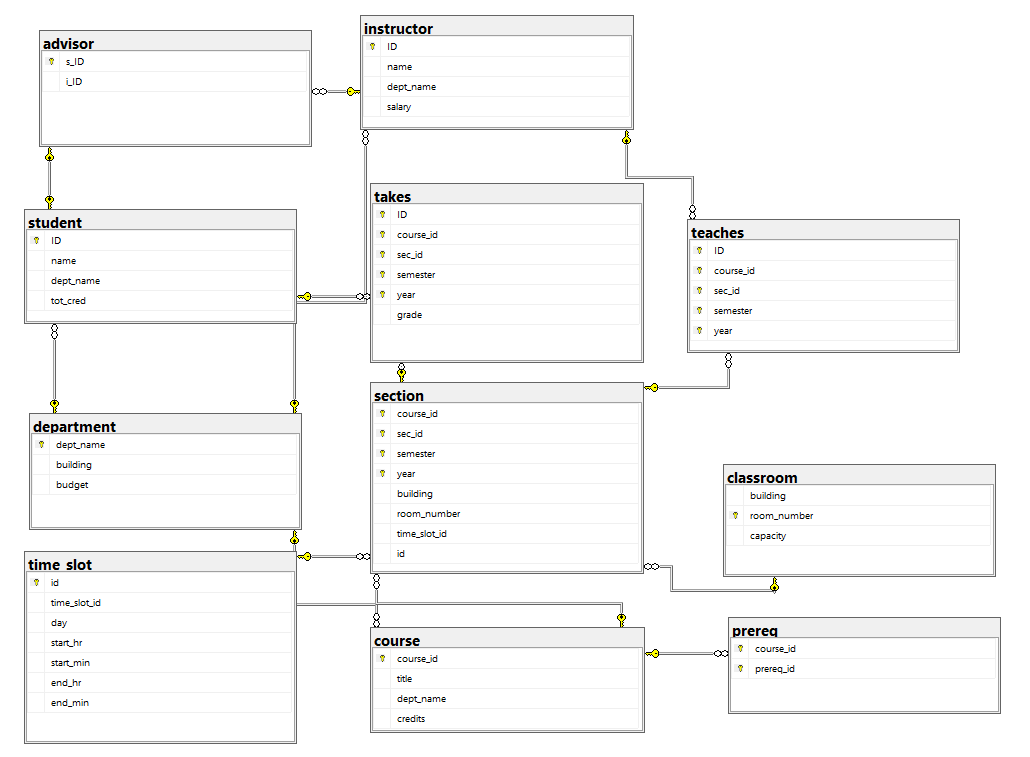
**SQL University assignment**

**Database Schema**

****

Initial queries for database, table creation and inserting values

Create and use database

create database university;

use university;

Create Tables

create table classroom

(building varchar(15),

room\_number varchar(7) primary key,

capacity numeric(4,0),

);

create table department

(dept\_name varchar(20),

building varchar(15),

budget numeric(12,2) check (budget > 0),

primary key (dept\_name)

);

create table course

(course\_id varchar(8),

title varchar(50),

dept\_name varchar(20),

credits numeric(2,0) check (credits > 0),

primary key (course\_id),

foreign key (dept\_name) references department (dept\_name)

on delete set null

);

create table instructor

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

salary numeric(8,2) check (salary > 29000),

primary key (ID),

foreign key (dept\_name) references department (dept\_name)

on delete set null

);

create table time\_slot

(id int primary key,

time\_slot\_id varchar(4),

day varchar(1),

start\_hr numeric(2) check (start\_hr >= 0 and start\_hr < 24),

start\_min numeric(2) check (start\_min >= 0 and start\_min < 60),

end\_hr numeric(2) check (end\_hr >= 0 and end\_hr < 24),

end\_min numeric(2) check (end\_min >= 0 and end\_min < 60)

