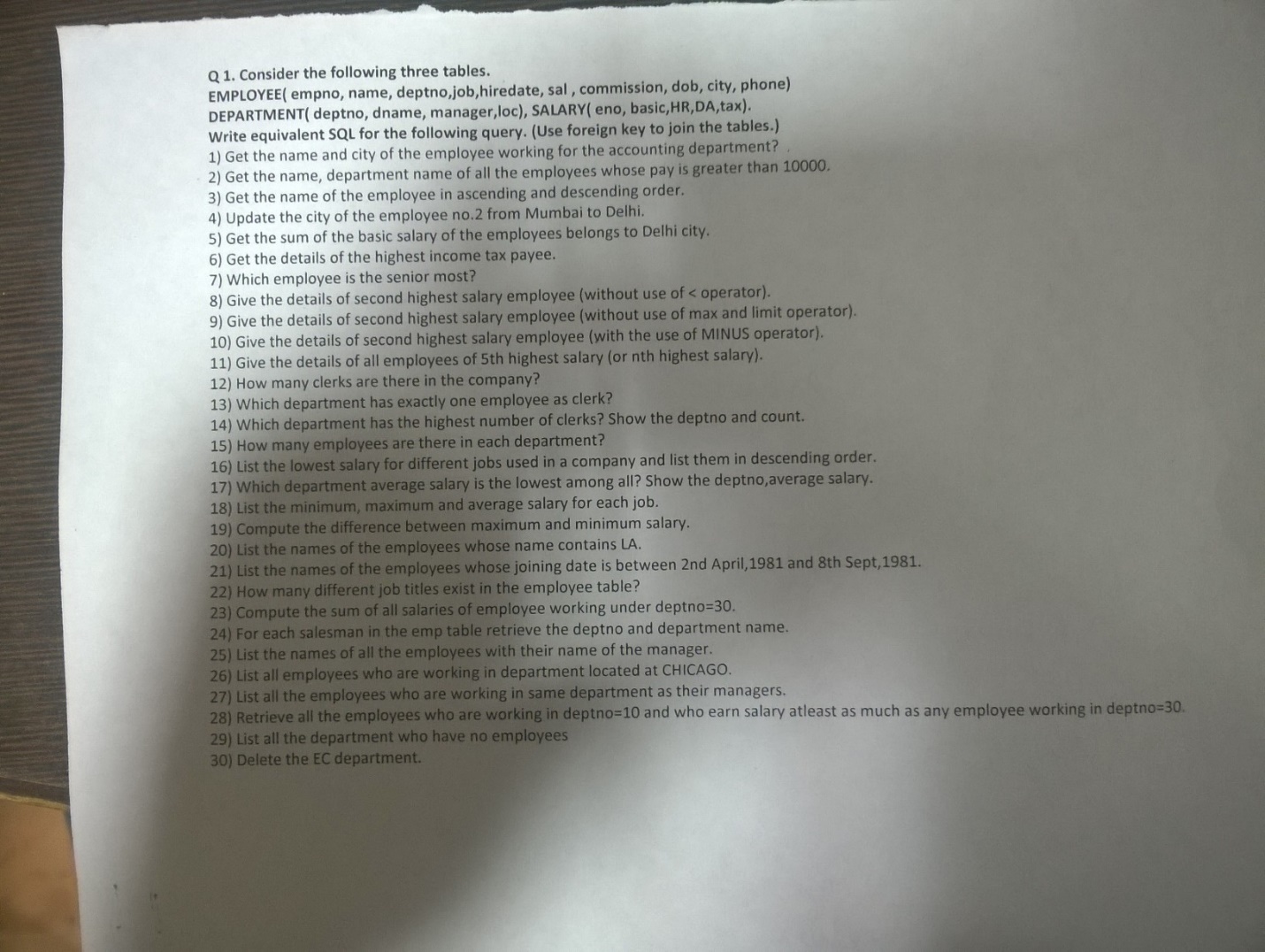
# Assignment 1:



-- Create the database if it does not exist and use it

CREATE DATABASE IF NOT EXISTS `Ashhar13BCS0015 - Q1`;

USE `Ashhar13BCS0015 - Q1`;

-- Delete existing tables

DROP TABLE IF EXISTS salary0015;

DROP TABLE IF EXISTS employee0015;

DROP TABLE IF EXISTS department0015;

-- Create Table Department

CREATE TABLE department0015 (

deptno INT PRIMARY KEY AUTO\_INCREMENT,

dname NVARCHAR(20),

manager NVARCHAR(20),

loc NVARCHAR(20)

);

-- Create Table Employee

CREATE TABLE employee0015 (

empno INT PRIMARY KEY AUTO\_INCREMENT,

name NVARCHAR(20) NOT NULL,

deptno INT NOT NULL,

job NVARCHAR(50) NOT NULL,

hiredate DATE,

sal INT,

commission INT,

dob DATE,

city NVARCHAR(20) NOT NULL,

phone INT,

FOREIGN KEY(deptno) REFERENCES department0015(deptno)

ON UPDATE CASCADE ON DELETE CASCADE

);

-- Create Table Salary

CREATE TABLE salary0015 (

eno INT,

basic INT,

HR INT,

DA INT,

tax INT,

FOREIGN KEY(eno) REFERENCES employee0015(empno)

ON UPDATE CASCADE ON DELETE CASCADE

);

INSERT INTO department0015 VALUES

(1, 'IT', 'Ashhar', 'Delhi'),

(2, 'CIVIL', 'Anas', 'Mumbai'),

(3, 'ELEC', 'Amit', 'Delhi'),

(4, 'HR', 'Aamir', 'Kanpur'),

(5, 'AC', 'Aayush', 'Kanpur'),

(6, 'Dummy', 'Random', 'Chicago');

INSERT INTO employee0015 VALUES

(1, 'Ashhar', 1, 'Design', '2013-01-01', 100000, 10000, '1995-09-07', 'Delhi', NULL),

(2, 'Anas', 2, 'Structure', '2013-09-07', 50000, 5000, '1995-08-08', 'Mumbai', NULL),

(3, 'Amit', 3, 'Drives', '2013-07-13', 80000, 8000, '1995-07-24', 'Delhi', NULL),

(4, 'Aamir', 4, 'Clerk', '2012-06-17', 60000, 6000, '1994-09-29', 'Kanpur', NULL),

(5, 'Aayush', 5, 'Clerk', '2013-02-18', 60000, 6000, '1995-07-14', 'Kanpur', NULL),

(6, 'Basit', 2, 'Manager', '2013-01-11', 38000, 3800, '1995-07-12', 'Delhi', NULL),

(7, 'Sonaal', 1, 'Dev', '2013-01-01', 30000, 3000, '1995-05-06', 'Delhi', NULL),

(8, 'Gaurav', 4, 'Manager', '2013-09-07', 18000, 1800, '1995-04-09', 'Delhi', NULL),

(9, 'Random', 2, 'Clerk', '2010-01-01', 20000, 1000, '1990-05-06', 'Chicago', NULL);

INSERT INTO salary0015 VALUES

(1, 100000, 1000, 2000, 3000),

(2, 50000, 8000, 2000, 4000),

(3, 80000, 1000, 1000, 1000),

(4, 60000, 5000, 8000, 9000),

(5, 60000, 3000, 5000, 5000),

(6, 38000, 800, 600, 1000),

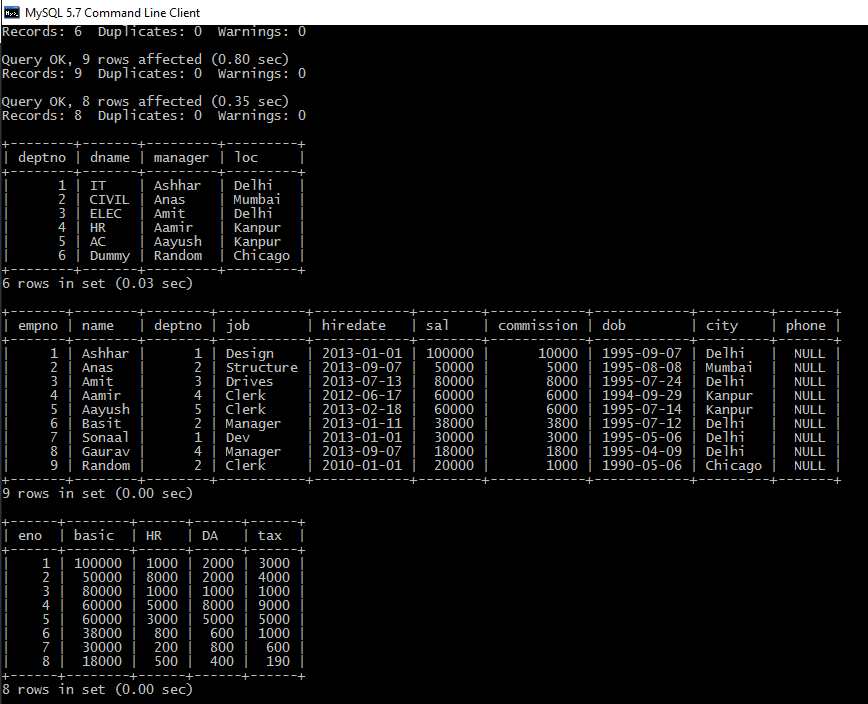
(7, 30000, 200, 800, 600),

(8, 18000, 500, 400, 190);

SELECT \* FROM department0015;

SELECT \* FROM employee0015;

SELECT \* FROM salary0015;



-- QUES 1

SELECT name, city

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE d.dname = 'AC';

-- QUES 2

SELECT name, dname

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE e.sal > 10000;

-- QUES 3

-- Part 1

SELECT name

FROM employee0015

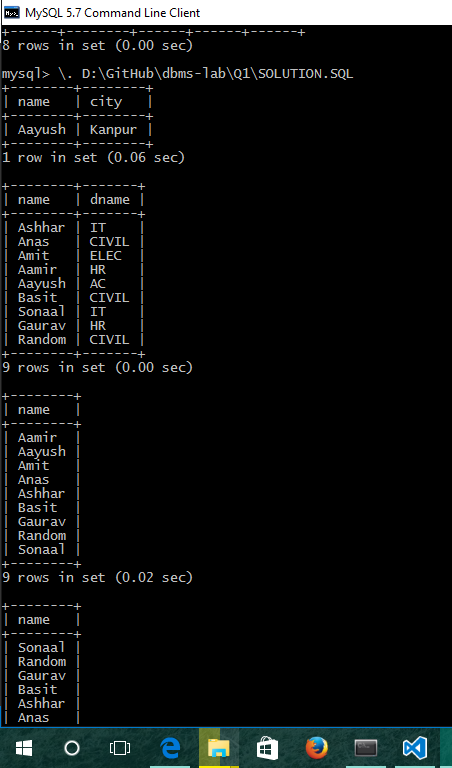
ORDER BY name ASC;

-- Part 2

SELECT name

FROM employee0015

ORDER BY name DESC;



-- QUES 4

UPDATE employee0015

SET city = 'Delhi'

WHERE empno = 2;

-- QUES 5

SELECT SUM(sal) AS 'Total Salary'

FROM employee0015 AS e

WHERE e.city = 'Delhi';

-- QUES 6

SELECT empno, name, deptno, job, hiredate, dob, city, phone,

basic, HR, DA, tax

FROM employee0015 AS e

INNER JOIN salary0015 AS s ON e.empno = s.eno

ORDER BY s.tax DESC LIMIT 1;

-- QUES 7

SELECT \*

FROM employee0015

ORDER BY hiredate DESC LIMIT 1;

