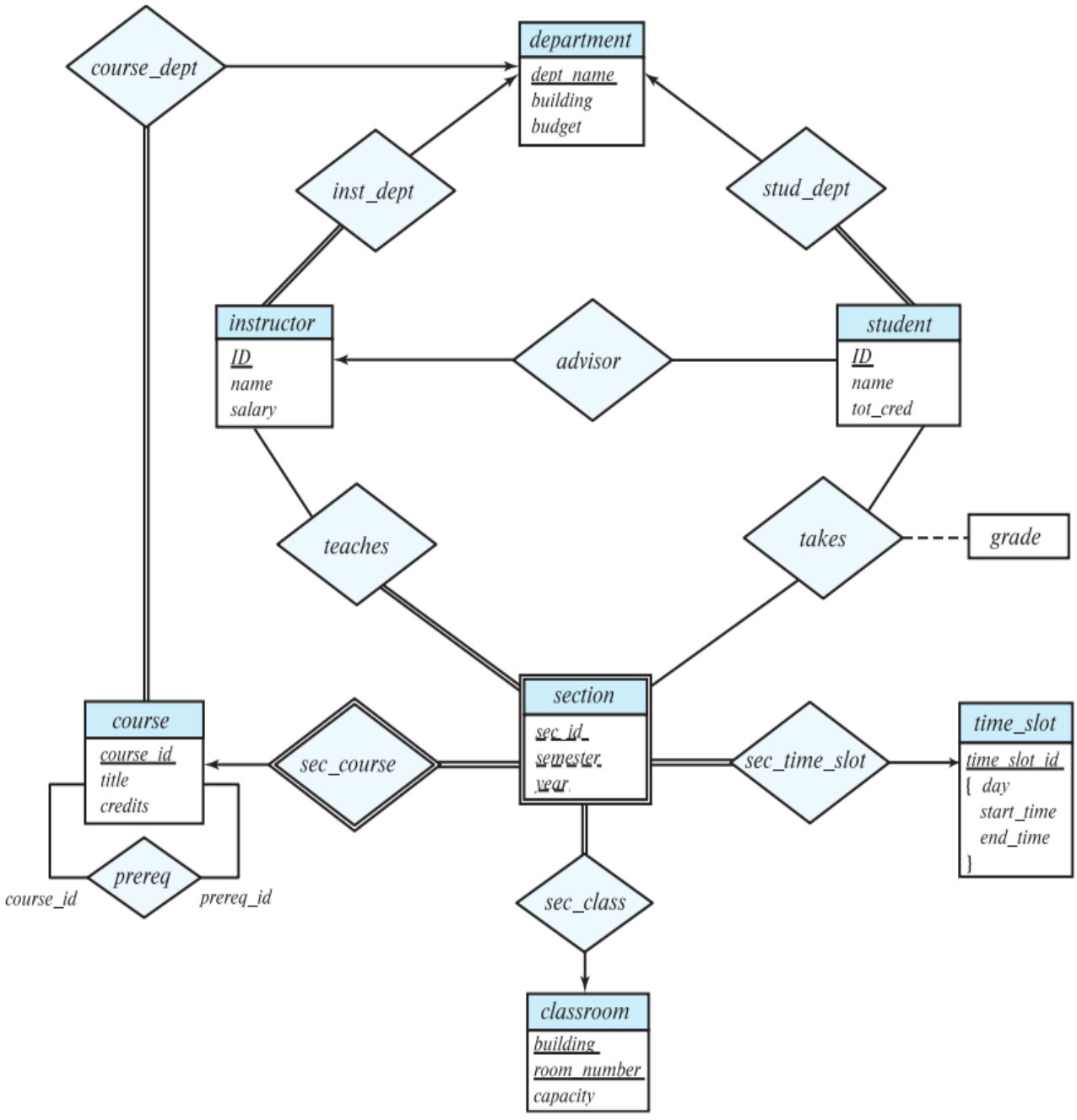
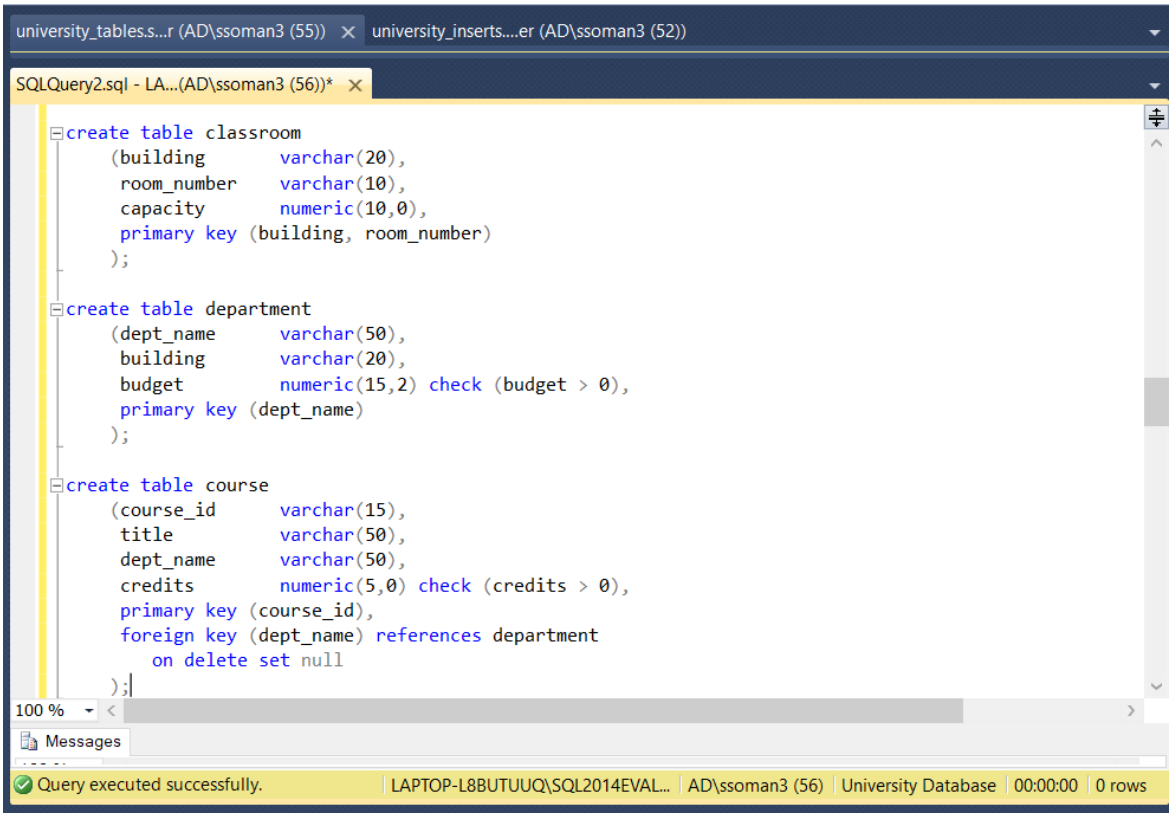
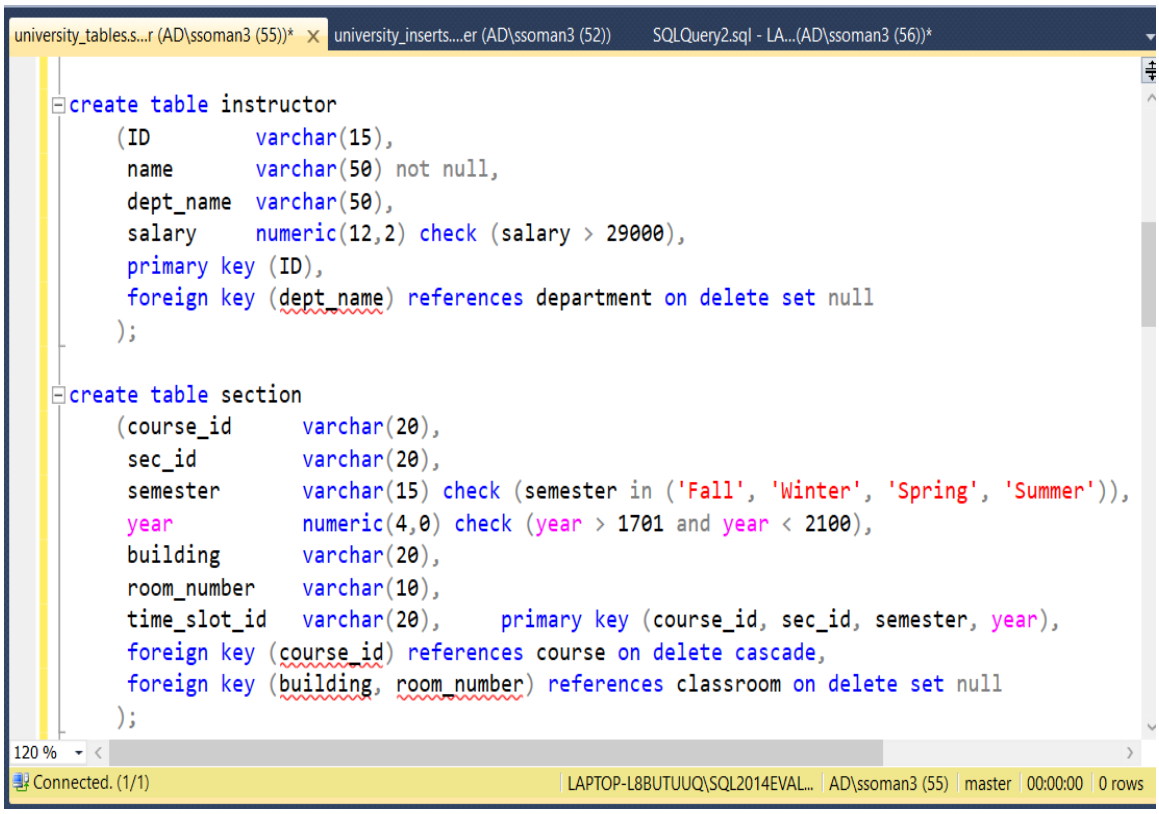
UNIVERSITY ENTERPRISE



