CS313, Assignment - 2

Shashank P, 200010048

August 30, 2022

1 Integrity Constraints in Univeristy Table

Table	**Primary Key**	**Domain of PK**	**Foreign key(Referencing table)**	**Not Null**
classroom	building, room_number	varchar	-	building, room_number
department	dept_name	varchar	-	dept_name
course	course_id	varchar	foreign key (dept_name) references department	course_id
instructor	ID	varchar	foreign key (dept_name) references department	name, ID
section	course_id, sec_id, semester, year	varchar varchar varchar numeric	foreign key (course_id) references course, foreign key (building, room_number) references classroom	course_id, sec_id, semester, year
teaches	ID, course_id, sec_id, semester, year	varchar varchar varchar varchar numeric	foreign key (course_id,sec_id, semester, year) references section, foreign key (ID) references instructor	ID, course_id, sec_id, semester, year

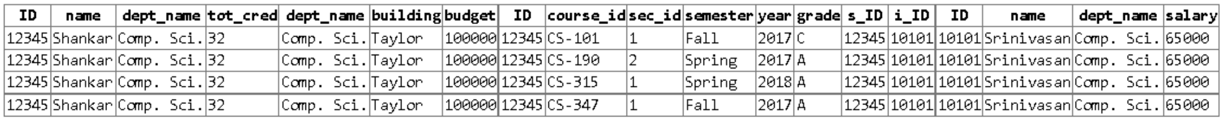
student	ID	varchar	foreign key (dept_name) references department	ID, name
takes	ID, course_id, sec_id, semester, year	varchar varchar varchar varchar numeric	foreign key (course_id,sec_id, semester, year) references section, foreign key (ID) references student	ID, course_id, sec_id, semester, year
advisor	s_ID	varchar	foreign key (i_ID) references instructor (ID), foreign key (s_ID) references student (ID)	s_ID
time_slot	time_slot_id, day, start_hr, start_min	varchar varchar numeric numeric	-	time_slot_id, day, start_hr, start_min
prereq	course_id, prereq_id	varchar varchar	foreign key (course_id) references course, foreign key (prereq_id) references course	course_id, prereq_id

Table 1: Integrity Constraints of all tables

2 One student's complete profile

Select complete profile of a student named 'Shankar' from multiple tables. Take a cartesian product of all tables, then combine relevent columns.

```
select * from student, department, takes, advisor, instructor
where student.name='Shankar'
  and student.dept_name=department.dept_name
  and student.ID=takes.ID
  and (student.ID=advisor.s_ID and instructor.ID=advisor.i_ID)
  and department.dept_name=instructor.dept_name;
```



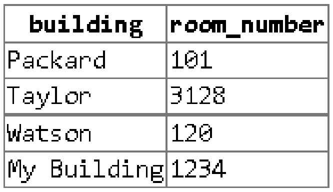
ID	name	dept_name	tot_cred	dept_name	building	budget	ID	course_id	sec_id	semester	year	grade	s_ID	i_ID	ID	name	dept_name	salary
12345	Shankar	Comp. Sci.	32	Comp. Sci.	Taylor	100000	12345	CS-101	1	Fall	2017	C	12345	10101	10101	Srinivasan	Comp. Sci.	65000
12345	Shankar	Comp. Sci.	32	Comp. Sci.	Taylor	100000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

Figure 1: Query output

3 Select and Insert queries

a Table: classroom

```
  insert into classroom values('My Building', 1234, 1000);
  select building, room_number from classroom where capacity>=50;
```

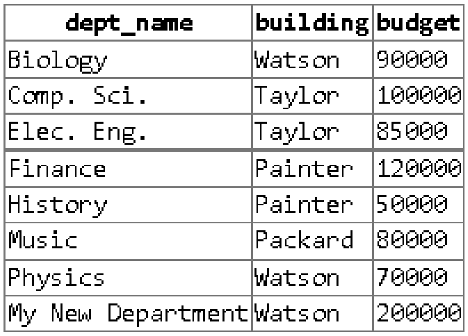


building	room_number
Packard	101
Taylor	3128
Watson	120
My Building	1234

Figure 2: Query output

b Table: department

```
  insert into department values('My New Department', 'Watson',
    200000);
  select * from department where budget between 50000 and 210000;
```

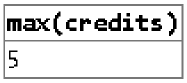


dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000
My New Department	Watson	200000

Figure 3: Query output

c Table: course

```sql
insert into course values('NN-101', 'A New Course', 'History', 5)
  ;
select max(credits) from course;
```

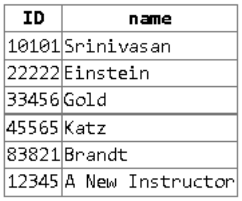


max(credits)
5

Figure 4: Query output

d Table: instructor

```sql
insert into instructor values('12345', 'A New Instructor', 'Comp.
   Sci.', 50000);
select ID, name from instructor where dept_name in ('Comp. Sci.',
   "Physics");
```