-- QUES 8

SELECT \*

FROM employee0015

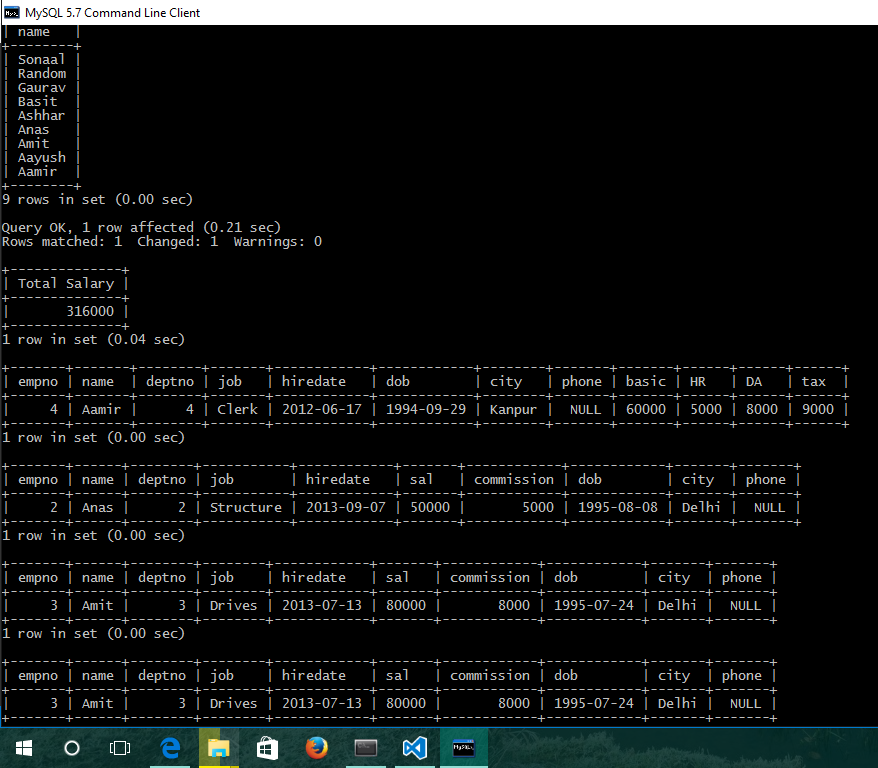
ORDER BY sal DESC LIMIT 1, 1;

-- QUES 9

SELECT \*

FROM employee0015 AS e

WHERE 2 = (SELECT COUNT(DISTINCT sal) FROM employee0015 WHERE e.sal <= sal);



-- TODO: QUES 10 (Second highest using MINUS)

-- QUES 11

SELECT \*

FROM employee0015

ORDER BY sal DESC LIMIT 6, 1;

-- QUES 12

SELECT COUNT(\*) AS 'Total Clerks'

FROM employee0015

WHERE job = 'Clerk';

-- QUES 13

SELECT dname, COUNT(\*) AS 'Total Clerks'

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE e.job = 'Clerk' GROUP BY d.deptno HAVING COUNT(\*) = 1;

-- QUES 14

SELECT d.dname, e.deptno, COUNT(name) AS 'Total Clerks'

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE e.job = 'Clerk' GROUP BY e.deptno ORDER BY name DESC LIMIT 0, 1;

-- QUES 15

SELECT d.dname, COUNT(\*) AS 'Total Employees'

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

GROUP BY d.deptno;



-- QUES 16

SELECT job, MIN(sal) AS 'Lowsest Salary'

FROM employee0015

GROUP BY job ORDER BY sal DESC;

-- QUES 17

SELECT deptno, AVG(sal) AS 'Lowest Average Salary'

FROM employee0015

GROUP BY deptno ORDER BY AVG(sal) ASC LIMIT 1;

-- QUES 18

SELECT job, MIN(sal) AS 'Lowest Salary', MAX(sal) AS 'Highest Salary', AVG(sal) AS 'Average Salary'

FROM employee0015

GROUP BY job;

-- QUES 19

SELECT MAX(sal) - MIN(sal) AS 'Difference between Maximum and Minimum Salaries'

FROM employee0015;

-- QUES 20

SELECT name AS 'Names starting with Aay'

FROM employee0015

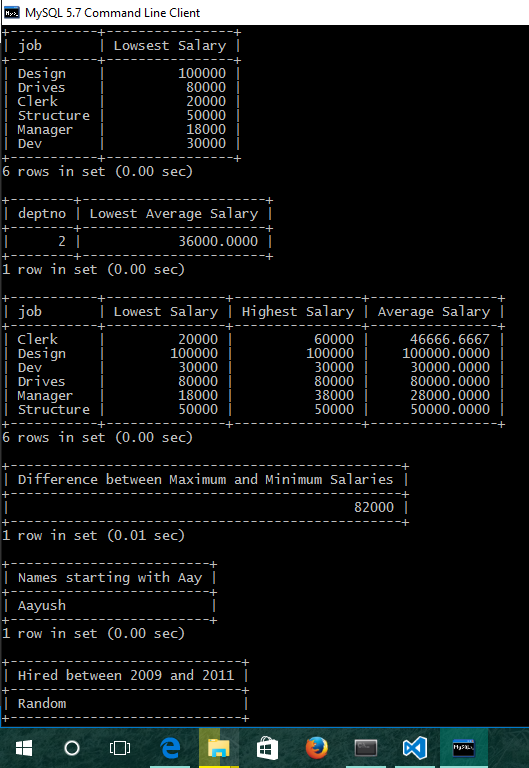
WHERE name LIKE 'Aay%';

-- QUES 21

SELECT name AS 'Hired between 2009 and 2011'

FROM employee0015

WHERE hiredate BETWEEN '2009-04-02' AND '2011-09-08';



-- QUES 22

SELECT job

FROM employee0015

GROUP BY job;

-- QUES 23

SELECT SUM(sal) AS 'Total salaries earned by department 3'

FROM employee0015

WHERE deptno = 3;

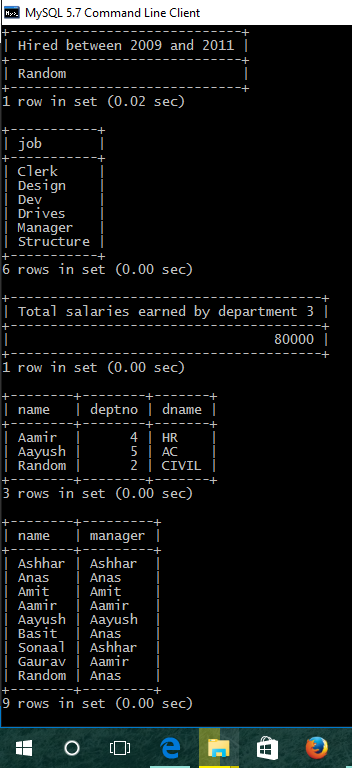
-- QUES 24

SELECT e.name, e.deptno, d.dname

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE e.job = 'Clerk';



-- QUES 25

SELECT e.name, d.manager

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno;

-- QUES 26

SELECT e.name

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE d.loc = 'Delhi';

-- QUES 27

SELECT e.name, d.dname

FROM employee0015 AS e

INNER JOIN department0015 AS d ON e.deptno = d.deptno

WHERE e.job = 'Manager';

-- QUES 28

SELECT name

FROM employee0015

WHERE deptno = 1 AND sal >= ANY(SELECT sal FROM employee0015 WHERE deptno = 3);

-- QUES 29

SELECT deptno, dname AS 'Empty Department'

FROM department0015

WHERE deptno NOT IN(SELECT deptno FROM employee0015);

-- QUES 30

ALTER TABLE employee0015

DROP FOREIGN KEY employee0015\_ibfk\_1;

ALTER TABLE salary0015 DROP

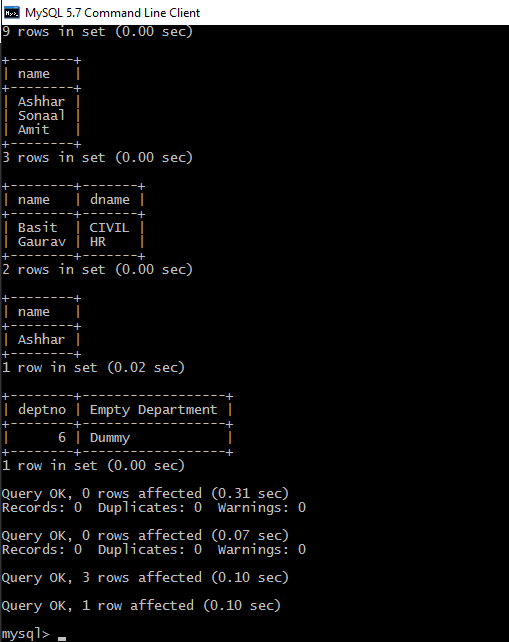
FOREIGN KEY salary0015\_ibfk\_1;

DELETE FROM employee0015

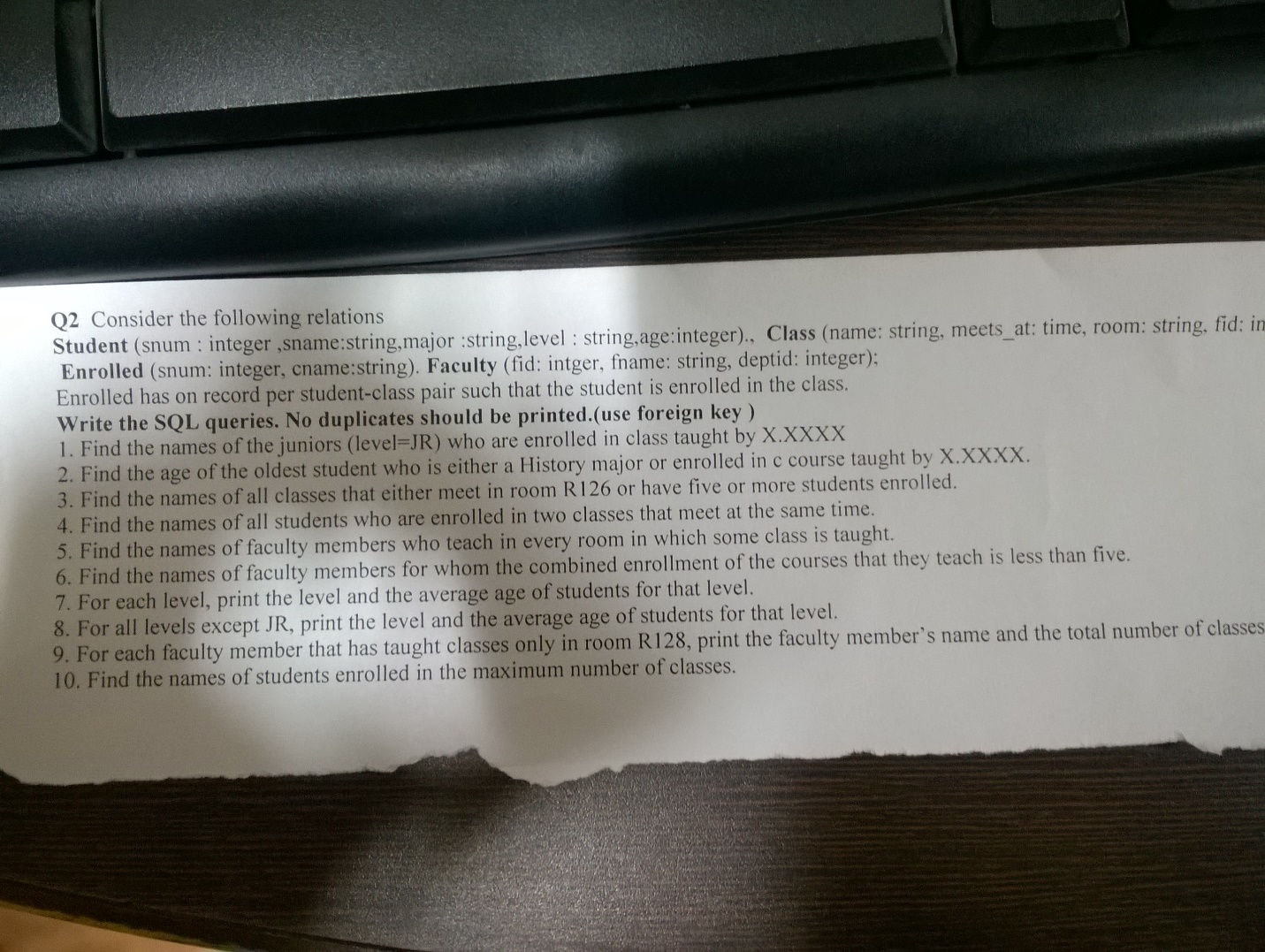
WHERE deptno = 2;

DELETE FROM department0015

WHERE deptno = 2;



# Assignment 2:



-- Create the database if it does not exist and use it

CREATE DATABASE IF NOT EXISTS `Ashhar13BCS0015 - Q2`;

USE `Ashhar13BCS0015 - Q2`;

-- Delete existing tables

DROP TABLE IF EXISTS enrolled0015;

DROP TABLE IF EXISTS class0015;

DROP TABLE IF EXISTS student0015;

DROP TABLE IF EXISTS faculty0015;

-- Create Table Student

CREATE TABLE student0015 (

snum INT PRIMARY KEY AUTO\_INCREMENT,

sname NVARCHAR(30) NOT NULL,

major NVARCHAR(30) NOT NULL,

level NVARCHAR(10) NOT NULL,

age INT NOT NULL

);

-- Create Table Faculty

CREATE TABLE faculty0015 (

fid INT PRIMARY KEY AUTO\_INCREMENT,

fname NVARCHAR(30) NOT NULL,

deptid INT NOT NULL

);

