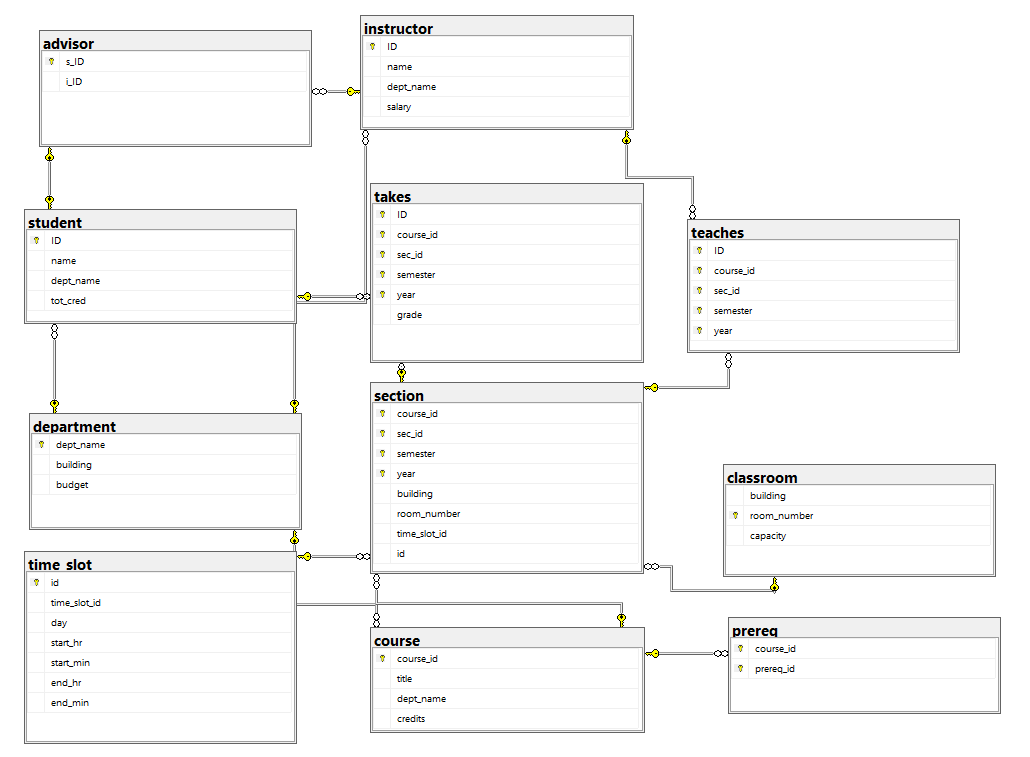
**SQL University assignment**

**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

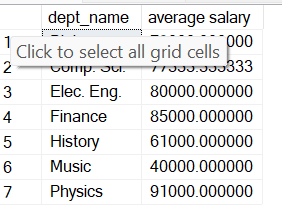
**1. Display average salary given by each department.**

SELECT dept\_name, AVG(salary) 'Average\_salary'

FROM instructor

GROUP BY dept\_name;

Output:



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

### SELECT student.name, takes.course\_id 'course\_id'

### FROM student,takes

### WHERE student.ID = takes.ID;

output:



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

SELECT student.name,COUNT(name) 'number\_of\_courses'

FROM student,takes

WHERE student.ID = takes.ID

GROUP BY name;

Output:

****

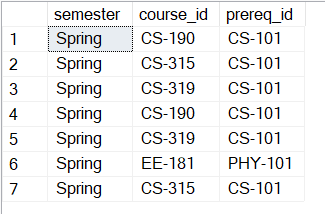
### 4. Get the prerequisites courses for courses in the Spring semester.