CREATION OF TABLES



```
create table classroom
    (building       varchar(20),
     room_number    varchar(10),
     capacity       numeric(10,0),
     primary key (building, room_number)
    );

create table department
    (dept_name      varchar(50),
     building       varchar(20),
     budget         numeric(15,2) check (budget > 0),
     primary key (dept_name)
    );

create table course
    (course_id      varchar(15),
     title          varchar(50),
     dept_name      varchar(50),
     credits        numeric(5,0) check (credits > 0),
     primary key (course_id),
     foreign key (dept_name) references department
        on delete set null
    );
```



```
create table instructor
    (ID         varchar(15),
     name       varchar(50) not null,
     dept_name  varchar(50),
     salary     numeric(12,2) check (salary > 29000),
     primary key (ID),
     foreign key (dept_name) references department on delete set null
    );

create table section
    (course_id      varchar(20),
     sec_id         varchar(20),
     semester       varchar(15) check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
     year           numeric(4,0) check (year > 1701 and year < 2100),
     building       varchar(20),
     room_number    varchar(10),
     time_slot_id   varchar(20),     primary key (course_id, sec_id, semester, year),
     foreign key (course_id) references course on delete cascade,
     foreign key (building, room_number) references classroom on delete set null
    );
```

university_tables.s...r (AD\ssoman3 (55))* | university_inserts....er (AD\ssoman3 (52)) | SQLQuery2.sql - LA...(AD\ssoman3 (56))*

```
create table teaches
    (ID             varchar(20),
     course_id      varchar(20),
     sec_id         varchar(20),
     semester       varchar(10),
     year           numeric(4,0),
     primary key (ID, course_id, sec_id, semester, year),
     foreign key (course_id,sec_id, semester, year) references section on delete cascade,
     foreign key (ID) references instructor on delete cascade
    );

create table student
    (ID             varchar(20),
     name           varchar(50) not null,
     dept_name      varchar(50),
     tot_cred       numeric(5,0) check (tot_cred >= 0),
     primary key (ID),
     foreign key (dept_name) references department on delete set null
    );
```

120 %

Connected. (1/1) | LAPTOP-L8BUTUUQ\SQL2014EVAL... | AD\ssoman3 (55) | master | 00:00:00 | 0 rows

university_tables.s...r (AD\ssoman3 (55))* | university_inserts....er (AD\ssoman3 (52)) | SQLQuery2.sql - LA...(AD\ssoman3 (56))*

```
create table takes
    (ID             varchar(20),
     course_id      varchar(20),
     sec_id         varchar(20),
     semester       varchar(10),
     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 on delete cascade,
     foreign key (ID) references student on delete cascade
    );

create table advisor
    (s_ID           varchar(20),
     i_ID           varchar(20),
     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
    );
```

120 %

Connected. (1/1) | LAPTOP-L8BUTUUQ\SQL2014EVAL... | AD\ssoman3 (55) | master | 00:00:00 | 0 rows

university_tables.s...r (AD\ssoman3 (55))* | university_inserts....er (AD\ssoman3 (52)) | SQLQuery2.sql - LA...(AD\ssoman3 (56))*

```
create table time_slot
    (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),
     primary key (time_slot_id, day, start_hr, start_min)
    );

create table prereq
    (course_id      varchar(20),
     prereq_id      varchar(20),
     primary key (course_id, prereq_id),
     foreign key (course_id) references course on delete cascade,
     foreign key (prereq_id) references course
    );
```

120 %

Connected. (1/1) | LAPTOP-L8BUTUUQ\SQL2014EVAL... | AD\ssoman3 (55) | master | 00:00:00 | 0 rows

INSERTION OF DATA

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
insert into classroom values ('Lecture_Center', 'A', '120');
insert into classroom values ('Lecture_Center', 'B', '100');
insert into classroom values ('Lecture_Center', 'C', '115');
insert into classroom values ('Lecture_Center', 'D', '85');
insert into classroom values ('ARC', '201', '85');
insert into classroom values ('ARC', '202', '90');
insert into classroom values ('ARC', '203', '95');
insert into classroom values ('ARC', '204', '100');
insert into classroom values ('Douglas_Hall', '101', '50');
insert into classroom values ('Douglas_Hall', '102', '90');
insert into classroom values ('Douglas_Hall', '103', '65');
insert into classroom values ('Douglas_Hall', '104', '70');
insert into classroom values ('Lincon_Hall', '301', '85');
insert into classroom values ('Lincon_Hall', '302', '45');
insert into classroom values ('Lincon_Hall', '303', '90');
insert into classroom values ('Lincon_Hall', '304', '120');
insert into classroom values ('Burnham_Hall', '001', '85');
insert into classroom values ('Burnham_Hall', '002', '65');
insert into classroom values ('Burnham_Hall', '003', '25');
insert into classroom values ('Burnham_Hall', '004', '120');
insert into classroom values ('Eng_Block', '111', '50');
insert into classroom values ('Eng_Block', '112', '80');
insert into classroom values ('Eng_Block', '113', '75');
insert into classroom values ('Eng_Block', '114', '45');


insert into department values ('IDS', 'ARC', '90000');
insert into department values ('MARKETING.', 'Douglas_Hall', '75000');
insert into department values ('ACCOUNTING', 'ARC', '85000');
insert into department values ('FINANCE', 'Lincon_Hall', '120000');
insert into department values ('BIOLOGY_SCIENCES', 'Burnham_Hall', '500000');
insert into department values ('BIOMEDICAL_ENG', 'Eng_Block', '720000');
insert into department values ('MECHANICAL_ENG', 'Eng_Block', '150000');
insert into department values ('COMPUTER_SCIENCE', 'Lincon_Hall', '150000');