-- Create Table Class

CREATE TABLE class0015 (

cname NVARCHAR(20) PRIMARY KEY,

meetsat NVARCHAR(10) NOT NULL,

room NVARCHAR(10) NOT NULL,

fid INT NOT NULL,

FOREIGN KEY(fid) REFERENCES faculty0015(fid)

);

-- Create Table Enrolled

CREATE TABLE enrolled0015 (

snum INT NOT NULL,

cname NVARCHAR(20) NOT NULL,

FOREIGN KEY(snum) REFERENCES student0015(snum),

FOREIGN KEY(cname) REFERENCES class0015(cname)

);

INSERT INTO student0015 VALUES

(101, 'Helen', 'CS', 'JR', 19),

(102, 'Charles', 'CS', 'SR', 21),

(103, 'Andy', 'CS', 'GR', 25),

(104, 'Bob', 'CS', 'SR', 23),

(105, 'Zorba', 'CS', 'GR', 31);

INSERT INTO faculty0015 VALUES

(201, 'S. Jackson', 301),

(202, 'M. Shanks', 301),

(203, 'I. Teach', 302);

INSERT INTO class0015 VALUES

('CSC343', 'W1', 'BA1080', 201),

('CSC443', 'T2', 'BA1170', 202),

('ECE300', 'M1', 'BA1080', 203),

('ECE201', 'F12', 'BA1060', 203),

('CSC165', 'R3', 'BA1170', 202),

('ECE301', 'T2', 'BA1080', 201);

INSERT INTO enrolled0015 VALUES

(101, 'CSC343'),

(101, 'CSC443'),

(101, 'ECE300'),

(102, 'CSC343'),

(102, 'ECE201'),

(103, 'CSC343'),

(103, 'CSC443'),

(103, 'ECE300'),

(103, 'ECE201'),

(105, 'CSC343'),

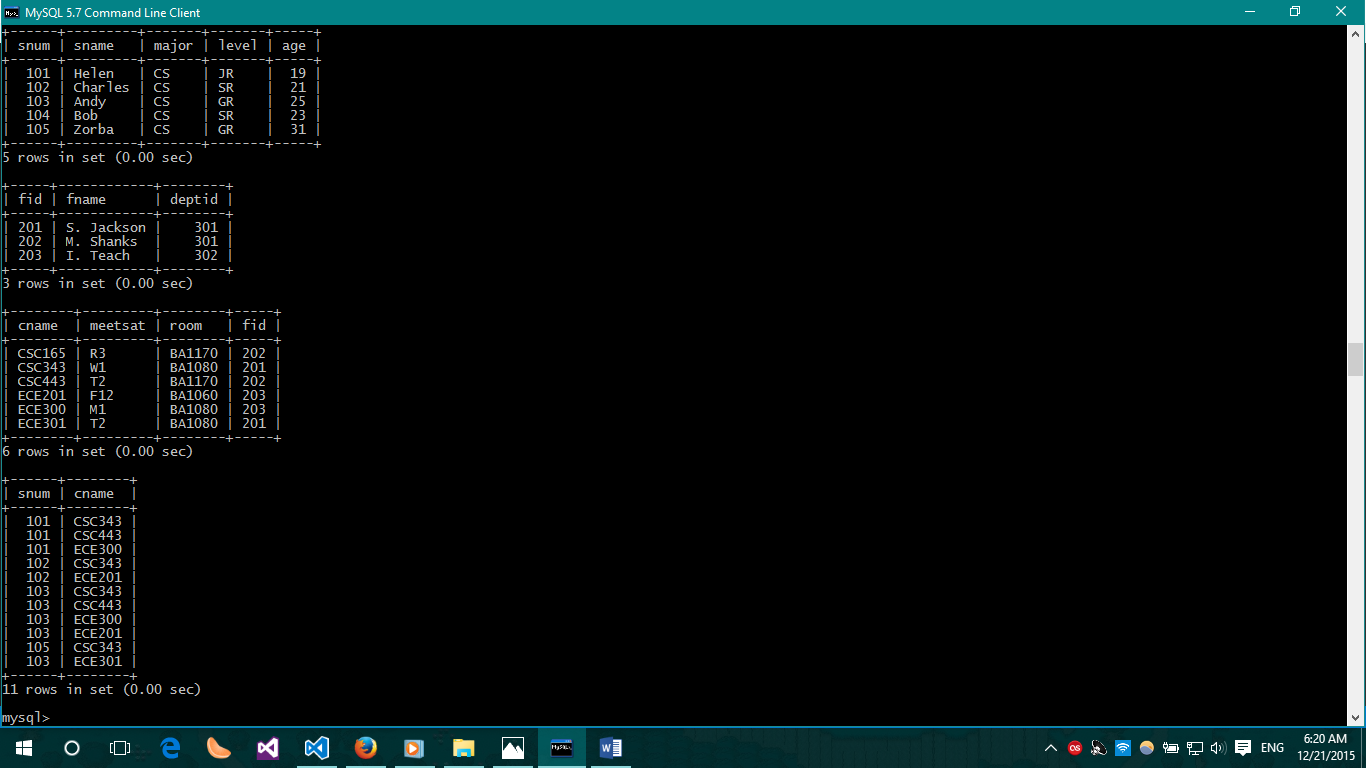
(103, 'ECE301');

SELECT \* FROM student0015;

SELECT \* FROM faculty0015;

SELECT \* FROM class0015;

SELECT \* FROM enrolled0015;



-- QUES 1

SELECT sname

FROM student0015 AS s

INNER JOIN enrolled0015 AS e ON s.snum = e.snum

INNER JOIN class0015 AS c ON c.cname = e.cname

INNER JOIN faculty0015 AS f ON f.fid = c.fid

WHERE level = 'JR' AND fname = 'S. Jackson';

-- QUES 2

SELECT sname, age

FROM student0015 AS s

INNER JOIN enrolled0015 AS e ON s.snum = e.snum

INNER JOIN class0015 AS c ON e.cname = c.cname

INNER JOIN faculty0015 AS f ON f.fid = c.fid

WHERE e.cname = 'CSC443' AND f.fname = 'M. Shanks' ORDER BY age DESC LIMIT 1;

-- QUES 3

SELECT c.cname

FROM class0015 AS c

INNER JOIN enrolled0015 AS e ON c.cname = e.cname

WHERE (room = 'R126' OR e.cname IN (SELECT cname FROM enrolled0015 HAVING COUNT(e.cname) >= 5)) GROUP BY e.cname;

-- QUES 4

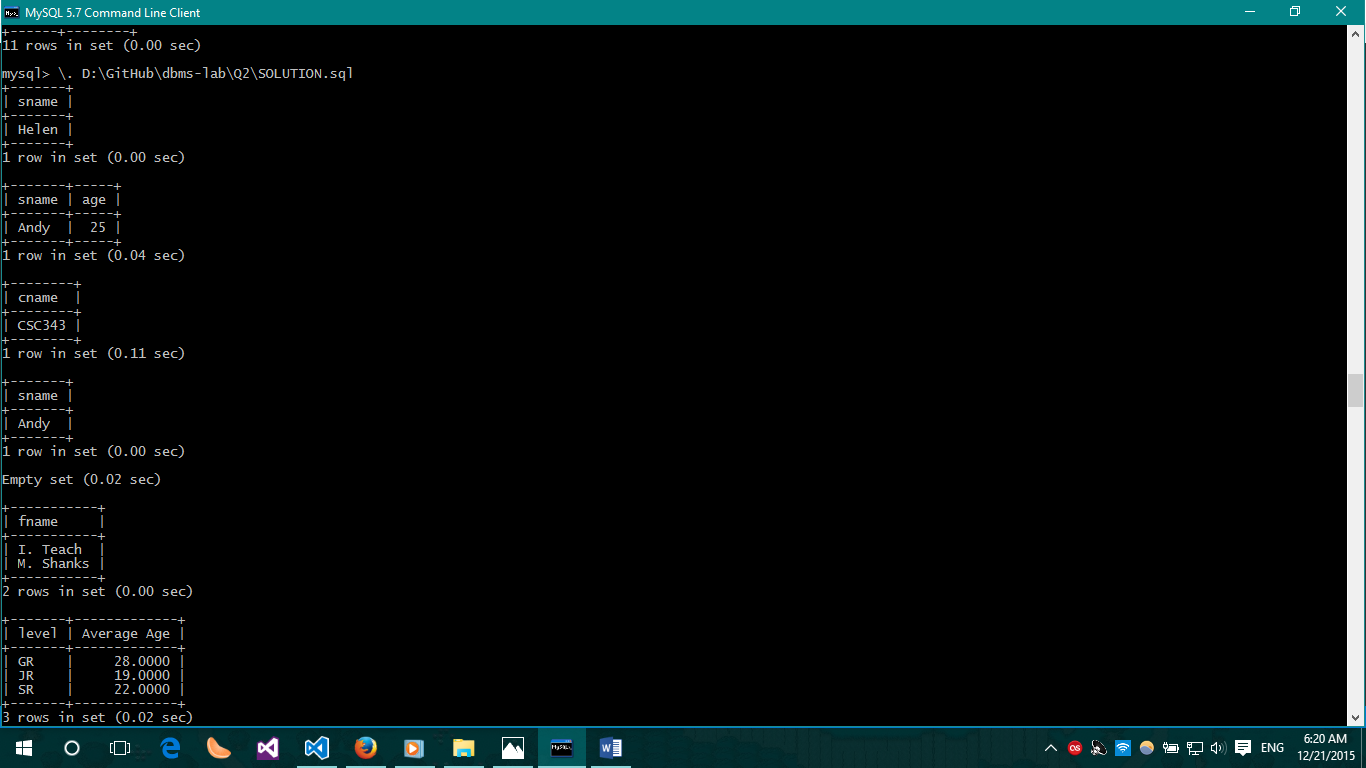
SELECT DISTINCT s.sname

FROM student0015 AS s

INNER JOIN enrolled0015 AS e ON s.snum = e.snum

INNER JOIN class0015 AS c ON c.cname = e.cname

GROUP BY c.meetsat, e.snum HAVING COUNT(e.cname) > 1;



-- QUES 5

SELECT fname

FROM faculty0015

WHERE fid IN(SELECT fid FROM class0015 GROUP BY fid HAVING COUNT(\*) = (SELECT COUNT(distinct(room)) FROM class0015));

-- QUES 6

SELECT f.fname

FROM class0015 AS c

INNER JOIN faculty0015 AS f ON f.fid = c.fid

INNER JOIN enrolled0015 AS e ON c.cname = e.cname

GROUP BY fname HAVING COUNT(\*) < 5;

-- QUES 7

SELECT level, AVG(age) AS 'Average Age'

FROM student0015

GROUP BY level;

-- QUES 8

SELECT level, AVG(age) AS 'Average Age'

FROM student0015

WHERE level NOT LIKE 'JR';

-- QUES 9

SELECT f.fname, COUNT(c.cname) AS 'No. of Classes'

