CS313 : DataBases and Information Systems Lab

Lab Assignment 2

Sourabh Bhosale
200010004

August 30, 2022

1 Table-wise all integrity constraints

Table	Primary Key	Domain of P.K.	Foreign keys (Referencing table)	Not Null
classroom	building, room_number	varchar, varchar	-	-
department	dept_name	varchar	-	-
course	course_id	varchar	(dept_name) references department	-
instructor	ID	varchar	(dept_name) references department	name
section	course_id, sec_id, semester, year	varchar, varchar, varchar, numeric	(course_id) references course, (building, room_number) references classroom	-
teaches	ID, course_id, sec_id, semester, year	varchar, varchar, varchar, varchar, numeric	(course_id,sec_id, semester, year) references section, (ID) references instructor	-
student	ID	varchar	(dept_name) references department	name
takes	ID, course_id, sec_id, semester, year	varchar, varchar, varchar, varchar, numeric	(course_id,sec_id, semester, year) references section, (ID) references student	-
advisor	s_ID	varchar	(i_ID) references instructor (ID), (s_ID) references student (ID)	-
time_slot	time_slot_id, day, start_hr, start_min	varchar, varchar, numeric, numeric	-	-
prereq	course_id, prereq_id	varchar, varchar	(course_id) references course, (prereq_id) references course	-

Figure 1: All integrity constraints for different tables
(Primary key, Domain of Primary key, Foreign key and Not Null)

2 Student profile

Here, full profile of the student using the tables student, department, takes, advisor and instructor is made.

2.1 First approach

Sorting of all the columns is done here without any repetitions using the given tables.

Query:

```
SELECT student.*, d.building, d.budget, t.course_id, t.sec_id, t.semester,
    t.year, t.grade, a.s_ID, a.i_ID, i.name, i.dept_name, i.salary
FROM student, department AS d, takes AS t, advisor AS a, instructor AS i
WHERE student.dept_name=d.dept_name AND student.ID=t.ID AND
    student.ID=a.s_ID AND a.i_ID=i.ID AND student.ID='12345';
```

Output:

ID	name	dept_name	tot_cred	building	budget	course_id	sec_id	semester	year	grade	s_ID	i_ID	name	dept_name	salary
12345	Shankar	Comp. Sci.	32	Taylor	100000	CS-101	1	Fall	2017	C	12345	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Taylor	100000	CS-190	2	Spring	2017	A	12345	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Taylor	100000	CS-315	1	Spring	2018	A	12345	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Taylor	100000	CS-347	1	Fall	2017	A	12345	10101	Srinivasan	Comp. Sci.	65000

2.2 Alternate approach

Sorting of all the columns is done here according to the example output given in the assignment question.

Query:

```
SELECT *
FROM student, department AS d, takes AS t, advisor AS a, instructor AS i
WHERE student.dept_name=d.dept_name AND student.ID=t.ID AND
student.ID=a.s_ID AND a.i_ID=i.ID AND student.ID='12345';
```

Output:

ID	name	dept_name	tot_cred	dept_name	building	budget	ID	course_id	sec_id	semester	year	grade	s_ID	i_ID	ID	name	dept_name	salary
12345	Shankar	Comp. Sci.	32	Comp. Sci.	Taylor	100000	12345	CS-101	1	Fall	2017	C	12345	10101	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Comp. Sci.	Taylor	100000	12345	CS-190	2	Spring	2017	A	12345	10101	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Comp. Sci.	Taylor	100000	12345	CS-315	1	Spring	2018	A	12345	10101	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Comp. Sci.	Taylor	100000	12345	CS-347	1	Fall	2017	A	12345	10101	10101	Srinivasan	Comp. Sci.	65000

In both places, you can change the *student.ID* value accordingly.

3 Trying out *select* and *insert* on all the tables

3.1 Query

```
INSERT INTO student
        VALUES('00002', 'Sourabh', 'Comp. Sci.', 125);
INSERT INTO course
        VALUES('CS-301', 'Computer Architecture', 'Comp. Sci.', 6);
INSERT INTO instructor
        VALUES('10001', 'Rajshekhar K', 'Comp. Sci.', 70000);
INSERT INTO teaches
        VALUES('10001', 'CS-301', '2', 'Spring', 2020);
INSERT INTO prereq
        VALUES('CS-301', 'CS-101');
INSERT INTO section
        VALUES('CS-301', '2', 'Spring', 2020, 'Watson', '120', 'B');
INSERT INTO classroom
        VALUES('Newton', '150', 100);
INSERT INTO department
        VALUES('MMAE', 'Newton', 95000);
INSERT INTO takes
        VALUES('00002', 'CS-301', '2', 'Spring', 2020, 'A');
INSERT INTO advisor
        VALUES('00002', '10001');
INSERT INTO time_slot
        VALUES('B', 'M', 8, 30, 9, 30);

SELECT * FROM student;
SELECT * FROM course;
SELECT * FROM instructor;
SELECT * FROM teaches;
SELECT * FROM prereq;
SELECT * FROM section;
SELECT * FROM classroom;
SELECT * FROM department;
SELECT * FROM takes;
SELECT * FROM advisor;
SELECT * FROM time_slot;
```

3.2 Results

You can see the new inputs in the last row of each table.