insert into course values ('BIO-101', 'Intro. to Biology', 'BIOLOGY_SCIENCES', '4');
insert into course values ('BIO-301', 'Genetics', 'BIOMEDICAL_ENG', '4');
insert into course values ('BIO-399', 'Computational Biology', 'BIOMEDICAL_ENG', '3');
insert into course values ('CS-101', 'Intro. to Computer Science', 'COMPUTER_SCIENCE', '4');
insert into course values ('CS-190', 'Game Design', 'COMPUTER_SCIENCE', '2');
insert into course values ('CS-315', 'Robotics', 'COMPUTER_SCIENCE', '2');
insert into course values ('CS-319', 'Image Processing', 'COMPUTER_SCIENCE', '3');
insert into course values ('IDS-200', 'Intro. to MIS', 'IDS', '4');
insert into course values ('IDS-532', 'Intro. to Operations Management', 'IDS', '4');
insert into course values ('IDS-270', 'Business Statistics', 'IDS', '4');
insert into course values ('IDS-572', 'Data Mining for Business', 'IDS', '4');
```

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
insert into instructor values ('10101', 'Bhatt', 'COMPUTER_SCIENCE', '65000');
insert into instructor values ('12121', 'Lee', 'FINANCE', '90000');
insert into instructor values ('15151', 'Hamid', 'MECHANICAL_ENG', '80000');
insert into instructor values ('22222', 'Singh', 'MECHANICAL_ENG', '95000');
insert into instructor values ('32343', 'Choi', 'IDS', '60000');
insert into instructor values ('33456', 'Yang', 'BIOMEDICAL_ENG', '87000');
insert into instructor values ('45565', 'Gupta', 'COMPUTER_SCIENCE', '75000');
insert into instructor values ('58583', 'Azad', 'IDS', '62000');
insert into instructor values ('76543', 'Agrawal', 'FINANCE', '80000');
insert into instructor values ('76766', 'Cooper', 'BIOMEDICAL_ENG', '72000');
insert into instructor values ('83821', 'Moseby', 'COMPUTER_SCIENCE', '92000');
insert into instructor values ('98345', 'Geller', 'BIOLOGY_SCIENCES', '80000');

insert into section values ('BIO-101', '1', 'Fall', '2022', 'Burnham_Hall', '001', 'B');
insert into section values ('BIO-301', '1', 'Spring', '2023', 'Burnham_Hall', '002', 'A');
insert into section values ('BIO-399', '1', 'Spring', '2023', 'Burnham_Hall', '002', 'F');
insert into section values ('CS-101', '1', 'Fall', '2022', 'Lincon_Hall', '301', 'H');
insert into section values ('CS-190', '2', 'Fall', '2022', 'Lincon_Hall', '304', 'B');
insert into section values ('CS-190', '2', 'Spring', '2023', 'Lincon_Hall', '302', 'E');
insert into section values ('CS-315', '1', 'Fall', '2022', 'Lincon_Hall', '302', 'D');
insert into section values ('CS-319', '1', 'Fall', '2022', 'Lincon_Hall', '303', 'B');
insert into section values ('CS-319', '2', 'Spring', '2023', 'Lincon_Hall', '303', 'C');
insert into section values ('FIN-200', '1', 'Fall', '2022', 'ARC', '201', 'C');
insert into section values ('FIN-200', '1', 'Spring', '2023', 'ARC', '201', 'C');
insert into section values ('FIN-302', '2', 'Fall', '2022', 'Lincon_Hall', '101', 'B');
insert into section values ('FIN-303', '2', 'Spring', '2023', 'Douglas_Hall', '104', 'C');
insert into section values ('FIN-320', '1', 'Spring', '2023', 'Lincon_Hall', '301', 'D');
insert into section values ('FIN-330', '2', 'Fall', '2022', 'Lincon_Hall', '304', 'A');
insert into section values ('FIN-419', '2', 'Spring', '2023', 'ARC', '202', 'B');
insert into section values ('IDS-200', '1', 'Fall', '2022', 'ARC', '201', 'A');
insert into section values ('IDS-270', '1', 'Spring', '2023', 'ARC', '201', 'A');
insert into section values ('IDS-521', '2', 'Fall', '2022', 'Lecture_Center', 'A', 'A');
insert into section values ('IDS-521', '1', 'Spring', '2023', 'Lecture_Center', 'A', 'A');
insert into section values ('IDS-532', '1', 'Fall', '2022', 'Lecture_Center', 'C', 'A');
insert into section values ('IDS-572', '2', 'Spring', '2023', 'Lecture_Center', 'D', 'G');
insert into section values ('IDS-575', '2', 'Fall', '2022', 'ARC', '202', 'F');
insert into section values ('IDS-594', '2', 'Spring', '2023', 'Lecture_Center', 'B', 'D');
insert into section values ('ME-101', '1', 'Fall', '2022', 'Eng_Block', '112', 'C');
insert into section values ('ME-101', '1', 'Spring', '2023', 'Eng_Block', '112', 'C');
insert into section values ('ME-205', '2', 'Fall', '2022', 'Eng_Block', '113', 'D');
insert into section values ('ME-211', '2', 'Spring', '2023', 'Eng_Block', '114', 'F');
insert into section values ('ME-220', '1', 'Spring', '2023', 'Eng_Block', '111', 'A');
insert into section values ('ME-306', '2', 'Fall', '2022', 'Eng_Block', '112', 'H');
```

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
insert into teaches values ('12121', 'FIN-200', '1', 'Spring', '2023');
insert into teaches values ('12121', 'FIN-200', '1', 'Fall', '2022');
insert into teaches values ('12121', 'FIN-303', '2', 'Spring', '2023');
insert into teaches values ('76543', 'FIN-302', '2', 'Fall', '2022');
insert into teaches values ('76543', 'FIN-320', '1', 'Spring', '2023');
insert into teaches values ('76543', 'FIN-330', '2', 'Fall', '2022');
insert into teaches values ('76543', 'FIN-419', '2', 'Spring', '2023');
insert into teaches values ('15151', 'ME-101', '1', 'Spring', '2023');
insert into teaches values ('15151', 'ME-101', '1', 'Fall', '2022');
insert into teaches values ('22222', 'ME-205', '2', 'Fall', '2022');
insert into teaches values ('22222', 'ME-211', '2', 'Spring', '2023');
insert into teaches values ('22222', 'ME-306', '2', 'Fall', '2022');
insert into teaches values ('22222', 'ME-220', '1', 'Spring', '2023');
insert into teaches values ('32343', 'IDS-521', '1', 'Spring', '2023');
insert into teaches values ('32343', 'IDS-521', '2', 'Fall', '2022');
insert into teaches values ('32343', 'IDS-200', '1', 'Fall', '2022');
insert into teaches values ('32343', 'IDS-594', '2', 'Spring', '2023');
insert into teaches values ('58583', 'IDS-270', '1', 'Spring', '2023');
insert into teaches values ('58583', 'IDS-532', '1', 'Fall', '2022');
insert into teaches values ('58583', 'IDS-572', '2', 'Spring', '2023');
insert into teaches values ('58583', 'IDS-575', '2', 'Fall', '2022');
insert into teaches values ('98345', 'BIO-101', '1', 'Fall', '2022');
insert into teaches values ('33456', 'BIO-301', '1', 'Spring', '2023');
insert into teaches values ('76766', 'BIO-399', '1', 'Spring', '2023');
insert into teaches values ('10101', 'CS-101', '1', 'Fall', '2022');
insert into teaches values ('10101', 'CS-315', '1', 'Fall', '2022');
insert into teaches values ('83821', 'CS-190', '2', 'Fall', '2022');
insert into teaches values ('83821', 'CS-190', '2', 'Spring', '2023');
insert into teaches values ('83821', 'CS-319', '2', 'Spring', '2023');
insert into teaches values ('45565', 'CS-319', '1', 'Fall', '2022');