FROM faculty0015 AS f

INNER JOIN class0015 AS c ON c.fid = f.fid

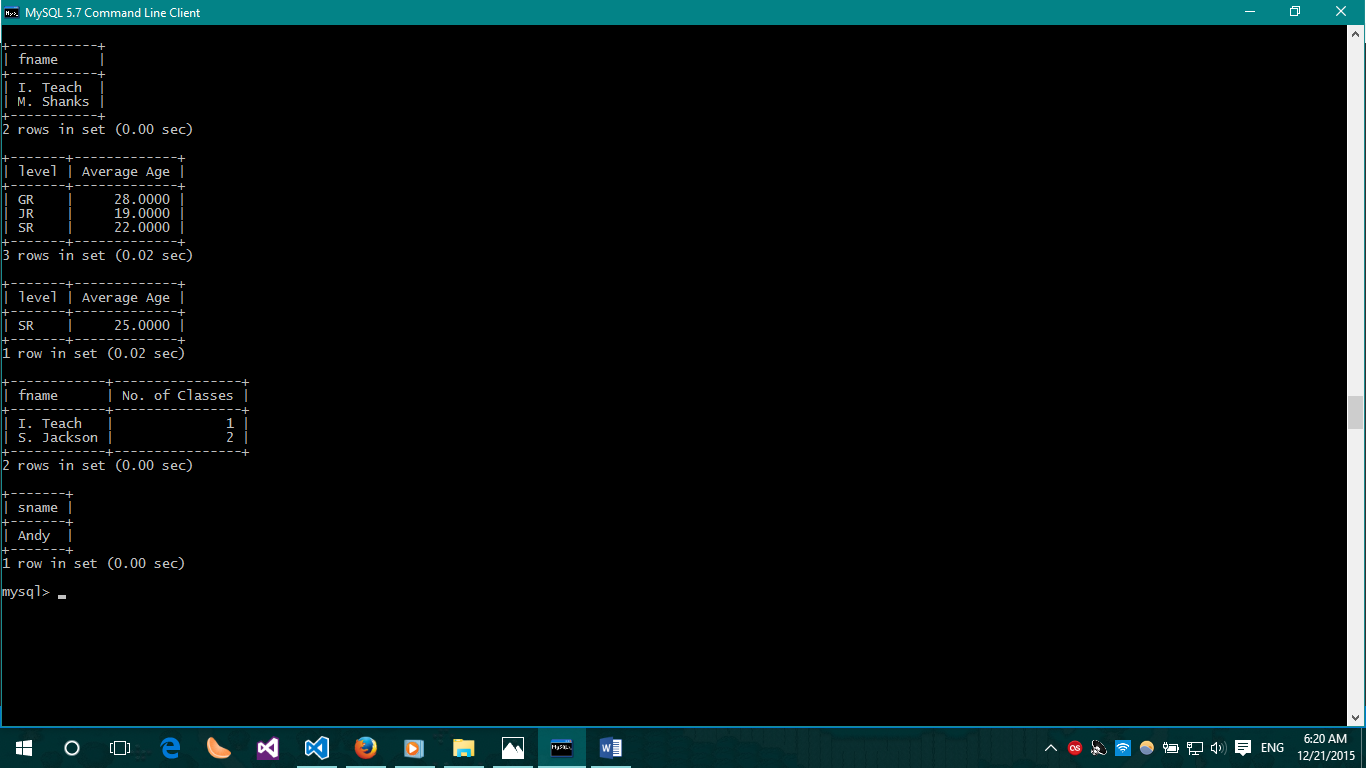
WHERE c.room = 'BA1080' GROUP BY f.fname;

-- QUES 10

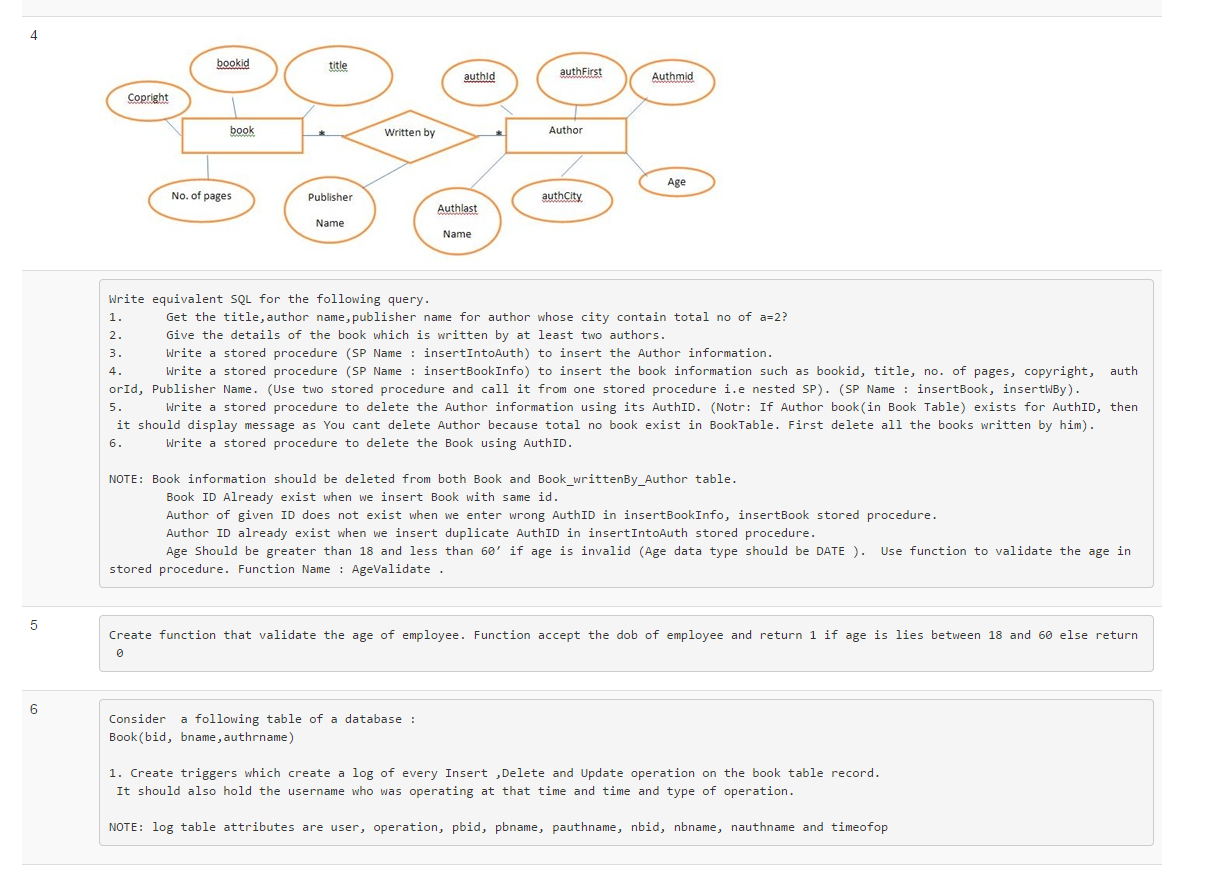
SELECT DISTINCT s.sname

FROM student0015 AS s

WHERE s.snum IN (SELECT e1.snum FROM enrolled0015 e1 GROUP BY e1.snum HAVING COUNT(\*) >= ALL (SELECT COUNT(\*) FROM enrolled0015 e2 GROUP BY e2.snum));



# Assignment 3:



-- Create the database IF it does not exist AND use it

CREATE DATABASE IF NOT EXISTS `Ashhar13BCS0015 - Q3`;

USE `Ashhar13BCS0015 - Q3`;

-- Delete existing tables

DROP TABLE IF EXISTS writtenby0015;

DROP TABLE IF EXISTS book0015;

DROP TABLE IF EXISTS author0015;

DROP TABLE IF EXISTS log0015;

-- Delete Procedures and Functions

DROP PROCEDURE insertIntoAuth0015;

DROP PROCEDURE insertBookInfo0015;

DROP PROCEDURE deleteAuthById0015;

DROP PROCEDURE deleteBookByAuthId0015;

DROP FUNCTION AgeValidate0015;

DROP TRIGGER trig\_insbook0015;

-- Create Table Book

CREATE TABLE book0015 (

bookid INT PRIMARY KEY AUTO\_INCREMENT,

title NVARCHAR(100),

copyright NVARCHAR(50),

pages INT

);

-- Create Table Faculty

CREATE TABLE author0015 (

authid INT PRIMARY KEY AUTO\_INCREMENT,

authFirst NVARCHAR(50),

authMid NVARCHAR(50),

authLast NVARCHAR(50),

age DATE,

authCity NVARCHAR(50)

);

-- Create Table WrittenBy

CREATE TABLE writtenby0015 (

publisher NVARCHAR(50),

bookid INT,

authid INT,

FOREIGN KEY (bookid) REFERENCES book0015(bookid) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (authid) REFERENCES author0015(authid) ON DELETE CASCADE ON UPDATE CASCADE

);

INSERT INTO book0015 VALUES

(1, 'Book 1', 'Copyright 1', 100),

(2, 'Book 2', 'Copyright 2', 120),

(3, 'Book 3', 'Copyright 3', 130),

(4, 'Book 4', 'Copyright 4', 140),

(5, 'Book 5', 'Copyright 1', 150),

(6, 'Book 6', 'Copyright 2', 160),

(7, 'Book 7', 'Copyright 3', 170),

(8, 'Book 8', 'Copyright 4', 180),

(9, 'Book 9', 'Copyright 5', 190),

(10, 'Book 10', 'Copyright 1', 200);

INSERT INTO author0015 VALUES

(1, 'Alan Wake', NULL, NULL, '1995-11-07', 'New York'),

(2, 'Dan Brown', NULL, NULL, '1993-11-07', 'New York'),

(3, 'Jeffrey Archer', NULL, NULL, '1991-11-07', 'Chicago'),

(4, 'Premchand', NULL, NULL, '1985-11-07', 'Patna'),

(5, 'Gulzar', NULL, NULL, '1975-11-07', 'Delhi');

INSERT INTO writtenby0015 VALUES

('Publisher 1', 1, 1),

('Publisher 1', 2, 1),

('Publisher 1', 3, 1),

('Publisher 2', 4, 2),

('Publisher 3', 5, 2),

('Publisher 4', 6, 3),

('Publisher 4', 7, 3),

('Publisher 4', 8, 4),

('Publisher 1', 9, 5),

('Publisher 1', 10, 5),

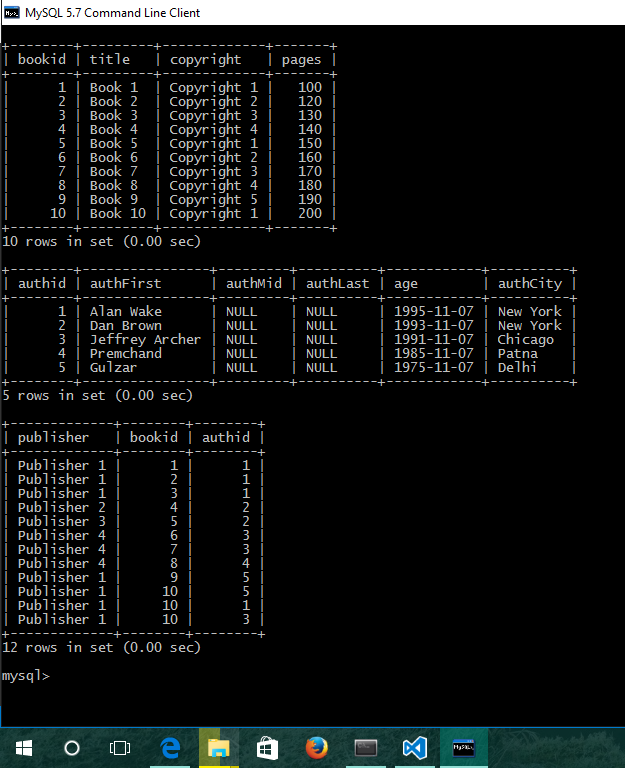
('Publisher 1', 10, 1),

('Publisher 1', 10, 3);

SELECT \* FROM book0015;

SELECT \* FROM author0015;

SELECT \* FROM writtenby0015;



-- QUES 1

SELECT \*

FROM (SELECT authCity, authFirst FROM author0015) AS a

GROUP BY authCity HAVING COUNT(authCity) = 2;

-- QUES 2

SELECT b.bookid, b.copyright, b.pages, w.publisher

FROM book0015 AS b

INNER JOIN writtenby0015 AS w ON b.bookid = w.bookid

GROUP BY w.bookid HAVING COUNT(\*) > 1;

-- QUES 3

DELIMITER //

CREATE PROCEDURE insertIntoAuth0015(IN authid INT, IN authFirst NVARCHAR(50), IN authMid NVARCHAR(50), IN age DATE, IN authCity NVARCHAR(50), IN authLast NVARCHAR(50))

BEGIN

DECLARE verifyAuthid INT DEFAULT 0;

SELECT COUNT(\*) INTO verifyAuthid

FROM author0015 AS a

WHERE a.authid = authid;

IF verifyAuthid = 0

THEN

INSERT INTO author0015 VALUES(authid, authFirst, authMid, authLast, age, authCity);

ELSE

SELECT 'This AuthorID already exists!';

END IF;

END//

DELIMITER ;

-- QUES 4

DELIMITER //

CREATE PROCEDURE insertBookInfo0015(IN bookid INT, IN title NVARCHAR(100), IN copyright NVARCHAR(50), IN pages INT)

BEGIN

DECLARE verifyBID INT DEFAULT 0;

SELECT COUNT(\*) INTO verifyBID

FROM book0015 AS b

WHERE b.bookid = bookid;

IF verifyBID = 0

THEN

INSERT INTO book0015 VALUES(bookid, title, copyright, pages);

ELSE

SELECT 'This Book ID already exists!';

END IF;

END//

DELIMITER ;

-- QUES 5

DELIMITER //

CREATE PROCEDURE deleteAuthById0015(IN \_authid INT)

BEGIN

DECLARE verifyExists INT DEFAULT 0;

SELECT COUNT(\*) INTO verifyExists

FROM book0015 AS b, writtenby0015 AS w

WHERE b.bookid = w.bookid AND w.authid = \_authid;