);

create table section

(course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6)

check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),

year numeric(4,0) check (year > 1701 and year < 2100),

building varchar(15),

room\_number varchar(7),

time\_slot\_id varchar(4),

id int,

primary key (course\_id, sec\_id, semester, year),

foreign key (course\_id) references course (course\_id)

on delete cascade,

foreign key (room\_number) references classroom (room\_number)

on delete set null,

foreign key (id) references time\_slot (id)

on delete set null,

);

create table teaches

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id, sec\_id, semester, year) references section (course\_id, sec\_id, semester, year)

on delete cascade,

foreign key (ID) references instructor (ID)

on delete cascade

);

create table student

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

tot\_cred numeric(3,0) check (tot\_cred >= 0),

primary key (ID),

foreign key (dept\_name) references department (dept\_name)

on delete set null

);

create table takes

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

grade varchar(2),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id, sec\_id, semester, year) references section (course\_id, sec\_id, semester, year)

on delete cascade,

foreign key (ID) references student (ID)

on delete cascade

);

create table advisor

(s\_ID varchar(5),

i\_ID varchar(5),

primary key (s\_ID),

foreign key (i\_ID) references instructor (ID)

on delete set null,

foreign key (s\_ID) references student (ID)

on delete cascade

);

create table prereq

(course\_id varchar(8),

prereq\_id varchar(8),

primary key (course\_id, prereq\_id),

foreign key (course\_id) references course (course\_id)

on delete cascade,

foreign key (prereq\_id) references course (course\_id)

);

Inserting Values

insert into classroom values ('Packard', '101', '500');

insert into classroom values ('Painter', '514', '10');

insert into classroom values ('Taylor', '3128', '70');

insert into classroom values ('Watson', '100', '30');

insert into classroom values ('Watson', '120', '50');

insert into classroom values ('Taylor', '112', '30');

insert into classroom values ('Painter', '234', '50');

insert into classroom values ('Packard', '303', '56');

insert into department values ('Biology', 'Watson', '90000');

insert into department values ('Comp. Sci.', 'Taylor', '100000');

insert into department values ('Elec. Eng.', 'Taylor', '85000');

insert into department values ('Finance', 'Painter', '120000');

insert into department values ('History', 'Painter', '50000');

insert into department values ('Music', 'Packard', '80000');

insert into department values ('Physics', 'Watson', '70000');

insert into course values ('BIO-101', 'Intro. to Biology', 'Biology', '4');

insert into course values ('BIO-301', 'Genetics', 'Biology', '4');

insert into course values ('BIO-399', 'Computational Biology', 'Biology', '3');

insert into course values ('CS-101', 'Intro. to Computer Science', 'Comp. Sci.', '4');

insert into course values ('CS-190', 'Game Design', 'Comp. Sci.', '4');

insert into course values ('CS-315', 'Robotics', 'Comp. Sci.', '3');

insert into course values ('CS-319', 'Image Processing', 'Comp. Sci.', '3');

insert into course values ('CS-347', 'Database System Concepts', 'Comp. Sci.', '3');

insert into course values ('EE-181', 'Intro. to Digital Systems', 'Elec. Eng.', '3');

insert into course values ('FIN-201', 'Investment Banking', 'Finance', '3');

insert into course values ('HIS-351', 'World History', 'History', '3');

insert into course values ('MU-199', 'Music Video Production', 'Music', '3');

insert into course values ('PHY-101', 'Physical Principles', 'Physics', '4');

insert into instructor values ('10101', 'Srinivasan', 'Comp. Sci.', '65000');

insert into instructor values ('12121', 'Wu', 'Finance', '90000');

insert into instructor values ('15151', 'Mozart', 'Music', '40000');

insert into instructor values ('22222', 'Einstein', 'Physics', '95000');

insert into instructor values ('32343', 'El Said', 'History', '60000');

insert into instructor values ('33456', 'Gold', 'Physics', '87000');

insert into instructor values ('45565', 'Katz', 'Comp. Sci.', '75000');

insert into instructor values ('58583', 'Califieri', 'History', '62000');

insert into instructor values ('76543', 'Singh', 'Finance', '80000');

insert into instructor values ('76766', 'Crick', 'Biology', '72000');

insert into instructor values ('83821', 'Brandt', 'Comp. Sci.', '92000');

insert into instructor values ('98345', 'Kim', 'Elec. Eng.', '80000');

insert into time\_slot values (1,'A', 'M', '8', '0', '8', '50');

insert into time\_slot values (2,'A', 'W', '8', '0', '8', '50');

insert into time\_slot values (3,'A', 'F', '8', '0', '8', '50');

insert into time\_slot values (4,'B', 'M', '9', '0', '9', '50');