insert into student values ('00128', 'Somani', 'IDS', '40');
insert into student values ('12345', 'Singh', 'ACCOUNTING', '36');
insert into student values ('19991', 'Sharma', 'FINANCE', '30');
insert into student values ('45678', 'Kubal', 'MARKETING.', '40');
insert into student values ('70557', 'Shukla', 'IDS', '56');
insert into student values ('76653', 'Kumar', 'ACCOUNTING', '32');
insert into student values ('98765', 'Sethi', 'FINANCE', '30');
insert into student values ('98988', 'Niar', 'IDS', '36');
insert into student values ('55739', 'Hegde', 'COMPUTER_SCIENCE', '38');
insert into student values ('76543', 'Abbas', 'COMPUTER_SCIENCE', '34');
insert into student values ('23121', 'Rai', 'MECHANICAL_ENG', '32');
insert into student values ('20202', 'Raj', 'MECHANICAL_ENG', '46');
insert into student values ('44553', 'Pachpor', 'BIOMEDICAL_ENG', '48');
insert into student values ('54321', 'Mistry', 'BIOLOGY_SCIENCES', '50');
```

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

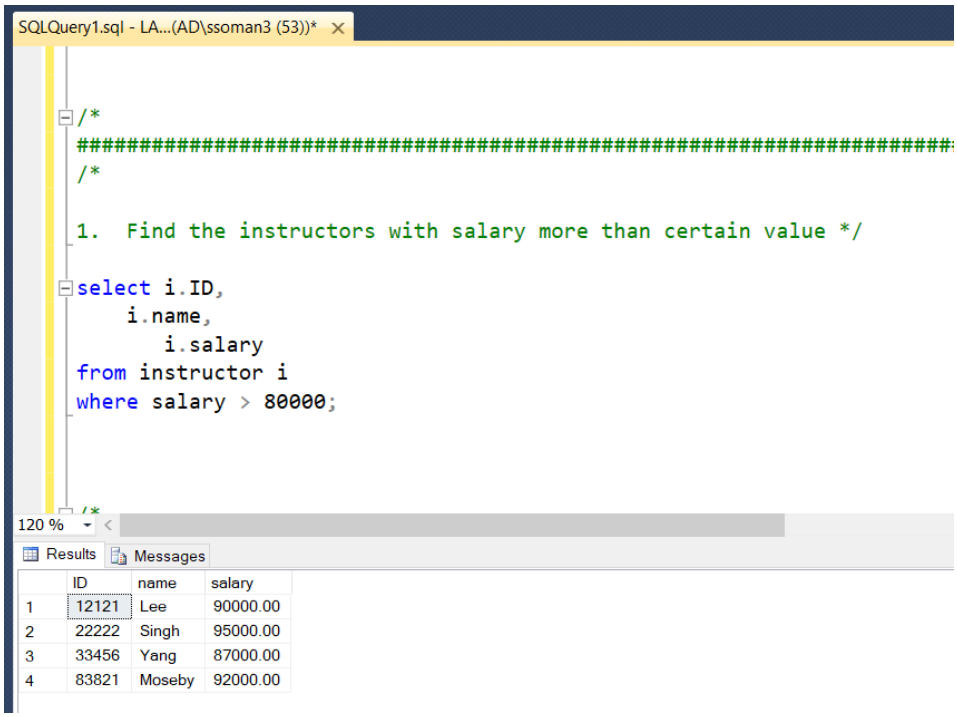
```
insert into takes values ('45678', 'FIN-200', '1', 'Fall', '2022', 'B');
insert into takes values ('45678', 'IDS-270', '1', 'Spring', '2023', 'B');
insert into takes values ('45678', 'IDS-200', '1', 'Fall', '2022', 'C');
insert into takes values ('45678', 'IDS-521', '1', 'Spring', '2023', 'A');
insert into takes values ('70557', 'FIN-302', '2', 'Fall', '2022', 'F');
insert into takes values ('70557', 'FIN-419', '2', 'Spring', '2023', 'B');
insert into takes values ('70557', 'IDS-532', '1', 'Fall', '2022', 'A');
insert into takes values ('70557', 'IDS-200', '1', 'Fall', '2022', 'B');
insert into takes values ('70557', 'IDS-572', '2', 'Spring', '2023', 'C');
insert into takes values ('76653', 'IDS-200', '1', 'Fall', '2022', 'B');
insert into takes values ('76653', 'IDS-575', '2', 'Fall', '2022', 'C');
insert into takes values ('76653', 'IDS-594', '2', 'Spring', '2023', 'B');
insert into takes values ('76653', 'FIN-330', '2', 'Fall', '2022', 'A');
insert into takes values ('76653', 'FIN-320', '1', 'Spring', '2023', 'A');
insert into takes values ('98765', 'FIN-200', '1', 'Fall', '2022', 'A');
insert into takes values ('98765', 'FIN-330', '2', 'Fall', '2022', 'C');
insert into takes values ('98765', 'IDS-270', '1', 'Spring', '2023', 'B');
insert into takes values ('98765', 'FIN-419', '2', 'Spring', '2023', 'A');
insert into takes values ('98765', 'IDS-575', '2', 'Fall', '2022', 'F');
insert into takes values ('98988', 'IDS-200', '1', 'Fall', '2022', 'B');
insert into takes values ('98988', 'FIN-302', '2', 'Fall', '2022', 'A');
insert into takes values ('98988', 'IDS-594', '2', 'Spring', '2023', 'C');
insert into takes values ('98988', 'IDS-532', '1', 'Fall', '2022', 'B');
insert into takes values ('98988', 'IDS-270', '1', 'Spring', '2023', 'A');
insert into takes values ('76543', 'CS-101', '1', 'Fall', '2022', 'B');
insert into takes values ('76543', 'CS-190', '2', 'Fall', '2022', 'C');
insert into takes values ('76543', 'CS-315', '1', 'Fall', '2022', 'B');
insert into takes values ('76543', 'CS-319', '2', 'Spring', '2023', 'B');
insert into takes values ('55739', 'CS-101', '1', 'Fall', '2022', 'A');
insert into takes values ('55739', 'CS-190', '2', 'Spring', '2023', 'B');
insert into takes values ('55739', 'CS-315', '1', 'Fall', '2022', 'A');
insert into takes values ('55739', 'CS-319', '2', 'Spring', '2023', 'A');
insert into takes values ('23121', 'ME-101', '1', 'Fall', '2022', 'A');
insert into takes values ('23121', 'ME-205', '2', 'Fall', '2022', 'B');
insert into takes values ('23121', 'ME-211', '2', 'Spring', '2023', 'C');
insert into takes values ('23121', 'ME-220', '1', 'Spring', '2023', 'B');
insert into takes values ('23121', 'ME-306', '2', 'Fall', '2022', 'B');
insert into takes values ('20202', 'ME-101', '1', 'Fall', '2022', 'C');
insert into takes values ('20202', 'ME-205', '2', 'Fall', '2022', 'B');
insert into takes values ('20202', 'ME-211', '2', 'Spring', '2023', 'A');
insert into takes values ('20202', 'ME-220', '1', 'Spring', '2023', 'C');
insert into takes values ('20202', 'ME-306', '2', 'Fall', '2022', 'C');
insert into takes values ('44553', 'BIO-101', '1', 'Fall', '2022', 'A');
insert into takes values ('44553', 'BIO-301', '1', 'Spring', '2023', 'A');
insert into takes values ('44553', 'BIO-399', '1', 'Spring', '2023', 'A');
insert into takes values ('54321', 'BIO-101', '1', 'Fall', '2022', 'A');
insert into takes values ('54321', 'BIO-301', '1', 'Spring', '2023', 'B');
```

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

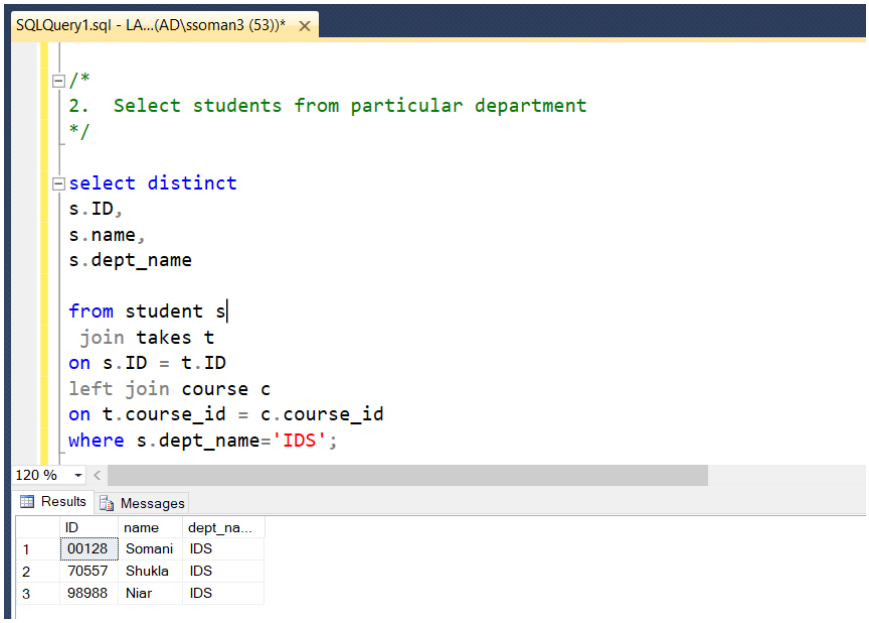
```
insert into takes values ('44553', 'BIO-301', '1', 'Spring', '2023', 'A');
insert into takes values ('44553', 'BIO-399', '1', 'Spring', '2023', 'A');
insert into takes values ('54321', 'BIO-101', '1', 'Fall', '2022', 'A');
insert into takes values ('54321', 'BIO-301', '1', 'Spring', '2023', 'B');
insert into takes values ('54321', 'BIO-399', '1', 'Spring', '2023', 'B');

