Simon Abhijet Biswas

CSE370: Database Systems

Brac University

Lab 01: Setting Up and Connecting to the MySQL Server

Lab 02: SQL Update, Delete & Basic Select Queries

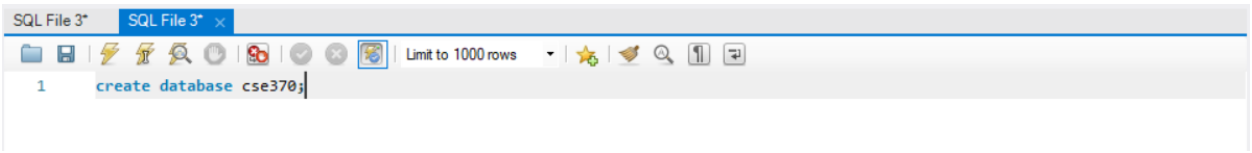
Lab 03: SQL Subqueries & Aggregate Functions

Lab 05: Introduction to Bank DB, SQL Joins and Constraints

Lab 1: Create and Insert

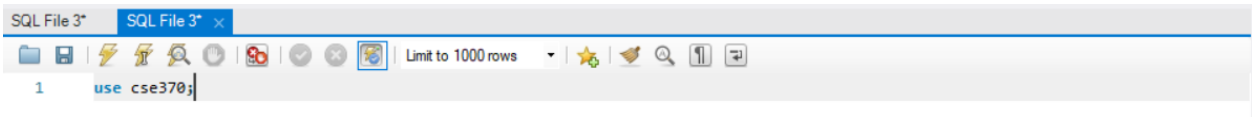
- **Create a database cse370 from Lab Grades**

create database cse370



- **Use cse370**

use cse370

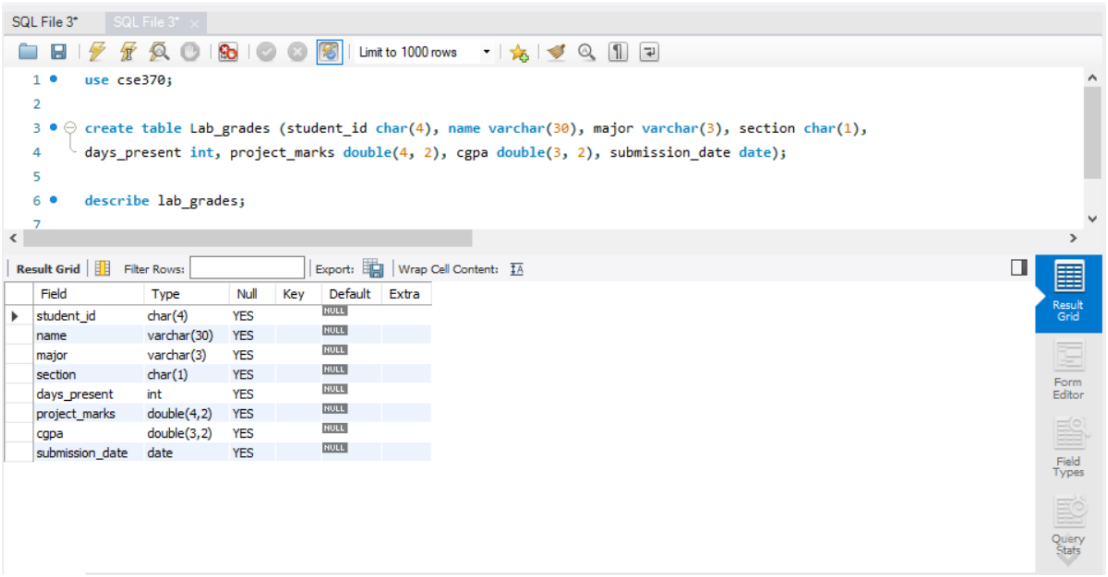


- **Create a Lab Grades table and Describe it.**

use cse370;

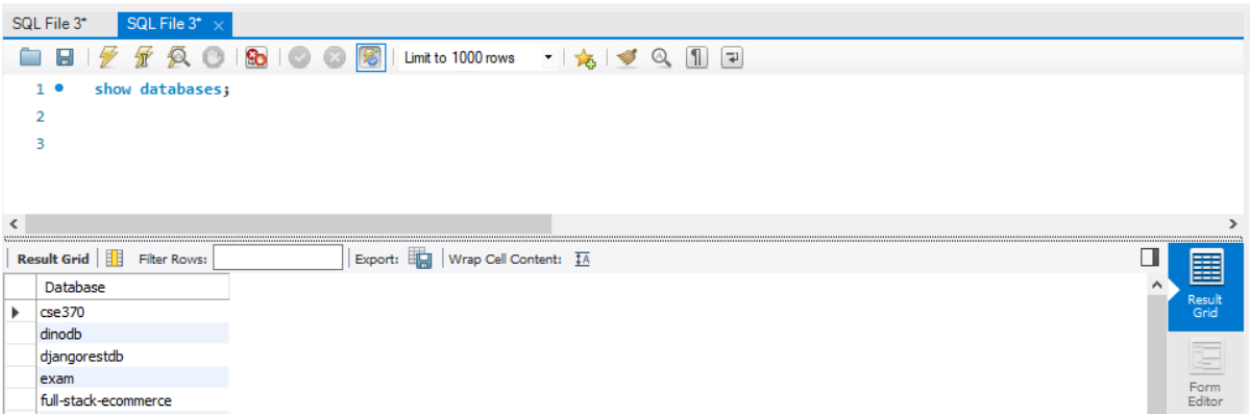
create table Lab_grades (student_id char(4), name varchar(30), major varchar(3), section char(1), days_present int, project_marks double(4, 2), cgpa double(3, 2), submission_date date);

describe lab_grades;



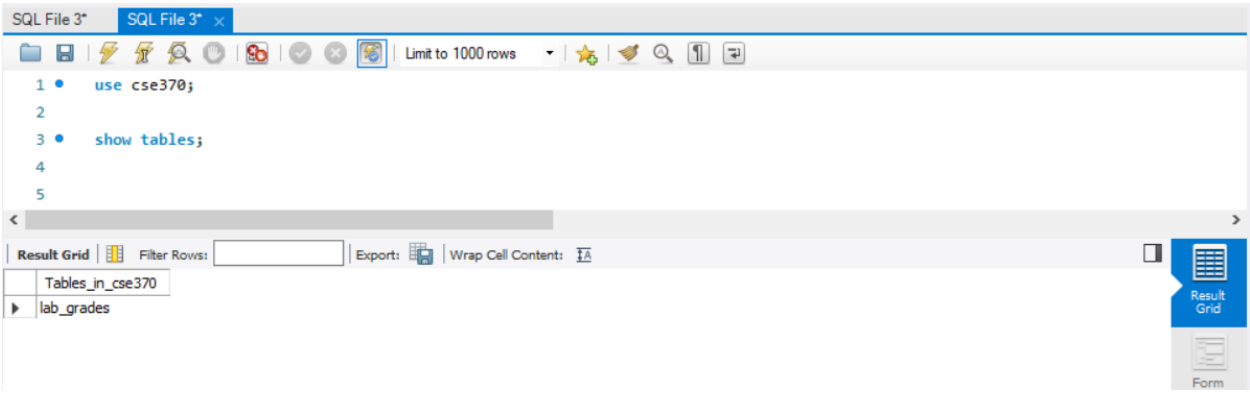
- **Show Databases and Tables**

show database;



use cse370;

show tables;



- **Insert data into the Lab Grades table**

use cse370;

insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s001', 'Abir', 'CS', '1', 10, 18.50, 3.91, '2018-09-15');

insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s002', 'Nafis', 'CSE', '1', 12, 20, 3.86, '2018-08-15');

insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s019', 'Naima', 'CSE', '2', 12, 20, 3.7, '2018-08-14');

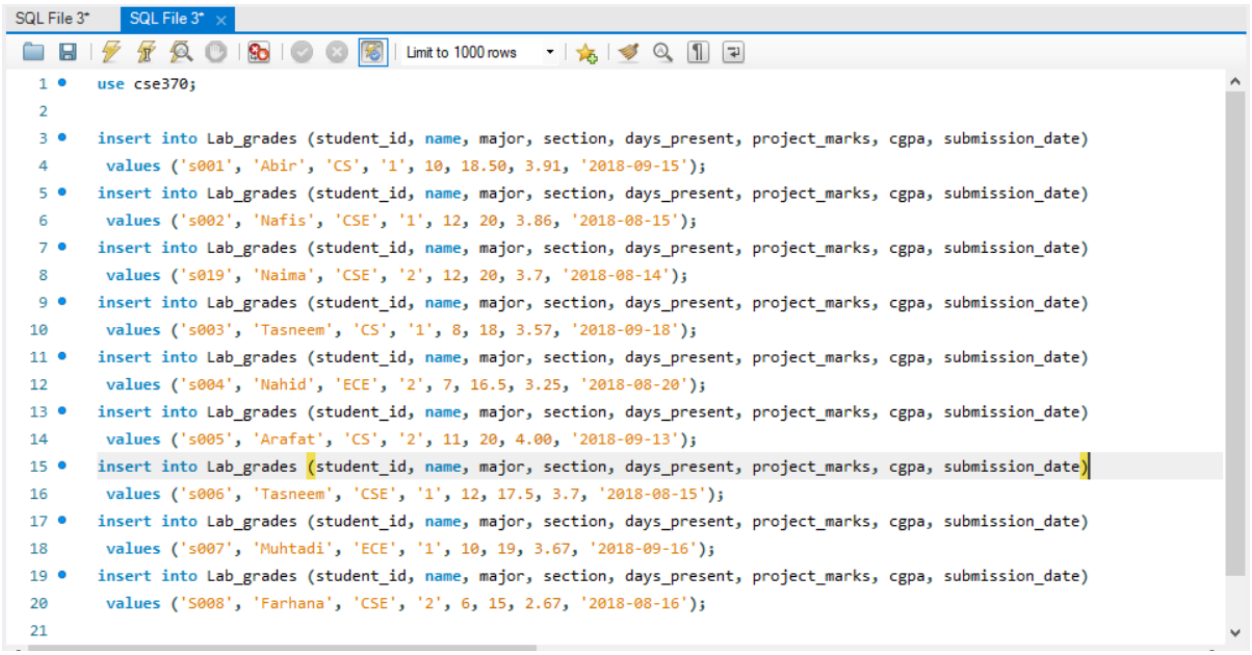
insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s003', 'Tasneem', 'CS', '1', 8, 18, 3.57, '2018-09-18');
insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s004', 'Nahid', 'ECE', '2', 7, 16.5, 3.25, '2018-08-20');

insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s005', 'Arafat', 'CS', '2', 11, 20, 4.00, '2018-09-13');

insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s006', 'Tasneem', 'CSE', '1', 12, 17.5, 3.7, '2018-08-15');

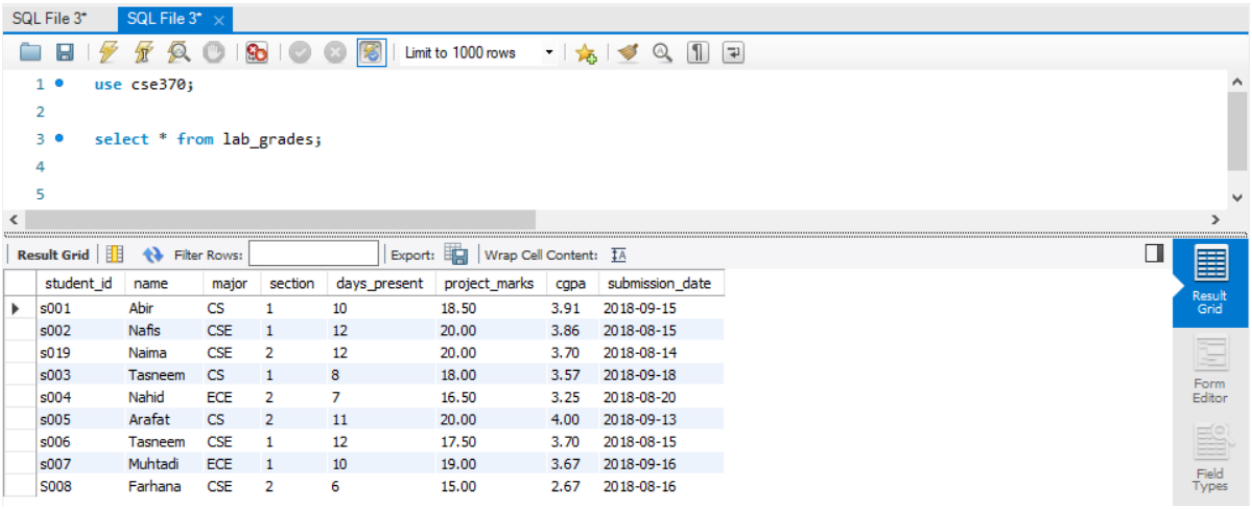
insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('s007', 'Muhtadi', 'ECE', '1', 10, 19, 3.67, '2018-09-16');

insert into Lab_grades (student_id, name, major, section, days_present, project_marks, cgpa, submission_date) values ('S008', 'Farhana', 'CSE', '2', 6, 15, 2.67, '2018-08-16');



- **Show the Lab Grades Table**

*select * from Lab_grades;*



student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s019	Naima	CSE	2	12	20.00	3.70	2018-08-14
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s004	Nahid	ECE	2	7	16.50	3.25	2018-08-20
s005	Arafat	CS	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16

The Lab Grades Table:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s019	Naima	CSE	2	12	20.00	3.70	2018-08-14
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s004	Nahid	ECE	2	7	16.50	3.25	2018-08-20
s005	Arafat	CS	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16

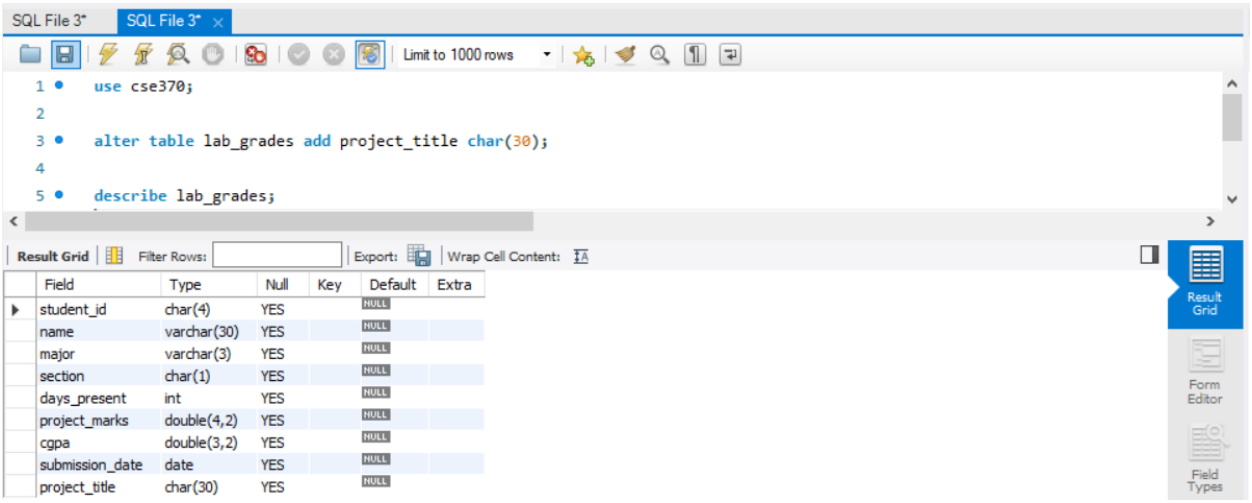
Lab 02: SQL Update, Delete & Basic Select Queries

- **Add Column Project Title to the Lab Grades table**

use cse370;

alter table lab_grades add project_title char(30);

describe lab_grades;