insert into time\_slot values (5,'B', 'W', '9', '0', '9', '50');

insert into time\_slot values (6,'B', 'F', '9', '0', '9', '50');

insert into time\_slot values (7,'C', 'M', '11', '0', '11', '50');

insert into time\_slot values (8,'C', 'W', '11', '0', '11', '50');

insert into time\_slot values (9,'C', 'F', '11', '0', '11', '50');

insert into time\_slot values (10,'D', 'M', '13', '0', '13', '50');

insert into time\_slot values (11,'D', 'W', '13', '0', '13', '50');

insert into time\_slot values (12,'D', 'F', '13', '0', '13', '50');

insert into time\_slot values (13,'E', 'T', '10', '30', '11', '45 ');

insert into time\_slot values (14,'E', 'R', '10', '30', '11', '45 ');

insert into time\_slot values (15,'F', 'T', '14', '30', '15', '45 ');

insert into time\_slot values (16,'F', 'R', '14', '30', '15', '45 ');

insert into time\_slot values (17,'G', 'M', '16', '0', '16', '50');

insert into time\_slot values (18,'G', 'W', '16', '0', '16', '50');

insert into time\_slot values (19,'G', 'F', '16', '0', '16', '50');

insert into time\_slot values (20,'H', 'W', '10', '0', '12', '30');

insert into section values ('BIO-101', '1', 'Summer', '2017', 'Painter', '514', 'B',1);

insert into section values ('BIO-301', '1', 'Summer', '2018', 'Painter', '514', 'A',2);

insert into section values ('CS-101', '1', 'Fall', '2017', 'Packard', '101', 'H',3);

insert into section values ('CS-101', '1', 'Spring', '2018', 'Packard', '101', 'F',4);

insert into section values ('CS-190', '1', 'Spring', '2017', 'Taylor', '3128', 'E',5);

insert into section values ('CS-190', '2', 'Spring', '2017', 'Taylor', '3128', 'A',6);

insert into section values ('CS-315', '1', 'Spring', '2018', 'Watson', '120', 'D',7);

insert into section values ('CS-319', '1', 'Spring', '2018', 'Watson', '100', 'B',8);

insert into section values ('CS-319', '2', 'Spring', '2018', 'Taylor', '3128', 'C',9);

insert into section values ('CS-347', '1', 'Fall', '2017', 'Taylor', '3128', 'A',10);

insert into section values ('EE-181', '1', 'Spring', '2017', 'Taylor', '3128', 'C',11);

insert into section values ('FIN-201', '1', 'Spring', '2018', 'Packard', '101', 'B',12);

insert into section values ('HIS-351', '1', 'Spring', '2018', 'Painter', '514', 'C',13);

insert into section values ('MU-199', '1', 'Spring', '2018', 'Packard', '101', 'D',14);

insert into section values ('PHY-101', '1', 'Fall', '2017', 'Watson', '100', 'A',15);

insert into teaches values ('10101', 'CS-101', '1', 'Fall', '2017');

insert into teaches values ('10101', 'CS-315', '1', 'Spring', '2018');

insert into teaches values ('10101', 'CS-347', '1', 'Fall', '2017');

insert into teaches values ('12121', 'FIN-201', '1', 'Spring', '2018');

insert into teaches values ('15151', 'MU-199', '1', 'Spring', '2018');

insert into teaches values ('22222', 'PHY-101', '1', 'Fall', '2017');

insert into teaches values ('32343', 'HIS-351', '1', 'Spring', '2018');

insert into teaches values ('45565', 'CS-101', '1', 'Spring', '2018');

insert into teaches values ('45565', 'CS-319', '1', 'Spring', '2018');

insert into teaches values ('76766', 'BIO-101', '1', 'Summer', '2017');

insert into teaches values ('76766', 'BIO-301', '1', 'Summer', '2018');

insert into teaches values ('83821', 'CS-190', '1', 'Spring', '2017');

insert into teaches values ('83821', 'CS-190', '2', 'Spring', '2017');

insert into teaches values ('83821', 'CS-319', '2', 'Spring', '2018');

insert into teaches values ('98345', 'EE-181', '1', 'Spring', '2017');

insert into student values ('00128', 'Zhang', 'Comp. Sci.', '102');

insert into student values ('12345', 'Shankar', 'Comp. Sci.', '32');