insert into advisor values ('00128', '58583');
insert into advisor values ('12345', '12121');
insert into advisor values ('23121', '22222');
insert into advisor values ('20202', '15151');
insert into advisor values ('44553', '33456');
insert into advisor values ('45678', '32343');
insert into advisor values ('54321', '98345');
insert into advisor values ('76543', '45565');
insert into advisor values ('76653', '76543');
insert into advisor values ('55739', '10101');
insert into advisor values ('98765', '76543');
insert into advisor values ('98988', '58583');

insert into time_slot values ('A', 'M', '8', '0', '8', '50');
insert into time_slot values ('A', 'W', '8', '0', '8', '50');
insert into time_slot values ('B', 'M', '9', '0', '9', '50');
insert into time_slot values ('B', 'F', '9', '0', '9', '50');
insert into time_slot values ('C', 'W', '11', '0', '11', '50');
insert into time_slot values ('C', 'F', '11', '0', '11', '50');
insert into time_slot values ('D', 'M', '13', '0', '13', '50');
insert into time_slot values ('D', 'F', '13', '0', '13', '50');
insert into time_slot values ('E', 'T', '10', '30', '11', '45 ');
insert into time_slot values ('F', 'T', '14', '30', '15', '45 ');
insert into time_slot values ('F', 'R', '14', '30', '15', '45 ');
insert into time_slot values ('G', 'M', '16', '0', '16', '50');
insert into time_slot values ('G', 'F', '16', '0', '16', '50');
insert into time_slot values ('H', 'W', '10', '0', '12', '30');

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');
```

QUERIES

1. **Find the instructors with salary more than certain value.**
   Displaying a list of instructors who earn more than $8000 per year by first selecting their ID, Name and Salary and then putting up a condition stating the same.



2. **Select students from a particular department.**
   Selecting students from the 'IDS' department. It is important for to DGS to know the names of the students in their respective department.



3. For each department find the maximum salary given to an instructor in that department.

The maximum salary in each department is displayed.

```
/*
3.  For each department find the maximum salary given to an instructor in that department.
*/

select dept_name,
    max(salary)
from  instructor
group by dept_name;
```

120 %

Results | Messages

	dept_name	(No column na...
1	BIOLOGY_SCIENCES	80000.00
2	BIOMEDICAL_ENG	87000.00
3	COMPUTER_SCIENCE	92000.00
4	FINANCE	90000.00
5	IDS	62000.00
6	MECHANICAL_ENG	95000.00

4. Print the class schedule for a particular student for a given semester & year.

Students must check their class schedule on a frequent basis, and this is a necessity for all students. We can access their class schedule with complete information such as ID, Name, Year, Semester, Grade, Course, Title, Credits, Building, Room Number, Course, Day, Start and End Times.

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
4.  Print the class schedule for a particular student for a given semester & year
*/
select  st.id,st.name,t.year,t.semester,t.grade,c.course_id,c.title,c.credits,s.building,s.room_number, ts.day,ts.start_hr,ts.start_min,
        ts.end_hr,ts.end_min

from takes t
join student st
on t.ID = st.ID
join course c
on c.course_id = t.course_id
 join section s
on s.course_id = c.course_id
join time_slot ts
on s.time_slot_id = ts.time_slot_id
where st.id='20202'
and t.year = 2022;
```

100 %

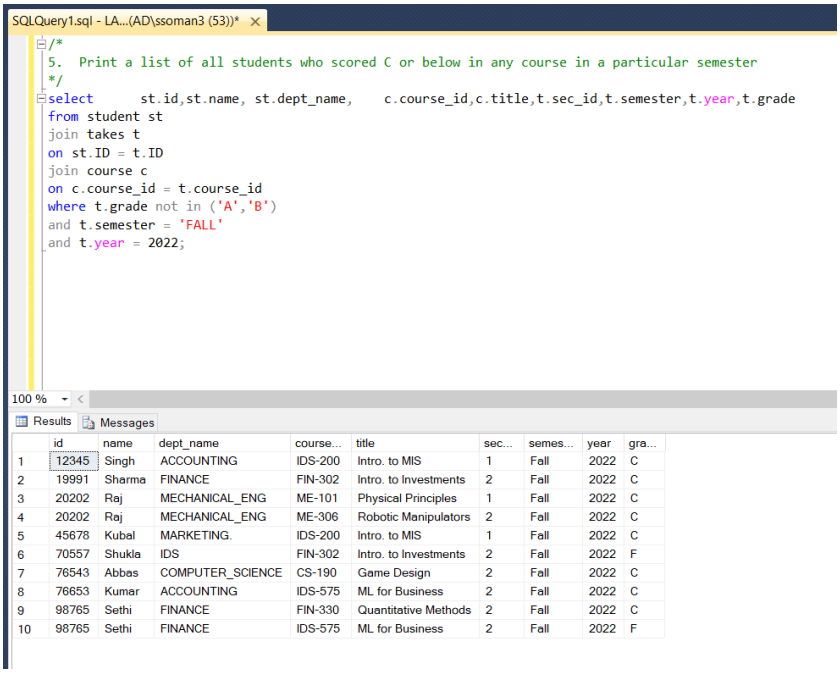
Results | Messages

	id	na...	year	semes...	gra...	course...	title	cred...	building	room_num...	d...	start...	start_...	end...	end_...
1	20202	Raj	2022	Fall	C	ME-101	Physical Principles	4	Eng_Block	112	F	11	0	11	50
2	20202	Raj	2022	Fall	C	ME-101	Physical Principles	4	Eng_Block	112	W	11	0	11	50
3	20202	Raj	2022	Fall	C	ME-101	Physical Principles	4	Eng_Block	112	F	11	0	11	50
4	20202	Raj	2022	Fall	C	ME-101	Physical Principles	4	Eng_Block	112	W	11	0	11	50
5	20202	Raj	2022	Fall	B	ME-205	Thermodynamics	4	Eng_Block	113	F	13	0	13	50
6	20202	Raj	2022	Fall	B	ME-205	Thermodynamics	4	Eng_Block	113	M	13	0	13	50
7	20202	Raj	2022	Fall	C	ME-306	Robotic Manipulators	4	Eng_Block	112	W	10	0	12	30

5. Print a list of all students who scored C or below in any course in a particular semester.

The institution wants to keep track of student progress and ensure that they receive the assistance and information they require. Students frequently take prerequisite classes for other courses, and they must maintain a minimum GPA to complete a semester. Advisors must evaluate student performance in such circumstances.

The following query will return a list of all students who received a C or lower in any of their courses in Fall 2021, along with all course data.



SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
5.  Print a list of all students who scored C or below in any course in a particular semester
*/
select      st.id,st.name, st.dept_name,    c.course_id,c.title,t.sec_id,t.semester,t.year,t.grade
from student st
join takes t
on st.ID = t.ID
join course c
on c.course_id = t.course_id
where t.grade not in ('A','B')
and t.semester = 'FALL'
and t.year = 2022;
```

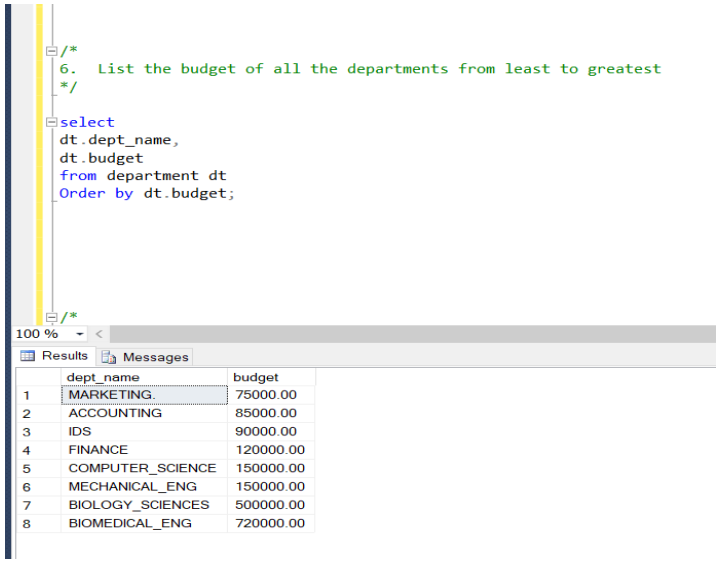
100 %

Results | Messages

	id	name	dept_name	course...	title	sec...	semes...	year	gra...
1	12345	Singh	ACCOUNTING	IDS-200	Intro. to MIS	1	Fall	2022	C
2	19991	Sharma	FINANCE	FIN-302	Intro. to Investments	2	Fall	2022	C
3	20202	Raj	MECHANICAL_ENG	ME-101	Physical Principles	1	Fall	2022	C
4	20202	Raj	MECHANICAL_ENG	ME-306	Robotic Manipulators	2	Fall	2022	C
5	45678	Kubal	MARKETING.	IDS-200	Intro. to MIS	1	Fall	2022	C
6	70557	Shukla	IDS	FIN-302	Intro. to Investments	2	Fall	2022	F
7	76543	Abbas	COMPUTER_SCIENCE	CS-190	Game Design	2	Fall	2022	C
8	76653	Kumar	ACCOUNTING	IDS-575	ML for Business	2	Fall	2022	C
9	98765	Sethi	FINANCE	FIN-330	Quantitative Methods	2	Fall	2022	C
10	98765	Sethi	FINANCE	IDS-575	ML for Business	2	Fall	2022	F

6. List the budget of all the departments from least to greatest.

We are printing all the departments in order of least to greatest budget that might help us in giving an insight into if a department is receiving adequate funding or not.



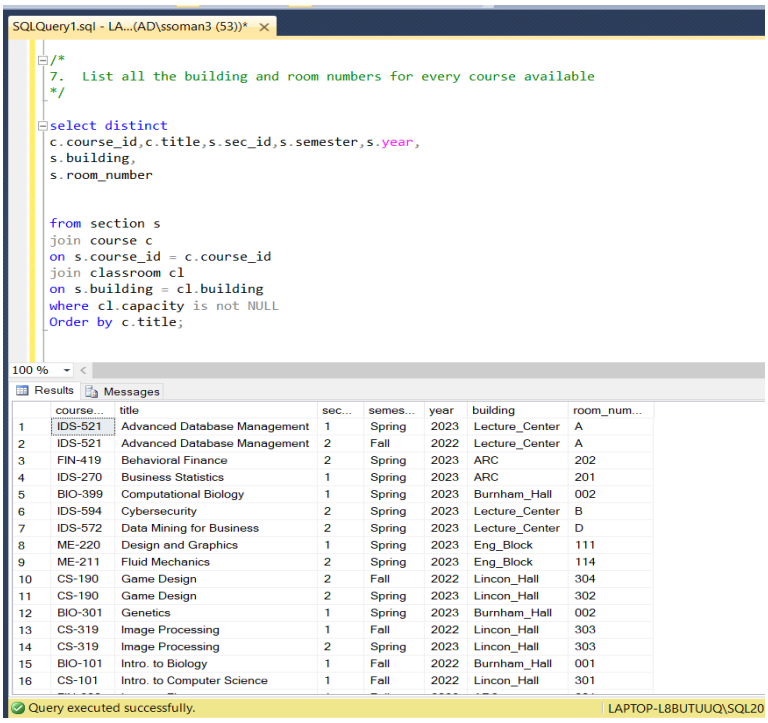
```
/*
6.  List the budget of all the departments from least to greatest
*/

select
dt.dept_name,
dt.budget
from department dt
Order by dt.budget;

/*
```

100 %

Results | Messages

	dept_name	budget
1	MARKETING.	75000.00
2	ACCOUNTING	85000.00
3	IDS	90000.00
4	FINANCE	120000.00
5	COMPUTER_SCIENCE	150000.00
6	MECHANICAL_ENG	150000.00
7	BIOLOGY_SCIENCES	500000.00
8	BIOMEDICAL_ENG	720000.00

7. List all the building and room numbers for every course available.
This can be used as a directory to assist students in locating their class.



```
/*
7.  List all the building and room numbers for every course available
*/

select distinct
c.course_id,c.title,s.sec_id,s.semester,s.year,
s.building,
s.room_number


from section s
join course c
on s.course_id = c.course_id
join classroom cl
on s.building = cl.building
where cl.capacity is not NULL
Order by c.title;
```

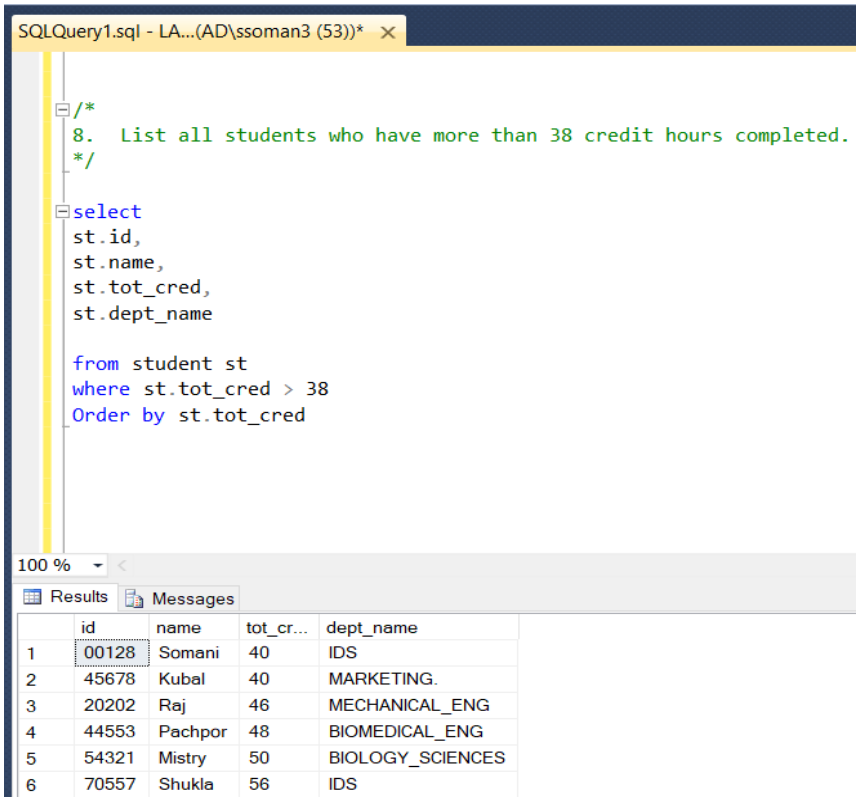
100 %

Results | Messages

	course...	title	sec...	semes...	year	building	room_num...
1	IDS-521	Advanced Database Management	1	Spring	2023	Lecture_Center	A
2	IDS-521	Advanced Database Management	2	Fall	2022	Lecture_Center	A
3	FIN-419	Behavioral Finance	2	Spring	2023	ARC	202
4	IDS-270	Business Statistics	1	Spring	2023	ARC	201
5	BIO-399	Computational Biology	1	Spring	2023	Burnham_Hall	002
6	IDS-594	Cybersecurity	2	Spring	2023	Lecture_Center	B
7	IDS-572	Data Mining for Business	2	Spring	2023	Lecture_Center	D
8	ME-220	Design and Graphics	1	Spring	2023	Eng_Block	111
9	ME-211	Fluid Mechanics	2	Spring	2023	Eng_Block	114
10	CS-190	Game Design	2	Fall	2022	Lincon_Hall	304
11	CS-190	Game Design	2	Spring	2023	Lincon_Hall	302
12	BIO-301	Genetics	1	Spring	2023	Burnham_Hall	002
13	CS-319	Image Processing	1	Fall	2022	Lincon_Hall	303
14	CS-319	Image Processing	2	Spring	2023	Lincon_Hall	303
15	BIO-101	Intro. to Biology	1	Fall	2022	Burnham_Hall	001
16	CS-101	Intro. to Computer Science	1	Fall	2022	Lincon_Hall	301

Query executed successfully. LAPTOP-L8BUTUUQ\SQL20

8. List all students who have more than 38 credit hours completed.
This can be useful to determine if the student has completed 38 hours or more than that.



