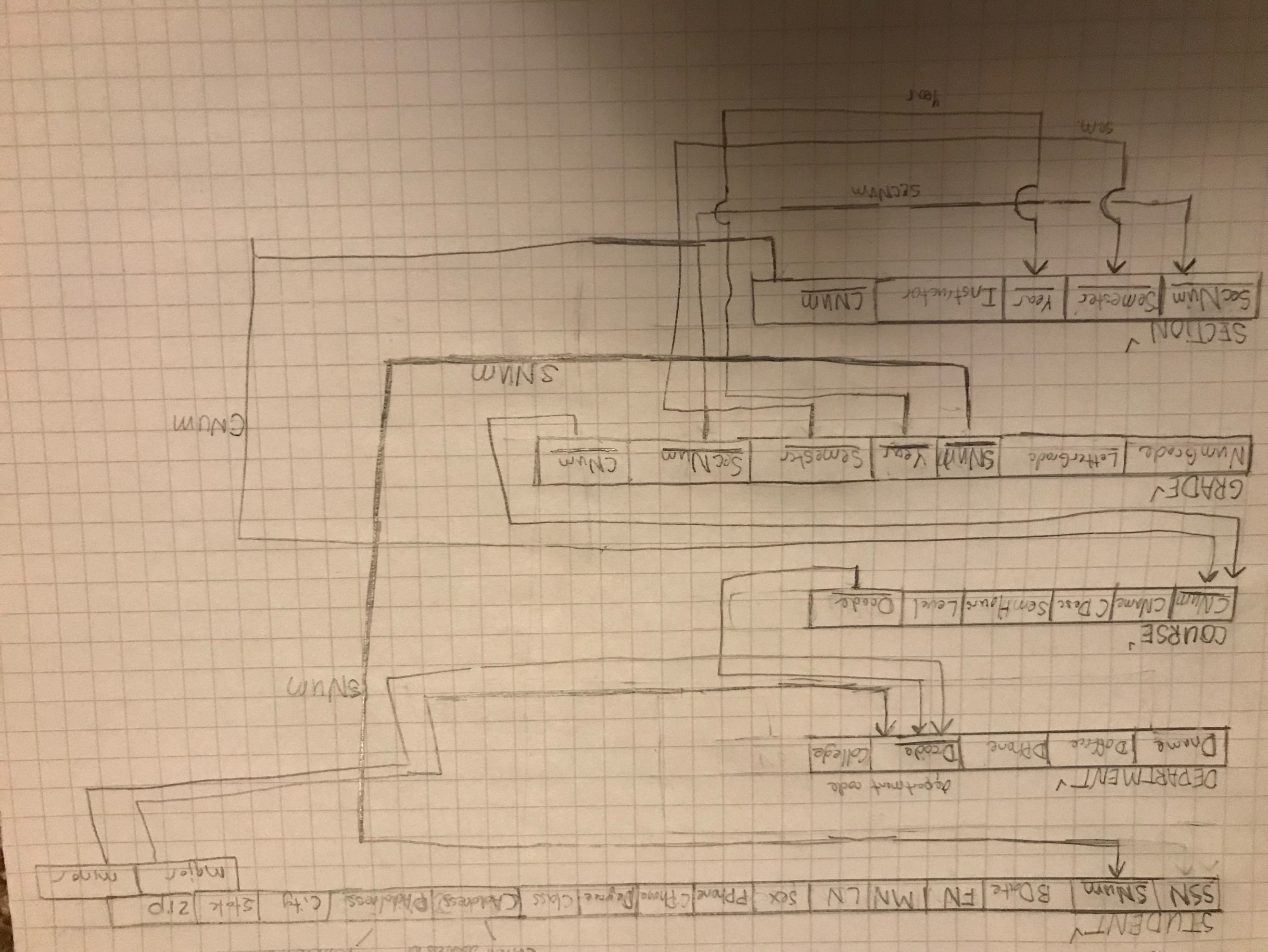
**CSS 475 A | Homework 3 |** Misha Ward

PART 1)

Relational Database Model: (please let me know if you need a better picture)



Other Constraints for DB (Sentences):

-Letter grades within normal grades (A, B, C, D, F)

-Zip within normal boundaries (5 digits)

-SSN within normal boundaries (9 digits)

-Class had to be within boundaries of freshman, sophmore, etc.