IF verifyExists = 0

THEN

DELETE FROM author0015 WHERE author0015.authid = \_authid;

ELSE

SELECT 'There are Books by the author in the Books Table, delete them first';

END IF;

END//

DELIMITER ;

-- QUES 6

DELIMITER //

CREATE PROCEDURE deleteBookByAuthId0015(IN \_authid INT)

BEGIN

DECLARE numBooks INT DEFAULT 0;

DECLARE \_bookid INT DEFAULT 0;

SELECT COUNT(\*) INTO numBooks

FROM book0015 AS b, writtenby0015 AS w

WHERE b.bookid = w.bookid AND w.authid = \_authid;

IF numBooks = 0

THEN

SELECT 'No book by the author exists.';

ELSE

label1: LOOP

IF numBooks > 0

THEN

SELECT bookid INTO \_bookid FROM writtenby0015 WHERE writtenby0015.authid = \_authid LIMIT 1;

DELETE FROM book0015 WHERE bookid = \_bookid;

SET numBooks = numBooks - 1;

ITERATE label1;

ELSE

LEAVE label1;

END IF;

DELETE FROM writtenby0015 WHERE writtenby0015.authid = \_authid;

END LOOP label1;

END IF;

END//

DELIMITER ;

-- QUES 7

DELIMITER //

CREATE FUNCTION AgeValidate0015(age DATE) RETURNS NVARCHAR(100)

BEGIN

DECLARE check\_age INT;

DECLARE message NVARCHAR(100);

SET check\_age = DATEDIFF(CURDATE(), age) / 365.25;

IF check\_age >= 18 AND check\_age <= 60

THEN

SET message = CONCAT('Age is Valid');

ELSE

IF check\_age < 18

THEN

SET message = CONCAT('Age is less than 18');

ELSE

SET message = CONCAT('Age is greater than 60');

END IF;

END IF;

RETURN message;

END//

DELIMITER ;

-- QUES 8

CREATE TABLE log0015 (

op NVARCHAR(30) NOT NULL,

previous NVARCHAR(100),

after NVARCHAR(100),

\_TIMESTAMP TIMESTAMP);

DELIMITER //

CREATE trigger trig\_insbook0015 AFTER INSERT ON book0015

FOR EACH ROW

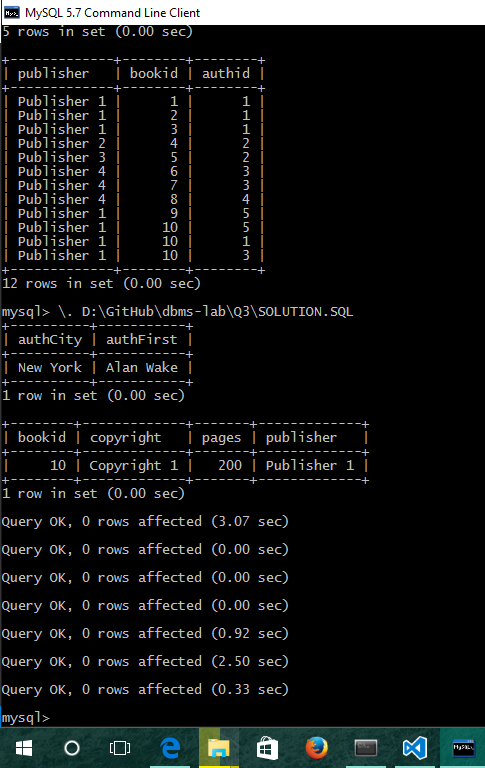
BEGIN

INSERT INTO log0015(op, AFTER, \_TIMESTAMP) VALUES

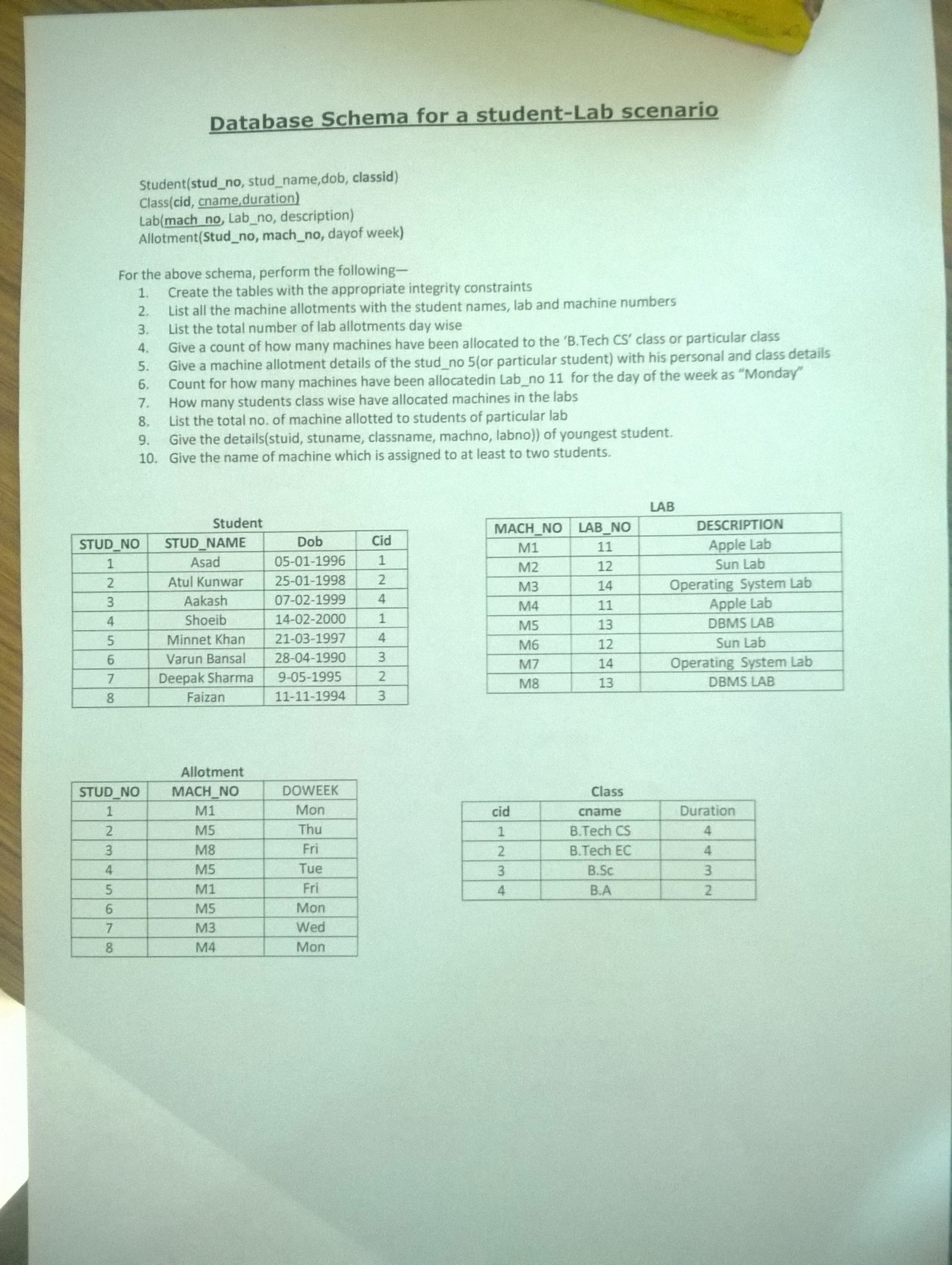
('INSERT', CONCAT(NEW.bookid, ' ', NEW.title, ' ', NEW.copyright, ' ', NEW.pages), current\_timestamp());

END//

DELIMITER ;



# Assignment 4:



-- Create the database IF it does not exist AND use it

CREATE DATABASE IF NOT EXISTS `Ashhar13BCS0015 - Q4`;

USE `Ashhar13BCS0015 - Q4`;

-- Drop tables if they exist

DROP TABLE IF EXISTS Allotment0015;

DROP TABLE IF EXISTS Student0015;

DROP TABLE IF EXISTS Class0015;

DROP TABLE IF EXISTS Lab0015;

-- Create Table Class

CREATE TABLE Class0015 (

cid INT PRIMARY KEY,

cname VARCHAR(50) NOT NULL,

duration INT NOT NULL

);

-- Create Table Student

CREATE TABLE Student0015 (

stud\_no INT PRIMARY KEY,

stud\_name VARCHAR(50) NOT NULL,

dob DATE NOT NULL,

cid INT NOT NULL,

FOREIGN KEY(cid) REFERENCES Class0015(cid) ON UPDATE CASCADE ON DELETE CASCADE

);

-- Create Table Lab

CREATE TABLE Lab0015 (

mach\_no VARCHAR(3) PRIMARY KEY,

lab\_no INT NOT NULL,

description VARCHAR(255)

);

-- Create Table Allotment

CREATE TABLE Allotment0015 (

stud\_no INT NOT NULL,

mach\_no VARCHAR(3) NOT NULL,

doweek VARCHAR(10) NOT NULL,

FOREIGN KEY(stud\_no) REFERENCES Student0015(stud\_no) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(mach\_no) REFERENCES Lab0015(mach\_no) ON UPDATE CASCADE ON DELETE CASCADE

);

INSERT INTO Class0015 VALUES

(1, 'B.Tech CS', 4),

(2, 'B.Tech EC', 4),

(3, 'B.Sc.', 3),

(4, 'B.A.', 2);

INSERT INTO Student0015 VALUES

(1, 'Asad', '1996-01-05', 1),

(2, 'Atul Kunwar', '1998-01-25', 2),

(3, 'Aakash', '1999-02-07', 4),

(4, 'Shoeib', '2000-02-14', 1),

(5, 'Minnet Khan', '1997-03-21', 4),

(6, 'Varun Bansal', '1990-04-28', 3),

(7, 'Deepak Sharma', '1995-05-09', 2),

(8, 'Faizan', '1994-11-11', 3);

INSERT INTO Lab0015 VALUES

('M1', 11, 'Apple Lab'),

('M2', 12, 'Sun Lab'),

('M3', 14, 'Operating System Lab'),

('M4', 11, 'Apple Lab'),

('M5', 13, 'DBMS Lab'),

('M6', 12, 'Sun Lab'),

('M7', 14, 'Operating System Lab'),

('M8', 13, 'DBMS Lab');

INSERT INTO Allotment0015 VALUES

(1, 'M1', 'Mon'),

(2, 'M5', 'Thu'),

(3, 'M8', 'Fri'),

(4, 'M5', 'Tue'),

(5, 'M1', 'Fri'),

(6, 'M5', 'Mon'),

(7, 'M3', 'Wed'),

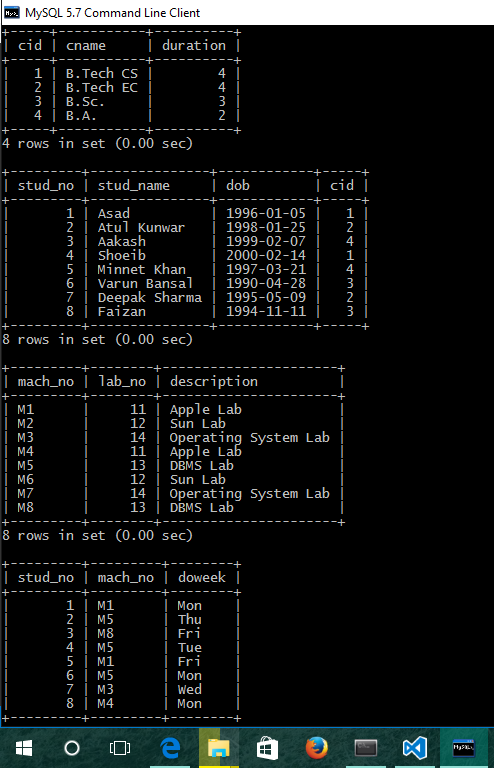
(8, 'M4', 'Mon');

SELECT \* FROM Class0015;

SELECT \* FROM Student0015;

SELECT \* FROM Lab0015;

SELECT \* FROM Allotment0015;



-- QUES 2

SELECT L.mach\_no, S.stud\_name, L.lab\_no, L.description

FROM Allotment0015 AS A

INNER JOIN Student0015 AS S ON S.stud\_no = A.stud\_no

INNER JOIN Lab0015 AS L ON L.mach\_no = A.mach\_no;

-- QUES 3

SELECT doweek, COUNT(\*) AS 'No. of Allotments'

FROM Allotment0015

GROUP BY doweek;

-- QUES 4

SELECT COUNT(\*) AS 'No. of machines allocated to B.Tech CS'