insert into student values ('19991', 'Brandt', 'History', '80');

insert into student values ('23121', 'Chavez', 'Finance', '110');

insert into student values ('44553', 'Peltier', 'Physics', '56');

insert into student values ('45678', 'Levy', 'Physics', '46');

insert into student values ('54321', 'Williams', 'Comp. Sci.', '54');

insert into student values ('55739', 'Sanchez', 'Music', '38');

insert into student values ('70557', 'Snow', 'Physics', '0');

insert into student values ('76543', 'Brown', 'Comp. Sci.', '58');

insert into student values ('76653', 'Aoi', 'Elec. Eng.', '60');

insert into student values ('98765', 'Bourikas', 'Elec. Eng.', '98');

insert into student values ('98988', 'Tanaka', 'Biology', '120');

insert into takes values ('00128', 'CS-101', '1', 'Fall', '2017', 'A');

insert into takes values ('00128', 'CS-347', '1', 'Fall', '2017', 'A-');

insert into takes values ('12345', 'CS-101', '1', 'Fall', '2017', 'C');

insert into takes values ('12345', 'CS-190', '2', 'Spring', '2017', 'A');

insert into takes values ('12345', 'CS-315', '1', 'Spring', '2018', 'A');

insert into takes values ('12345', 'CS-347', '1', 'Fall', '2017', 'A');

insert into takes values ('19991', 'HIS-351', '1', 'Spring', '2018', 'B');

insert into takes values ('23121', 'FIN-201', '1', 'Spring', '2018', 'C+');

insert into takes values ('44553', 'PHY-101', '1', 'Fall', '2017', 'B-');

insert into takes values ('45678', 'CS-101', '1', 'Fall', '2017', 'F');

insert into takes values ('45678', 'CS-101', '1', 'Spring', '2018', 'B+');

insert into takes values ('45678', 'CS-319', '1', 'Spring', '2018', 'B');

insert into takes values ('54321', 'CS-101', '1', 'Fall', '2017', 'A-');

insert into takes values ('54321', 'CS-190', '2', 'Spring', '2017', 'B+');

insert into takes values ('55739', 'MU-199', '1', 'Spring', '2018', 'A-');

insert into takes values ('76543', 'CS-101', '1', 'Fall', '2017', 'A');

insert into takes values ('76543', 'CS-319', '2', 'Spring', '2018', 'A');

insert into takes values ('76653', 'EE-181', '1', 'Spring', '2017', 'C');

insert into takes values ('98765', 'CS-101', '1', 'Fall', '2017', 'C-');

insert into takes values ('98765', 'CS-315', '1', 'Spring', '2018', 'B');

insert into takes values ('98988', 'BIO-101', '1', 'Summer', '2017', 'A');

insert into takes values ('98988', 'BIO-301', '1', 'Summer', '2018', null);

insert into advisor values ('00128', '45565');

insert into advisor values ('12345', '10101');

insert into advisor values ('23121', '76543');

insert into advisor values ('44553', '22222');

insert into advisor values ('45678', '22222');

insert into advisor values ('76543', '45565');

insert into advisor values ('76653', '98345');

insert into advisor values ('98765', '98345');

insert into advisor values ('98988', '76766');

insert into prereq values ('BIO-301', 'BIO-101');

insert into prereq values ('BIO-399', 'BIO-101');

insert into prereq values ('CS-190', 'CS-101');

insert into prereq values ('CS-315', 'CS-101');

insert into prereq values ('CS-319', 'CS-101');

insert into prereq values ('CS-347', 'CS-101');

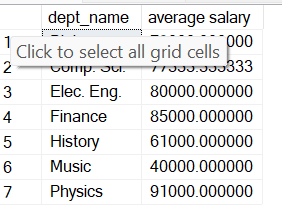
insert into prereq values ('EE-181', 'PHY-101');

Assignment Questions

**1. Display average salary given by each department.**

select dept\_name ,avg(salary) as 'average salary' from instructor group by dept\_name;

Output:



### 2. Display the name of students and their corresponding course IDs.

select s.name,t.course\_id from student s ,takes t where s.ID=t.ID;

output:



**3. Display number of courses taken by each student.**

select s.name,count(t.course\_id) from student s ,takes t where s.ID=t.ID group by s.name;