```
/*
8.  List all students who have more than 38 credit hours completed.
*/

select
st.id,
st.name,
st.tot_cred,
st.dept_name

from student st
where st.tot_cred > 38
Order by st.tot_cred
```

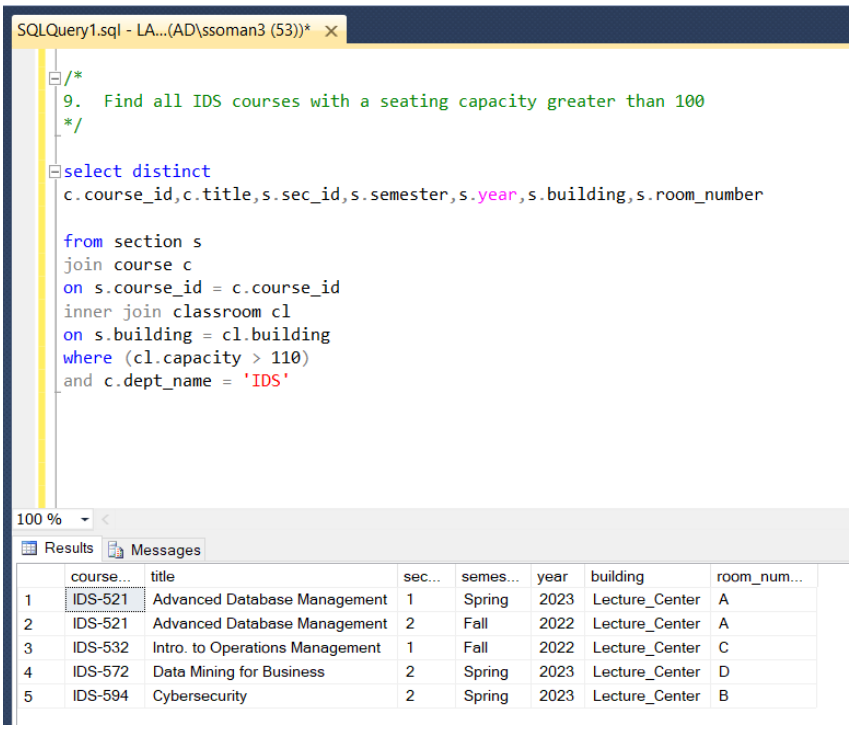
100 %

Results | Messages

	id	name	tot_cr...	dept_name
1	00128	Somani	40	IDS
2	45678	Kubal	40	MARKETING.
3	20202	Raj	46	MECHANICAL_ENG
4	44553	Pachpor	48	BIOMEDICAL_ENG
5	54321	Mistry	50	BIOLOGY_SCIENCES
6	70557	Shukla	56	IDS

9. Find all IDS courses with a seating capacity greater than 100

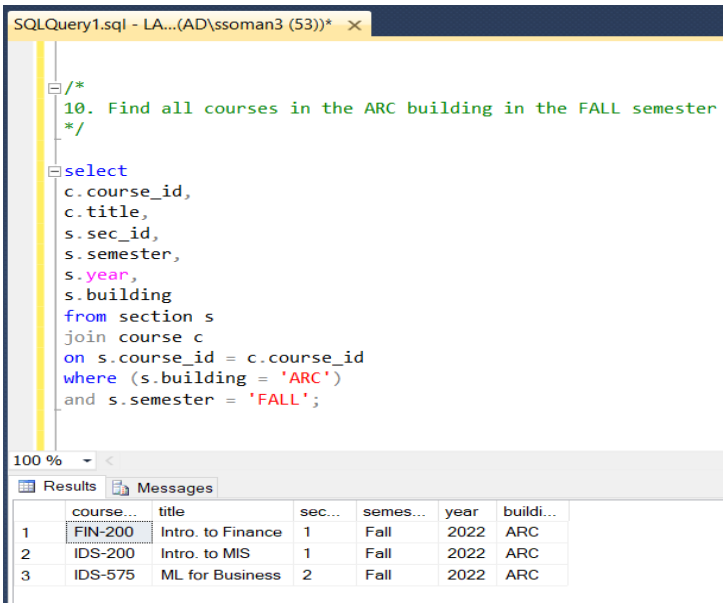
Determine which classes can accommodate more than 100 number of students.



	course...	title	sec...	semes...	year	building	room_num...
1	IDS-521	Advanced Database Management	1	Spring	2023	Lecture_Center	A
2	IDS-521	Advanced Database Management	2	Fall	2022	Lecture_Center	A
3	IDS-532	Intro. to Operations Management	1	Fall	2022	Lecture_Center	C
4	IDS-572	Data Mining for Business	2	Spring	2023	Lecture_Center	D
5	IDS-594	Cybersecurity	2	Spring	2023	Lecture_Center	B

10. Find all courses in the ARC building in the FALL semester.

This is helpful in finding all the courses that would be conducted in ARC building in Fall semester



	course...	title	sec...	semes...	year	buildi...
1	FIN-200	Intro. to Finance	1	Fall	2022	ARC
2	IDS-200	Intro. to MIS	1	Fall	2022	ARC
3	IDS-575	ML for Business	2	Fall	2022	ARC

11. Find all professors with a salary between $60000 and $79000

See which professors are getting paid between a certain range.

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
11. Find all professors with a salary between $60000 and $79000
*/

select i.id,
     i.name,
     i.salary,
     i.dept_name
from instructor i
where salary between 60000 and 79000
```

100 %

Results | Messages

	id	name	salary	dept_name
1	10101	Bhatt	65000.00	COMPUTER_SCIENCE
2	32343	Choi	60000.00	IDS
3	45565	Gupta	75000.00	COMPUTER_SCIENCE
4	58583	Azad	62000.00	IDS
5	76766	Cooper	72000.00	BIOMEDICAL_ENG

12. Update the salary of the instructors from a certain department.

This query can be used to inflict any changes in the database values. Instructors salaries, departments or other details can be updated according to the requirements.

```
/*
12. Update the salary of the instructors from a certain department to certain value
*/

update instructor
set salary = 75000
where ID = '32343';

select * from instructor
```

100 %

Results | Messages

	ID	name	dept_name	salary
1	10101	Bhatt	COMPUTER_SCIENCE	65000.00
2	12121	Lee	FINANCE	90000.00
3	15151	Hamid	MECHANICAL_ENG	80000.00
4	22222	Singh	MECHANICAL_ENG	95000.00
5	32343	Choi	IDS	75000.00
6	33456	Yang	BIOMEDICAL_ENG	87000.00
7	45565	Gupta	COMPUTER_SCIENCE	75000.00
8	58583	Azad	IDS	62000.00
9	76543	Agrawal	FINANCE	80000.00
10	76766	Cooper	BIOMEDICAL_ENG	72000.00
11	83821	Moseby	COMPUTER_SCIENCE	92000.00
12	98345	Geller	BIOLOGY_SCIENCES	80000.00

13. List all students who got A in Fall 2021

This query can be used to find list of students doing well in their respective departments.

SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
13.  List all students who got A in Fall 2021
*/

select      st.id,st.name, st.dept_name,    c.course_id,c.title,t.sec_id,t.semester,t.year,t.grade
from student st
join takes t
on st.ID = t.ID
join course c
on c.course_id = t.course_id
where t.grade = 'A'
and t.semester = 'FALL'
and t.year = 2022;
```

100 %

Results | Messages

	id	name	dept_name	course...	title	sec...	semes...	year	gra...
1	00128	Somani	IDS	FIN-200	Intro. to Finance	1	Fall	2022	A
2	00128	Somani	IDS	IDS-575	ML for Business	2	Fall	2022	A
3	12345	Singh	ACCOUNTING	IDS-575	ML for Business	2	Fall	2022	A
4	19991	Sharma	FINANCE	IDS-532	Intro. to Operations Management	1	Fall	2022	A
5	23121	Rai	MECHANICAL_ENG	ME-101	Physical Principles	1	Fall	2022	A
6	44553	Pachpor	BIOMEDICAL_ENG	BIO-101	Intro. to Biology	1	Fall	2022	A
7	54321	Mistry	BIOLOGY_SCIENCES	BIO-101	Intro. to Biology	1	Fall	2022	A
8	55739	Hegde	COMPUTER_SCIENCE	CS-101	Intro. to Computer Science	1	Fall	2022	A
9	55739	Hegde	COMPUTER_SCIENCE	CS-315	Robotics	1	Fall	2022	A
10	70557	Shukla	IDS	IDS-532	Intro. to Operations Management	1	Fall	2022	A
11	76653	Kumar	ACCOUNTING	FIN-330	Quantitative Methods	2	Fall	2022	A
12	98765	Sethi	FINANCE	FIN-200	Intro. to Finance	1	Fall	2022	A
13	98988	Niar	IDS	FIN-302	Intro. to Investments	2	Fall	2022	A

14. Print a list of sections and courses with no student enrollment

Checking if there are any sections which has no student enrollment. If there is no enrollment, the department or the program can make decisions to shift or remove the section.

HW3 - SQL Query.s...(AD\ssoman3 (51)) | SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
14. Print a list of sections and courses with no student enrollment
*/

select      s.sec_id,
            s.semester,
            s.year,
            s.course_id,
            c.title,
            c.dept_name,
            count(e.id) as student_count
from section s
left join takes e
on s.sec_id = e.sec_id
join course c
on s.course_id = c.course_id
group by s.sec_id, s.semester, s.year, s.course_id, c.title, c.dept_name
having count(e.id) < 1;
```

100 %

Results | Messages

sec_id	semester	ye...	course...	title	dept_na...	student_co...

