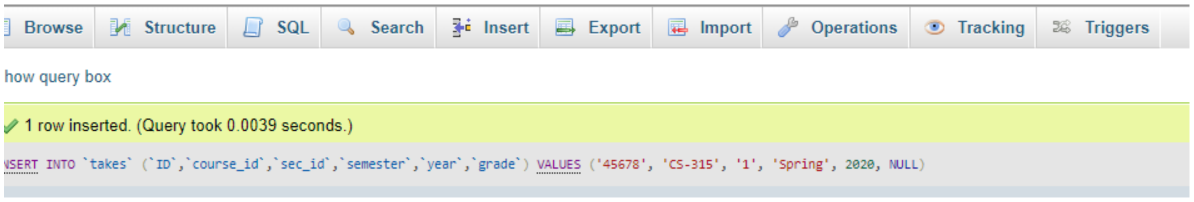
SQL query:

```
INSERT INTO `takes` (`ID`, `course_id`, `sec_id`, `semester`, `year`, `grade`) VALUES ('00001', 'CS-315', '1', 'Spring', 2020, NULL)
```

MySQL said:

```
#1644 - Student 00001 did not fulfil the prerequisite requirement of CS-315.
```

INSERT INTO `takes` (`ID`,`course_id`,`sec_id`,`semester`,`year`,`grade`) VALUES ('45678', 'CS-315', '1', 'Spring', 2020, NULL)



1 row inserted. (Query took 0.0039 seconds.)

```
NSERT INTO `takes` (`ID`,`course_id`,`sec_id`,`semester`,`year`,`grade`) VALUES ('45678', 'CS-315', '1', 'Spring', 2020, NULL)
```