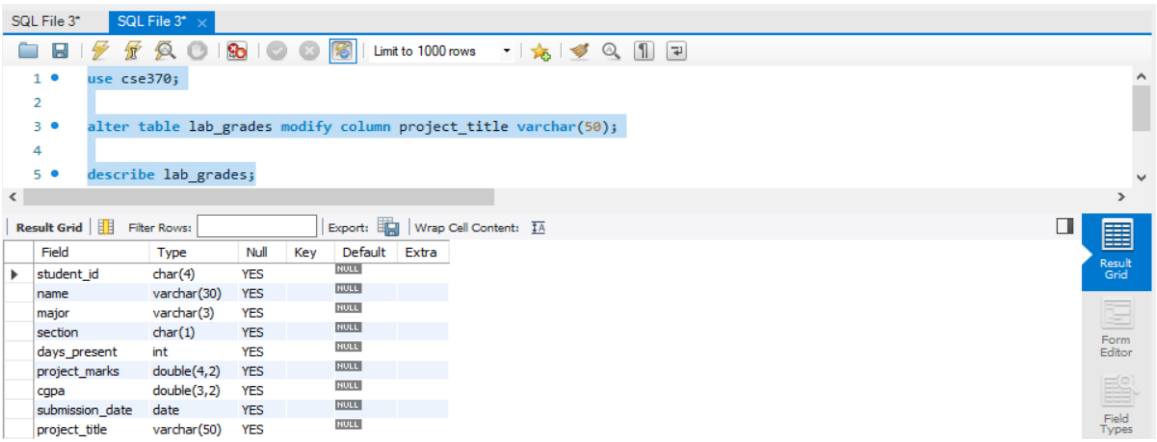


- **Change the Data type of Project Title column from char to varchar**

use cse370;

alter table lab_grades modify column project_title varchar(50);

describe lab_grades;

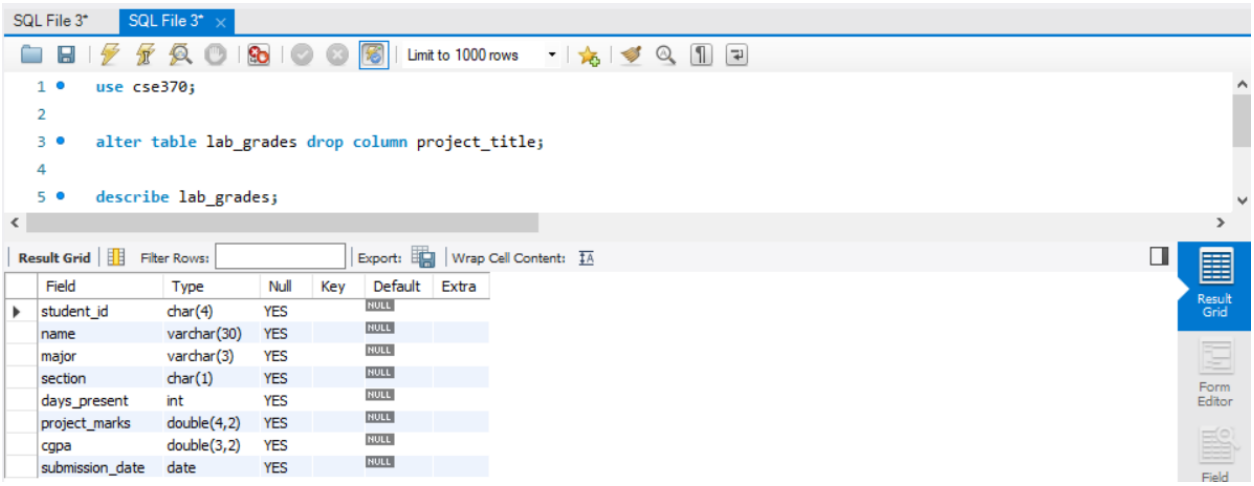


- **Delete Project Title column from table**

use cse370;

alter table lab_grades drop column project_title;

describe lab_grades;

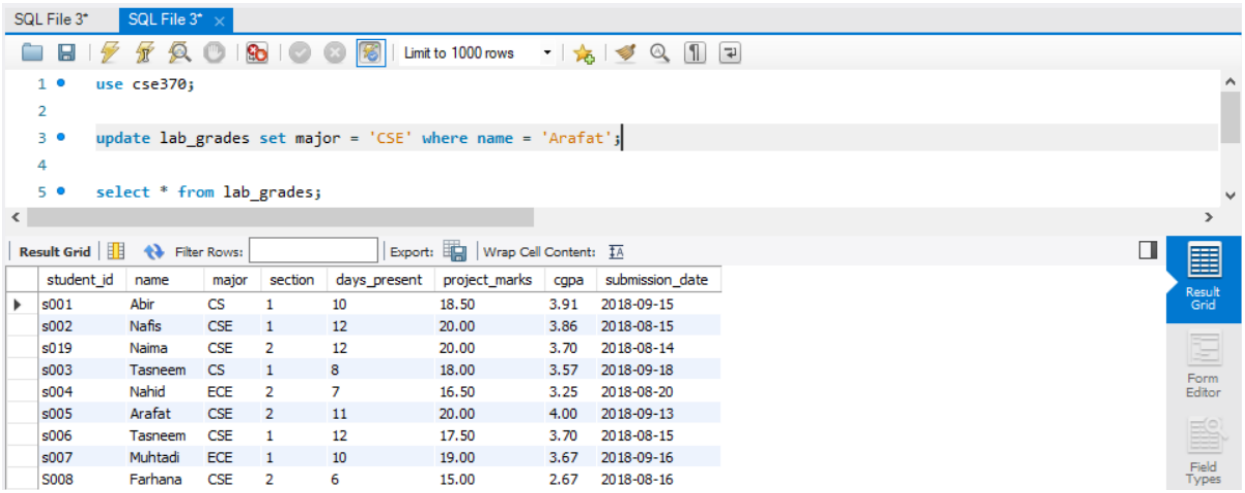


- **Change Arafat's Major from CS to CSE**

use cse370;

update lab_grades set major = 'CSE' where name = 'Arafat';

*select * from lab_grades;*

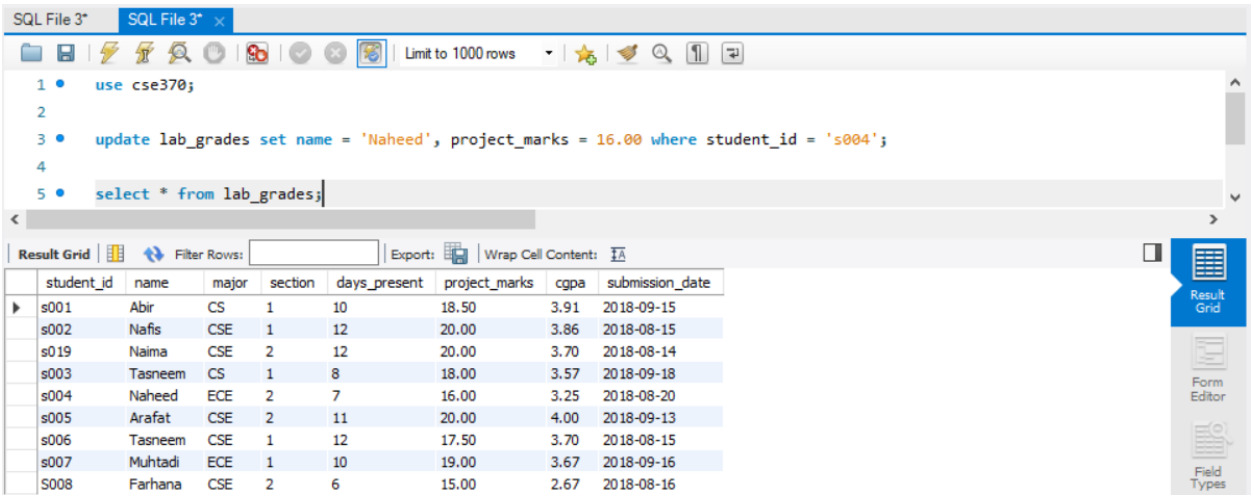


- **Change Nahid name to Naheed and his project mark is 16**

use cse370;

update lab_grades set name = 'Naheed', project_marks = 16.00 where student_id = 's004';

*select * from lab_grades;*



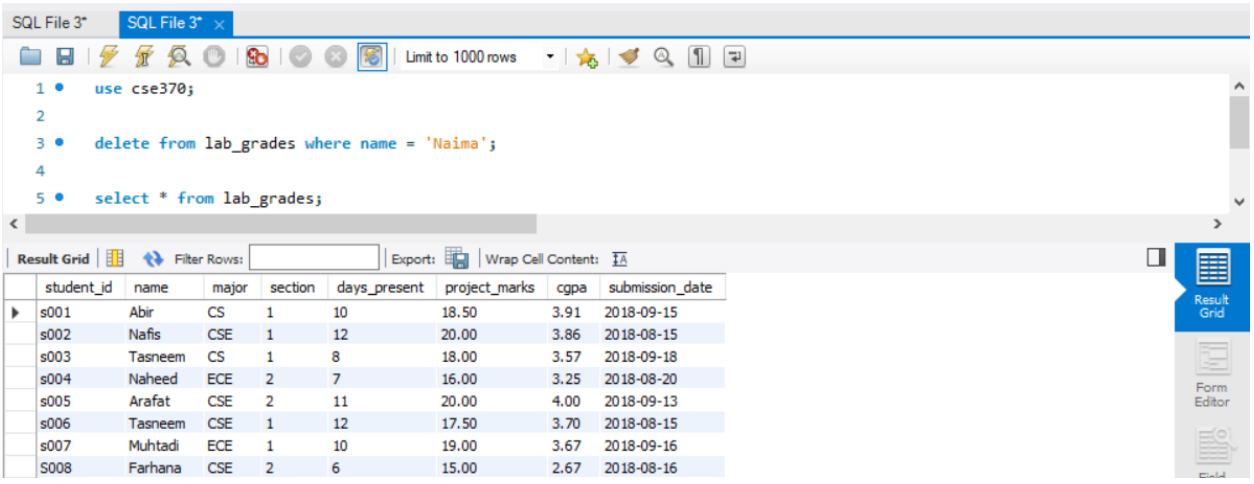
student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s019	Naima	CSE	2	12	20.00	3.70	2018-08-14
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s004	Naheed	ECE	2	7	16.00	3.25	2018-08-20
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16

- **Naima has been dropped out so delete her data**

use cse370;

delete from lab_grades where name = 'Naima';

*select * from lab_grades;*

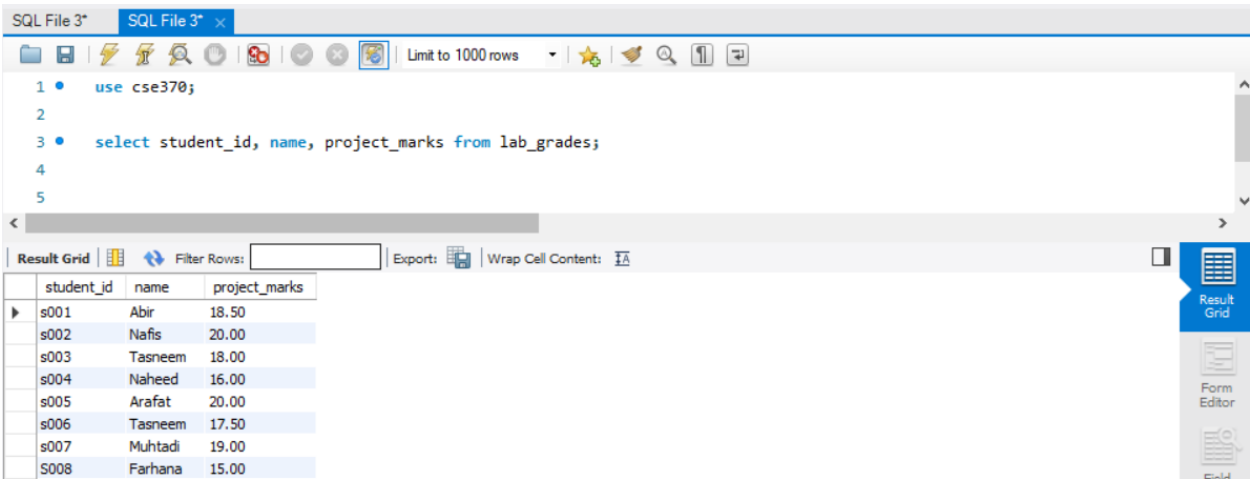


student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s004	Naheed	ECE	2	7	16.00	3.25	2018-08-20
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16

- **Retrieve only Student Id, Name and Project Marks**

use cse370;

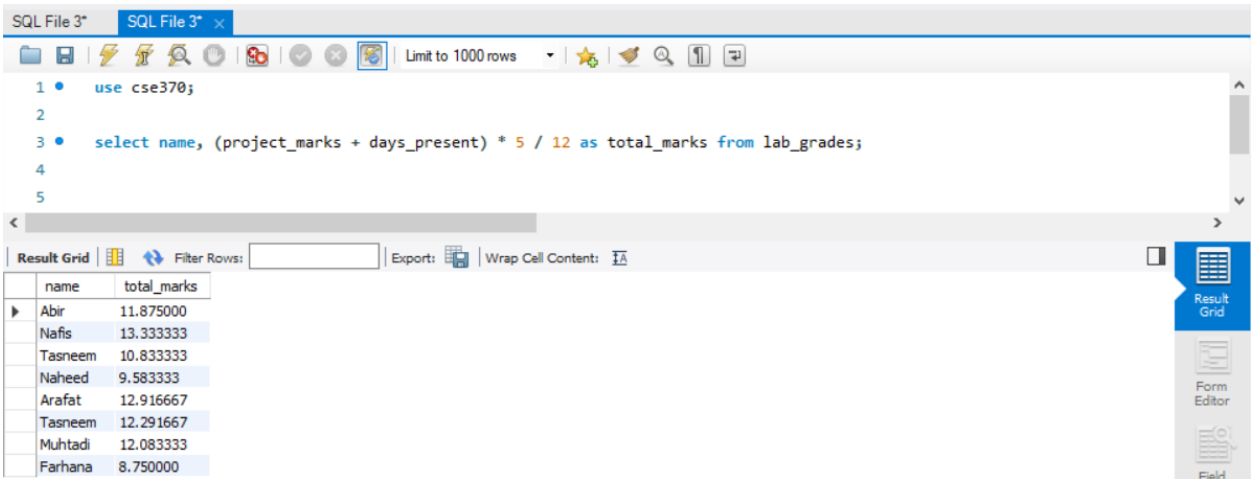
select student_id, name, project_marks from lab_grades;



- **Retrieve Student Name and Total Marks of Students out of 25 (Project + Attendance)**

use cse370;

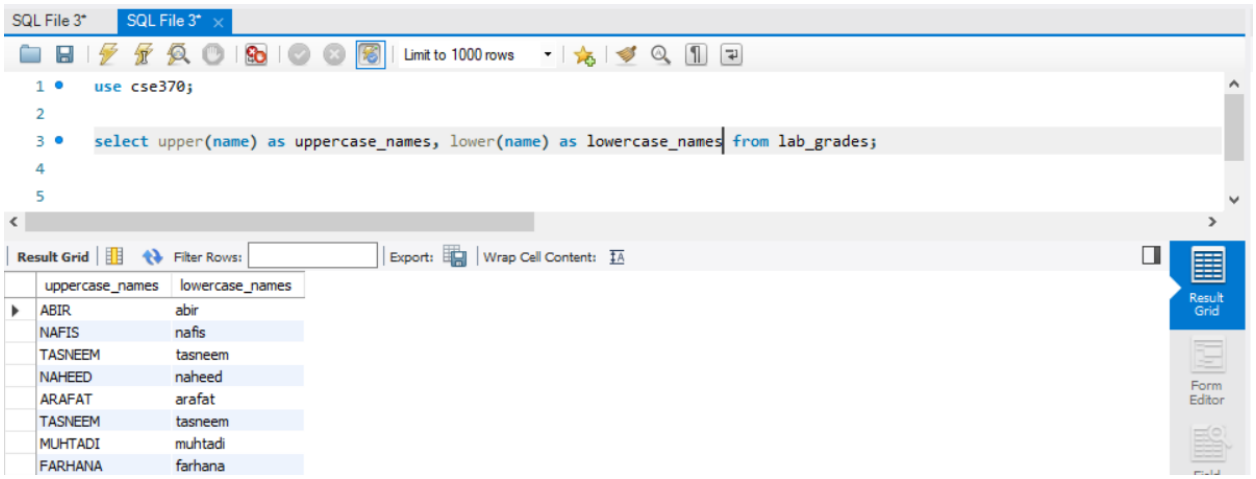
*select name, (project_marks + days_present) * 5 / 12 as total_marks from lab_grades;*