-Degree program had to be within normal boundaries like MBA, B.A., B.S., Ph.D.

-Grade should be decimal and between 0 to 4.0

-Many more, please look at SQL code for more details.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Creation of the five tables for the UNIVERSITY DB\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

-- please copy code below to test database

CREATE TABLE STUDENT

(

Ssn char(9) NOT NULL UNIQUE,

snum CHAR(10) NOT NULL UNIQUE,

bdate DATE,

Fname VARCHAR(15) NOT NULL ,

Mname VARCHAR(15) NOT NULL ,

Lname VARCHAR(15) NOT NULL ,

sex VARCHAR(2) NOT NULL,

PPhone CHAR(10) NOT NULL,

CPhone CHAR(10) NOT NULL,

Degree VARCHAR(15) NOT NULL ,

Class VARCHAR(15) NOT NULL ,

PAddress VARCHAR(30) NOT NULL ,

CAddress VARCHAR(30) NOT NULL ,

City VARCHAR(15) NOT NULL ,

State VARCHAR(15) NOT NULL ,

Zip CHAR(5) NOT NULL,

major CHAR(5),

minor CHAR(5),

CHECK (class == 'freshman' OR class == 'sophmore'

OR class == 'junior' OR class == 'senior'),

CHECK (zip >= 10000 AND zip <= 99999),

CHECK (ssn >= 100000000 AND ssn <= 999999999),

CHECK (degree == 'B.A.' OR degree == 'B.S.' OR degree == 'PH.D.'),

CHECK (sex == 'm' OR sex == 'f' OR sex == 'other'),

primary key (Ssn, snum)

FOREIGN KEY(major) REFERENCES DEPARTMENT(dcode) on update cascade on delete cascade,

FOREIGN KEY(minor) REFERENCES DEPARTMENT(dcode) on update cascade on delete cascade

);

CREATE TABLE DEPARTMENT

(

dname VARCHAR(15) NOT NULL,

doffice CHAR(3),

dphone CHAR(10) NOT NULL,

dcode snum CHAR(5) UNIQUE,

college VARCHAR(30) NOT NULL,

primary key (Dname, Dcode)

);

CREATE TABLE COURSE

(

Cnum CHAR(3) NOT NULL UNIQUE,

cname VARCHAR(15) NOT NULL ,

cdesc VARCHAR(30) NOT NULL ,

semHours CHAR(2) NOT NULL,

level CHAR(3) NOT NULL,

Dcode CHAR(3) NOT NULL,

primary key(Cnum, Dcode)

FOREIGN KEY(dcode) REFERENCES DEPARTMENT(dcode) on update cascade on delete cascade

);

CREATE TABLE SECTION

(

instructor VARCHAR(30) NOT NULL ,

Cnum CHAR(3) NOT NULL,

year CHAR(4) NOT NULL,

semester CHAR(1) NOT NULL,

secnum CHAR(3) NOT NULL,

CHECK (year >= 2000 AND year <= 2020),

CHECK (semester == 1 OR semester == 2),

FOREIGN KEY (cnum) REFERENCES COURSE(cnum) on update cascade on delete cascade

);

CREATE TABLE GRADE

(

numgrade DECIMAL(2,1) NOT NULL,

lettergrade VARCHAR(1) NOT NULL,

snum CHAR(10) NOT NULL,

year CHAR(4) NOT NULL,

semester CHAR(1) NOT NULL,

secnum CHAR(3) NOT NULL,

cnum CHAR(3) NOT NULL,

CHECK (numgrade >= 0 AND numgrade <= 4),

CHECK (lettergrade == 'A' OR lettergrade == 'B' OR lettergrade == 'C' OR lettergrade == 'D' OR lettergrade == 'F'),

primary key(numgrade, lettergrade)

