First Name: ____________________ Last Name: ___________________ Student ID: __________

BBM 471 Database Management Systems

Midterm 1 - March, 30st, 2018

ANSWER KEY

Instructions

This exam contains **6** multi-part problems and you have **90 minutes** to earn 100 points. When the exam begins, please write your name on every page of this exam booklet. Check that the exam booklet contains **10 pages** to the exam, including this one. This exam is a **closed book and notes exam (except for the handwritten A4 sized sheet)**. Show all work, as partial credit will be given since you will be graded not only on the correctness and efficiency of your answers, but also on your clarity that you express it. Be neat.

Good luck!

Problem	Points	Score
1	15	
2	15	
3	15	
4	25	
5	15	
6	15	
Total	100	

It is a violation of the Academic Integrity Code to look at any exam paper other than your own, any other reference material (books, lecture notes) except for the handwritten A4 sized sheet, or to give inappropriate help to someone or to receive unauthorized aid by someone. Please also not discuss this exam with the students who are scheduled to take a makeup exam.

Academic Integrity is expected of all students of Hacettepe University at all times, whether in the presence or absence of members of the faculty. Do NOT sign nor take this exam if you do not agree with the honor code.

Understanding this, I declare I shall not give, use or receive unauthorized aid in this examination.

Signature: _________________________

First Name: ______________________ Last Name: _____________________ Student ID: ___________

Question 1 (15 points): Consider the *Person* and *PersonFriend* relations below:

Person (id, name)
PersonFriend (pid, fid)

The *Person* relation represents people in some social media platform. The *PersonFriend* relation records the friendship relation between two people, i.e., both *pid* and *fid* are references to *ids* in the *Person* relation.

Assume "Fuat" is the name of a person in the *Person* relation. Write an SQL query to return names of Fuat's friends.

SELECT p1.name
FROM Person p1, PersonFriend, Person p2
Where PersonFriend.fid = p1.id AND PersonFriend.pid = p2.id
WHERE p2.name = 'Fuat'

Question 2 (15 points): Definition of the NVL function is given in the Oracle's reference manual as follows: "`NVL` lets you replace null (returned as a blank) with a string in the results of a query. If `expr1` is null, then `NVL` returns `expr2`. If `expr1` is not null, then `NVL` returns `expr1`."

According to this definition, what are the equivalent functions to NVL (or, how can you achieve the same thing) in *MYSQL*, *Postgres* and *Microsoft SQL Server* databases?

MySQL : IFNULL or COALESCE
SQL Server: ISNULL
Postgres: COALESCE

Question 3 (15 points): Content of a *Customer* table is given below. Each row in the table shows a customer's name and where that customer is from (City and Country).

CustomerID	**Customer Name**	**City**	**Country**
1	Rancho grande	Buenos Aires	Argentina
2	Océano Atlántico Ltda.	Buenos Aires	Argentina
3	Cactus Comidas para llevar	Buenos Aires	Argentina
4	Ernst Handel	Graz	Austria
5	Piccolo und mehr	Salzburg	Austria
6	Maison Dewey	Bruxelles	Belgium
7	Suprêmes délices	Charleroi	Belgium
8	Gourmet Lanchonetes	Campinas	Brazil
9	Wellington Importadora	Resende	Brazil
10	Que Delícia	Rio de Janeiro	Brazil
11	Hanari Carnes	Rio de Janeiro	Brazil
12	Ricardo Adocicados	Rio de Janeiro	Brazil
13	Familia Arquibaldo	São Paulo	Brazil

First Name: ______________________ Last Name: _____________________ Student ID: ___________

14	Queen Cozinha	São Paulo	Brazil
15	Comércio Mineiro	São Paulo	Brazil
16	Tradição Hipermercados	São Paulo	Brazil

What will be the output of the query below when it was executed against the *Customer* table? If you think the query cannot be executed at all, explain why?

```
SELECT Country, COUNT(CustomerID) As Customer
FROM Customers
GROUP BY Country, City
Order By Country, City;
```

Country	Customers
Argentina	3
Austria	1
Austria	1
Belgium	1
Belgium	1
Brazil	1
Brazil	1
Brazil	3
Brazil	4

Question 4 (15+10=25 points): For a Patient-Diagnosis database, the following relation definitions are given:

Patient: Holds patient's id, name and birthdate.
Diagnosis: Holds the id and name of diagnosis e.g. Flu.
Diagnosed: Holds the date of diagnosis (*This relation should not have an artifial key like an auto incremental integer.*).
Visits: Holds the follow-up visits (only the dates of visits) of patients after being diagnosed/for a specific diagnosis.

a) According to definitions above, draw the ER diagram of the Patient-Diagnosis database by using the Chen notation.

Drawn in the lecture.

b) Write necessary SQL statements to implement the Patient-Diagnosis database schema (*Note: Your SQL statements do not need to be 100 % accurate in terms of syntax. They must show that you are familiar with SQL though.*).

Create Table Patient();
Create Table Diagnosis();
Create Table Diagnosed(pid, did, date_of_diagnosis);
Create Table Visit (pid, did, visit_date is the key, pid, did is the Foreign Key)

First Name: ____________________ Last Name: ____________________ Student ID: __________

Question 5 (15 points): A database schema containing two *tables A* and *B* are created by using the *create.sql* script given below. To populate records in *tables A* and *B*, the *populate.sql* script is wanted to be used.

Can you use the *populate.sql* script as it is to populate records in tables? If no, propose **three** alternative (and different) ways to populate tables.

Hint: You can make modifications in scripts without changing the data.

create.sql	**populate.sql**

`Create Table A (`
`      c1 integer Primary Key Not Null,`
`      c2 char(2) Not Null,`
`);`
`Create Table B (`
`      c1 integer Primary Key Not Null,`
`      c2 char(2) Not Null,`
`      c3 integer,`
`      Constraint B_FK`
`        Foreign Key`
`           c3 References A(c1)`
`);`
	`Insert into B Values (1, "A1", 1);`
	`Insert into A Values (1, "B1");`
	`Insert into B Values (2, "A2", 3);`
	`Insert into A Values (2, "B2");`
	`Insert into B Values (3, "A3", 2);`
	`Insert into A Values (3, "B3");`
	`Insert into B Values (4, "A4", 1);`

We can't, because of dependencies.
1) Sort statements in populate.sql
2) Drop FK constraint. Execute script. Add FK constraint.
3) Disable FK constraint checks in database. Execute script. Enable FK constraint checks in database.

Question 6 (15 points): Assume that you have a *table A* with attributes *a1, a2* and *a3*, i.e. *A (a1, a2, a3).* Further assume that the types of attributes (e.g., alphanumeric or numeric) are not known to you. Write **two** different SQL scripts (or set of statements) to create a *table B* with attributes *a1* and *a3* of the *table A*, **including** the data in those attributes, i.e. *B (a1, a3)*.

1)
Create Table B LIKE A
Drop Column B.a2
Insert into B Select a1, a3 From A

2)
Create Table B As Select a1, a3 From A