- **Select Student Name in Upper and Lower Case**

use cse370;

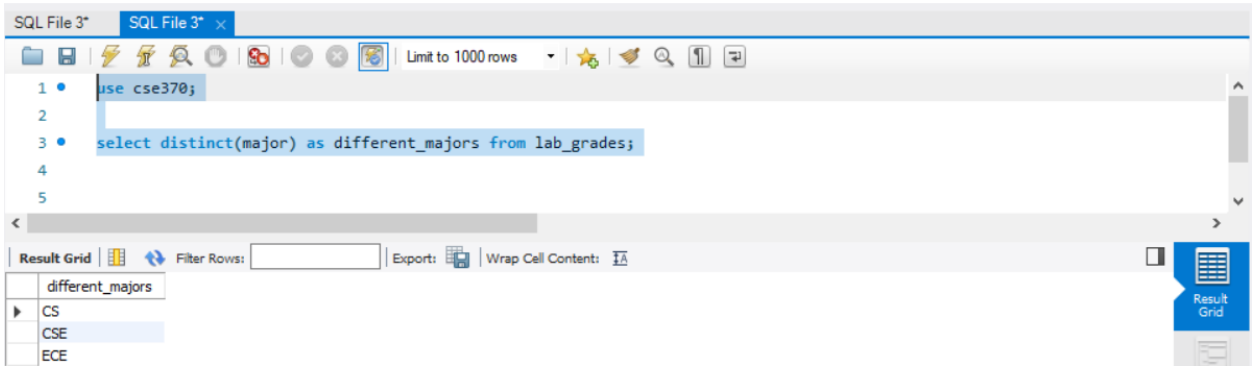
select upper(name) as uppercase_names, lower(name) as lowercase_names from lab_grades;



- **Select Distinct Majors**

use cse370;

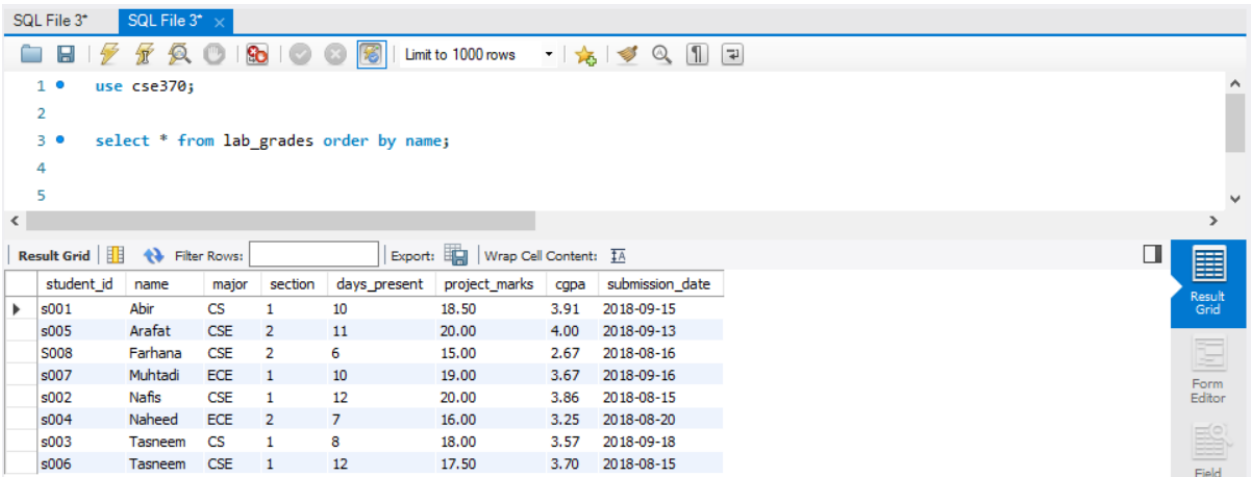
select distinct(major) as different_majors from lab_grades;



- **View all the details sorted by Name**

use cse370;

*select * from lab_grades order by name;*

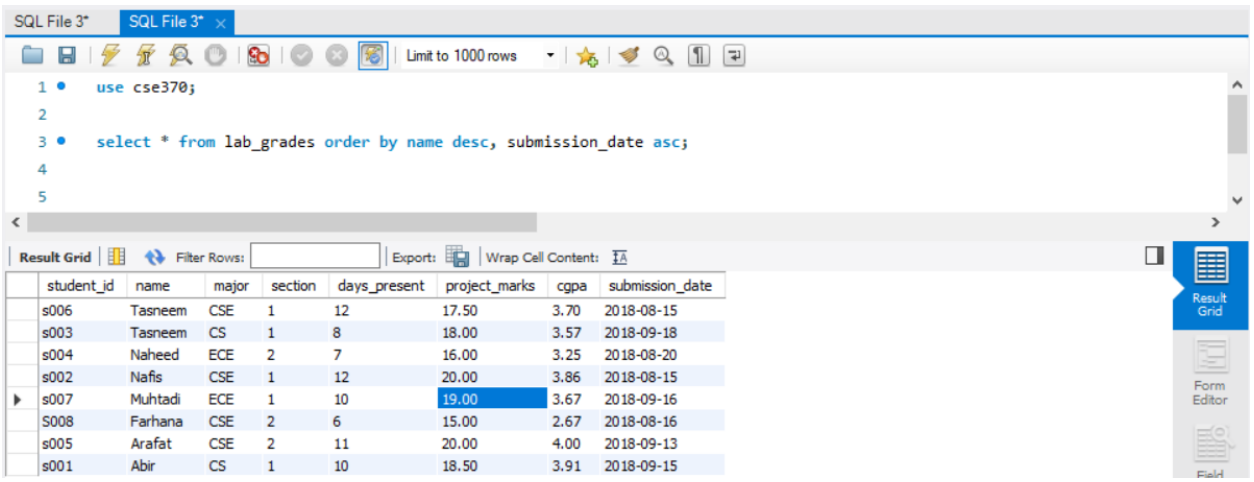


student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s004	Naheed	ECE	2	7	16.00	3.25	2018-08-20
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15

- **Sort all details according to Name and then by Submission Date. There are two students named Tasneem**

use cse370;

*select * from lab_grades order by name desc, submission_date asc;*

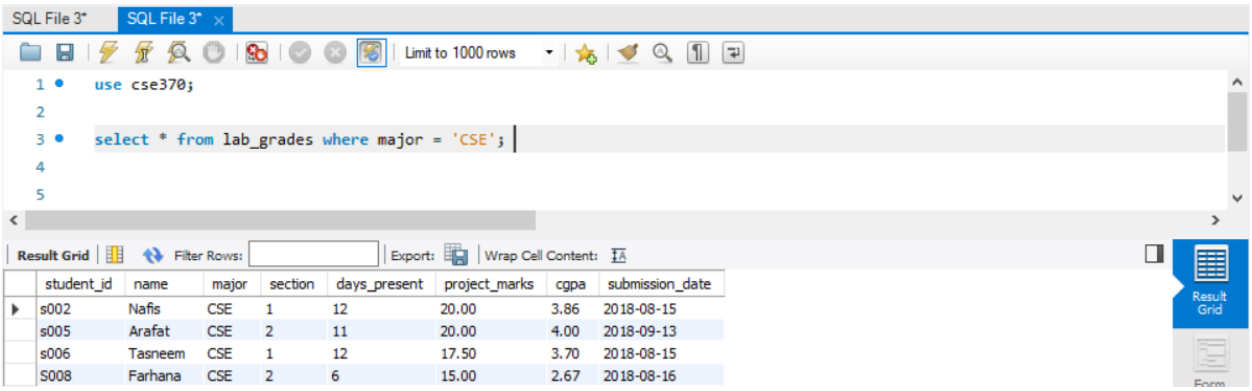


student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s004	Naheed	ECE	2	7	16.00	3.25	2018-08-20
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s001	Abir	CS	1	10	18.50	3.91	2018-09-15

- **View the Name and Project marks for only CSE students**

use cse370;

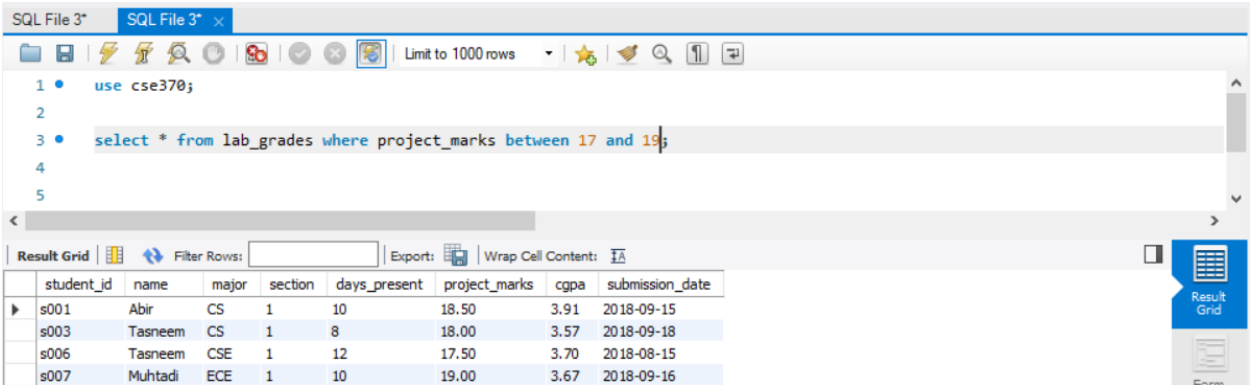
*select * from lab_grades where major = 'CSE';*



- **Retrieve the Name and Project Marks of students whose Project Marks is between 17 and 19**

use cse370;

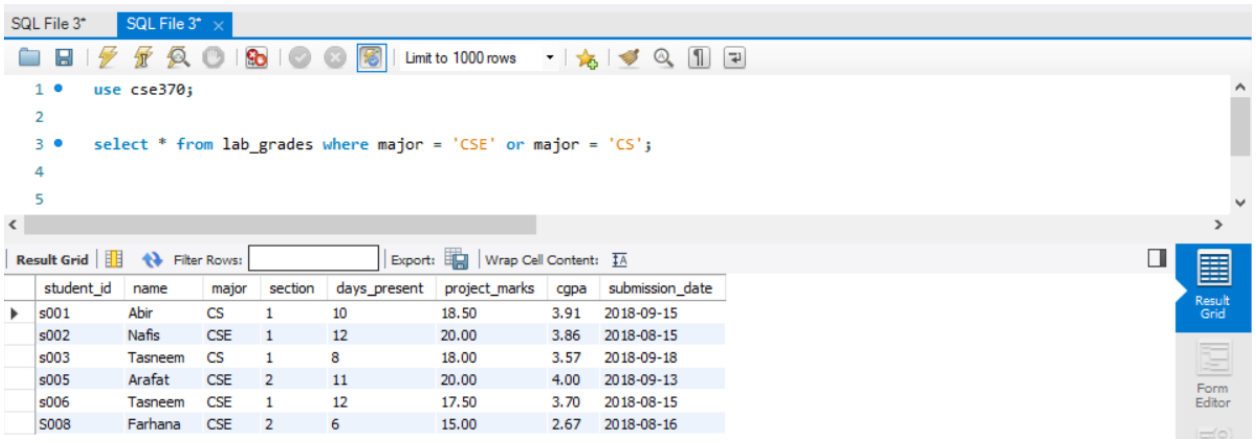
*select * from lab_grades where project_marks between 17 and 19;*



- **Retrieve the details of students who are majoring in either CS or CSE**

use cse370;

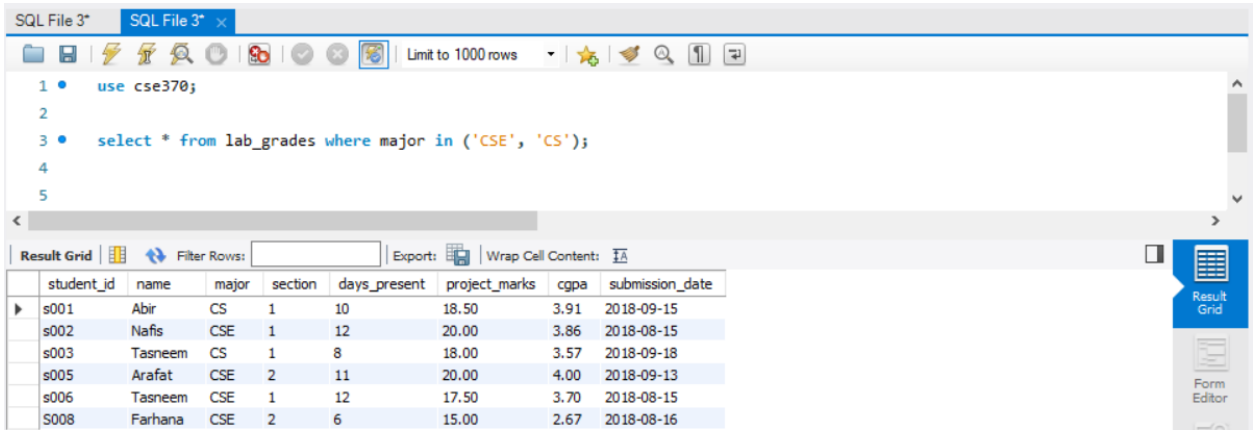
*select * from lab_grades where major = 'CSE' or major = 'CS';*



student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16

use cse370;

*select * from lab_grades where major in ('CSE', 'CS');*

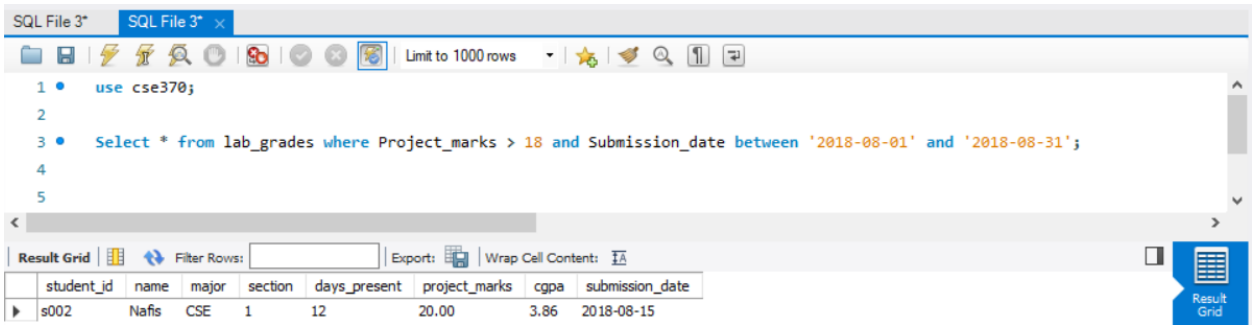


student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
S008	Farhana	CSE	2	6	15.00	2.67	2018-08-16

- **Retrieve the details of the students who submitted their Project in August and whose marks is greater than 18**

use cse370;

