King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Information and Computer Science Department

ICS324 Database Systems

Final Exam (131)

Name:

.

ID#:

Question #	Maximum Grade	Obtained Grade
1	**20**	
2	**30**	
3	**30**	
4	**20**	
Total	**100**	

Wednesday January 1, 2014

Question 1: [Mark the following as True or False – 20 points]

F	In the database approach of data management, each end user defines and implements the files needed for a specific software application.
T	Enforcing integrity constraints is one of the advantages of using the database approach in data management.
T	The database state refers to the actual data stored in a database at a particular moment in time.
T	In the three-schema architecture, the conceptual level describes the structure of the whole database for a community of users.
F	Data independence in a DBMS refers to the ability to change the DB state without having to change the schema at a certain level of a database system.
F	The ordering of tuples in relations is important, unlike the ordering of values within a tuple.
T	Duplicated tuples are not allowed in the relational data model.
T	In the relational model, domain constraints, key constraints, and entity-integrity constraints are sample schema-based constraints that can be specified using DDL/SQL.
F	Requiring the primary key of a relation not to be equal to NULL is an example referential integrity-based constraint.
F	Enforcing the prohibition of allowing an unmarried employee from claiming spouse (wife) insurance can be achieved at the schema level.
T	One of the options available when deleting a tuple in a database is to reject the delete operation altogether.
F	The SQL statement: "`UPDATE employee SET dno = 2;`" sets the dno value of only the first tuple in the "employee" relation to 2.
F	A view in Oracle DBMS cannot be updated, since it may consist of a join of more than one relation.
T	In SQL, the outer join operations may retain rows that do not satisfy the join condition.
T	SQL, to a great extent, corresponds to relational algebra, with possible differences in the notation.
T	Relational algebra is capable of specifying aggregate functions.
F	The following relational domain calculus expression is not valid due to the use of "NOT". `{ qs	employee(qrstuvwxyz)` `AND NOT (∃k) (dependent(kmnop)` `AND t = k)` `}`
T	*Query-by-Example* (QBE) is largely based on domain relational calculus.
F	The *Entity-Relationship model* (ER) can specify generalization and specialization relationships between entities.
T	Hashing for disk files is usually referred to as *External Hashing*.

Question 2: [General Concepts in DB Modeling and Design - 30 points]

I. (5 points) Explain the difference between a shared class and a union

A shared subclass member must exist in all of its superclasses whereas a category member must exist in at least one of its superclasses

II. (5 points) Give two main reasons why DB projects fail.

1. **Lack of complete requirements specification.**
2. **Lack of appropriate development methodology.**
3. **Poor decomposition of design into manageable components.**

III. (6 points) Write 3 factors affecting the choice of DBMS.

1. **Technical**
2. **Cost**
3. **Political**

IV. (3 points) What is the advantage of denormalization?

To improve the performance of frequently occurring queries and transactions.

V. (6 points) When processing an SQL statement the query processor first scans, parses and validates the query. Briefly explain what is accomplished during

a. Scanning

Identification of the query tokens, such as, SQL keywords, attribute names, and relation names.

b. Parsing

Checking of the query syntax to determine whether it is formulated according to the grammar of the query language.

c. validation

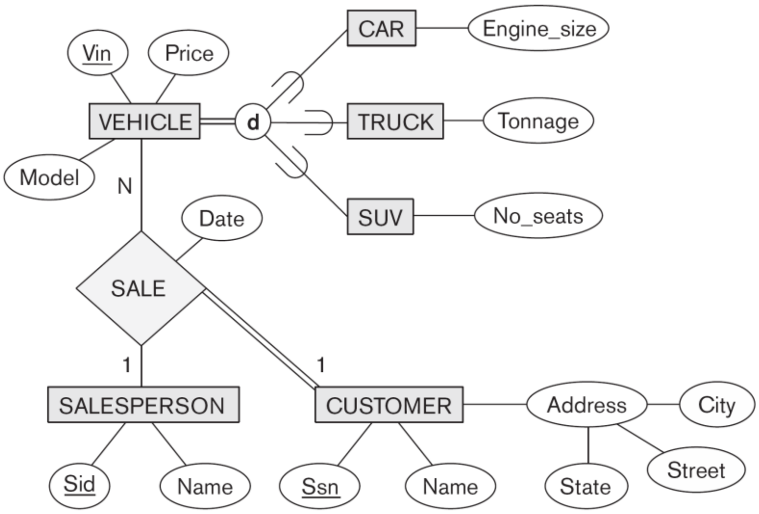
Checking if all attribute names and relation names are valid according to the schema of the particular database being queried.

VI. (5 points) Write two goals of DB tuning.

1. **To make application run faster**
2. **To lower the response time of queries/transactions**
3. **To improve the overall throughput of transactions**

Question 3: [Conceptual Modeling and Relational Database Design – 30 points]

I. (15 points) Consider the following extended entity-relationship diagram:



a. (5 points) Assume that customers can either be regular persons, companies or government agencies, and that they are all disjoint. The following information is available regarding each type:

   i. Person: <u>Ssn</u>, Name, Address (City, Street, State)

   ii. Company: <u>Cid</u>, Name, Address (City, Street, State), Type

   iii. Government Agency: <u>Aid</u>, Name, Address (City, Street, State), Division

   Update the above EER to include this information.

b. (10 points) Convert the updated EER diagram of part (a) into a relational schema.