FROM Student0015 AS S

INNER JOIN Allotment0015 AS A ON S.stud\_no = A.stud\_no

INNER JOIN Class0015 AS C ON S.cid = C.cid

WHERE C.cname = 'B.Tech CS';

-- QUES 5

SELECT S.stud\_no, S.stud\_name, S.dob, C.cid, C.cname, C.duration, L.mach\_no, L.lab\_no, L.description, A.doweek

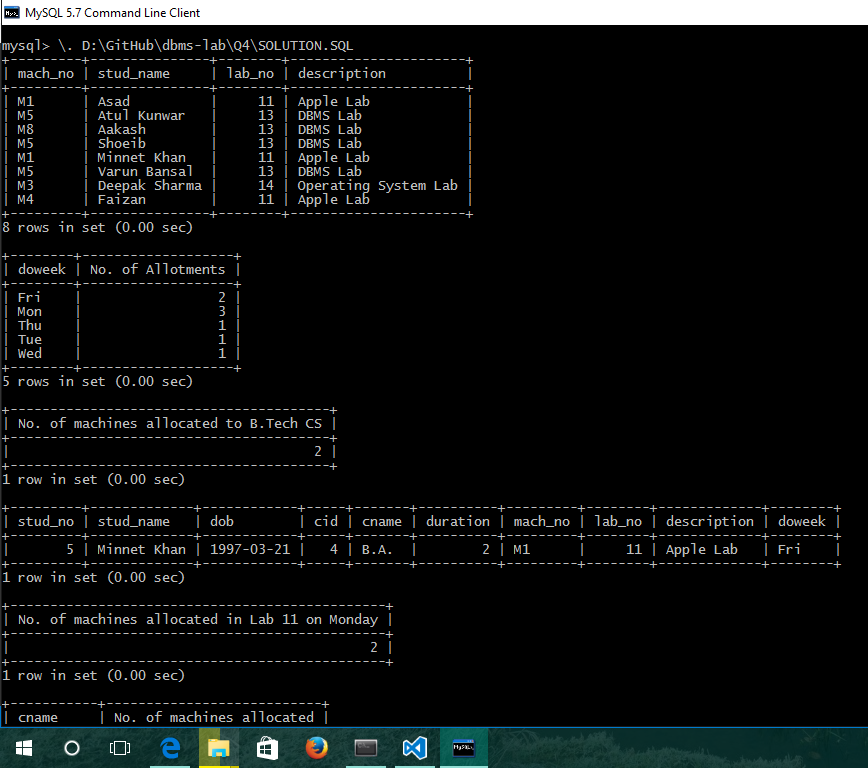
FROM Student0015 AS S

INNER JOIN Allotment0015 AS A ON S.stud\_no = A.stud\_no

INNER JOIN Class0015 AS C ON S.cid = C.cid

INNER JOIN Lab0015 AS L ON A.mach\_no = L.mach\_no

WHERE S.stud\_no = 5;



-- QUES 6

SELECT COUNT(\*) AS 'No. of machines allocated in Lab 11 on Monday'

FROM Allotment0015 AS A

INNER JOIN Lab0015 AS L ON L.mach\_no = A.mach\_no

WHERE A.doweek = 'Mon' AND L.lab\_no = 11;

-- QUES 7

SELECT C.cname, COUNT(\*) AS 'No. of machines allocated'

FROM Class0015 AS C

INNER JOIN Student0015 AS S ON S.cid = C.cid

INNER JOIN Allotment0015 AS A ON A.stud\_no = S.stud\_no;

-- QUES 8

SELECT L.description, COUNT(DISTINCT A.mach\_no) AS 'No. of machines allocated'

FROM Lab0015 AS L

INNER JOIN Allotment0015 AS A ON A.mach\_no = L.mach\_no

INNER JOIN Student0015 AS S ON S.stud\_no = A.stud\_no

GROUP BY L.lab\_no;

-- QUES 9

SELECT S.stud\_no, S.stud\_name, C.cname, A.mach\_no, L.lab\_no

FROM Student0015 AS S

INNER JOIN Class0015 AS C ON S.cid = C.cid

INNER JOIN Allotment0015 AS A ON A.stud\_no = S.stud\_no

INNER JOIN Lab0015 AS L ON L.mach\_no = A.mach\_no

ORDER BY S.dob DESC LIMIT 1;

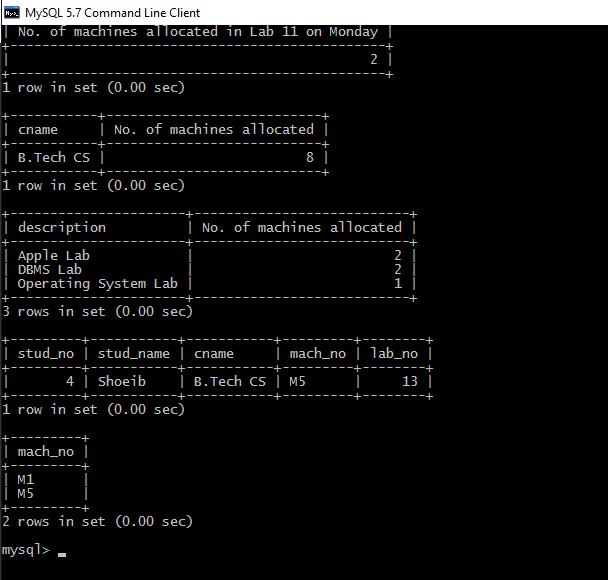
-- QUES 10

SELECT A.mach\_no

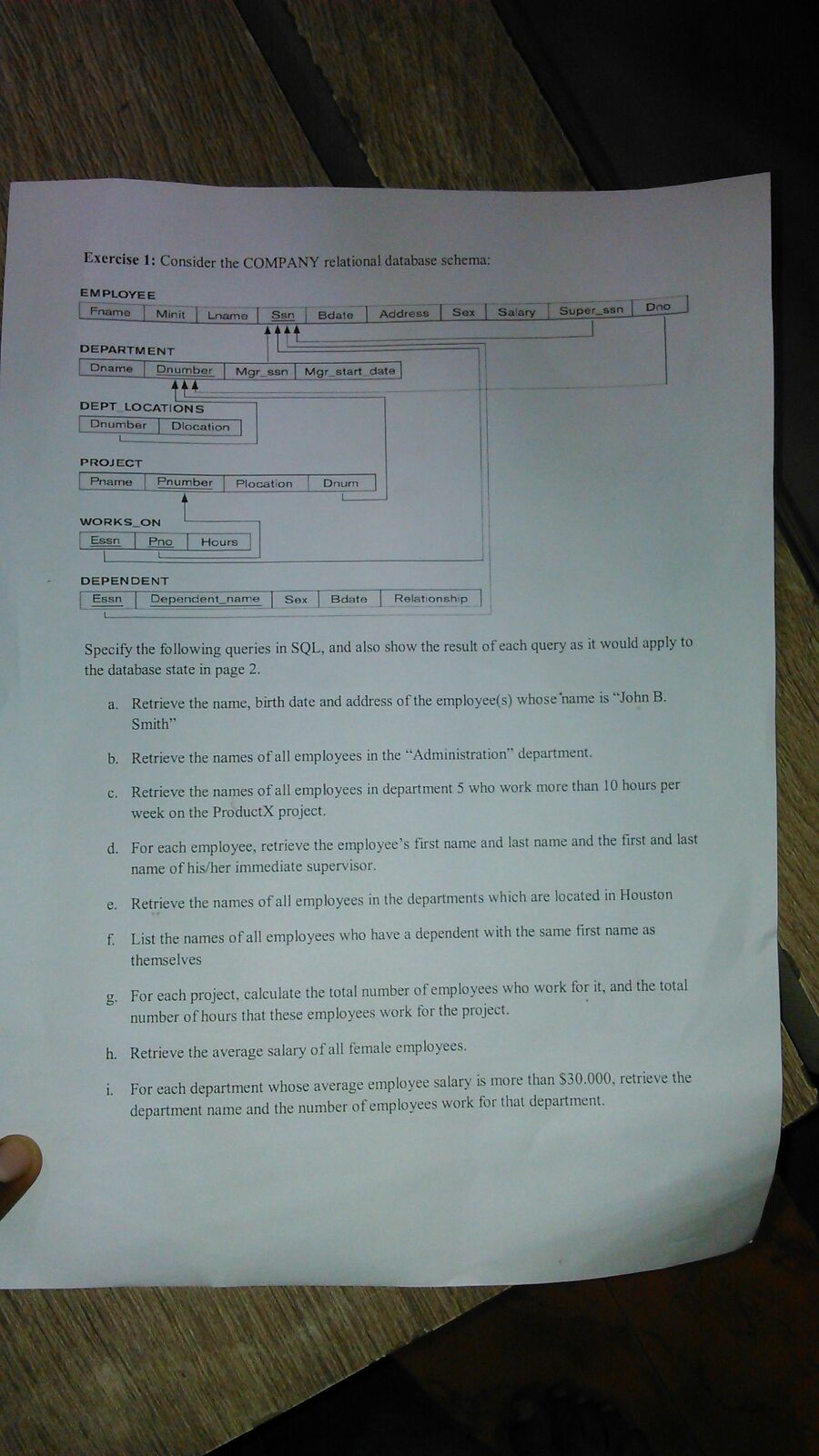
FROM Allotment0015 AS A

INNER JOIN Student0015 AS S ON S.stud\_no = A.stud\_no

WHERE A.mach\_no IN (SELECT a.mach\_no FROM Allotment0015 AS a GROUP BY a.mach\_no HAVING COUNT(a.mach\_no) > 1) GROUP BY A.mach\_no;



# Assignment 5:



-- Create Database if it does not exist and then use it

DROP DATABASE `Ashhar13BCS0015 - Q5`;

CREATE DATABASE `Ashhar13BCS0015 - Q5`;

USE `Ashhar13BCS0015 - Q5`;

-- Create Table Employee

CREATE TABLE Employee0015 (

fname VARCHAR(50),

minit VARCHAR(50),

lname VARCHAR(50),

ssn INT PRIMARY KEY,

bdate DATE,

address VARCHAR(500),

sex VARCHAR(7),

salary INT,

super\_ssn INT,

dno INT

);

-- Create Table Department

CREATE TABLE Department0015 (

dname VARCHAR(100),

dnumber INT PRIMARY KEY,

mgr\_ssn INT,

mgr\_start\_date DATE,

FOREIGN KEY(mgr\_ssn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE

);

-- Create Table Dept\_Locations

CREATE TABLE Dept\_Locations0015 (

dnumber INT,

dlocation VARCHAR(100),

FOREIGN KEY(dnumber) REFERENCES Department0015(dnumber) ON UPDATE CASCADE ON DELETE CASCADE

);

-- Create Table Project

CREATE TABLE Project0015 (

pname VARCHAR(100),

pnumber INT PRIMARY KEY,

plocation VARCHAR(100),

dnum INT,

FOREIGN KEY(dnum) REFERENCES Department0015(dnumber) ON UPDATE CASCADE ON DELETE CASCADE

);

-- Create Table Works\_On

CREATE TABLE Works\_On0015 (

essn INT,

pno INT,

hours INT,

FOREIGN KEY(essn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(pno) REFERENCES Project0015(pnumber) ON UPDATE CASCADE ON DELETE CASCADE

);

-- Create Table Dependent

CREATE TABLE Dependent0015 (

essn INT,

dependent\_name VARCHAR(50) PRIMARY KEY,

sex VARCHAR(7),

bdate DATE,

relationship VARCHAR(50),

FOREIGN KEY(essn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE

);

INSERT INTO Employee0015 VALUES

('John', 'B.', 'Smith', 1, NULL, 'JMI, New Delhi', 'M', 10000, 2, 1),

('Ashhar', NULL, 'Hasan', 2, NULL, 'Texas, HO', 'M', 20000, NULL, 1),

('Fatima', NULL, 'Munawwar', 3, NULL, 'New Delhi', 'F', 10000, 2, 5);

INSERT INTO Department0015 VALUES

('CS', 1, 2, NULL),

('Administration', 5, 3, NULL);

INSERT INTO Dept\_Locations0015 VALUES

(1, 'Houston'),

(5, 'New Delhi');

INSERT INTO Project0015 VALUES

('ProjectX', 1, 'New Delhi', 1),

('ProjectY', 2, 'Houston', 5);

INSERT INTO Works\_On0015 VALUES

(1, 1, 12),

(2, 1, 10),

(3, 2, 12);

INSERT INTO Dependent0015 VALUES

(3, 'Dependent1', 'M', NULL, 'Reference'),

(1, 'John A. Kool', 'M', NULL, 'Reference');