FOREIGN KEY (snum) REFERENCES STUDENT(snum) on update cascade on delete cascade,

FOREIGN KEY (year) REFERENCES SECTION(year) on update cascade on delete cascade,

FOREIGN KEY (semester) REFERENCES SECTION(semester) on update cascade on delete cascade,

FOREIGN KEY (secnum) REFERENCES SECTION(secnum) on update cascade on delete cascade,

FOREIGN KEY (cnum) REFERENCES COURSE(cnum) on update cascade on delete cascade

);

--INSERT STATEMENTS

insert into STUDENT values (111222444, 1234567890, '1990-10-10', 'Michael', 'Robert', 'Ward',

'm' , '2069827777' , '2069827777' , 'B.S.' , 'junior' , 'Panda Drive' , 'Trots Way',

'Seattle' , 'Washington' , '98008' , '1' , '2');

insert into DEPARTMENT values ('CS', 231, 2069999999, 1, 'UW Bothell');

insert into COURSE values (475, 'Database', 'Teaches about databases', 5, 400, 1);

insert into GRADE values (4, 'A', 1234567890, 2017, 1, 4, 475);

insert into SECTION values ('Mr. Giles', 475, 2017, 1, 4);

--UPDATE STATEMENTS

UPDATE STUDENT

SET ssn = 111222445

WHERE snum = 1234567890;

UPDATE DEPARTMENT

SET dname = 'Computer Science'

WHERE dcode = 1;

UPDATE COURSE

SET cname = 'Database and SQL'

WHERE cnum = 475;

UPDATE GRADE

SET numgrade = 3.3

WHERE snum = 1234567890;

UPDATE SECTION

SET instructor = 'Min Chen'

WHERE cnum = 475;

--SELECT STATEMENTS

SELECT \* FROM student;

SELECT \* FROM department;

SELECT \* FROM course;

SELECT \* FROM grade;

SELECT \* FROM section;

--OTHER REQUIRED QUERIES...

--Include at least one SELECT statement provide historical information for a student

SELECT Fname, Mname, LName, CName, semHours, lettergrade, Year

FROM student s, course c, grade g

WHERE g.snum = s.snum AND

c.cnum = g.Cnum AND

g.year < 2018 ;

/\*

Results:

Fname Mname Lname cname semHours lettergrade year

---------- ---------- ---------- ---------------- ---------- ----------- ----------

Michael Robert Ward Database and SQL 5 A 2017

\*/

--Include at least one SELECT statement providing current registration information for a student

SELECT stu.FName, stu.LName, g.secnum, g.cnum, sect.instructor

FROM STUDENT stu, SECTION sect, GRADE g

WHERE stu.snum = g.snum AND

g.year = sect.year AND

g.semester = sect.semester AND

g.year = sect.year AND

g.year = 2018 ; -- could possibly use current date to get year for dynamic use

-- no results as no data for this selection, otherwise this should work to pull

--current registration.

--Include at least one SELECT statement providing all courses’

-- information (including the number of sections they each have) for the university

SELECT c.cnum, c.cname, c.cdesc, c.semHours, c.level, c.dcode, s.secnum

FROM course c, section s

WHERE s.cnum = c.cnum

GROUP BY dcode;

/\*

Results:

Cnum cname cdesc semHours level Dcode secnum

---------- ---------------- ----------------------- ---------- ---------- ---------- ----------

475 Database and SQL Teaches about databases 5 400 1 4

\*/

--Include at least one SELECT statement providing all the students majored in a certain department

SELECT Fname, Lname, dname

FROM student, department

WHERE major = dcode;

/\*

Results:

Fname Lname dname

---------- ---------- ----------------

Michael Ward Computer Science

\*/

--Include at least one SELECT statement providing information on how many

-- students a instructor teaches (including all the sections that the instructor is teaching)

SELECT instructor, count(snum) as Total\_Students

FROM section s, grade g

WHERE s.secnum = g.secnum AND

s.year = g.year AND

s.semester = g.semester

GROUP BY instructor;

/\*

Results:

instructor Total\_Students

---------- --------------

Min Chen 1

\*/

--DELETE STATEMENTS

DELETE FROM STUDENT

WHERE snum = 1234567890;

--Should return no results

DELETE FROM DEPARTMENT

WHERE Dcode = 1;

--Should return no results

DELETE FROM COURSE

WHERE cnum = 475;

--Should return no results

DELETE FROM GRADE

WHERE lettergrade = 'A';

--Should return no results

DELETE FROM SECTION

WHERE instructor = 'Min Chen';

--Should return no results

-- please do not copy further than this point.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of PART 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

PART 2)

a.