II. (15 points) Consider the relation *Courses* (C, T, H, R, S, G), whose attributes may be thought of informally as course, teacher, hour, room, student, and grade. Let the set of FD's for *Courses* be $C \rightarrow T$, $HR \rightarrow C$, $HT \rightarrow R$, $HS \rightarrow R$, and $CS \rightarrow G$. Intuitively, the first says that a course has a unique teacher, and the second says that only one course can meet in a given room at a given hour. The third says that a teacher can be in only one room at a given hour, and the fourth says the same about students. The last says that students get only one grade in a course.

a. (4 points) What is the key for *Courses*?

Any key will be a subset of {C,H,R,T,S}

The key is HS since {HS}+ = {H,S,R,C,T,G}. Note that no other key exists.

b. (3 points) Is this relation in 1NF? Justify.

Yes. All tuples are atomic (no multi-valued attributes) and no repeating groups.

c. (3 points) Is this relation in 2NF? Justify.

Yes. No non-key value can be determined using a proper subset of the primary key.

d. (5 points) Convert the above relation *Courses* into 3NF relations.

R_1=<C,T>, R_2=<H,R,C>, R_3=<H,T,R>, R_4=<H,S,R>, R_5=<C,S,G>

Question 4: [Database Indexing and Query Optimization – 20 points]

I. (10 points) Show all the steps of heuristically optimizing the following SQL statement.

```
SELECT ssn, fname, dname, pname
FROM employee, department, project, works_on
WHERE salary > 10000
AND hours  >  4
AND pnumber = 1
AND  pno = pnumber
AND dnumber = dno
AND dnum = dnumber
```

$$(\prod_{ssn,fname,dname,pname}($$

$$(\prod_{ssn,fname,dno}(\sigma_{salary>10000}(\textbf{employee})))$$

$$\bowtie_{dno=dnumber}$$

$$(\prod_{pno,dnumber,dname,pname}($$

$$(\prod_{pno}(\sigma_{hours>4}(\textbf{Works\_on})))$$

$$\bowtie_{pno=pnumber}$$

$$(\prod_{dnumber,pno,pname,dname}($$

$$(\prod_{dnumber,dname}(\textbf{department}))$$

$$\bowtie_{dno=dnum}$$

$$(\prod_{pnumber,pname,dnum}(\sigma_{pnumber=1}(\textbf{project}))))))$$

II. (10 points) Assume the following create table statement in answering the following question

```
CREATE TABLE TAB01 (  Col1 CHAR(4),
                      Col2 CHAR(4),
                      Col3 CHAR(4),
                      Col4 CHAR(4)
                      PRIMARY KEY(Col1),
                      UNIQUE(Col2)
                      );
```

Assume TAB01 is a sequential file ordered by Col3.

a. (5 points) Write "X" on the cell where the type is the correct one of the corresponding index.

Index on	Primary	Clustering	Secondary	Sparse	Dense
Col1			X		X
Col2			X		X
Col3		X		X	
Col4			X	X	X

b. (5 points) According to your answer in Part a, calculate the size of the index created on Col2, assuming that:

Disk block size = 1024 bytes
Record pointer = 8bytes
Block pointer = 6 bytes
Records in TAB01 = 10,000

Size of level 2 index record = 4 + 8 = 12

Bfr = Floor(1024/12) = 85

Level 2 index blocks = Ceiling(10,000/85) = 118

Size of level 1 index record = 4 + 6 = 10

Bfr = Floor(1024/10) = 102

Level 1 index blocks = ceiling(118/102) = 2

Level 0 index blocks = ceiling(2/102) = 1

Total number of index blocks = 118 + 2 + 1 = 121

Size in Kilobytes = 121

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

CREATE TABLE <table name> (<column name> <column type> [<attribute constraint>] { , <column name> <column type> [<attribute constraint>] } [<table constraint> { , <table constraint> }])
DROP TABLE <table name>
ALTER TABLE <table name> ADD <column name> <column type>
SELECT [DISTINCT] <attribute list>
FROM (<table name> { <alias> } | <joined table>) { , (<table name> { <alias> } | <joined table>) }
[WHERE <condition>]
[GROUP BY <grouping attributes> [HAVING <group selection condition>]]
[ORDER BY <column name> [<order>] { , <column name> [<order>] }]
<attribute list> ::= (* | (<column name> | <function> (([DISTINCT] <column name> | *)))
{ , (<column name> | <function> (([DISTINCT] <column name> | *)) }))
<grouping attributes> ::= <column name> { , <column name> }
<order> ::= (ASC | DESC)
INSERT INTO <table name> [(<column name> { , <column name> })]
(VALUES (<constant value> , { <constant value> }) { , (<constant value> { , <constant value> }) }
| <select statement>)
DELETE FROM <table name> [WHERE <selection condition>]
UPDATE <table name> SET <column name> = <value expression> { , <column name> = <value expression> }
[WHERE <selection condition>]
CREATE VIEW <view name> [(<column name> { , <column name> })] AS <select statement>
DROP VIEW <view name>