ALTER TABLE Employee0015 ADD CONSTRAINT FOREIGN KEY(super\_ssn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Employee0015 ADD CONSTRAINT FOREIGN KEY(dno) REFERENCES Department0015(dnumber) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Department0015 add constraint FOREIGN KEY(mgr\_ssn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Dept\_Locations0015 ADD constraint FOREIGN KEY(dnumber) REFERENCES Department0015(dnumber) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Project0015 ADD CONSTRAINT FOREIGN KEY(dnum) REFERENCES Department0015(dnumber) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Works\_On0015 ADD CONSTRAINT FOREIGN KEY(essn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Works\_On0015 ADD CONSTRAINT FOREIGN KEY(pno) REFERENCES Project0015(pnumber) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE Dependent0015 ADD CONSTRAINT FOREIGN KEY(essn) REFERENCES Employee0015(ssn) ON UPDATE CASCADE ON DELETE CASCADE;

SELECT \* FROM Employee0015;

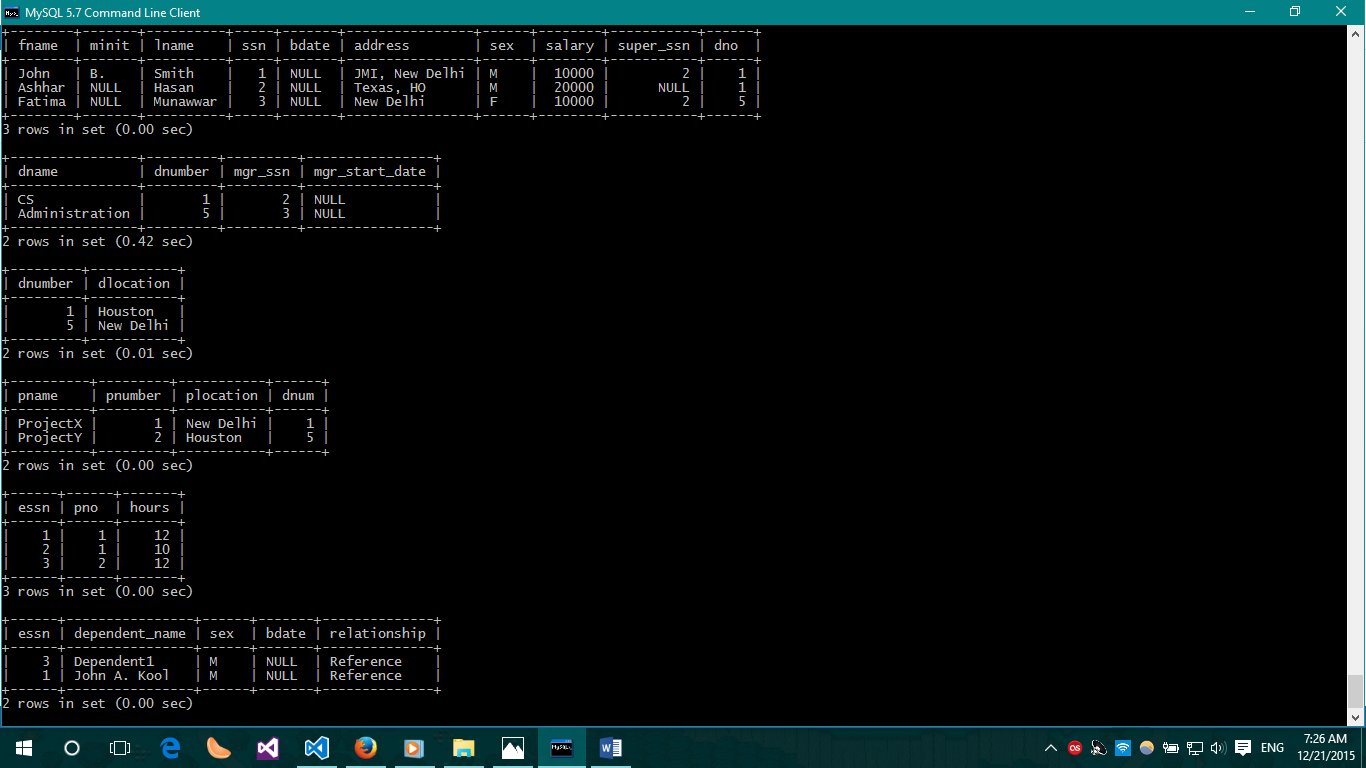
SELECT \* FROM Department0015;

SELECT \* FROM Dept\_Locations0015;

SELECT \* FROM Project0015;

SELECT \* FROM Works\_On0015;

SELECT \* FROM Dependent0015;



-- QUES 1

SELECT fname, minit, lname, bdate, address

FROM Employee0015

WHERE fname = 'John' AND minit = 'B.' AND lname = 'Smith'

;

-- QUES 2

SELECT E.fname, E.lname

FROM Employee0015 AS E

INNER JOIN Department0015 AS D ON E.dno = D.dnumber

WHERE D.dname = 'Administration';

-- QUES 3

SELECT E.fname, E.lname

FROM Employee0015 AS E

INNER JOIN Department0015 AS D ON E.dno = D.dnumber

INNER JOIN Works\_On0015 AS W ON W.essn = E.ssn

INNER JOIN Project0015 AS P

WHERE D.dnumber = 5 AND W.hours > 10 AND P.pname = 'ProjectY';

-- QUES 4

SELECT E.fname, E.lname, S.fname AS 'Supervisor', S.lname AS 'Supervisor'

FROM Employee0015 AS E, Employee0015 AS S

WHERE E.super\_ssn = S.ssn;

-- QUES 5

SELECT E.fname, E.lname

FROM Employee0015 AS E

INNER JOIN Department0015 AS D ON D.dnumber = E.dno

INNER JOIN Dept\_Locations0015 AS DL ON DL.dnumber = D.dnumber

WHERE DL.dlocation = 'Houston';

-- QUES 6

SELECT E.fname, E.lname

FROM Employee0015 AS E

INNER JOIN Dependent0015 AS D ON D.essn = E.ssn

WHERE D.dependent\_name LIKE "John%";

-- QUES 7

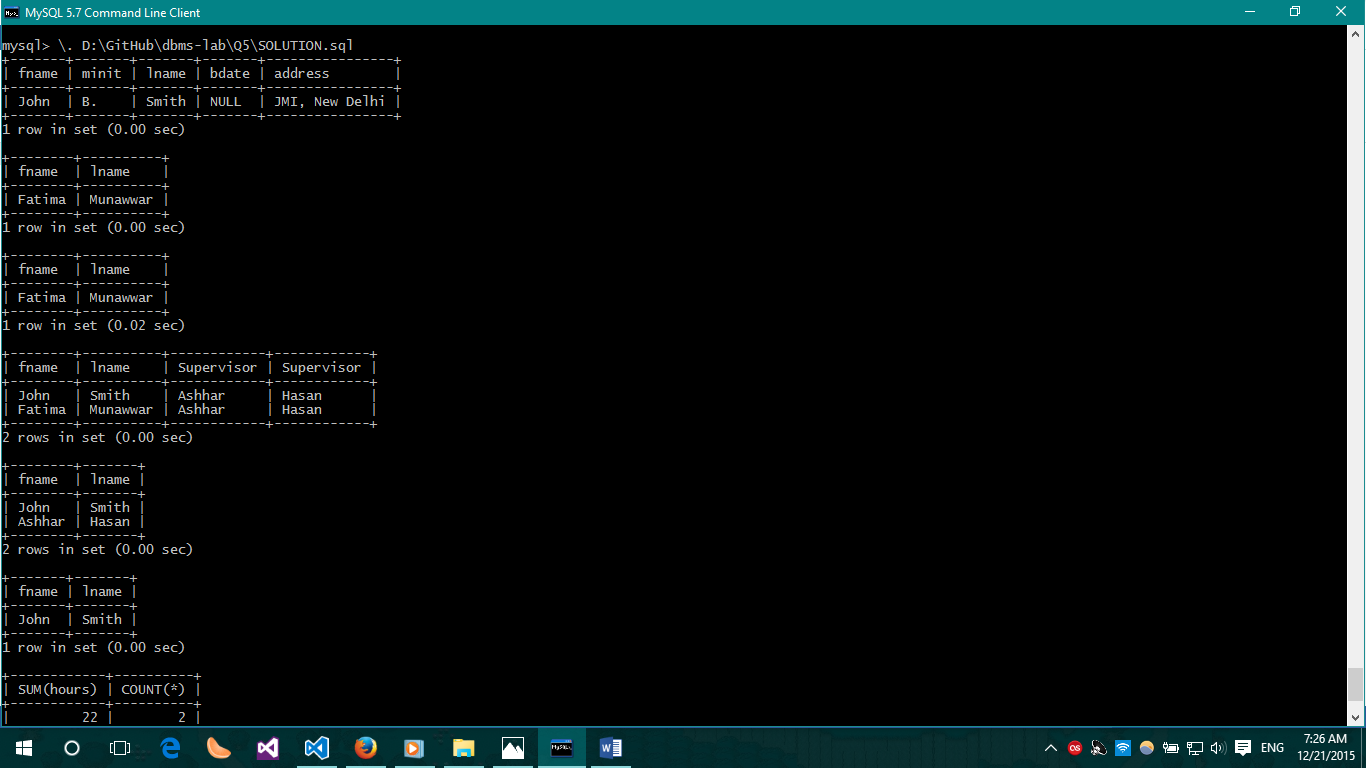
SELECT SUM(hours), COUNT(\*)

FROM Employee0015 AS E

INNER JOIN Works\_On0015 AS W ON W.essn = E.ssn

INNER JOIN Project0015 AS P ON W.pno = P.pnumber

GROUP BY P.pname;

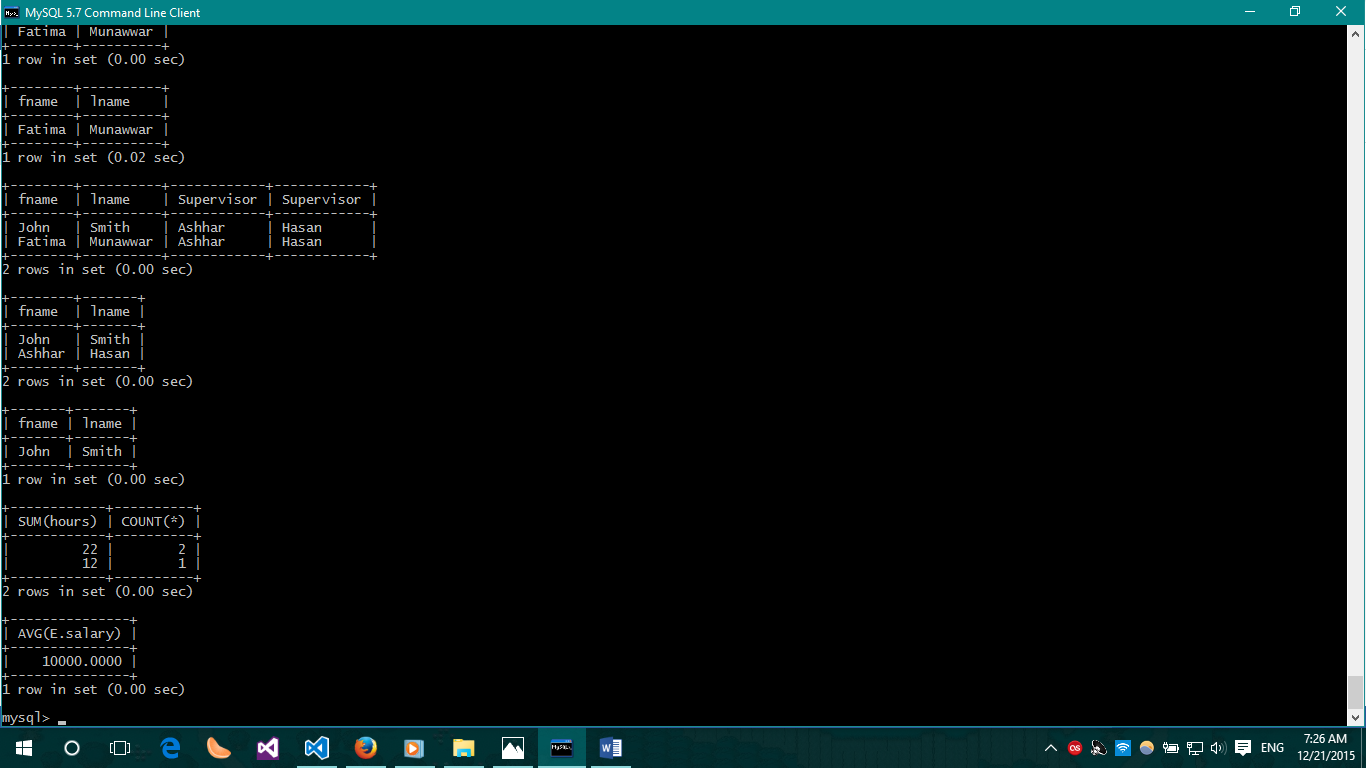


-- QUES 8

SELECT AVG(E.salary)