### SELECT t.semester,p.course\_id ,p.prereq\_id

### FROM takes t, prereq p

### WHERE p.course\_id = t.course\_id AND t.semester = 'Spring';

### output:

****

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

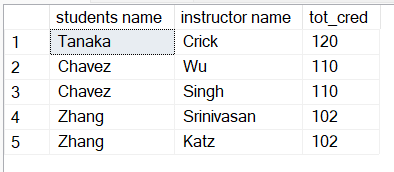
### SELECT s.name 'students\_name',i.name 'instructor\_name',s.tot\_cred

### FROM instructor i,student s

### WHERE i.dept\_name = s.dept\_name

### ORDER BY s.tot\_cred DESC LIMIT 5;

### Output:

****

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

### SELECT sec.semester , c.dept\_name, COUNT(sec.course\_id) AS 'Max'

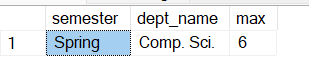
### FROM section sec , course c

### WHERE sec.course\_id = c.course\_id

### GROUP BY sec.semester,c.dept\_name

### ORDER BY Max DESC LIMIT 1;

### Output:

****

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

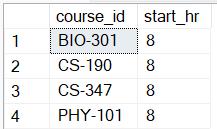
### SELECT s.course\_id,t.start\_hr

### FROM section s,time\_slot t

### WHERE s.time\_slot\_id = t.time\_slot\_id and start\_hr = 8

### GROUP BY s.course\_id,t.start\_hr;

### Output:

****

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

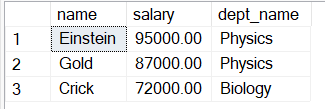
### SELECT i.name,i.salary,i.dept\_name

### FROM instructor i,department d

### WHERE i.dept\_name = d.dept\_name AND d.building = 'Watson'

### ORDER BY i.salary DESC;

### Output:

****

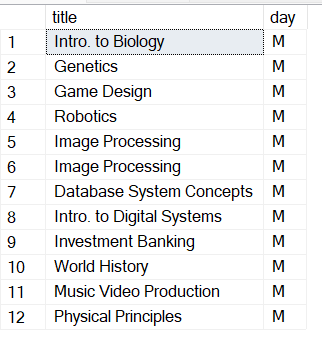
### 9. Show the title of courses available on Monday.

### SELECT c.title, t.day

### FROM course c , time\_slot t, section s

### WHERE c.course\_id = s.course\_id AND s.time\_slot\_id = t.time\_slot\_id AND t.day = 'M'

### output:

****

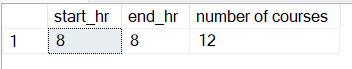
### 10. Find the number of courses that start at 8 and end at 8.

### SELECT t.start\_hr , t.end\_hr,COUNT(c.course\_id) 'number\_of\_courses'

### FROM course c, time\_slot t, section s

### WHERE c.course\_id = s.course\_id AND s.time\_slot\_id = t.time\_slot\_id AND t.start\_hr = 8 AND t.end\_hr = 8;

### Output:

****

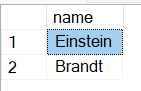
### 11. Find instructors having salary more than 90000.

SELECT name

FROM instructor

WHERE salary >90000;

Output:

****

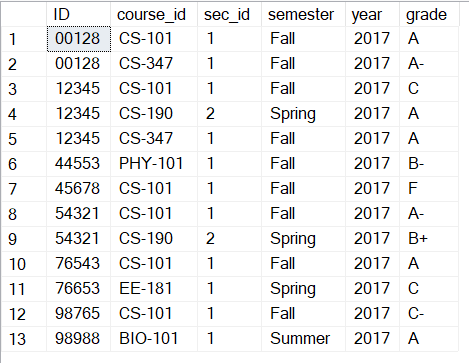
### 12. Find student records taking courses before 2018.

### SELECT t.ID,t.course\_id,t.sec\_id,t.semester,t.year,t.grade

### FROM takes t

### WHERE t.year < 2018;

### Output:

****

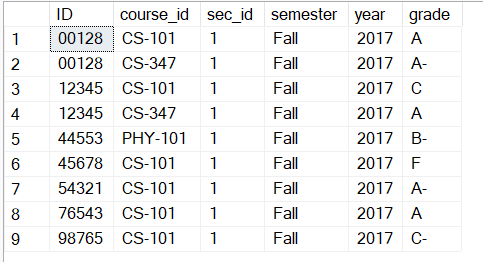
### 13. Find student records taking courses in the fall semester and coming under first section.