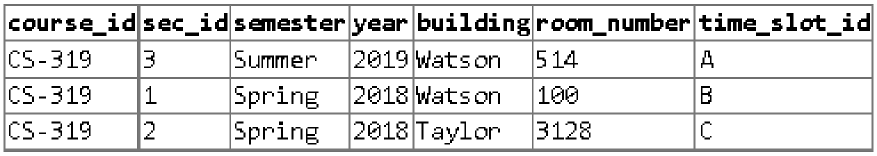


ID	name
10101	Srinivasan
22222	Einstein
33456	Gold
45565	Katz
83821	Brandt
12345	A New Instructor

Figure 5: Query output

e Table: section

```sql
insert into section values('CS-319', '3', 'Summer', 2019, 'Watson', '514', 'A');
select * from section where course_id='CS-319' order by time_slot_id asc
```

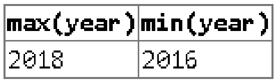


course_id	sec_id	semester	year	building	room_number	time_slot_id
CS-319	3	Summer	2019	Watson	514	A
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C

Figure 6: Query output

f Table: teaches

```sql
insert into teaches values('45565', 'CS-101', '1', 'Fall', 2016);
select max(year), min(year) from teaches where ID='45565';
```

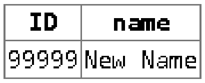


max(year)	min(year)
2018	2016

Figure 7: Query output

g Table: student

```sql
insert into student values('99999', 'New Name', 'Comp. Sci.', 250);
select ID, name from student where tot_cred in (select max(tot_cred) from student);
```

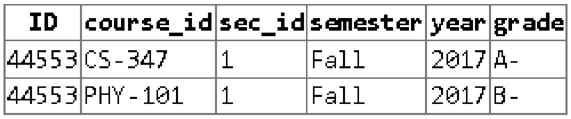


ID	name
99999	New Name

Figure 8: Query output

h Table: takes

```sql
insert into takes values('44553', 'CS-347', '1', 'Fall', 2017, 'A-');
select * from takes where ID=='44553';
```

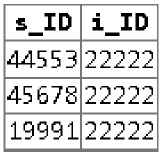


ID	course_id	sec_id	semester	year	grade
44553	CS-347	1	Fall	2017	A-
44553	PHY-101	1	Fall	2017	B-

Figure 9: Query output

i Table: advisor

```
insert into advisor values('19991', '22222');
select * from advisor where i_ID='22222';
```

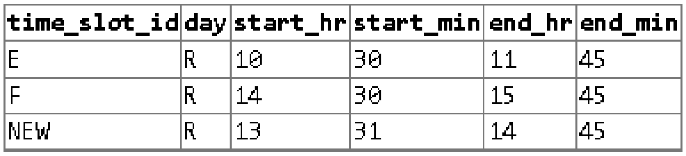


s_ID	i_ID
44553	22222
45678	22222
19991	22222

Figure 10: Query output

j Table: time_slot

```
insert into time_slot values('NEW', 'R', 13, 31, 14, 45);
select * from time_slot where day='R';
```

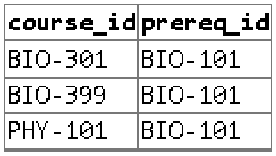


time_slot_id	day	start_hr	start_min	end_hr	end_min
E	R	10	30	11	45
F	R	14	30	15	45
NEW	R	13	31	14	45

Figure 11: Query output

k Table: prereq

```
insert into prereq values('PHY−101', 'BIO−101');
select * from prereq where prereq_id='BIO−101';
```



course_id	prereq_id
BIO-301	BIO-101
BIO-399	BIO-101
PHY-101	BIO-101

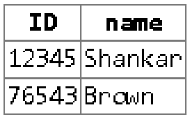
Figure 12: Query output

4 Specific Queries

a Stduent taking Comp. Sci. from Watson building

Take cartesian product of student, takes and section table.
Then combine relevent columns, also add condintions for department name and building.

```
  select student.ID, student.name from student, takes, section
  where student.dept_name='Comp. Sci.'
    and student.ID=takes.ID
    and takes.course_id=section.course_id
    and section.building='Watson'
```



ID	name
12345	Shankar
76543	Brown

Figure 13: Query output

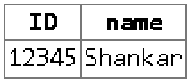
b Student having both 'A' and 'C' grade

Find tables of student names for grade 'A' and grade 'C' and then take their **intersection**

```
select ID, name
from student natural join takes
where takes.grade='A'

intersect

select ID, name
from student natural join takes
where takes.grade='C'
```



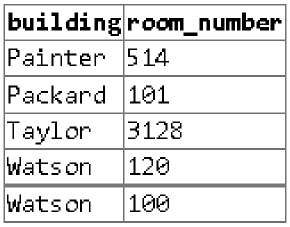
ID	name
12345	Shankar

Figure 14: Query output

c Buildings that have classes on Wednesday

Natural join section and time_slot tables. Then select the distinct building and room_number with time_slot wednesday.

```
  select distinct building, room_number
  from section natural join time_slot
  where time_slot.day='W'
```



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

Figure 15: Query output