**Metropolitan State University**

**ICS 311-03**

**Mid-Term Exam**

**Total Points: 80**

**Name: Nalongsone Danddank Date: 10/21/2020**

\_

**Question 1 (Total 10 points):**

**1.1) (5 Points) Highlight) ONLY ONE correct answer. (If more than one answer is highlighted then your answer will not be considered even if it includes the correct answer).**

1. All of the following terms represent the same concept, except:

a. Tuple b. Record

c. Row d. Entity

2. Which of the following is NOT a valid form of join:

a. Right Outer Join b. Natural Join

c. Union Join d. Self Join

3. A table in which the intersection of every column and record contains only one value is in which normalization form:

a. 1NF b. 2NF

c. Boyce-Codd NF d. 3NF

4. The following rule must be maintained by the DBMS to ensure data consistency:

a. Entity integrity constraints b. Referential integrity constraints

c. Attribute domain constraints d. All of the above

5. Data Manipulation Language (DML) contains all of the following except (Choose one):

a. Insert b. Delete

c. Update d. Create

**1.2) (5 Points) For each of the following statements, mark T (True) or F (False)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | | |
| SELECT DISTINCT is used if a user does not want to see duplicate rows in a query. | | | T | F |
| Consider the following SQL statement:  SELECT prod\_price \* prod\_quant as total\_cost FROM Product;  The output of this query is a table with two columns. | | | T | F |
| You can create a View if you do not want users to see all the attributes of a single or joined tables | | | T | F |
| A super key identifies unique rows in the table. | | | T | F |
| If a salesperson attempts to enter information in INVOICE table for a customer not in the CUSTOMER table, the database will typically generate an error message. This message indicates that a referential integrity constraint has been violated. | | | T | F |

**Question 2 (Total 5 Points):**

Refer to Schema diagram of University database. Write the relational algebra expressions for the following queries. *Hint: Use Projection ∏ and Selection σ*

2.1 (2 Points) Find the name and salary of all instructors in the Music department

*∏name,salary(σ* department.*dept\_name = ‘Music’(σ instructors.id = department.id (instructors*  department*))*

2.1 (2 Points) Find the name of all students with tot\_cred > 15

*∏name(σ tot\_cred>15(students))*

2.3 (1 Point) For the following relational algebra expression, explain the output of the expression in words:

∏*course\_id* (σ *semester=“Spring” Λ year=2019* (*section*))

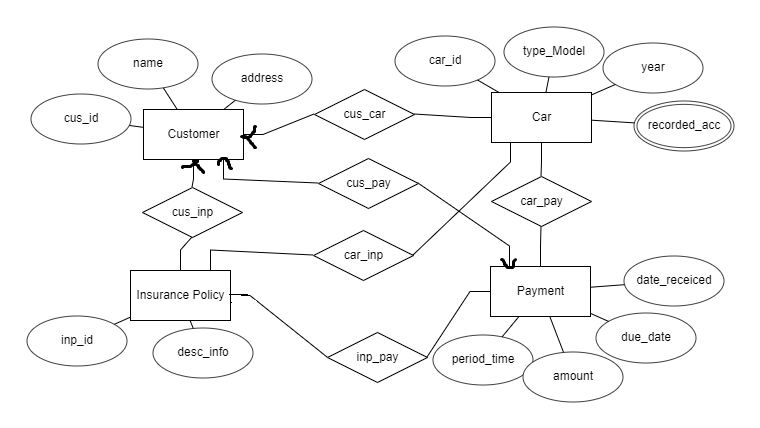
U   
∏*course\_id* (σ *semester=“Fall” Λ year=2020* (*section*))

Find all the course id of section in the Spring 2019 or Fall 2020 semesters in section table.

**Question 3 (Total 10 Points)**

Construct an ER diagram for a car insurance company where:

* Customers own one or more cars each
* Each car is associated with zero to any number of recorded accidents.
* Each insurance policy covers one or more cars, has one or more premium payments associated with it.
* Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.