### SELECT ID, course\_id,sec\_id,semester,year, grade

### 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 ID, course\_id,sec\_id,semester,year, grade

### 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 ID, course\_id,sec\_id,semester,year, grade

FROM takes

WHERE semester = 'summer' AND sec\_id = 1 AND year = 2017;

Output:

****

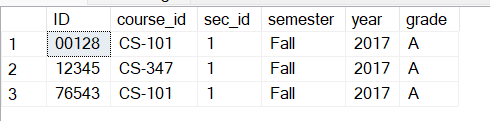
### 16. Find student records taking courses in the fall semester and having A grade.

### SELECT ID, course\_id,sec\_id,semester,year, grade

### FROM takes

### WHERE semester = 'Fall' AND grade = 'A';

### Output:

****

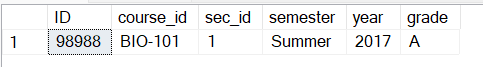
### 17. Find student records taking courses in the summer semester and having A grade.

### SELECT ID, course\_id,sec\_id,semester,year, grade

### FROM takes

### WHERE semester = 'Summer' AND grade = 'A';

### Output:

****

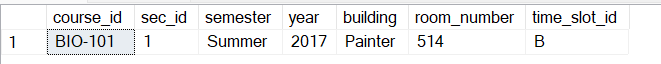
### 18. Display section details with B time slot, room number 514 and in the Painter building.

### SELECT course\_id,sec\_id,semester,year,building,room\_number,time\_slot\_id

### FROM section

### WHERE time\_slot\_id = 'B' AND room\_number = 514 AND building = 'Painter';

### Output:

****

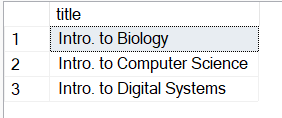
**19. Find all course titles which have a string "Intro.".**

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 dept\_name = 'Comp. Sci.' 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 c.course\_id,c.title

FROM course c,instructor i

where c.dept\_name = i.dept\_name AND i.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) as 'maximum 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) 'number\_of\_students',semester,year

FROM takes

WHERE semester = 'Fall' AND year = 2017

GROUP BY sec\_id,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 + salary \* 0.10

### 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 name ,course\_id,title

### FROM(

### SELECT s.name,t.course\_id,c.title

### FROM student s , takes t , course c

### WHERE s.ID = t.ID AND t.course\_id = c.course\_id

### ) AS new\_t

### WHERE name = 'Shankar';

### Output:

### 

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

### SELECT dept\_name,SUM(credits)

### FROM course c

### GROUP BY dept\_name

### Output:

### 

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

SELECT s.building, COUNT(s.course\_id), t.grade

FROM section s, takes t

WHERE s.course\_id = t.course\_id AND t.grade = 'A'

GROUP BY s.building

Output:

### 

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

SELECT s.dept\_name,COUNT(s.name)

FROM student s,course c

WHERE s.dept\_name = c.dept\_name AND s.tot\_cred%c.credits = 0

GROUP BY s.dept\_name

Output:

### 

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

### SELECT d.building 'building', COUNT(c.dept\_name) 'number\_of\_course'

### FROM department d, course c

### WHERE d.dept\_name = c.dept\_name

### GROUP BY d.building

### Output:

### 

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

### SELECT dept\_name ,COUNT(name) 'number\_of\_instructor'

### 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 c.room\_number,c.capacity,COUNT(s.course\_id) 'number of courses'

### FROM classroom c

### INNER JOIN section s

### ON c.room\_number = s.room\_number

### WHERE c.capacity >20

### GROUP BY c.room\_number

### 

### 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.

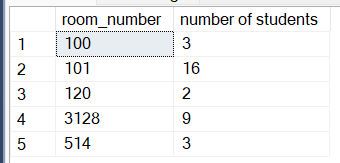
### SELECT s.room\_number, COUNT(t.ID) 'number\_of\_students'

