CMPE 351 Database Systems Final Examination

27 December 2019 Friday 15:00　　　　*Duration is 75 mins*

ID:　　　　　　　　　　# SOLUTION & RUBRIC
Name:
Signature:

Q1) Following *StdCourse* Relation has the composite primary key set (<u>StdID</u>, <u>CourseID</u>). You see three records in the relation, which are just an examples. Assume that more than 1 million records exist in the table.

StdCourse

<u>StdID</u>	<u>CourseID</u>	Grade	StdName	Cname	StdDeptID	StdDeptName
1234	CMPE 351	B+	Mary	Database	999	Computer Engineering
3456	CMPE 351	A-	Alice	Database	888	Psychology
5555	PH 101	D+	Adam	Physics	999	Computer Engineering
… + 1 millon more	….	...	...	...	...	...

Attribute Descriptions
- **StdID**: Student ID
- **Course ID:** Course ID
- **Grade**: Student grade for the corresponding course
- **StdName**: Student Name
- **Cname**: Course name
- **StdDeptID**: ID of the student's department
- **StdDeptName**: Name of the student's department

A) Provide the functional dependencies in the form　**(e.g. A → {B} or {A,B} → {C} or {A} → {C,D} and so forth)**

There exist 5 functional dependencies as follows:
1. **{StdID, CourseID} → Grade**
2. **StdID → StdName**
3. **CourseID → Cname**
4. **StdId → StdDeptID**
5. **StdDepthID → stdDeptName**

Rubric:
Each FD 2 points, each incorrect FD costs -2 points (Total you can get up to 10 points from this part A)

B) Normalize the relation into 2nd Normal Form (2NF) and provide functional dependencies.

2nd Normal Form indicates that each non-prime attributes need to be fully depend on prime keys.
We can craft three relations

R1(StdId, CourseID, Grade)
  FD:(StdId, CourseID) -> Grade)

R2(StdId, StdName, StdDeptID, StdDeptName)
  FD: StdId → {StdName, StdDeptID}
  FD: StdDeptID → StdDeptName

R3(CourseID, Cname)
  FD: CourseID → Cname

RUBRIC: each table 3 points. Total points 9
Each incorrect table costs -3

C) Normalize the relation into 3rd Normal Form (3NF)

R2 above has transitive dependencies such as StdID→ StdDeptID→ StdDeptName
We can decompose the relation as follows

R2.1(StdId, StdName, StdDeptID)
R2.2(StdDeptID, StdDeptName)

RUBRIC: each table 4 points. 8 points

YOU CAN GET AT MOST <u>**25 POINTS**</u> FROM THE Q1

Q2) Extract all functional dependencies in the table below, by taking only one-to-one relation into consideration such as (S → T)

Apple	Banana	Grape	Milk
1	a	p	x
1	a	r	y
2	b	n	z
2	b	n	x
2	c	n	z

There are two one-to-one FD as follows

Banana → Apple
Grape → Apple

RUBRIC points = max((# of correct FDs - # of inccorect FD), 0) * 12
+1 if the points ==24

Q3) A) Match the followings

A.	Pig	**4**	**C**	1	Library and runtime environment for efficiently moving data between relational databases and HDFS
B.	Oozie	**3**	**D**	2	Column-oriented key-value store that uses HDFS
C.	Sqoop	**1**	**B**	3	Service for scheduling and running workflows of jobs
D.	HBase	**2**	**A**	4	Provides higher-level interface for working with Hadoop framework

RUBRIC = # correct matching * 3 points

B) What is the CAP Theorem in the distributed database management?

Q4) Apply Inverted Index Method (as one of the information retrieval methods) to the following documents

D1: "The cat sat on the mat"
D2: "The black cat will sit on the mat"
D3: "The dog sit on the table"

ID	Term	Document Positions
0	the	D1:0, D1:4, D2:0, D2:6, D3:0,D3:4
1	cat	D1:1, D2:2
2	sat	D1:2
3	on	D1:3, D2:5, D3:3
4	mat	D1:5, D2:7
5	black	D2:1
6	will	D2:3
7	sit	D2:4, D3:2
8	dog	D3:1
9	table	D3:5