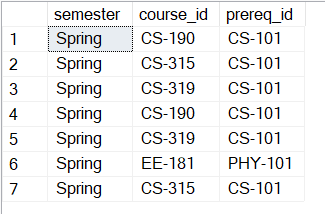
Output:

****

### 4. Get the prerequisites courses for courses in the Spring semester.

### select se.semester,se.course\_id,pr.prereq\_id from section se,prereq pr where se.course\_id=pr.course\_id;

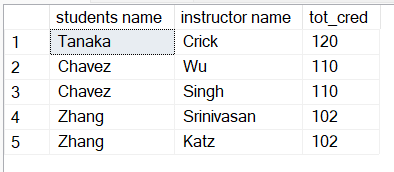
### output:

****

### 5. Display the instructor name who teaches student with highest 5 credits.

### select top 5 s.name , i.name,s.tot\_cred from student s, instructor i where s.dept\_name=i.dept\_name order by s.tot\_cred desc;

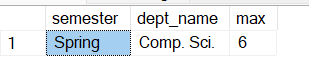
### Output:

****

### 6. Which semester and department offers maximum number of courses.

### select top 1 se.semester , c.dept\_name , count(c.course\_id) as max from course c,section se where c.course\_id=se.course\_id group by se.semester,c.dept\_name order by count(c.course\_id) desc;

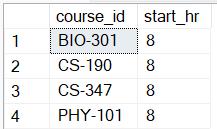
### Output:

****

### 7. Display course and department whose time starts at 8.

### select se.course\_id,ts.start\_hr from time\_slot ts ,section se where se.time\_slot\_id=ts.time\_slot\_id and start\_hr=8 group by se.course\_id,ts.start\_hr;

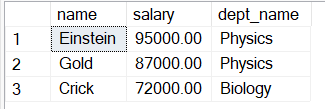
### Output:

****

### 8. Display the salary of instructors from Watson building.

### select name,salary,dept\_name from instructor where dept\_name in (select dept\_name from department where building = 'Watson');

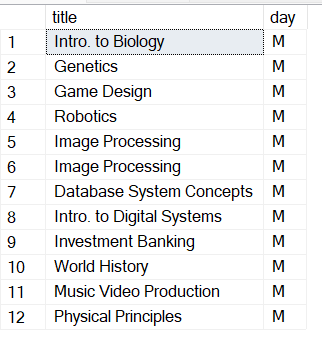
### Output:

****

### 9. Show the title of courses available on Monday.

### select title,day from course c,time\_slot ts,section s where s.time\_slot\_id=ts.time\_slot\_id and s.course\_id=c.course\_id and ts.day='M';

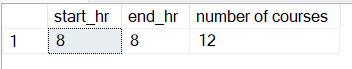
### output:

****

### 10. Find the number of courses that start at 8 and end at 8.

### select start\_hr,end\_hr,count(course\_id) as 'no of course' from time\_slot ts,section s where ts.time\_slot\_id=s.time\_slot\_id and ts.start\_hr=8 and ts.end\_hr=8 group by start\_hr,end\_hr;

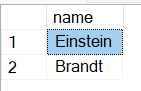
### Output:

****

### 11. Find instructors having salary more than 90000.

select name,salary from instructor where salary > 90000;

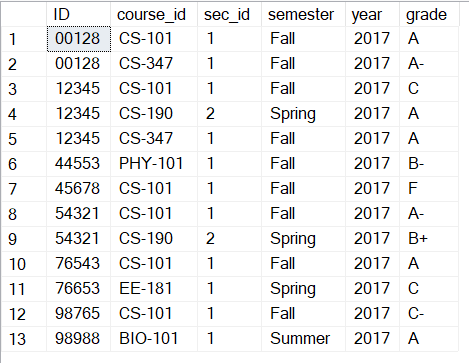
Output:

****

### 12. Find student records taking courses before 2018.

### select \* from takes where year < 2018;

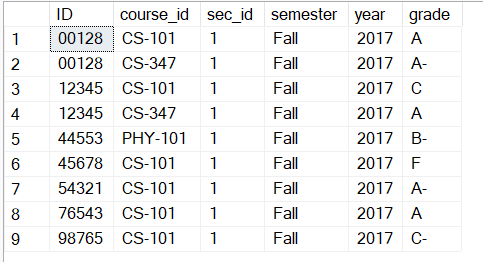
### Output:

****

### 13. Find student records taking courses in the fall semester and coming under first section.

### select \* from takes where semester='fall' and sec\_id=1;

### Output:

****

### 14. Find student records taking courses in the fall semester and coming under second section.

### select \* from takes where semester='fall' and sec\_id=2;

### Output:

****

**15. Find student records taking courses in the summer semester, coming under first section in the year 2017.**

select \* from takes where semester='summer' and sec\_id=1 and year=2017;

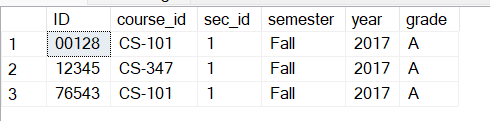
Output:

****

### 16. Find student records taking courses in the fall semester and having A grade.

### select \* from takes where semester='fall' and grade='A';

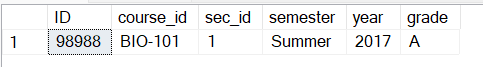
### Output:

****

### 17. Find student records taking courses in the summer semester and having A grade.

### select \* from takes where semester='summer' and grade='A';

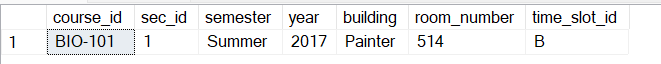
### Output:

****

### 18. Display section details with B time slot, room number 514 and in the Painter building.

### select \* from section where time\_slot\_id='B' and room\_number=514 and building='Painter';

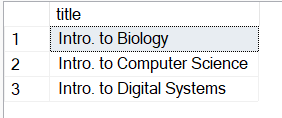
### Output:

****

**19. Find all course titles which have a string "Intro.".**

select title from course where title like '%Intro.%';

Output:

****

### 20. Find the titles of courses in the Computer Science department that have 3 credits.

### select title from course where course\_id like 'CS%' and credits=3;

### Output:

### 

### 21. Find IDs and titles of all the courses which were taught by an instructor named Einstein. Make sure there are no duplicates in the result.

select course\_id,name from course,instructor where course.dept\_name=instructor.dept\_name and name='Einstein' ;

Output:

### 

### 22. Find all course IDs which start with CS

### select course\_id from course where course\_id like 'CS%';

### Output:

### 

**23. For each department, find the maximum salary of instructors in that department.**

select dept\_name, max(salary) from instructor group by dept\_name;

Output:

### 

**24. Find the enrollment (number of students) of each section that was offered in Fall 2017.**

select sec\_id ,count(ID) as 'No of students',semester,year from takes where year=2017 and semester = 'Fall' group by sec\_id,semester,year;

Output:

### 

### 25. Increase(update) the salary of each instructor by 10% if their current salary is between 0 and 90000.

### update instructor set salary=salary\*1.1 where salary between 0 and 90000;

### Output:

### 

### select name, salary from instructor;

### 

### 26. Find the names of instructors from Biology department having salary more than 50000.

### select name ,salary from instructor where dept\_name='Biology' and salary >50000;

### Output:

### 

### 27. Find the IDs and titles of all courses taken by a student named Shankar.

### select s.name,t.course\_id,c.title from student s inner join takes t on s.ID=t.ID inner join course c on t.course\_id=c.course\_id where s.name='Shankar';

### Output:

### 

### 28. For each department, find the total credit hours of courses in that department.

### select dept\_name , sum(credits) as 'Total no of credits' from course group by dept\_name;

### Output:

### 

### 29. Find the number of courses having A grade in each building.

select s.building ,count(t.course\_id),t.grade from section s,takes t where s.course\_id=t.course\_id and t.grade='A' group by building,grade;

Output:

### 

### 30. Display number of students in each department having total credits divisible by course credits.

select s.dept\_name,count(s.ID) as 'No of students' from student s,course c where s.dept\_name=c.dept\_name and s.tot\_cred % c.credits=0 group by s.dept\_name,c.dept\_name;

Output:

### 

### 31. Display number of courses available in each building.

### select building,count(course\_id) as 'No of cources' from department d,course c where c.dept\_name=d.dept\_name group by building;