SELECT fname, lname, hours

FROM department d, employee e, works\_on w, project p

WHERE e.dno = d.dnumber AND

e.ssn = w.essn AND

e.dno = '4' AND

w.hours < 20 AND

w.pno = p.pnumber AND

pname = 'Computerization';

Results:

Fname Lname

---------- ----------

Alicia Zelaya

b.

SELECT innerTable.fname, innerTable.lname, innerTable.salary, e.fname, e.lname

FROM (

SELECT fname, lname, salary, ssn, super\_ssn

FROM employee e, department d

WHERE e.dno = d.dnumber AND

d.dname = 'Research' ) innerTable, employee e

WHERE innerTable.super\_ssn = e.ssn ;

Results:

fname lname salary Fname Lname

---------- ---------- ---------- ---------- ----------

Franklin Wong 40000 James Borg

John Smith 30000 Franklin Wong

Ramesh Narayan 38000 Franklin Wong

Joyce English 25000 Franklin Wong

c.

SELECT dname, count(ssn)

FROM department, employee

WHERE salary > 32000 AND

dno = dnumber

GROUP BY dname;

Results:

dname count(ssn)

-------------- ----------

Administration 1

Headquarter 1

Research 2

d.

SELECT dname, count

FROM (

SELECT dname, count(ssn) as count, avg(salary) as avgSalary

FROM department, employee

WHERE dno = dnumber

GROUP BY dname )

WHERE avgSalary > 32000

GROUP BY dname;

Results:

dname count

----------- ----------

Headquarter 1

Research 4

e.

SELECT pname, dname, count(\*), SUM(hours)

FROM project, department, works\_on, employee

WHERE dnum = dnumber AND

pno = pnumber AND

ssn = essn

GROUP BY pnumber;

Results:

pname dname count(\*) SUM(hours)

---------- ---------- ---------- ----------

ProductX Research 2 52.5

ProductY Research 3 37.5

ProductZ Research 2 50

Computeriz Administra 3 55

Reorganiza Headquarte 3 25

Newbenefit Administra 3 55

f.

SELECT e.fname, e.lname

FROM (

SELECT dname, fname, lname, MAX(salary) as maxSal, ssn

FROM employee, department

WHERE dno = dnumber ) innerTable, employee e, department d

WHERE innerTable.dname = d.dname AND

innerTable.ssn = e.ssn

GROUP BY e.fname;

Results:

Fname Lname

---------- ----------

James Borg

g.

SELECT e.fname, e.lname, e.ssn

FROM (

SELECT e.fname, e.lname, e.ssn

FROM (

SELECT fname, lname, ssn

FROM employee

WHERE ssn = '888665555' ) innerMostTable, employee e

WHERE innerMostTable.ssn = e.super\_ssn

) innerTable, employee e

WHERE innerTable.ssn = e.super\_ssn ;

Results:

Fname Lname Ssn

---------- ---------- ----------

John Smith 123456789

Alicia Zelaya 999887777

Ramesh Narayan 666884444

Joyce English 453453453

Ahmad Jabbar 987987987

h.

SELECT e.fname, e.lname, e.salary

FROM (

SELECT fname, lname, dname, MIN(salary) as minSalary

FROM employee, department

GROUP BY dname

ORDER BY dname asc ) innerTable, employee e

WHERE (e.salary) > minSalary

GROUP BY e.fname ;

Results:

Fname Lname salary

---------- ---------- ----------

Franklin Wong 40000

James Borg 55000

Jennifer Wallace 43000

John Smith 30000

Ramesh Narayan 38000