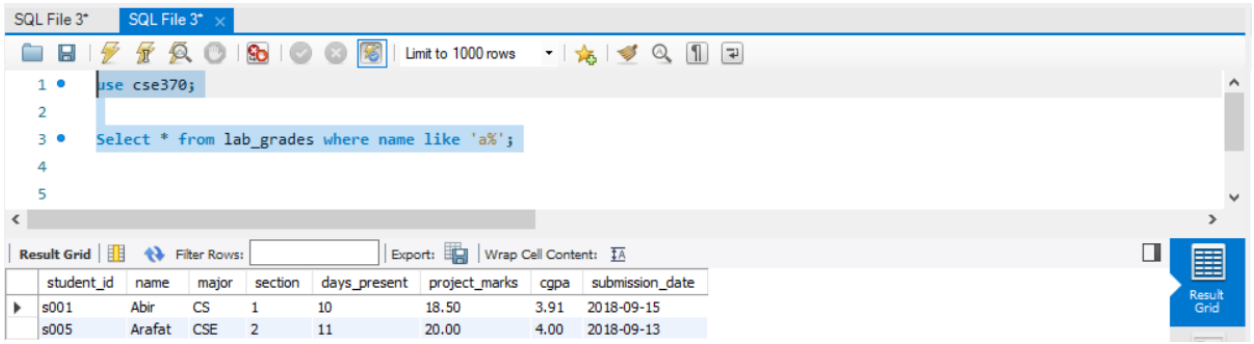
*select * from lab_grades where Project_marks > 18 and submission_date between '2018-08-01' and '2018-08-31';*



- **Retrieve the details of students whose name start with ‘a’**

use cse370;

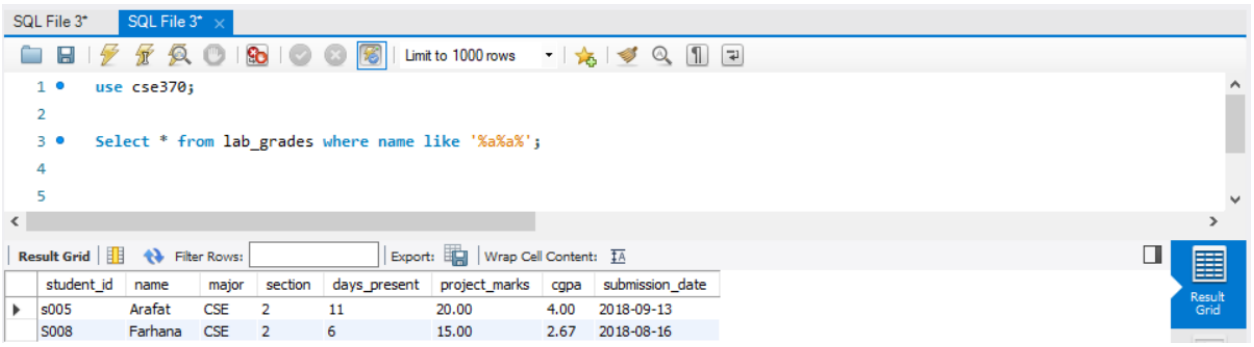
*select * from lab_grades where name like 'a%';*



- **Retrieve the details of students whose name contains at least 2 a's**

use cse370;

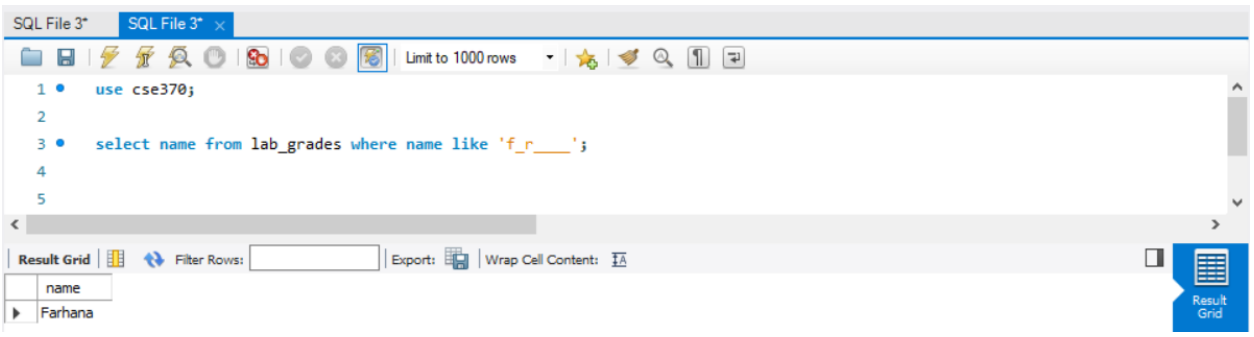
*Select * from lab_grades where name like '%a%a%';*



- **Understand the pattern**

use cse370;

select name from lab_grades where name like 'f_r____';



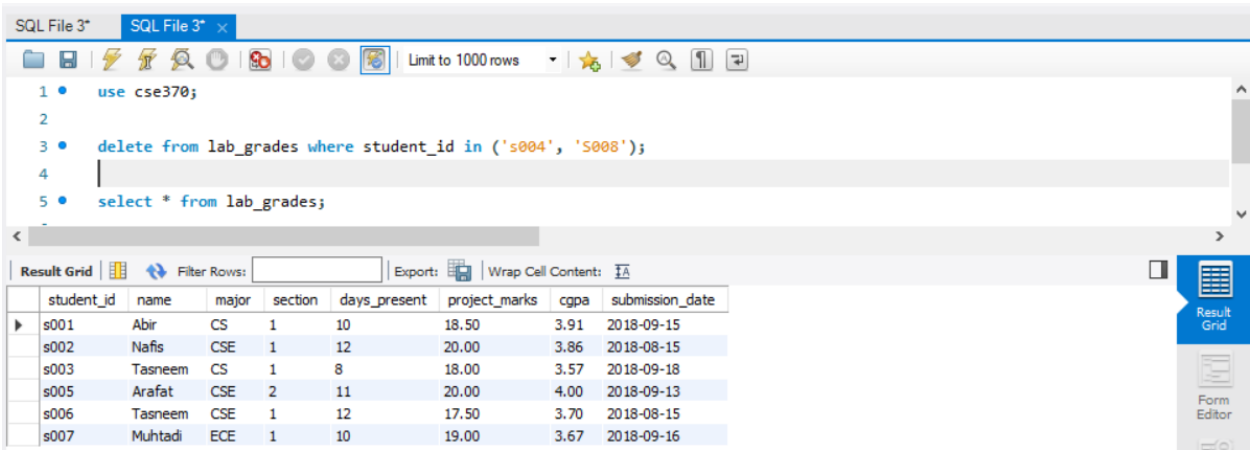
Lab 3: SQL Subqueries & Aggregate Functions

For This Lab Grades table has been refactored

use cse370;

delete from lab_grades where student_id in ('s004', 'S008');

*select * from lab_grades;*



Current Table:

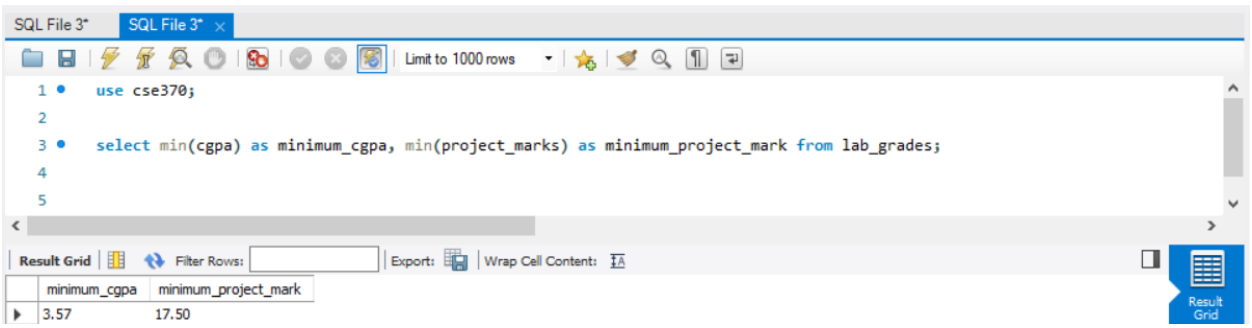
student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16

Aggregate Functions, Group By and Having:

- **Retrieve the minimum CGPA/Project_marks from the table**

use cse370;

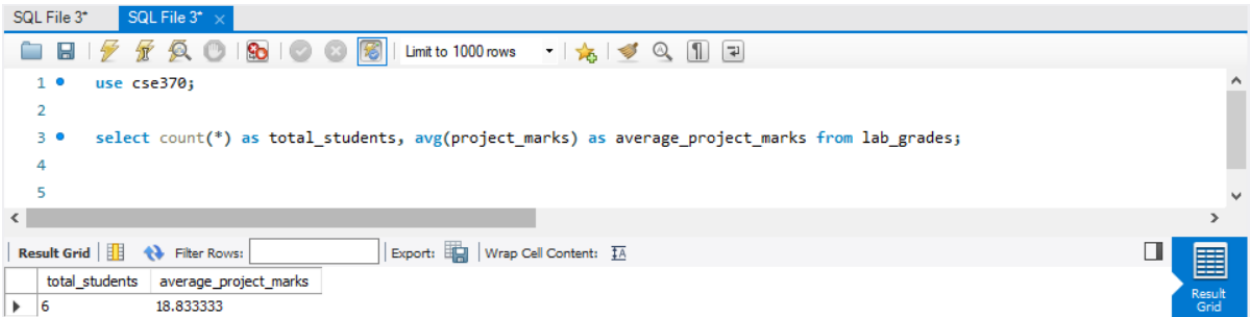
select min(cgpa) as minimum_cgpa, min(project_marks) as minimum_project_mark from lab_grades;



- **Retrieve the total number of students and the average Projects Marks**

use cse370;

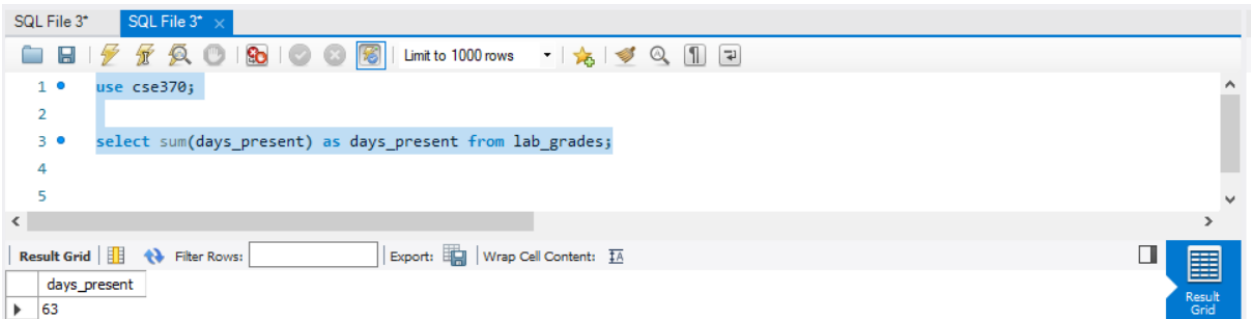
select count(*) as total_students, avg(project_marks) as average_project_marks from lab_grades;



- **Find the sum of the number of Days Present**

use cse370;

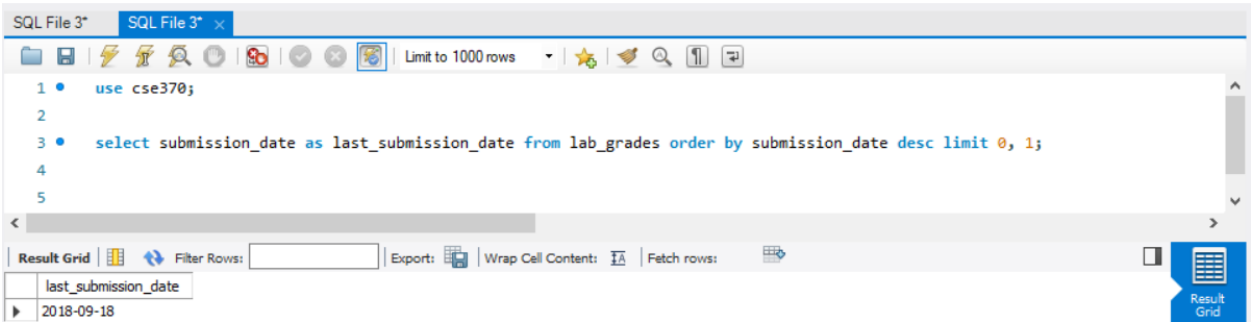
select sum(days_present) as days_present from lab_grades;



- ***Retrieve the last Submission Date***

use cse370;

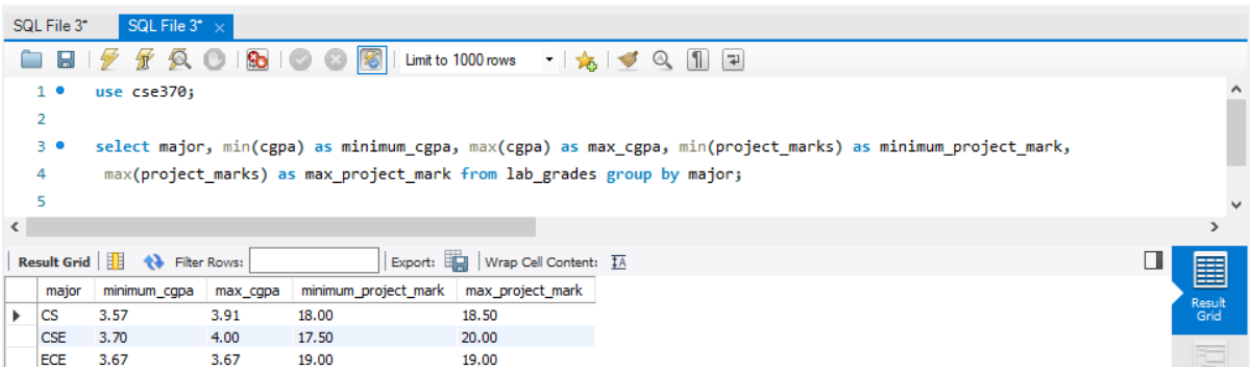
select submission_date as last_submission_date from lab_grades order by submission_date desc limit 0, 1;



- **Find Minimum and Maximum CGPA / Project Marks of each Major**

use cse370;

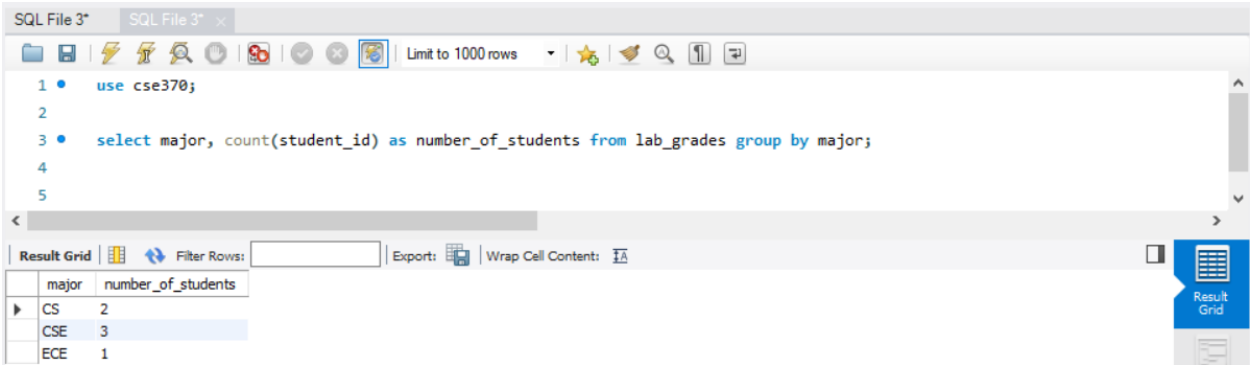
select major, min(cgpa) as minimum_cgpa, max(cgpa) as max_cgpa, min(project_marks) as minimum_project_mark, max(project_marks) as max_project_mark from lab_grades group by major;



- **Retrieve total number of students for each Major**

use cse370;