### Output:

### 

### 32. Find number of instructors in each department having 'a' and 'e' in their name.

### select dept\_name, count(name) from instructor where name like '%a%e%' group by dept\_name;

### Output:

### 

### 33. Display number of courses being taught in classroom having capacity more than 20.

### select s.room\_number,c.capacity,count(s.course\_id) as 'No of cources' from section s,classroom c where s.room\_number=c.room\_number and c.capacity>20 group by c.room\_number,s.room\_number,c.capacity;

### 

### 34. Update the budget of each department by Rs. 1000

### update department set budget=budget+1000;

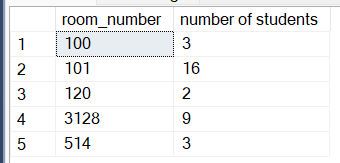
### Output:

### 

### 35. Find number of students in each room.

### select s.room\_number,count(t.ID) as 'No of students' from section s,takes t where s.course\_id=t.course\_id group by s.room\_number;

### Output:

****

### 36. Give the prerequisite course for each student.

### select s.name,p.prereq\_id from takes t,student s,prereq p where s.ID=t.ID and t.course\_id=p.course\_id ;

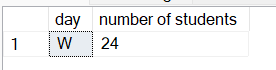
### Output:

****

### 37. Display number of students attending classes on Wednesday.

select ts.day,count(t.ID) as 'No of students' from takes t ,time\_slot ts, section s where ts.time\_slot\_id=s.time\_slot\_id and s.course\_id=t.course\_id and ts.day='W' group by ts.day;

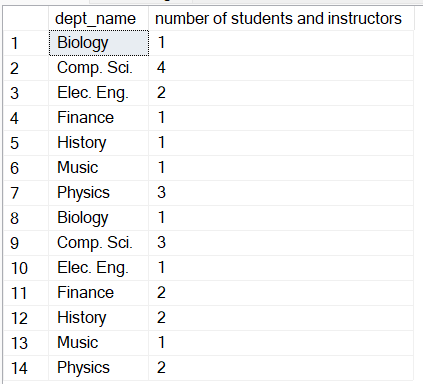
**Output:**

****

### 38. Display number of students and instructors in each department

select s.dept\_name ,count(s.ID) as 'Total num of students and Instructors' from student s group by s.dept\_name union all select i.dept\_name ,count(i.ID) as 'Total num of Instructors' from instructor i group by i.dept\_name ;

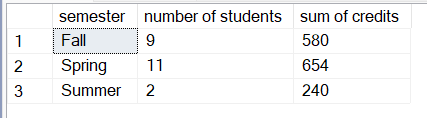
Output:



### 39. Display number of students in each semester and their sum of credits.

select t.semester,count(t.ID) as 'no of students',sum(s.tot\_cred) as 'No of credits' from takes t,student s where t.ID=s.ID group by t.semester;

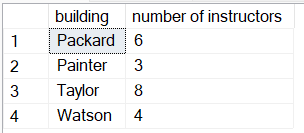
Output:



### 40. Give number of instructors in each building.

select s.building,count(t.ID) as 'no of instructors' from section s , teaches t where s.course\_id=t.course\_id group by s.building;

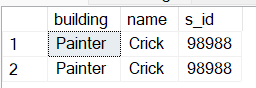
Output:



### 41. Display advisor IDs for instructors in Painter building.

select building,i.name,s\_id from advisor a,instructor i,section s,teaches t where i.ID=t.ID and t.course\_id=s.course\_id and a.i\_ID=i.ID and building='Painter';

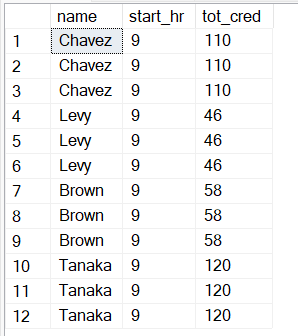
Output:



### 42. Find total credits earned by students coming at 9am

select s.name,ts.start\_hr,s.tot\_cred from student s,time\_slot ts,section se , takes t where s.ID=t.ID and t.course\_id=se.course\_id and se.time\_slot\_id=ts.time\_slot\_id and start\_hr=9 ;

Output:

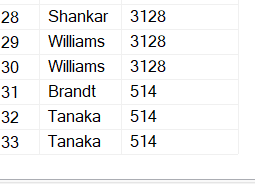


### 43. Display student names ordered by room number

### select s.name,se.room\_number from student s ,section se ,takes t where t.ID=s.ID and se.course\_id=t.course\_id order by se.room\_number;