### FROM section s, takes t

### WHERE s.course\_id = t.course\_id

### GROUP BY room\_number;

### Output:

****

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

### SELECT s.name,p.prereq\_id

### FROM student s,takes t,prereq p

### WHERE s.ID = t.ID AND t.course\_id = p.course\_id

### Output:

****

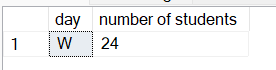
### 37. Display number of students attending classes on Wednesday.

SELECT ts.day,COUNT(t.ID) 'number\_of\_students'

FROM takes t, section s,time\_slot ts

WHERE t.course\_id = s.course\_id AND s.time\_slot\_id = ts.time\_slot\_id AND ts.day = 'W'

**Output:**

****

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

SELECT s.dept\_name,COUNT(s.ID)

FROM student s

GROUP BY s.dept\_name

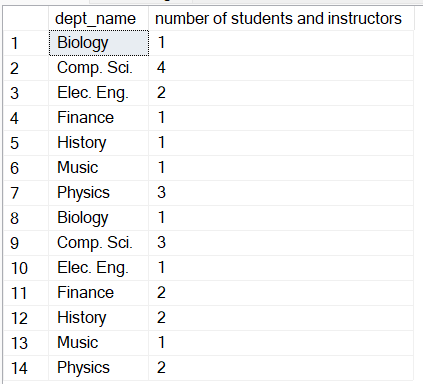
UNION ALL

SELECT i.dept\_name, COUNT(i.ID)

FROM instructor i

GROUP BY i.dept\_name

Output:



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

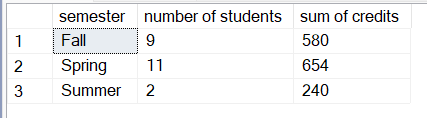
SELECT t.semester ,COUNT(s.ID) 'number of students',SUM(s.tot\_cred) 'sum of credits'

FROM takes t, student s

WHERE t.ID = s.ID

GROUP BY t.semester

Output:



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

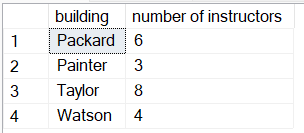
SELECT s.building,COUNT(i.ID) 'number\_of\_instructors'

FROM instructor i,teaches t,section s

WHERE t.ID = i.ID AND t.course\_id = s.course\_id

GROUP BY s.building

Output:



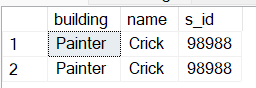
### 41. Display advisor IDs for instructors in Painter building.

SELECT s.building,i.name,a.s\_ID

FROM advisor a, teaches t,instructor i,section s

WHERE i.ID = t.ID AND t.course\_id = s.course\_id AND a.i\_ID = i.ID AND s.building = 'Painter';

Output:



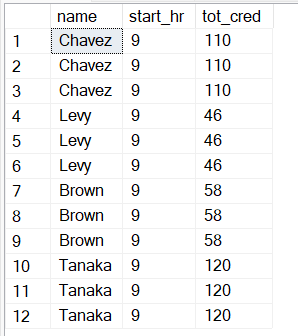
### 42. Find total credits earned by students coming at 9am

SELECT s.name,ts.start\_hr, s.tot\_cred

FROM student s,time\_slot ts, section sec ,takes t

WHERE s.ID = t.ID AND t.course\_id = sec.course\_id AND sec.time\_slot\_id = ts.time\_slot\_id AND ts.start\_hr = 9;

Output:



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

### SELECT s.name,sec.room\_number

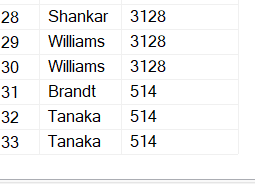
### FROM student s,section sec,takes t

### WHERE s.ID = t.ID AND t.course\_id = sec.course\_id

### ORDER BY sec.room\_number

Output:





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

SELECT sec.room\_number,cL.capacity - COUNT(s.name) 'remaining\_capacity'

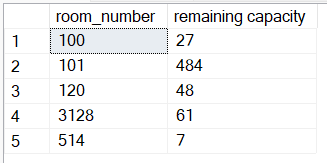
FROM student s,section sec,takes t,classroom cl

WHERE s.ID = t.ID AND t.course\_id = sec.course\_id AND sec.room\_number = cl.room\_number

GROUP BY sec.room\_number

ORDER BY sec.room\_number

Output:



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

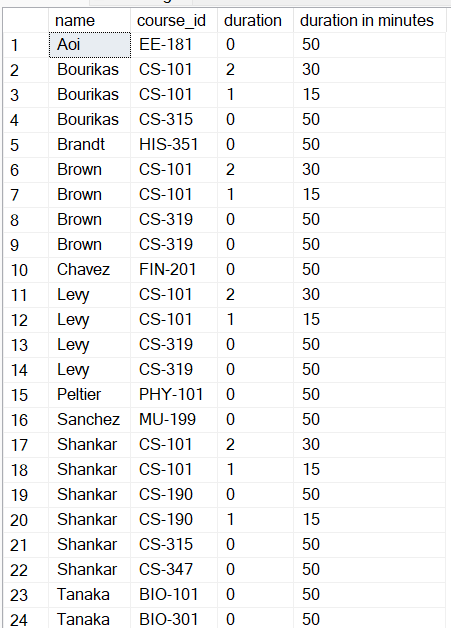
SELECT st.name,t.course\_id,ts.end\_hr - ts.start\_hr 'duration', ts.end\_min - ts.start\_min 'duration\_in\_minutes'

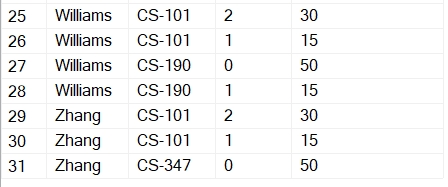
FROM student st,takes t, time\_slot ts,section s

WHERE st.ID = t.ID AND s.course\_id = t.course\_id AND s.time\_slot\_id = ts.time\_slot\_id

GROUP BY st.name,t.course\_id,ts.end\_hr,ts.start\_hr,ts.end\_min ,ts.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 'course\_name',AVG(i.salary) 'avg\_salary'

### FROM course c, instructor i,teaches t

### WHERE i.ID = t.ID AND t.course\_id = c.course\_id

### GROUP BY c.title

### ORDER BY AVG(i.salary) DESC

### Output:

### 

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

### with time\_slot\_duration AS

### (

### SELECT time\_slot\_id, end\_min - start\_min 'duration'

### FROM time\_slot

### )

### SELECT s.course\_id, AVG (tsd.duration)

### FROM time\_slot\_duration tsd, section s

### WHERE tsd.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, salary

### FROM instructor

### ORDER BY name

### Output:

### 

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

### SELECT i.name, SUM(st.tot\_cred) 'sum\_of\_credits'

### FROM instructor i, student st,advisor a

### WHERE a.i\_ID = i.ID AND a.s\_ID = st.ID

### GROUP BY i.name

### Output:

### 

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

### with dept\_creds AS

### (

### SELECT dept\_name,SUM(credits) 'credits'

### FROM course

### GROUP BY dept\_name

### )

### SELECT st.name, st.tot\_cred / dc.credits

### FROM student st, dept\_creds dc

### WHERE dC.dept\_name = st.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\_room\_data AS

### (

### SELECT building ,count(room\_number) 'num\_rooms'

### FROM classroom

### GROUP BY building

### )

### SELECT brd.building, SUM(d.budget)/num\_rooms 'room\_budget'

### from building\_room\_data brd , department d

### WHERE d.building = brd.building

### GROUP BY brd.building

### output:

### 