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120
00002	Sourabh	Comp. Sci.	125

Figure 2: Student table after insertion

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4
CS-301	Computer Architecture	Comp. Sci.	6

Figure 3: Course table after insertion

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
10001	Rajshekhar K	Comp. Sci.	70000

Figure 4: Instructor table after insertion

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017
10001	CS-301	2	Spring	2020

Figure 5: Teaches table after insertion

course_id	prereq_id
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101
CS-301	CS-101

Figure 6: Prereq table after insertion

course_id	sec_id	semester	year	building	room_number	time_slot_id
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A
CS-301	2	Spring	2020	Watson	120	B

Figure 7: Section table after insertion

building	room_number	capacity
Packard	101	500
Painter	514	10
Taylor	3128	70
Watson	100	30
Watson	120	50
Newton	150	100

Figure 8: Classroom table after insertion

dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000
MMAE	Newton	95000

Figure 9: Department table after insertion

ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2017	A
00128	CS-347	1	Fall	2017	A-
12345	CS-101	1	Fall	2017	C
12345	CS-190	2	Spring	2017	A
12345	CS-315	1	Spring	2018	A
12345	CS-347	1	Fall	2017	A
19991	HIS-351	1	Spring	2018	B
23121	FIN-201	1	Spring	2018	C+
44553	PHY-101	1	Fall	2017	B-
45678	CS-101	1	Fall	2017	F
45678	CS-101	1	Spring	2018	B+
45678	CS-319	1	Spring	2018	B
54321	CS-101	1	Fall	2017	A-
54321	CS-190	2	Spring	2017	B+
55739	MU-199	1	Spring	2018	A-
76543	CS-101	1	Fall	2017	A
76543	CS-319	2	Spring	2018	A
76653	EE-181	1	Spring	2017	C
98765	CS-101	1	Fall	2017	C-
98765	CS-315	1	Spring	2018	B
98988	BIO-101	1	Summer	2017	A
98988	BIO-301	1	Summer	2018	
00002	CS-301	2	Spring	2020	A

Figure 10: Takes table after insertion

s_ID	i_ID
00128	45565
12345	10101
23121	76543
44553	22222
45678	22222
76543	45565
76653	98345
98765	98345
98988	76766
00002	10001

Figure 11: Advisor table after insertion

time_slot_id	day	start_hr	start_min	end_hr	end_min
A	M	8	0	8	50
A	W	8	0	8	50
A	F	8	0	8	50
B	M	9	0	9	50
B	W	9	0	9	50
B	F	9	0	9	50
C	M	11	0	11	50
C	W	11	0	11	50
C	F	11	0	11	50
D	M	13	0	13	50
D	W	13	0	13	50
D	F	13	0	13	50
E	T	10	30	11	45
E	R	10	30	11	45
F	T	14	30	15	45
F	R	14	30	15	45
G	M	16	0	16	50
G	W	16	0	16	50
G	F	16	0	16	50
H	W	10	0	12	30
B	M	8	30	9	30

Figure 12: Time_slot table after insertion

4 Additional queries

4.1 Students (ID and names) from xxx department who have done courses from a room in building yyy

4.1.1 First Approach

Here, we will refer *section* and *takes* table to see which courses a student takes then will filter out building-wise.
Query:

```
SELECT DISTINCT A.ID, A.name, A.dept_name, sec.building
FROM(
  SELECT *
  FROM student AS s, takes AS t
  WHERE s.ID = t.ID ) AS A NATURAL JOIN section AS sec
WHERE sec.building = 'Taylor' AND A.dept_name = 'Comp. Sci.';
```

Output:

ID	name	dept_name	building
00128	Zhang	Comp. Sci.	Taylor
12345	Shankar	Comp. Sci.	Taylor
54321	Williams	Comp. Sci.	Taylor
76543	Brown	Comp. Sci.	Taylor

4.1.2 Another Approach

Here, we will refer only *student* and *department* to collect data regarding buildings.
Query:

```
SELECT s.ID, s.name, s.dept_name, d.building
FROM student AS s, department AS d
WHERE s.dept_name=d.dept_name AND s.dept_name='Comp. Sci.' AND
    d.building='Taylor';
```

Output:

ID	name	dept_name	building
00128	Zhang	Comp. Sci.	Taylor
12345	Shankar	Comp. Sci.	Taylor
54321	Williams	Comp. Sci.	Taylor
76543	Brown	Comp. Sci.	Taylor

4.2 Students who have A grade as well as a C grade

Query:

```
SELECT DISTINCT A_table.ID,A_table.name
FROM(SELECT s.*, t.*
          FROM student AS s, takes AS t
    WHERE s.ID=t.ID AND t.grade='A') as A_table,
    (SELECT s.*, t.*
          FROM student AS s, takes AS t
    WHERE s.ID=t.ID AND t.grade='C') AS C_table
WHERE  A_table.ID=C_table.ID;
```

Output:

ID	name
12345	Shankar

4.3 All buildings and rooms which have classes on Wednesday

Query:

```
SELECT DISTINCT sec.building, sec.room_number, ts.day
FROM section AS sec, time_slot AS ts
WHERE sec.time_slot_id=ts.time_slot_id AND ts.day='W';
```

Output:

building	room_number	day
Painter	514	W
Packard	101	W
Taylor	3128	W
Watson	120	W
Watson	100	W