15. Find all the courses in a department with certain number of credits

List of courses in IDS department which has 4 course credits. Since there usually a lot of courses offered by each department, this query can be used to easily filter out the courses with the desired outcomes.

```
/*
15. Find all the courses in a department with certain number of credits
*/

select title,
       credits
from course
where dept_name = 'IDS'
and credits = '4';
```

100 %

Results | Messages

	title	cred...
1	Intro. to MIS	4
2	Business Statistics	4
3	Advanced Database Management	4
4	Intro. to Operations Management	4
5	Data Mining for Business	4
6	ML for Business	4

16. Create a new course.

Listing a new course in the table and defining each parameter for that course.

HW3 - SQL Query.s...(AD\ssoman3 (51)) | SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
16. Create a new course.
*/

BEGIN TRANSACTION
INSERT INTO course (course_id, title, credits, dept_name)
VALUES ('IDS-550', 'Python Programming', '2', 'IDS');
END

select * from course
```

100 %

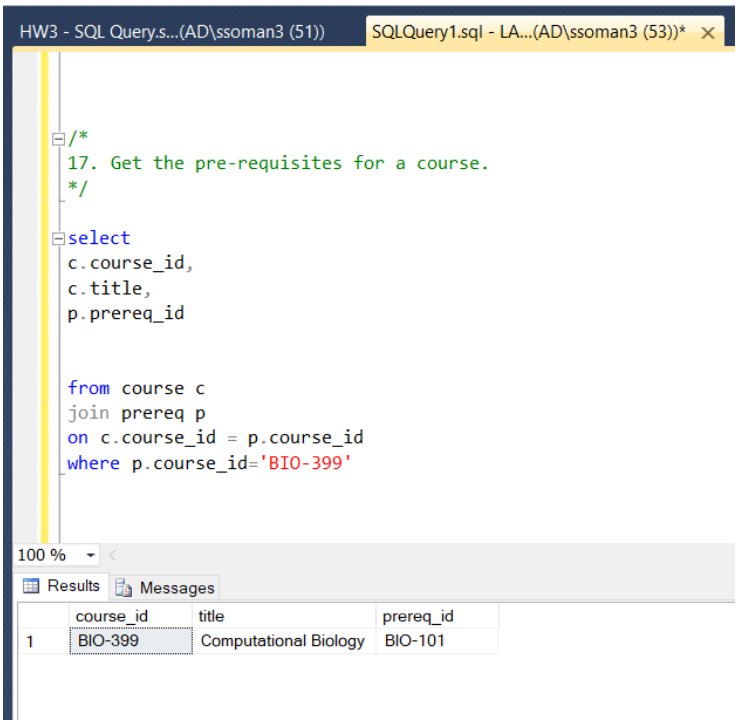
Results | Messages

	course...	title	dept_name	cred...
8	FIN-200	Intro. to Finance	FINANCE	4
9	FIN-302	Intro. to Investments	FINANCE	4
10	FIN-303	Investment Banking	FINANCE	3
11	FIN-320	Managerial Finance	FINANCE	2
12	FIN-330	Quantitative Methods	FINANCE	2
13	FIN-419	Behavioral Finance	FINANCE	2
14	IDS-200	Intro. to MIS	IDS	4
15	IDS-270	Business Statistics	IDS	4
16	IDS-521	Advanced Database Management	IDS	4
17	IDS-532	Intro. to Operations Management	IDS	4
18	IDS-550	Python Programming	IDS	2
19	IDS-572	Data Mining for Business	IDS	4
20	IDS-575	ML for Business	IDS	4
21	IDS-594	Cybersecurity	IDS	2
22	ME-101	Physical Principles	MECHANICAL_ENG	4
23	ME-205	Thermodynamics	MECHANICAL_ENG	4

Query executed successfully.

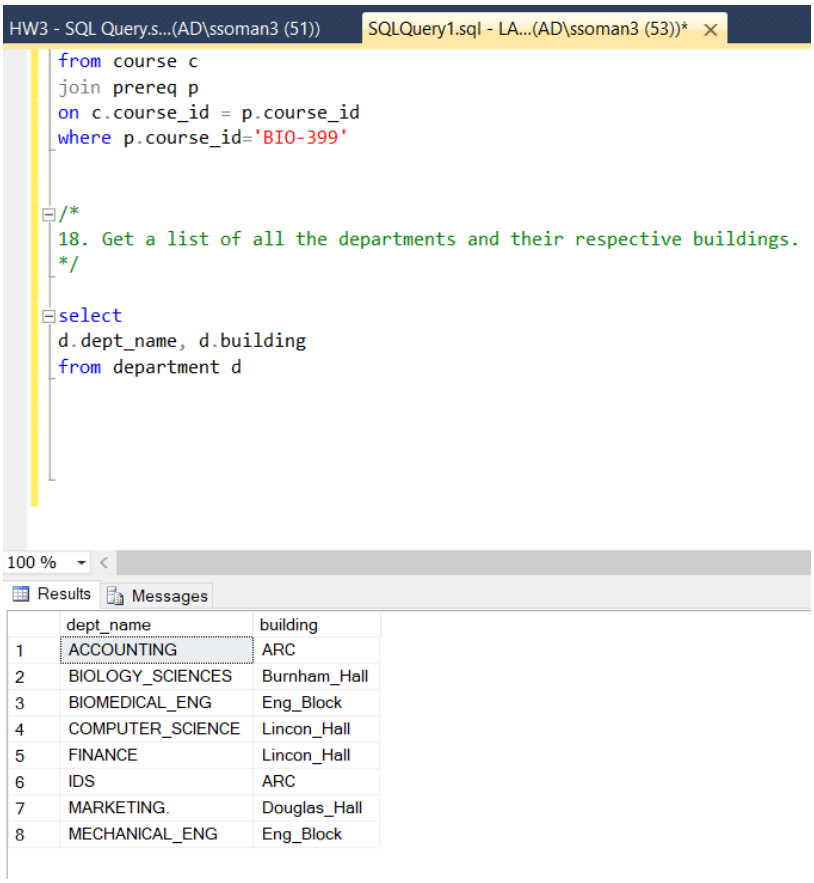
17. Get the pre-requisites for a course.

Entering the pre-requisite course and its information. It is important for students to know if they have completed the pre-requisite courses for the course which they want to take



18. Get a list of all the departments and their respective buildings.

Obtaining the list of all departments along with their respective buildings



19. For each department find the Average salary in that department

Displaying the Average salary for each department.

```
/*
19. For each department find the Average salary in that department
*/

select dept_name,
    avg(salary)
from  instructor
group by dept_name;
```

100 %

Results | Messages

	dept_name	(No column name)
1	BIOLOGY_SCIENCES	80000.000000
2	BIOMEDICAL_ENG	79500.000000
3	COMPUTER_SCIENCE	77333.333333
4	FINANCE	85000.000000
5	IDS	68500.000000
6	MECHANICAL_ENG	87500.000000

20. Select instructors from a particular department

This can be used to select instructors from a department.

HW3 - SQL Query.s...(AD\ssoman3 (51)) | SQLQuery1.sql - LA...(AD\ssoman3 (53))*

```
/*
20. Select instructors from particular department
*/

select distinct
s.ID,
s.name,
s.dept_name

from instructor s
 join teaches t
on s.ID = t.ID
left join course c
on t.course_id = c.course_id
where s.dept_name='IDS';
```

100 %

Results | Messages

	ID	name	dept_name
1	32343	Choi	IDS
2	58583	Azad	IDS