****

# Question 4 (Total 10 points): Normalization

Consider following table - Convert it to 3NF

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| StdSSN | StdCity | StdClass | OfferNo | OffTerm | OffYear | EnrGrade | CourseNo | CrsDesc |
| S1 | SEATTLE | JUN | O1 | FALL | 2006 | 3.5 | C1 | DB |
| S1 | SEATTLE | JUN | O2 | FALL | 2006 | 3.3 | C2 | VB |
| S2 | BOTHELL | JUN | O3 | SPRING | 2007 | 3.1 | C3 | OO |
| S2 | BOTHELL | JUN | O2 | FALL | 2006 | 3.4 | C2 | VB |

Solution:

|  |  |  |
| --- | --- | --- |
| StdSSN | StdCity | StdClass |
| S1 | SEATTLE | JUN |
| S2 | BOTHELL | JUN |
| S2 | BOTHELL | JUN |

|  |  |
| --- | --- |
| CourseNo | CrsDesc |
| C1 | DB |
| C2 | VB |
| C3 | OO |

|  |  |  |
| --- | --- | --- |
| OfferNo | OffTerm | OffYear |
| O1 | FALL | 2006 |
| O2 | FALL | 2006 |
| O3 | SPRING | 2007 |

|  |  |  |  |
| --- | --- | --- | --- |
| StdSSN | OfferNo | CourseNo | EnrGrade |
| S1 | O1 | C1 | 3.5 |
| S1 | O2 | C2 | 3.3 |
| S2 | O3 | C3 | 3.1 |
| S2 | O2 | C2 | 3.4 |

**Question 5 (Total 10 Points):**

Answer questions based on the following schema. Note that primary keys are **BOLD** and foreign keys are underlined.

**Branch**(**branchName:** *varchar(15),* street:*varchar(30),* city:*varchar(20))*

**Account**(**accountNum:** *integer,* branchName: *varchar(15),* balance: *integer*)

**Customer**(**customerSSN:** *integer,* street: *varchar(30),* city:*varchar(20)*)

**Deposit(customerSSN:** *integer****,* accountNum:** *integer****,*** amount**:** *integer****)***

5.1) (5 Points) Write SQL statements to create all the tables.

* Make sure to include entity integrity and referential integrity constraints (if any) in your statements.
* Use “on delete cascade” to ensure referential integrity is maintained

create table Branch(branchName varchar(15), street varchar(30), city varchar(20),

primary key(branchName));

create table Account(accountNum int, branchName varchar(15), balance int,

primary key(accountNum));

create table Customer(customerSSN int, street varchar(30), city varchar(20),

primary key(customerSSN));

create table Deposit(customerSSN int, accountNum int, amount int,

primary key(customerSSN, accountNum),

foreign key(customerSSN) references Customer(customerSSN) on delete cascade,

foreign key(accountNum) references Account(accountNum) on delete cascade);

3.2) (5 Points) Write SQL statements to Insert following data into the tables specified below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Branch |  |  |  | Account |  |  |
| **branchName** | **street** | **city** |  | **accountNum** | **branchName** | **balance** |
| First Bank | First Street | Crystal |  | 1 | First Bank | 100 |
| Second Bank | Second Street | New Hope |  | 2 | Second Bank | 200 |
| Third Bank | Third Street | Golden Valley |  | 3 | Third Bank | 300 |
|  |  |  |  |  |  |  |
| Customer |  |  |  | Deposit |  |  |
| **customerSSN** | **street** | **city** |  | **customerSSN** | **accountNum** | **amount** |
| 123 | Fourth Street | Edina |  | 123 | 1 | 100 |
| 456 | Fifth Street | Plymouth |  | 456 | 2 | 200 |
| 789 | Sixth Street | Wayzata |  | 789 | 3 | 300 |

insert into Branch values("First Bank", "first Street", "Crystal"),

("Second Bank", "Second Street", "New Hope"),

("Third Bank", "Third Street", "Golden Valley");

insert into Account values(1, "First Bank", 100),

(2, 'Second Bank', 200), (3,"Third Bank", 300);

insert into Customer values (123, "Fourth Street", "Edina"),

(456, "Fifth Street", "Plymouth" ), (789, "Sixth Street", "Wayzata" );

insert into Deposit values(123, 1, 100), (456, 2, 200), (789, 3,300);

**Question 6 (Total 15 Points):**

Write SQL statements to answer questions based on the following schema (data types are not needed).

**Customer(cid, cname, age, gender)**

**Buys**(**cid, pid**, quantity)

**Product**(**pid**, pname, type, manufacturer, price)

6.1) (2 Points) Find name and price of all products that are manufactured by ‘Samsung’.

Select pname, price from Product where manufacturer = ‘Samsung’;

6.2) (4 Points) Find names of all customers who have purchased a product that is manufactured by ‘Target’ and price is between $25 and $50.