Output:

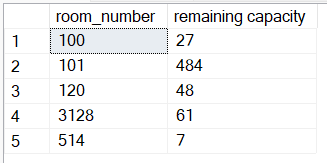




### 44. Find the number of capacity left after occupying all the students.

select cr.room\_number,cr.capacity-count(t.ID) as 'remaining capacity' from classroom cr,takes t,section s where cr.room\_number=s.room\_number and s.course\_id=t.course\_id group by cr.room\_number,cr.capacity,s.room\_number;

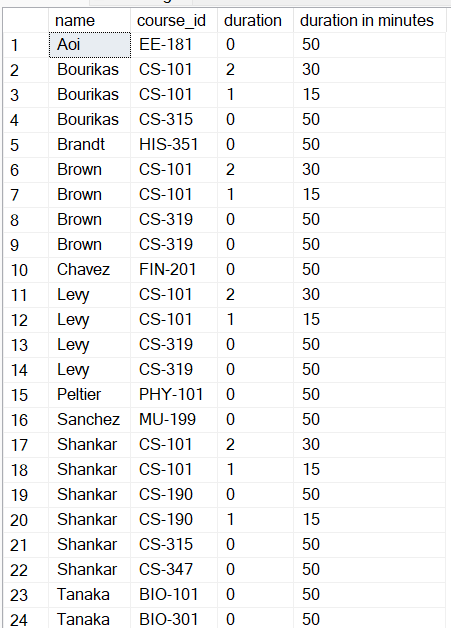
Output:

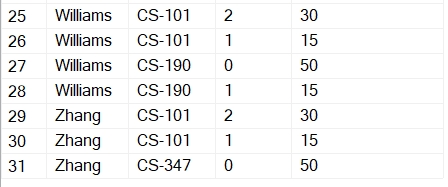


### 45. Find the duration for which each student has to attend each lecture.

select s.name,t.course\_id,end\_hr-start\_hr as 'duration',end\_min-start\_min as'duration in min' from student s,takes t , time\_slot ts,section se where s.ID=t.ID and t.course\_id=se.course\_id and se.time\_slot\_id=ts.time\_slot\_id group by s.name,t.course\_id,end\_hr,start\_hr,end\_min,start\_min;

Output:





### 46. Create a timetable for the university.

select ts.day,s.building,s.room\_number,s.course\_id from time\_slot ts , section s where ts.time\_slot\_id=s.time\_slot\_id;

Output:

### 

### 

### 47. Find the average salary that's distributed to teachers for each course and sort them in descending order

### select c.title,avg(i.salary) from instructor i , course c where c.dept\_name=i.dept\_name group by c.title order by avg(i.salary) desc;

### Output:

### 

### 48. Find the average duration of classes for each course id

### select s.course\_id , avg(ts.end\_min-ts.start\_min) as 'Duration' from section s , time\_slot ts where ts.time\_slot\_id=s.time\_slot\_id group by s.course\_id;

### Output:

### 

### 49 Get the name of the instructor with highest salary from each department.

### select dept\_name,name,max(salary) from instructor group by dept\_name ,name;

### Output:

### 

### 50. Get the sum of the total credits of students that is dealt by the instructors along with their names

with student\_advisor\_data as ( select \* from student s join advisor a on s.id=a.s\_ID)

select i.name, tot\_cred\_data.sum\_of\_credits

from ( select

i\_id, sum(tot\_cred) as 'sum\_of\_credits' from student\_advisor\_data group by i\_id) as tot\_cred\_data join instructor i

### on tot\_cred\_data.i\_id = i.id;

### Output:

### 

### 51. Perform division between student credits and department total credits

### with base as (select dept\_name,sum(credits) as credits from course group by dept\_name) select s.name,s.tot\_cred/b.credits from student s join base b on s.dept\_name=b.dept\_name;

### output:

### 

### 52. If the department budget was to be distributed among the buildings, how much amount can be allocated to each room in a building

### with building as (select building,count(room\_number)as 'num\_room' from classroom group by building) select b1.building,b1.budget/b2.num\_room from building b2 join (select building,sum(budget)as budget from department group by building) as b1 on b1.building=b2.building;

### output:

### 