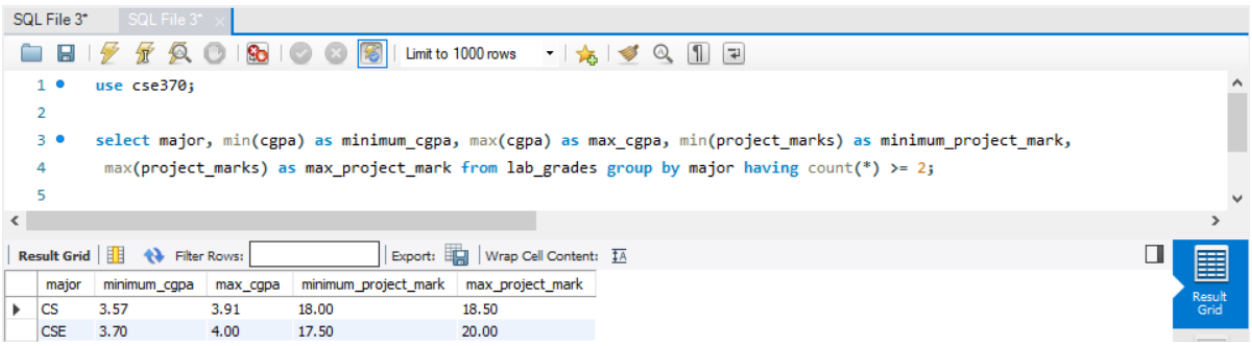
select major, count(student_id) as number_of_students from lab_grades group by major;



- **For each major find the minimum and maximum CGPA / Project Marks, but only if there were at least 2 students in the Major**

use cse370;

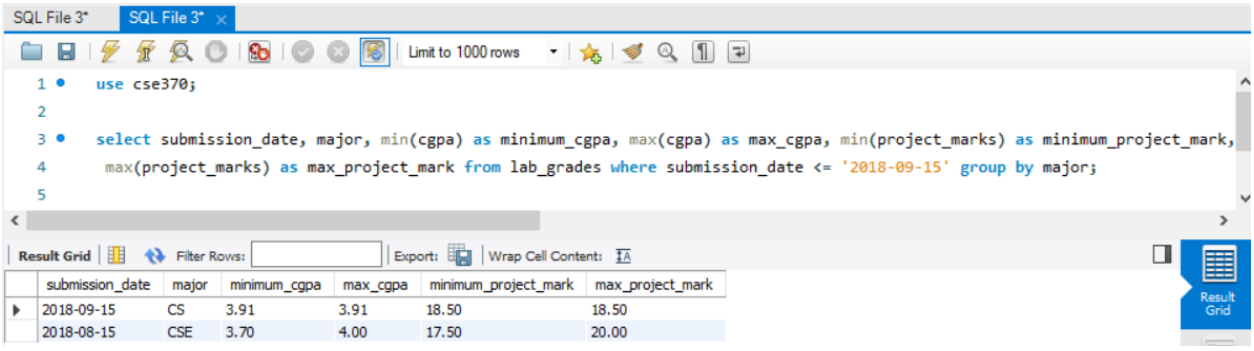
select major, min(cgpa) as minimum_cgpa, max(cgpa) as max_cgpa, min(project_marks) as minimum_project_mark, max(project_marks) as max_project_mark from lab_grades group by major having count(*) >= 2;



- **For each major find the minimum and maximum CGPA / Project Marks, but consider only students who submitted before or on 15th September**

use cse370;

select submission_date, major, min(cgpa) as minimum_cgpa, max(cgpa) as max_cgpa, min(project_marks) as minimum_project_mark, max(project_marks) as max_project_mark from lab_grades where submission_date <= '2018-09-15' group by major;



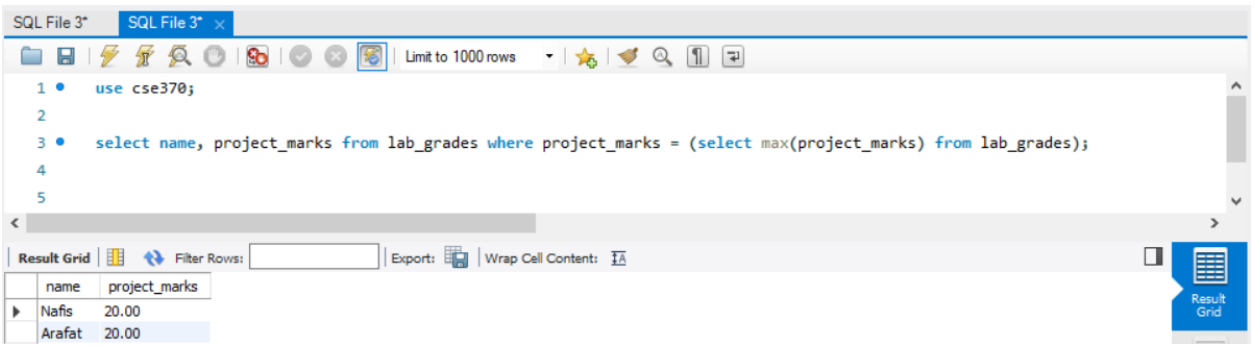
A HAVING clause is like a WHERE clause, but applies only to groups as a whole (that is, to the rows in the result set representing groups), whereas the WHERE clause applies to individual rows.

Sub Queries / Nested Queries, Any and All:

- **Name of the Student who received highest Project Mark**

use cse370;

select name, project_marks from lab_grades where project_marks = (select max(project_marks) from lab_grades);



- **For each major find the Name of the student who has the lowest attendance**

use cse370;

select name, major, days_present from lab_grades where (major, days_present) in (select major, min(days_present) from lab_grades group by major);

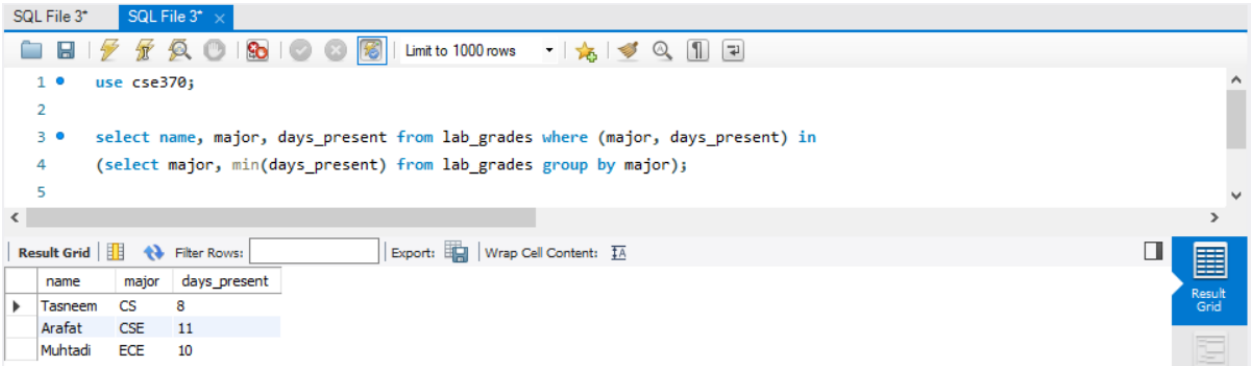


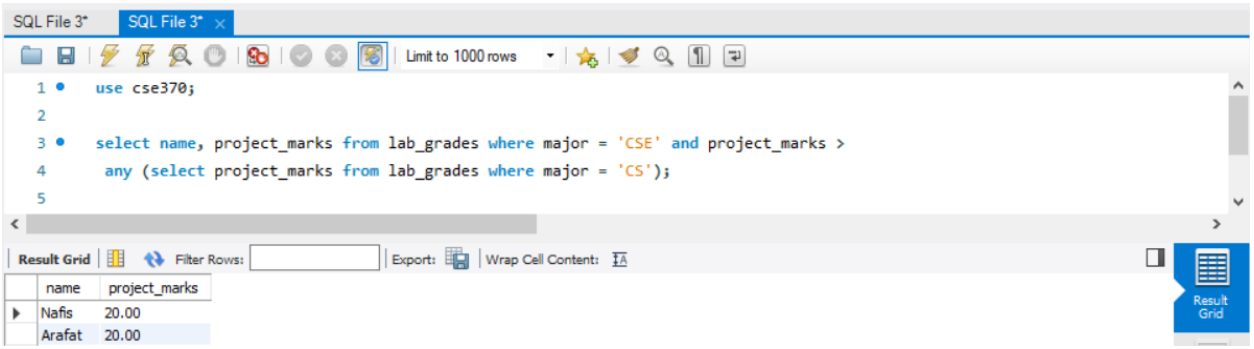
Table:

student_id	name	major	section	days_present	project_marks	cgpa	submission_date
s001	Abir	CS	1	10	18.50	3.91	2018-09-15
s002	Nafis	CSE	1	12	20.00	3.86	2018-08-15
s003	Tasneem	CS	1	8	18.00	3.57	2018-09-18
s005	Arafat	CSE	2	11	20.00	4.00	2018-09-13
s006	Tasneem	CSE	1	12	17.50	3.70	2018-08-15
s007	Muhtadi	ECE	1	10	19.00	3.67	2018-09-16

- **Retrieve the CSE students whose CGPA / Project Marks is higher than at least 1 CS students**

use cse370;

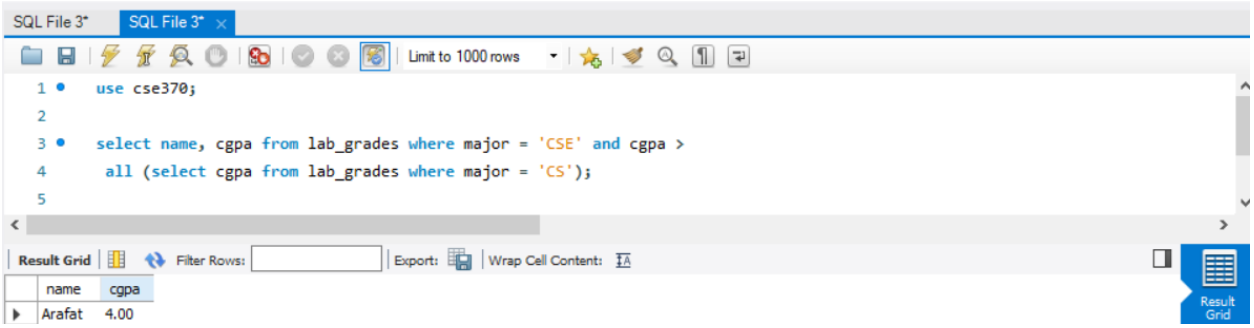
select name, project_marks from lab_grades where major = 'CSE' and project_marks > any (select project_marks from lab_grades where major = 'CS');



- ***Retrieve the CSE students whose* CGPA / Project Marks *is higher than all CS students***

use cse370;

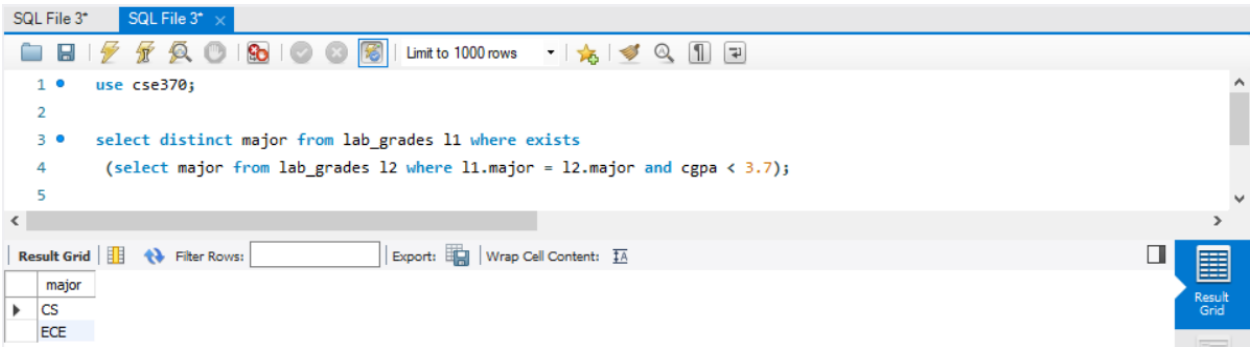
select name, cgpa from lab_grades where major = 'CSE' and cgpa > all (select cgpa from lab_grades where major = 'CS');



- **Select those Majors for which at least 1 student has CGPA lower than 3.7 / Project Marks < 18**

use cse370;

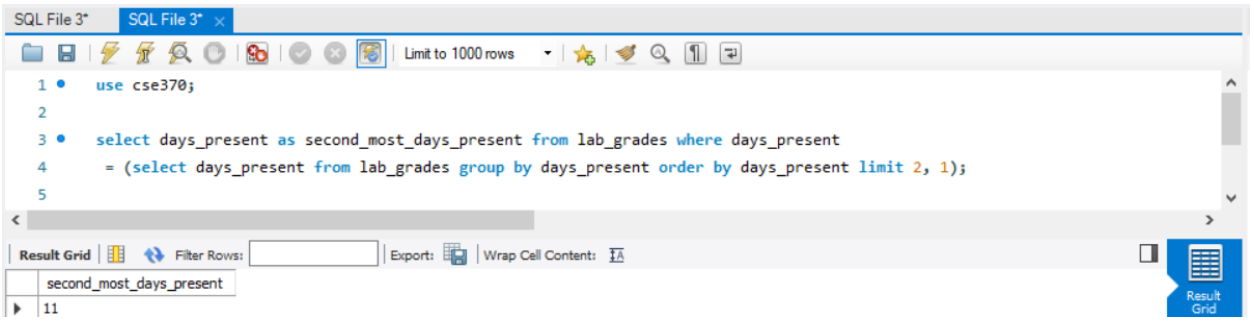
select distinct major from lab_grades l1 where exists (select major from lab_grades l2 where l1.major = l2.major and cgpa < 3.7);



- **Select 2nd most Days Present**

use cse370;

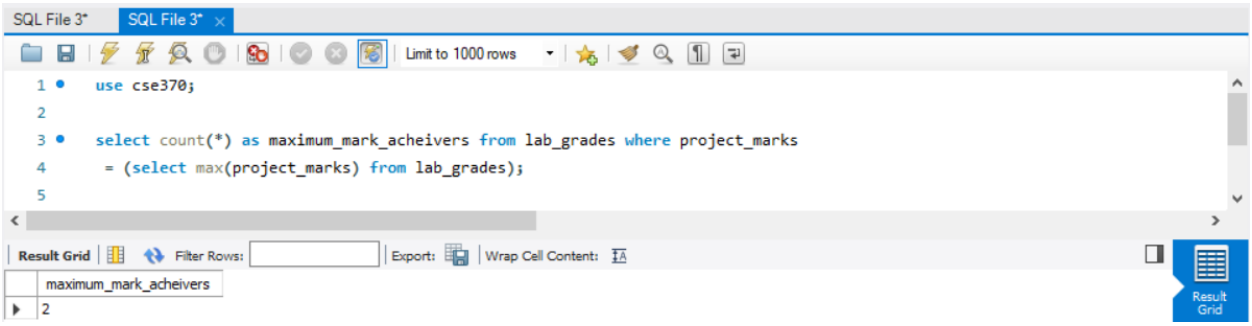
select days_present as second_most_days_present from lab_grades where days_present = (select days_present from lab_grades group by days_present order by days_present limit 2, 1);



- **Retrieve the total number of students who obtained the maximum Marks**

use cse370;

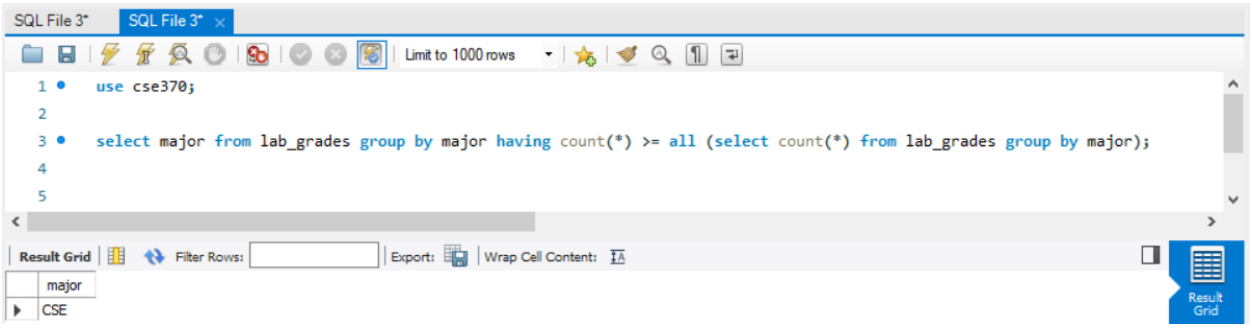
select count(*) as maximum_mark_acheivers from lab_grades where project_marks = (select max(project_marks) from lab_grades);