Select CBP.cname from

(select \* from Customer C ,Buys B , Product P where B.pid = P.pid and C.cid = B.cid) CBP

Where (CBP.manufactrued = ‘Target’) and (CBP.price between 25 and 50);

6.3) (4 Points) Find type and price of all products that have been purchased by a customer who is older than 32 years and quantity > 100.

select CBP.type and CBP.price from

(select \* from Customer C ,Buys B , Product P where B.pid = P.pid and C.cid = B.cid) CBP

where (CBP.age > 32) and (CBP.quantity > 100);

6.4) (3 Points) Write a SQL statement to decrease the amount of all price in Product table by 3%

update Product set price = price - price\*0.03;

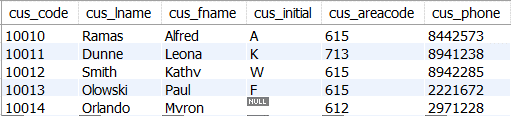
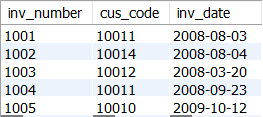
6.5) (2 Points) Write a SQL statement to select all Customers Name whose Age has a null value

select cname from Customer where age is null;

**Question 7 (Total 10 points):**

Answer questions based on the following tables:

**Customer** **Invoice**



7.1) (4 Points) Find customers residing in area codes ‘713’, ‘615’ or both. Your select statements must include customer last name, first name, area code and invoice number.

Use Set Operations (E.g. Union/Intersect/Except)

select \* from

((select C.cus\_lname, C.cus\_fname, C.cus\_areacode, I.inv\_number

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 713)

union

(select C.cus\_lname, C.cus\_fname, C.cus\_areacode, I.inv\_number

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 615)) PD;

7.2) (2 Points) Find customers residing in area code ‘713’ but not in ‘615’ . Your select statements must include customer code, last name, first name, phone and invoice date.

Use Set Operations (E.g. Union/Intersect/Except)

-- in SQL server

(select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 713)

except

(select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 615);

-- in Oracle

(select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 713)

MINUS

(select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 615);

-- in MySQL

select cus\_code, cus\_lname, cus\_fname, cus\_areacode, cus\_phone, inv\_date from

(select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 713) C

left join

(select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 615) D

on C.cus\_code = D.cus\_code ;

7.3) (4 Points) Answer question 7.2) using Nested Subqueries

select C.cus\_code, C.cus\_lname, C.cus\_fname, C.cus\_areacode, C.cus\_phone, I.inv\_date

from Customer C, Invoice I where C.cus\_code = I.cus\_code and C.cus\_areacode = 713

and (C.cus\_areacode not in (select cus\_areacode from Customer where cus\_areacode = 615));

**Question 8 (Total 10 points):**

Write SQL statements to answer questions 8.1, 8.2, 8.3 based on the following schema. Primary keys are( BOLD and italicized) and attributes with the same names in different tables are foreign keys.

**Doctors**(***doctorId***, doctorName, specialty)

**Patients**(***patientId***, patientName, sex, patientBalance, numberOfVisits)

**Records**(***doctorId, patientId***, ***date***, diagnosis, visitCost)

8.1) (2 Points) Find the cost of the most expensive visit (Remember: Aggregate functions)

select max(visitCost) from Recoeds;

8.2) (3 Points) For each doctor, list doctor identifier, doctor name and the total number of visits that were administered by the doctor.

select DPR.doctorId, DPR.doctorName, sum(DPR.numberOfVisits) from

(select \* from Doctors D, Patients P, Records R where D.doctorId = R.doctorId and P.patientId = R.patientId) DPR

group by DPR.doctorId;

8.3) (5 Points) Create a view called DoctorSmithsPatients for cost of visit for each patient of Dr. Smith. Note that a patient may visit the Dr multiple times on multiple days. The view must list DoctorID, patient name and Visit Cost.

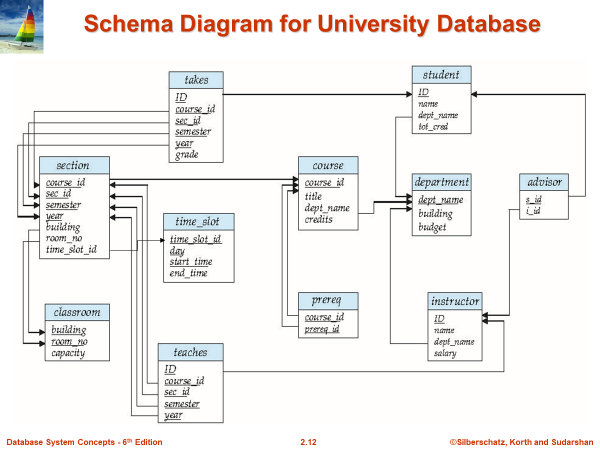
create view DoctorSmithsPatients as

select doctorId, patientName, visitCost from

(select \* from Doctors D, Patients P, Records R where D.doctorId = R.doctorId and P.patientId = R.patientId) DPR

where DPR.doctorName = 'Smith';

select \* from DoctorSmithsPatients;

****