FROM Employee0015 AS E

WHERE E.sex = 'F';



-- QUES 9: TODO

# Assignment 6:



-- Create the database if it does not exist and use it

DROP DATABASE `Ashhar13BCS0015 - Q6`;

CREATE DATABASE IF NOT EXISTS `Ashhar13BCS0015 - Q6`;

USE `Ashhar13BCS0015 - Q6`;

-- Drop the existing tables

DROP TABLE IF EXISTS Product0015;

DROP TABLE IF EXISTS Customer0015;

DROP TABLE IF EXISTS OrderDetail0015;

-- Create Table Product

CREATE TABLE Product0015 (

pid INT PRIMARY KEY,

pname VARCHAR(100) NOT NULL,

price INT NOT NULL,

discount DECIMAL(5,2),

pcompany VARCHAR(100) NOT NULL

);

-- Create Table Customer

CREATE TABLE Customer0015 (

cid INT PRIMARY KEY,

cname VARCHAR(100),

cadd VARCHAR(500),

ccity VARCHAR(50)

);

-- Create Table OrderDetail

CREATE TABLE OrderDetail0015 (

oid INT NOT NULL,

pid INT NOT NULL,

cid INT NOT NULL,

odate DATE NOT NULL,

qty INT NOT NULL,

status VARCHAR(100),

CONSTRAINT pk\_OD PRIMARY KEY (oid, pid),

FOREIGN KEY(cid) REFERENCES Customer0015(cid) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(pid) REFERENCES Product0015(pid) ON UPDATE CASCADE ON DELETE CASCADE

);

INSERT INTO Product0015 VALUES

(1, 'Pendrive', 700, 5, 'Kingston'),

(5, 'DVD', 20, 2, 'Moserbaer'),

(7, 'HeadPhone', 500, 25, 'Intex'),

(9, 'Modem', 1500, 10, 'Intex'),

(10, 'Hard Disk', 4500, 7, 'Seagate'),

(12, 'Speaker', 1000, 5, 'Creative'),

(101, 'RAM', 2000, 6, 'Hynix');

INSERT INTO Customer0015 VALUES

(1, 'Sonaal', 'Dwarka', 'New Delhi'),

(2, 'Abhinav', 'Mehrauli', 'New Delhi'),

(3, 'Abdul Rauf', 'Jamia Hostel', 'New Delhi'),

(5, 'Insaan', 'Jamia Hostel', 'New Delhi'),

(6, 'Javed Ahmed', 'Allahabad', 'Uttar Pradesh'),

(7, 'Deepak', 'Borivali', 'Mumbai'),

(8, 'Minnet Khan', 'Allahabad', 'Uttar Pradesh');

INSERT INTO OrderDetail0015 VALUES

(1, 1, 1, '2015-07-23', 2, 'success'),

(1, 7, 1, '2015-07-23', 2, 'success'),

(2, 10, 5, '2015-08-28', 5, 'success'),

(2, 12, 5, '2015-08-28', 5, 'success'),

(3, 5, 2, '2015-08-31', 20, 'success'),

(4, 9, 6, '2015-09-01', 2, 'success'),

(5, 10, 7, '2015-09-02', 1, 'success'),

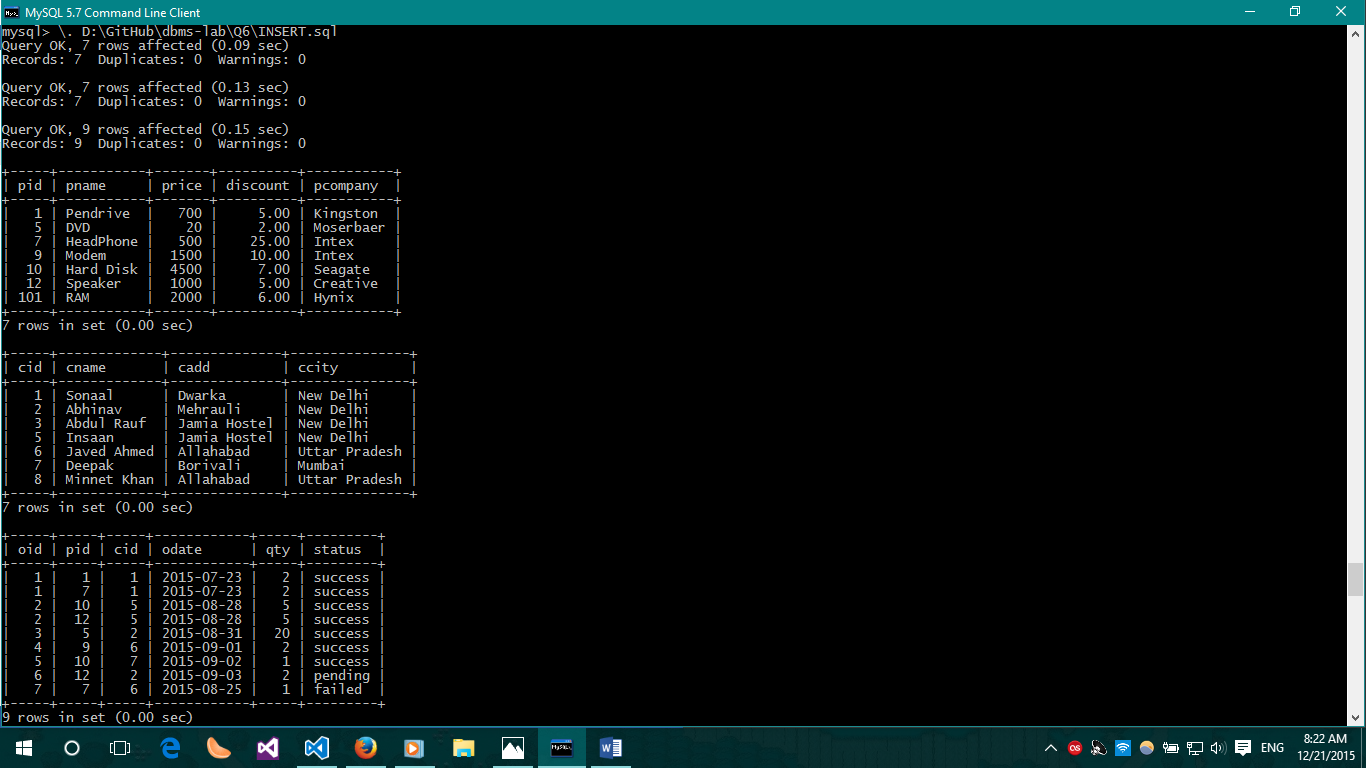
(6, 12, 2, '2015-09-03', 2, 'pending'),

(7, 7, 6, '2015-08-25', 1, 'failed');

SELECT \* FROM Product0015;

SELECT \* FROM Customer0015;

SELECT \* FROM OrderDetail0015;



-- QUES 2

SELECT \*

FROM OrderDetail0015

WHERE status = 'success';

-- QUES 3

SELECT SUM(P.price \* P.discount / 100 \* O.qty) AS 'Discount given on 2015-08-28'

FROM Product0015 AS P

INNER JOIN OrderDetail0015 AS O

WHERE O.odate = '2015-08-28';

-- QUES 4

SELECT DISTINCT(P.pname)

FROM Product0015 AS P

INNER JOIN OrderDetail0015 AS O ON O.pid = P.pid

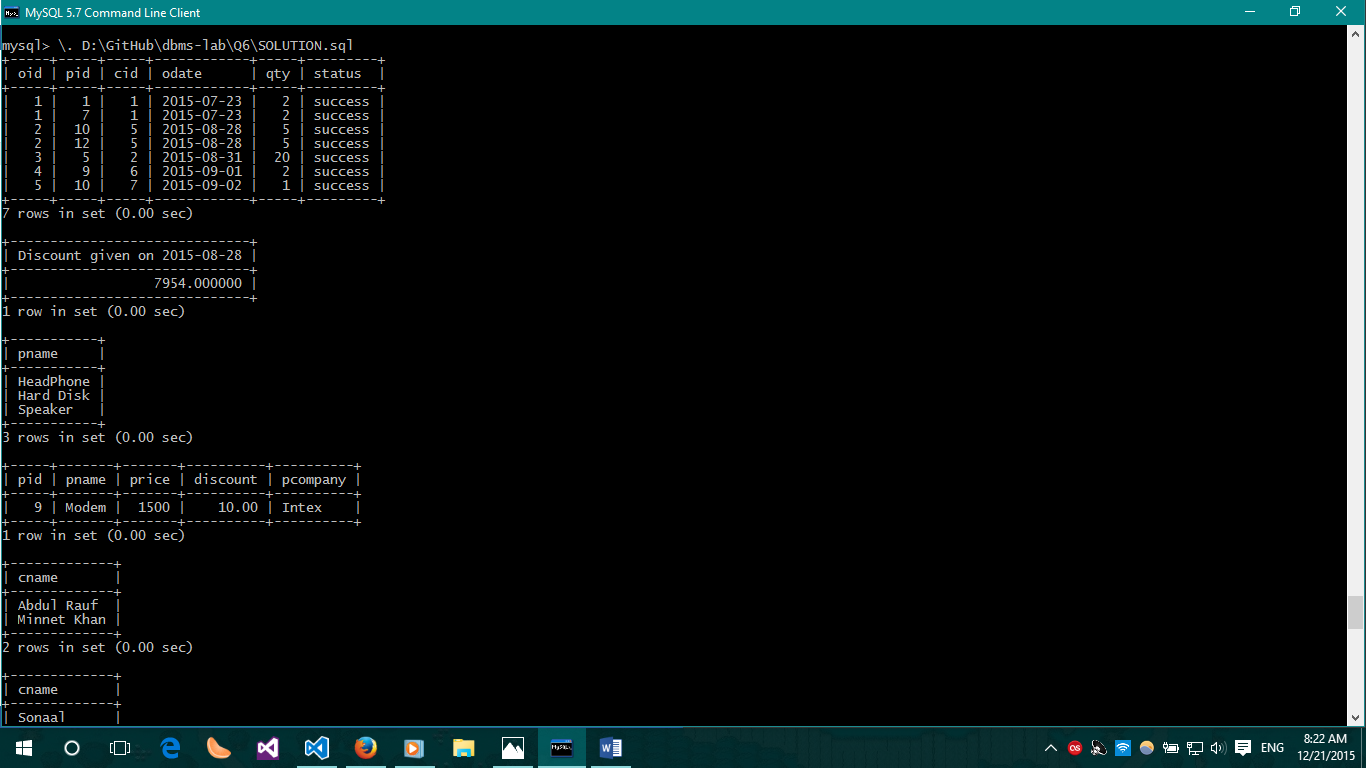
WHERE P.pname IN (SELECT pname FROM OrderDetail0015 AS o, Product0015 AS p WHERE p.pid = o.pid GROUP BY o.pid HAVING COUNT(\*) > 1);

-- QUES 5

SELECT \*

FROM Product0015 AS P

ORDER BY P.price DESC LIMIT 2, 1;



-- QUES 6

SELECT DISTINCT(C.cname)

FROM Customer0015 AS C

WHERE C.cid NOT IN (SELECT cid FROM OrderDetail0015);

-- QUES 7

SELECT DISTINCT(C.cname)

FROM Customer0015 AS C

INNER JOIN OrderDetail0015 AS O ON O.cid = C.cid

GROUP BY O.cid HAVING COUNT(O.cid) > 1;

-- QUES 8

SELECT DISTINCT(C.cname)

FROM Customer0015 AS C

INNER JOIN OrderDetail0015 AS O ON O.cid = C.cid

WHERE O.qty > 1 GROUP BY O.cid;

-- QUES 9

SELECT C.cname, O.qty, P.price, SUM(P.price \* (1 - P.discount / 100) \* O.qty) AS 'Payable Amount'

FROM OrderDetail0015 AS O

INNER JOIN Customer0015 AS C ON C.cid = O.cid

INNER JOIN Product0015 AS P ON P.pid = O.pid

GROUP BY O.cid;

-- QUES 10

SELECT C.cname, SUM(O.qty)

FROM OrderDetail0015 AS O

INNER JOIN Customer0015 AS C ON C.cid = O.cid

GROUP BY O.cid;