- **Retrieve the Major which has the highest number of students enrolled**

use cse370;

select major from lab_grades group by major having count(*) >= all (select count(*) from lab_grades group by major);



Order

SELECT *column_name(s)*
FROM *table_name(s)*
WHERE *conditions*
GROUP BY *column_name(s)*
HAVING *conditions*
ORDER BY *column_name(s)*;

Introduction to Bank DB, SQL Joins and Constraints

- **Create Tables**

use cse370;

create table customer (customer_id varchar(10) not null, customer_name varchar(20) not null, customer_street varchar(30), customer_city varchar(30), primary key (customer_id)) ;

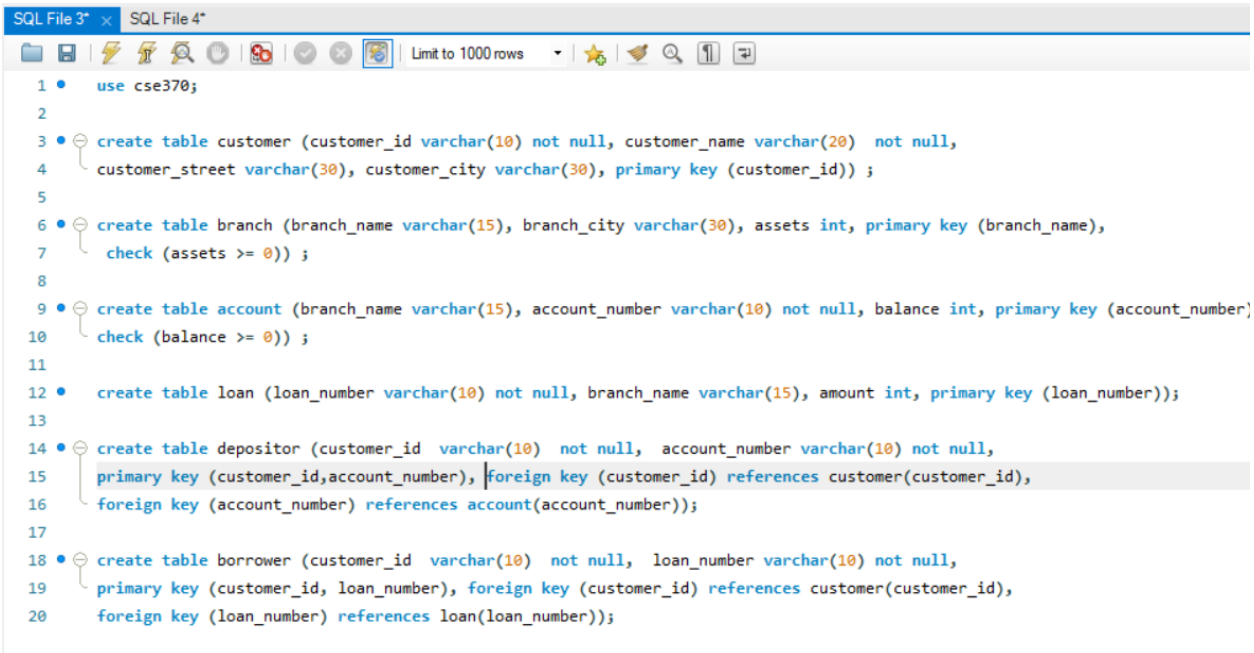
create table branch (branch_name varchar(15), branch_city varchar(30), assets int, primary key (branch_name), check (assets >= 0)) ;

create table account (branch_name varchar(15), account_number varchar(10) not null, balance int, primary key (account_number), check (balance >= 0)) ;

create table loan (loan_number varchar(10) not null, branch_name varchar(15), amount int, primary key (loan_number));

create table depositor (customer_id varchar(10) not null, account_number varchar(10) not null, primary key (customer_id,account_number), foreign key (customer_id) references customer(customer_id), foreign key (account_number) references account(account_number));

create table borrower (customer_id varchar(10) not null, loan_number varchar(10) not null, primary key (customer_id, loan_number), foreign key (customer_id) references customer(customer_id), foreign key (loan_number) references loan(loan_number));



- **Insert data into the tables**

use cse370;

insert into customer values ('C-101','Jones', 'Main', 'Harrison'), ('C-201','Smith', 'North', 'Rye'), ('C-211','Hayes', 'Main', 'Harrison'), ('C-212','Curry', 'North', 'Rye'), ('C-215','Lindsay', 'Park', 'Pittsfield'), ('C-220','Turner', 'Putnam', 'Stamford'), ('C-222','Williams', 'Nassau', 'Princeton'), ('C-225','Adams', 'Spring', 'Pittsfield'), ('C-226','Johnson', 'Alma', 'Palo Alto'), ('C-233','Glenn', 'Sand Hill', 'Woodside'), ('C-234','Brooks', 'Senator', 'Brooklyn'), ('C-255','Green', 'Walnut', 'Stamford');

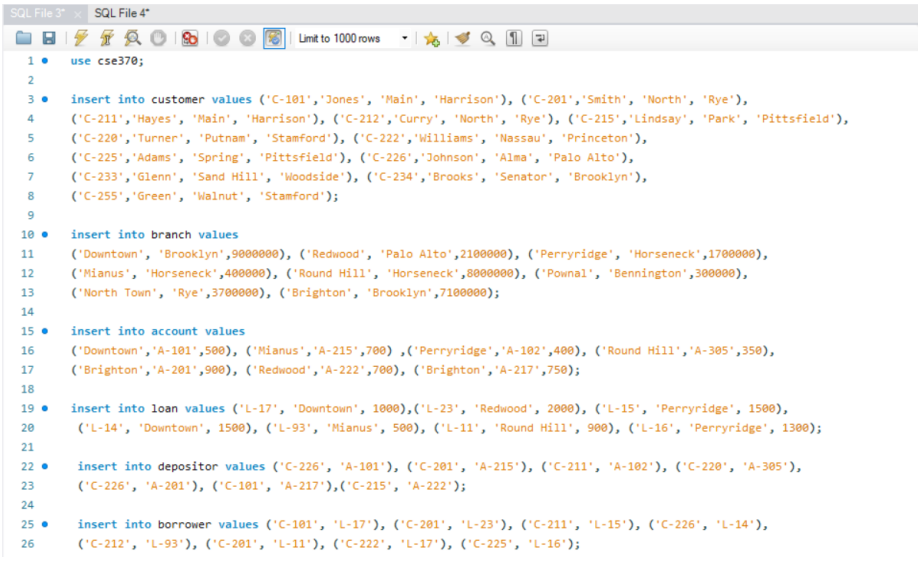
insert into branch values
('Downtown', 'Brooklyn',9000000), ('Redwood', 'Palo Alto',2100000), ('Perryridge', 'Horseneck',1700000), ('Mianus', 'Horseneck',400000), ('Round Hill', 'Horseneck',8000000), ('Pownal', 'Bennington',300000), ('North Town', 'Rye',3700000), ('Brighton', 'Brooklyn',7100000);

insert into account values
('Downtown','A-101',500), ('Mianus','A-215',700) ,('Perryridge','A-102',400), ('Round Hill','A-305',350), ('Brighton','A-201',900), ('Redwood','A-222',700), ('Brighton','A-217',750);

insert into loan values ('L-17', 'Downtown', 1000),('L-23', 'Redwood', 2000), ('L-15', 'Perryridge', 1500), ('L-14', 'Downtown', 1500), ('L-93', 'Mianus', 500), ('L-11', 'Round Hill', 900), ('L-16', 'Perryridge', 1300);

insert into depositor values ('C-226', 'A-101'), ('C-201', 'A-215'), ('C-211', 'A-102'), ('C-220', 'A-305'), ('C-226', 'A-201'), ('C-101', 'A-217'),('C-215', 'A-222');

insert into borrower values ('C-101', 'L-17'), ('C-201', 'L-23'), ('C-211', 'L-15'), ('C-226', 'L-14'), ('C-212', 'L-93'), ('C-201', 'L-11'), ('C-222', 'L-17'), ('C-225', 'L-16');



Customer Table:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(10)	NO	PRI	NULL	
customer_name	varchar(20)	NO		NULL	
customer_street	varchar(30)	YES		NULL	
customer_city	varchar(30)	YES		NULL	

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford
NULL	NULL	NULL	NULL

Branch Table:

Field	Type	Null	Key	Default	Extra
branch_name	varchar(15)	NO	PRI	NULL	
branch_city	varchar(30)	YES		NULL	
assets	int	YES		NULL	

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000
NULL	NULL	NULL

Account Table:

Field	Type	Null	Key	Default	Extra
branch_name	varchar(15)	YES		NULL	
account_number	varchar(10)	NO	PRI	NULL	
balance	int	YES		NULL	

branch_name	account_number	balance
Downtown	A-101	500
Perryridge	A-102	400
Brighton	A-201	900
Mianus	A-215	700
Brighton	A-217	750
Redwood	A-222	700
Round Hill	A-305	350
NULL	NULL	NULL

Loan Table:

Field	Type	Null	Key	Default	Extra
loan_number	varchar(10)	NO	PRI	NULL	
branch_name	varchar(15)	YES		NULL	
amount	int	YES		NULL	

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500
NULL	NULL	NULL

Depositor Table:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(10)	NO	PRI	NULL	
account_number	varchar(10)	NO	PRI	NULL	

customer_id	account_number
C-226	A-101
C-211	A-102
C-226	A-201
C-201	A-215
C-101	A-217
C-215	A-222
C-220	A-305
NULL	NULL

Borrower Table:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(10)	NO	PRI	NULL	
loan_number	varchar(10)	NO	PRI	NULL	

customer_id	loan_number
C-201	L-11
C-226	L-14
C-211	L-15
C-225	L-16
C-101	L-17
C-222	L-17
C-201	L-23
C-212	L-93
NULL	NULL

Customer, Account, Depositor Table Joined:

customer_id	customer_name	customer_street	customer_city	customer_id	account_number	branch_name	account_number	balance
C-226	Johnson	Alma	Palo Alto	C-226	A-101	Downtown	A-101	500
C-211	Hayes	Main	Harrison	C-211	A-102	Perryridge	A-102	400
C-226	Johnson	Alma	Palo Alto	C-226	A-201	Brighton	A-201	900
C-201	Smith	North	Rye	C-201	A-215	Mianus	A-215	700
C-101	Jones	Main	Harrison	C-101	A-217	Brighton	A-217	750
C-215	Lindsay	Park	Pittsfield	C-215	A-222	Redwood	A-222	700
C-220	Turner	Putnam	Stamford	C-220	A-305	Round Hill	A-305	350

Customer, Account, Depositor Table Joined:

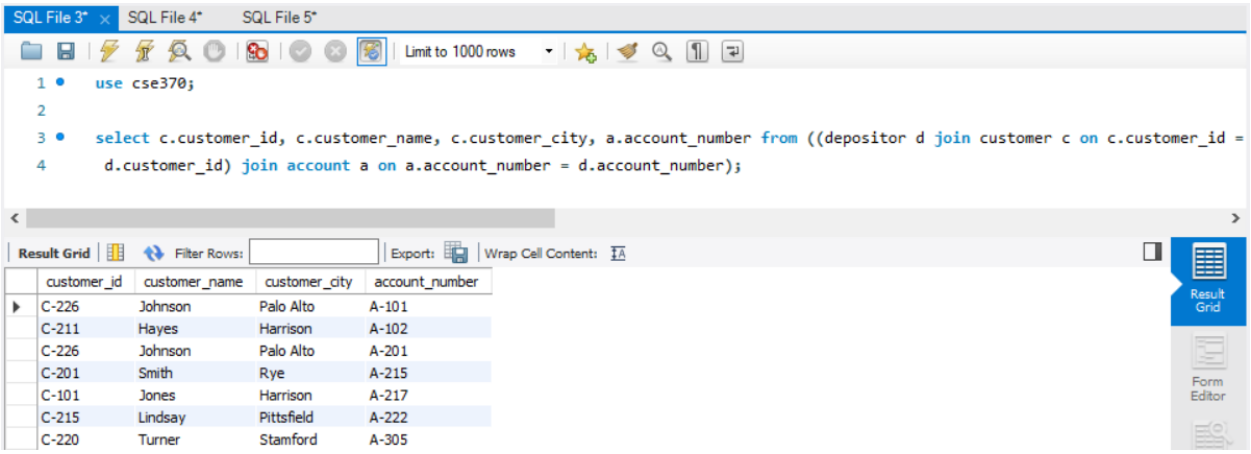
customer_id	customer_name	customer_street	customer_city	customer_id	loan_number	loan_number	branch_name	amount
C-201	Smith	North	Rye	C-201	L-11	L-11	Round Hill	900
C-226	Johnson	Alma	Palo Alto	C-226	L-14	L-14	Downtown	1500
C-211	Hayes	Main	Harrison	C-211	L-15	L-15	Perryridge	1500
C-225	Adams	Spring	Pittsfield	C-225	L-16	L-16	Perryridge	1300
C-222	Williams	Nassau	Princeton	C-222	L-17	L-17	Downtown	1000
C-101	Jones	Main	Harrison	C-101	L-17	L-17	Downtown	1000
C-201	Smith	North	Rye	C-201	L-23	L-23	Redwood	2000
C-212	Curry	North	Rye	C-212	L-93	L-93	Mianus	500

Retrieve all customer's Id, Name, City and Account Number using

Inner Join:

use cse370;

select c.customer_id, c.customer_name, c.customer_city, a.account_number from ((customer c inner join depositor d on c.customer_id = d.customer_id) inner join account a on a.account_number = d.account_number);

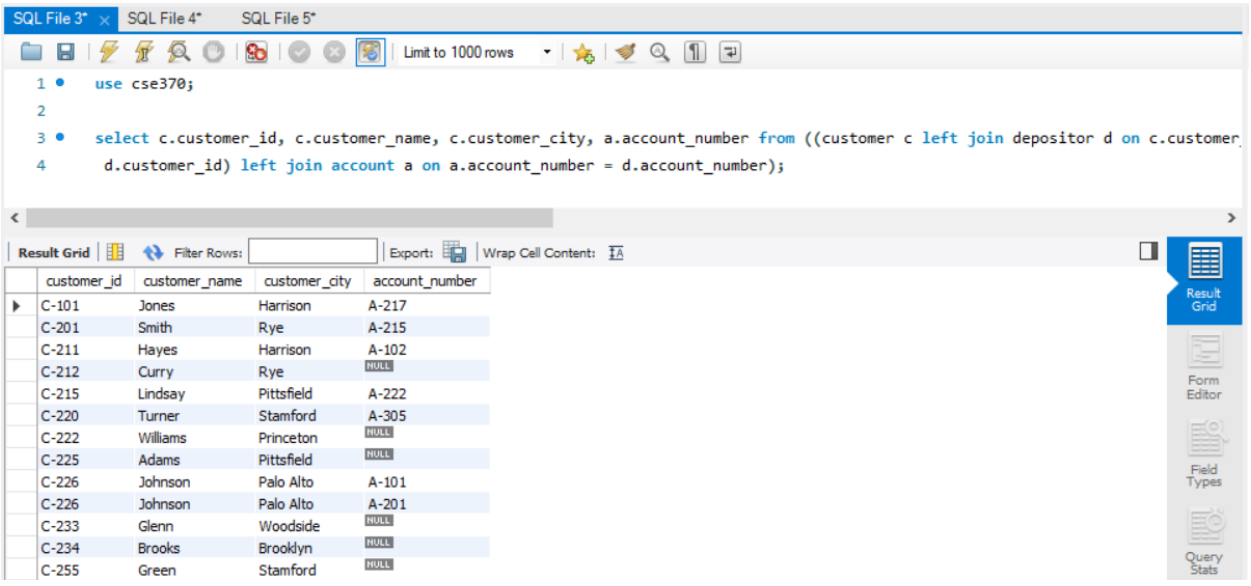


Outer Joins:

Left Join:

use cse370;

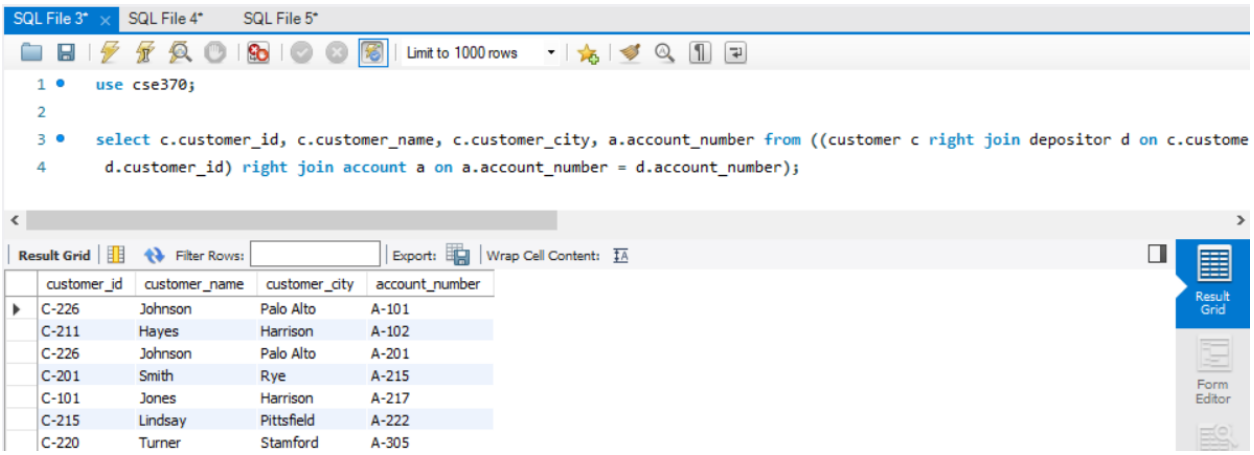
select c.customer_id, c.customer_name, c.customer_city, a.account_number from ((customer c left join depositor d on c.customer_id = d.customer_id) left join account a on a.account_number = d.account_number);



Right Join:

use cse370;

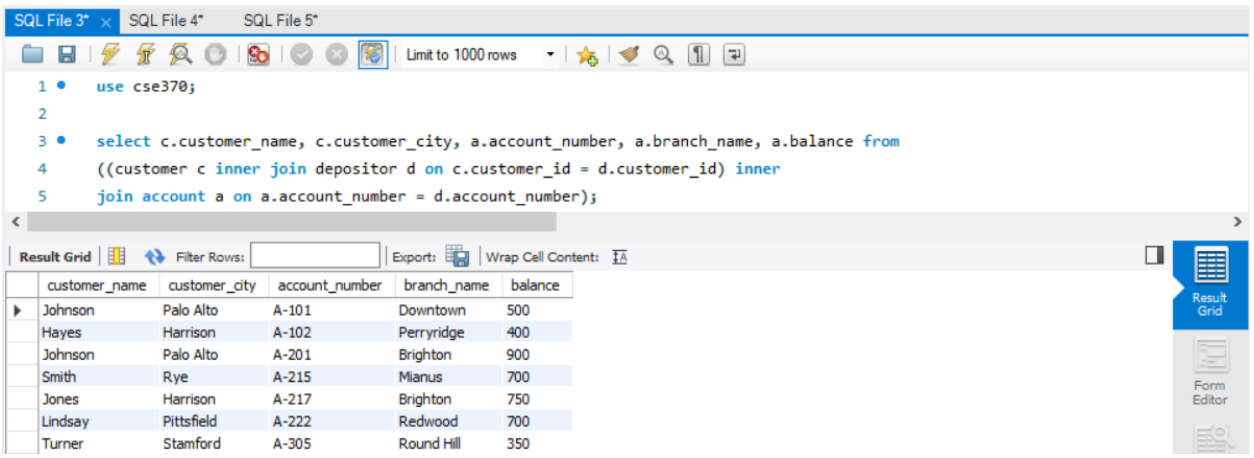
select c.customer_id, c.customer_name, c.customer_city, a.account_number from ((customer c right join depositor d on c.customer_id = d.customer_id) right join account a on a.account_number = d.account_number);



- **Retrieve the following information from your database using "join": Customer Name, City, Account Number, Balance and Branch name**

use cse370;

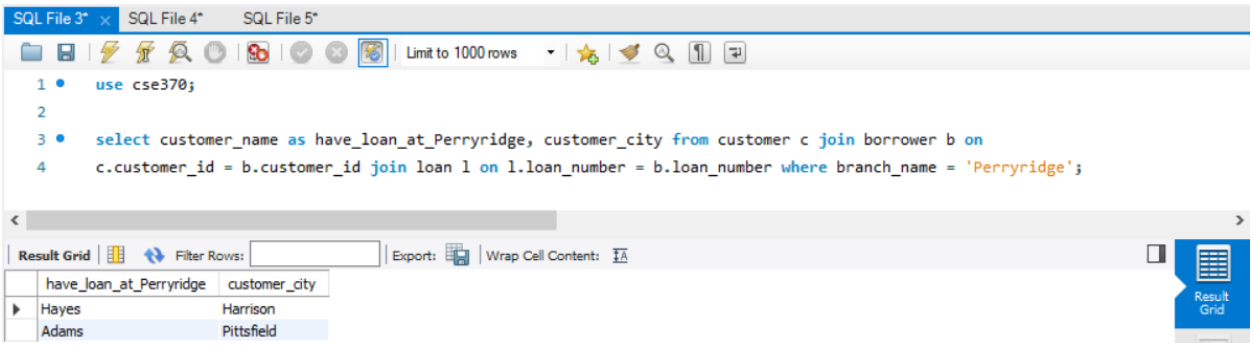
select c.customer_name, c.customer_city, a.account_number, a.branch_name, a.balance from ((customer c inner join depositor d on c.customer_id = d.customer_id) inner join account a on a.account_number = d.account_number);



- **Find Names and Cities of Customers who have a Loan at Perryridge branch**

use cse370;

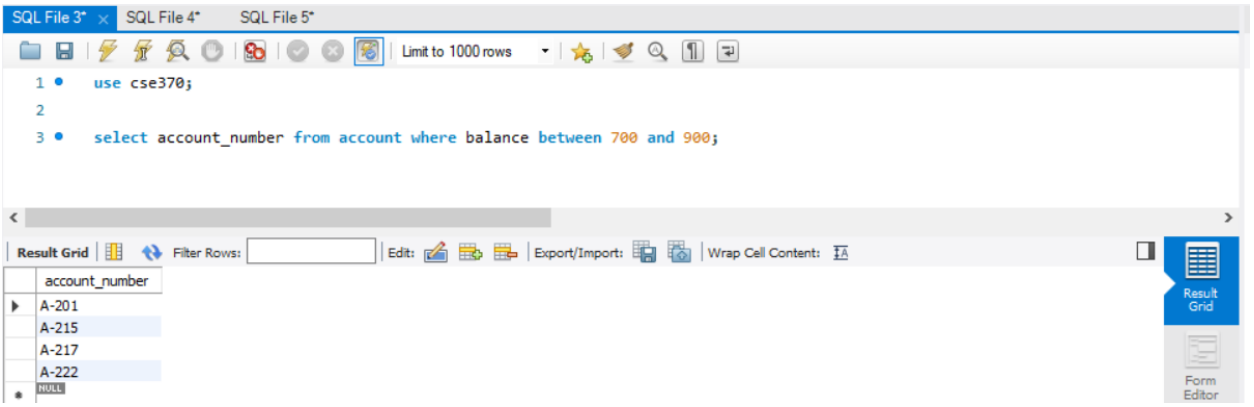
select customer_name as have_loan_at_Perryridge, customer_city from customer c join borrower b on c.customer_id = b.customer_id join loan l on l.loan_number = b.loan_number where branch_name = 'Perryridge'



- **Find which Accounts with Balances between 700 and 900**

use cse370;

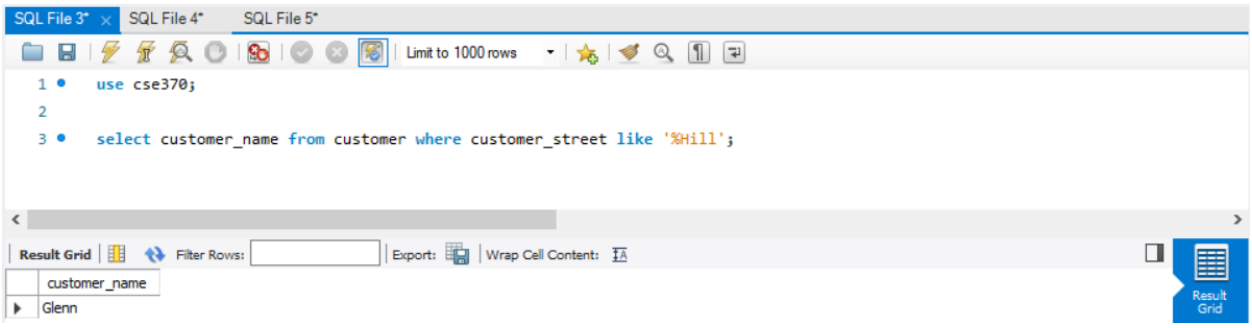
select account_number from account where balance between 700 and 900;



- **Find the Names of Customers on streets with Names ending in "Hill"**

use cse370;

select customer_name from customer where customer_street like '%Hill';

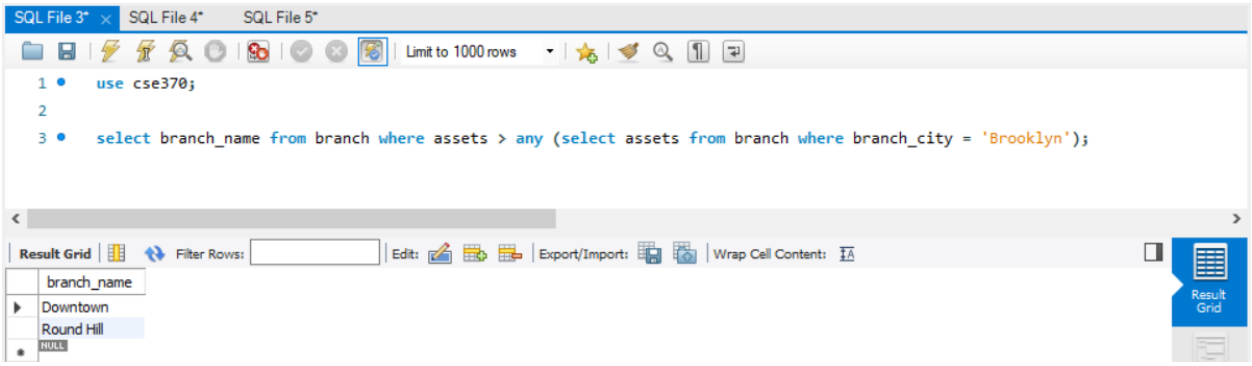


- **Find the Names of Branches whose assets are greater than the assets of some Branch in Brooklyn**

use cse370;

select branch_name from branch where assets > any (select assets from branch where branch_city = 'Brooklyn');

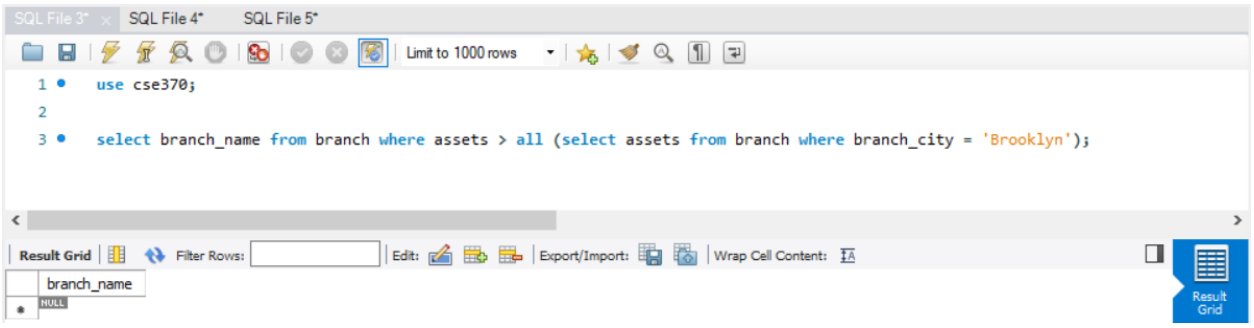
[Booklyn has two Branches Downtown (90000000) and Brighton (7100000). Here the Downtown branch and Round Hill (8000000) Branch is bigger than any of the two Branches which is Brighton.]



use cse370;

select branch_name from branch where assets > all (select assets from branch where branch_city = 'Brooklyn');

[In terms of 'all' both Downtown (90000000) and Brighton (7100000) have to considered and no other branch has more assets than Downtown Branch. That's why result is null]

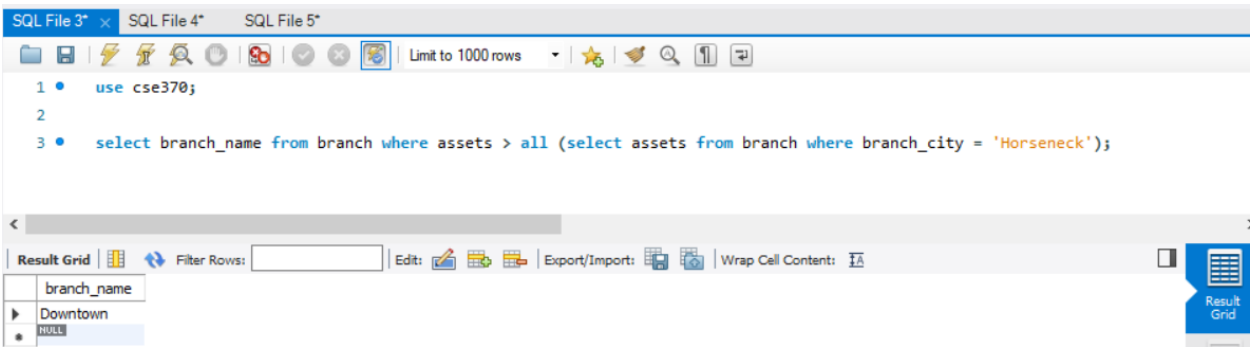


- **<u>Find the set of Names of branches whose Assets are greater than the Assets of all branches in Horseneck</u>**

use cse370;

select branch_name from branch where assets > all (select assets from branch where branch_city = 'Horseneck');

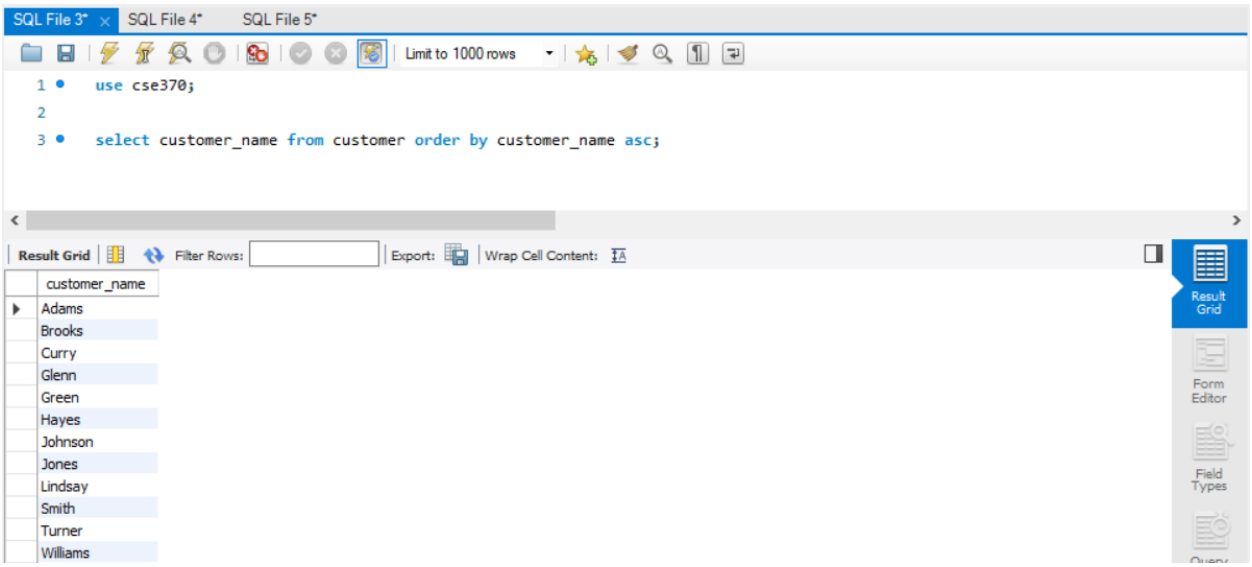
[There are three Branches in Horseneck City Mianus (400000), Perryridge (1700000) and RoundHill (8000000). Since 'all' is used all Branches need to be considered. Only other Branch that has more assets than RoundHill is Downtown Branch]



- **Find the set of names of customers at Brighton branch, in alphabetical order**

use cse370;

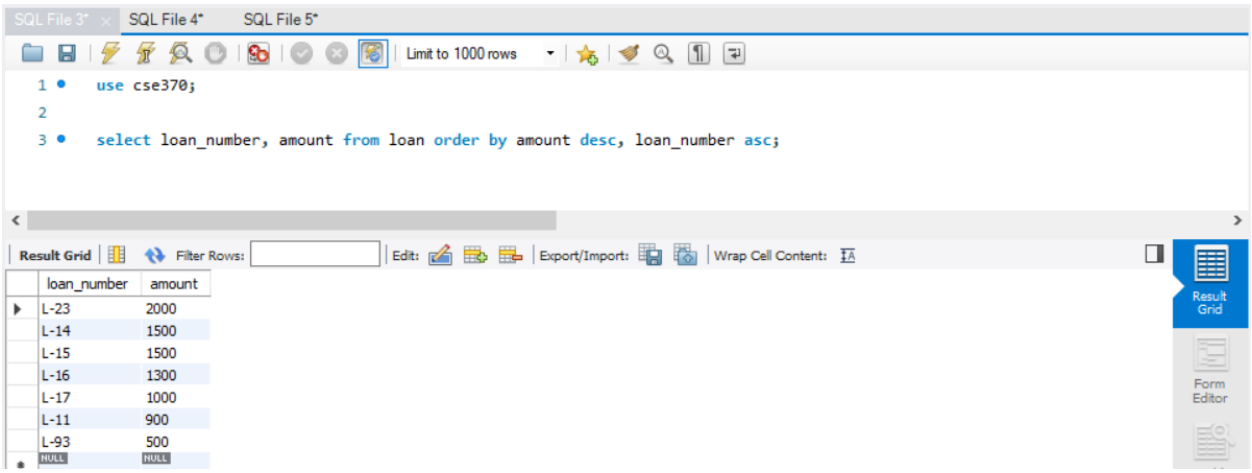
select customer_name from customer order by customer_name asc;



- **Show the loan data, ordered by decreasing Amounts, then increasing Loan numbers**

use cse370;

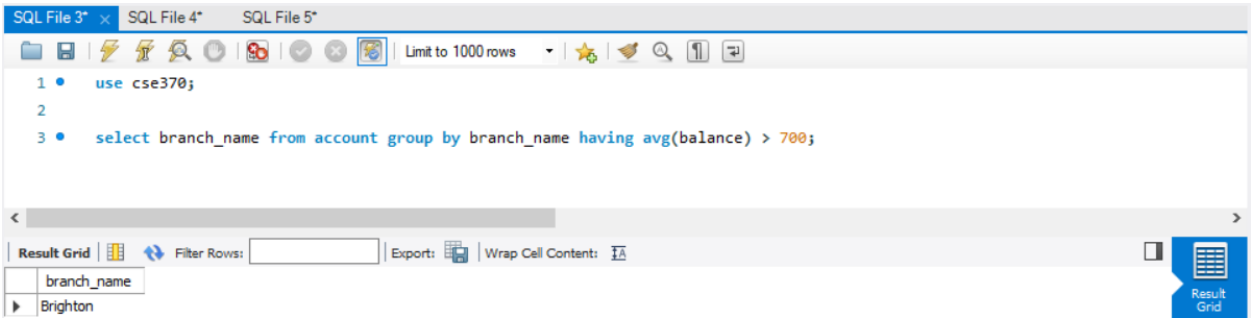
select loan_number, amount from loan order by amount desc, loan_number asc;



- **Find the Names of Branches having at least one Account, with average Balances greater than or equal 700**

use cse370;

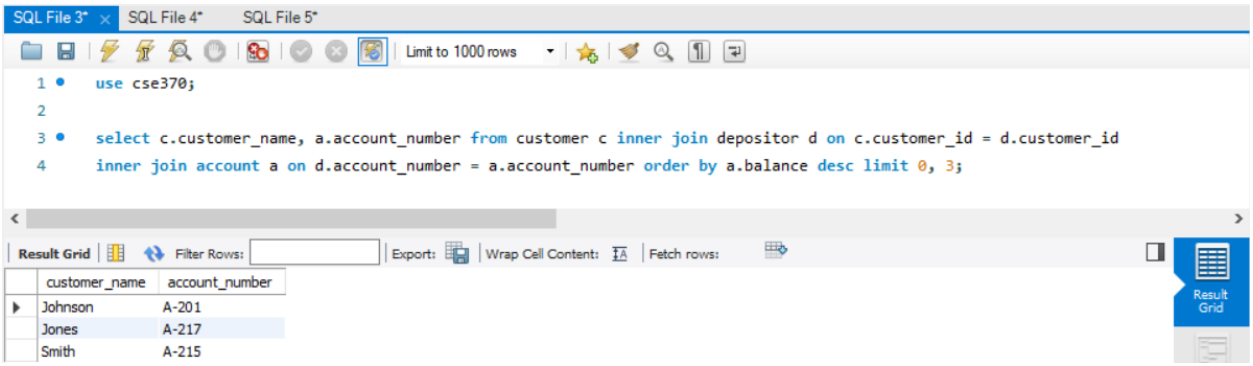
select branch_name from account group by branch_name having avg(balance) > 700;



- **Find the Names and Account number of Customers who have the 3 highest Balances in their Accounts**

use cse370;

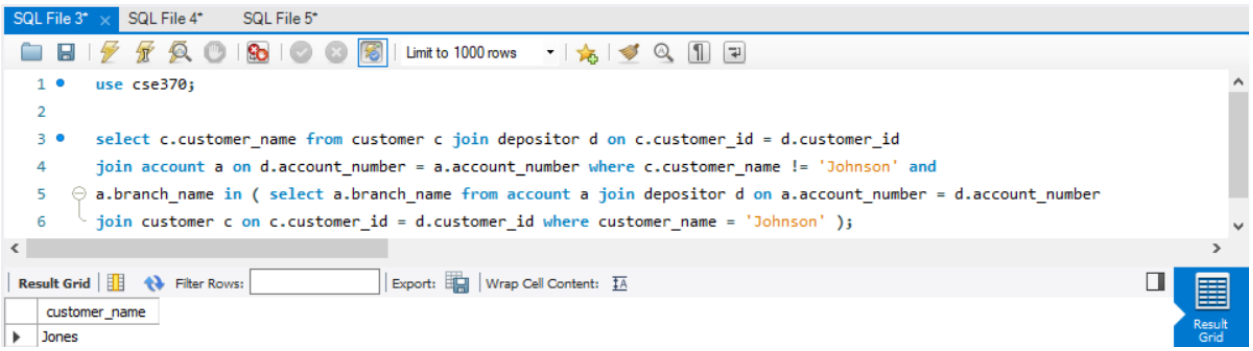
select c.customer_name, a.account_number from customer c inner join depositor d on c.customer_id = d.customer_id inner join account a on d.account_number = a.account_number order by a.balance desc limit 0, 3;



- **Find the Names of Customers with Accounts at a Branch where Johnson has an Account**

use cse370;

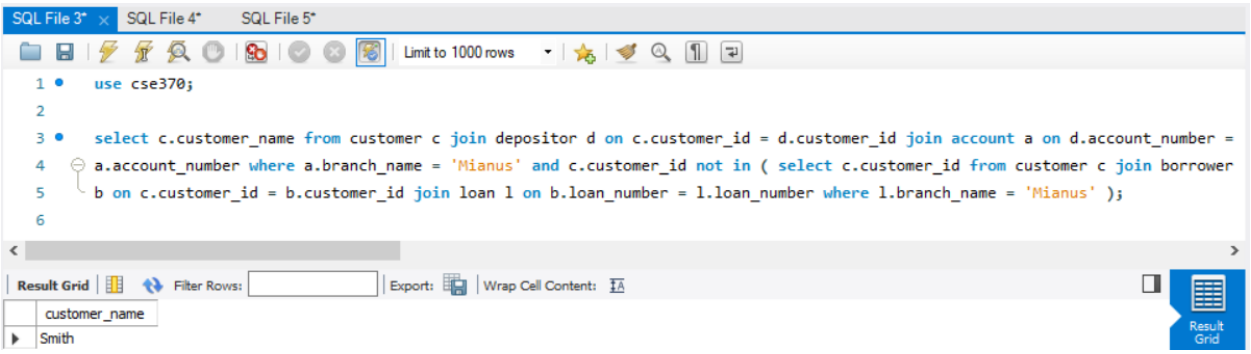
select c.customer_name from customer c join depositor d on c.customer_id = d.customer_id join account a on d.account_number = a.account_number where c.customer_name != 'Johnson' and a.branch_name in (select a.branch_name from account a join depositor d on a.account_number = d.account_number join customer c on c.customer_id = d.customer_id where customer_name = 'Johnson');



- **Find the Names of Customers with an Account on Mianus but not a Loan at Mianus branch**

use cse370;

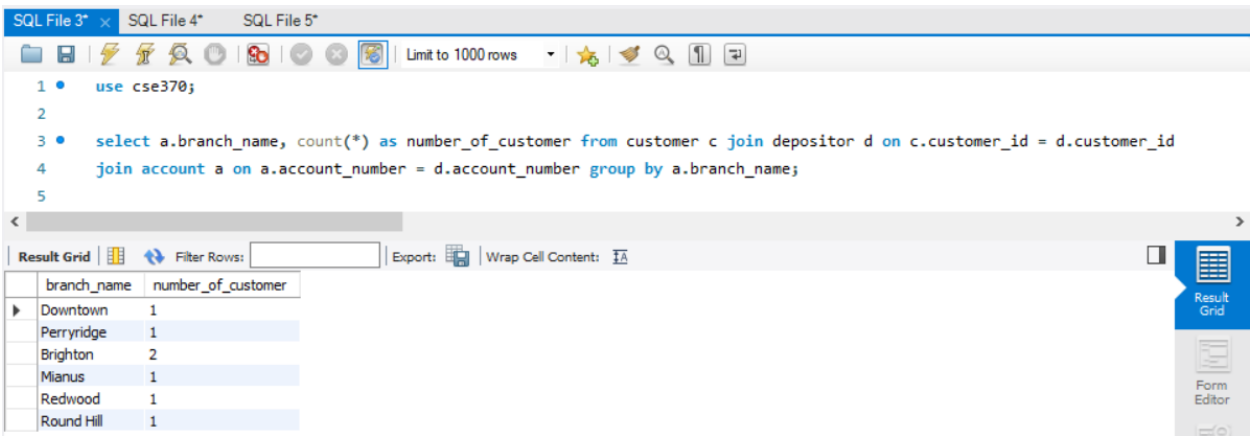
select c.customer_name from customer c join depositor d on c.customer_id = d.customer_id join account a on d.account_number = a.account_number where a.branch_name = 'Mianus' and c.customer_id not in (select c.customer_id from customer c join borrower b on c.customer_id = b.customer_id join loan l on b.loan_number = l.loan_number where l.branch_name = 'Mianus');



- **Find the Names of each Branch and the number of Customers having at least one Account at that Branch**

use cse370;

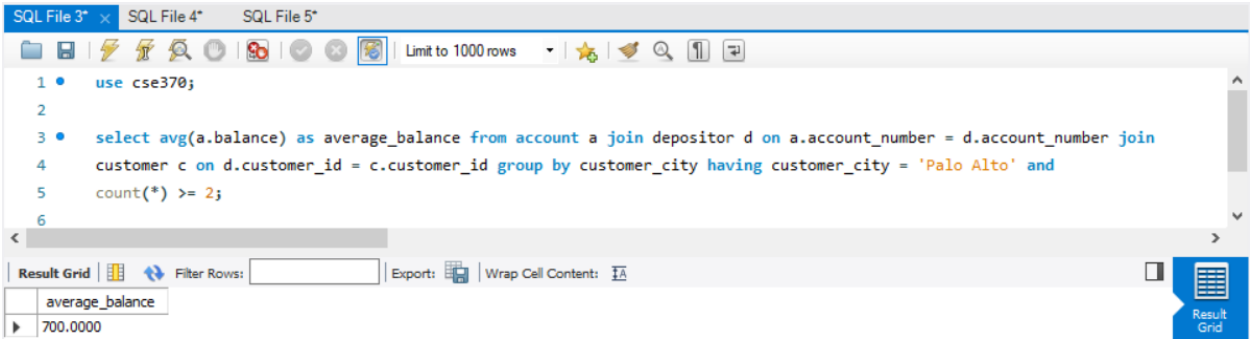
select a.branch_name, count(*) as number_of_customer from customer c join depositor d on c.customer_id = d.customer_id join account a on a.account_number = d.account_number group by a.branch_name;



- **Find the average Balance of all Customers in 'Palo Alto' having at least 2 Accounts**

use cse370;

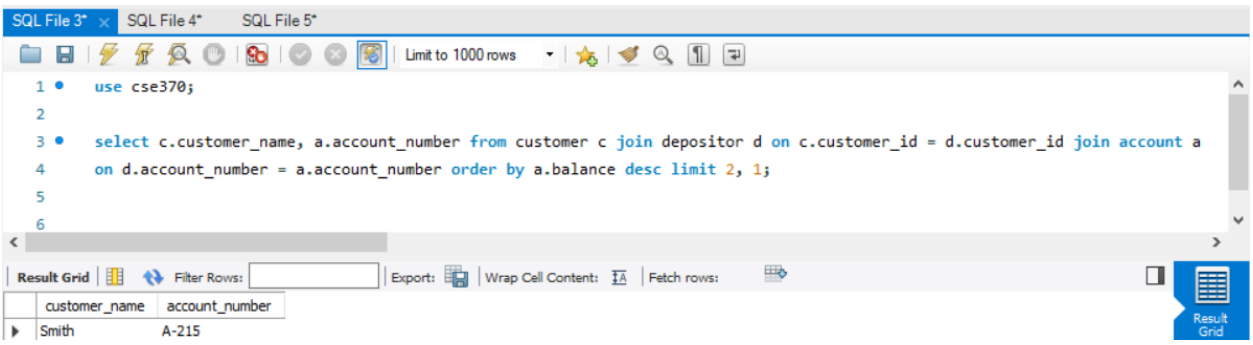
select avg(a.balance) as average_balance from account a join depositor d on a.account_number = d.account_number join customer c on d.customer_id = c.customer_id group by customer_city having customer_city = 'Palo Alto' and count(*) >= 2;



- **Find the Name and Account number of the customer who has the 3rd highest Balance in their Account**

use cse370;

select c.customer_name, a.account_number from customer c join depositor d on c.customer_id = d.customer_id join account a on d.account_number = a.account_number order by a.balance desc limit 2, 1;



Thank You