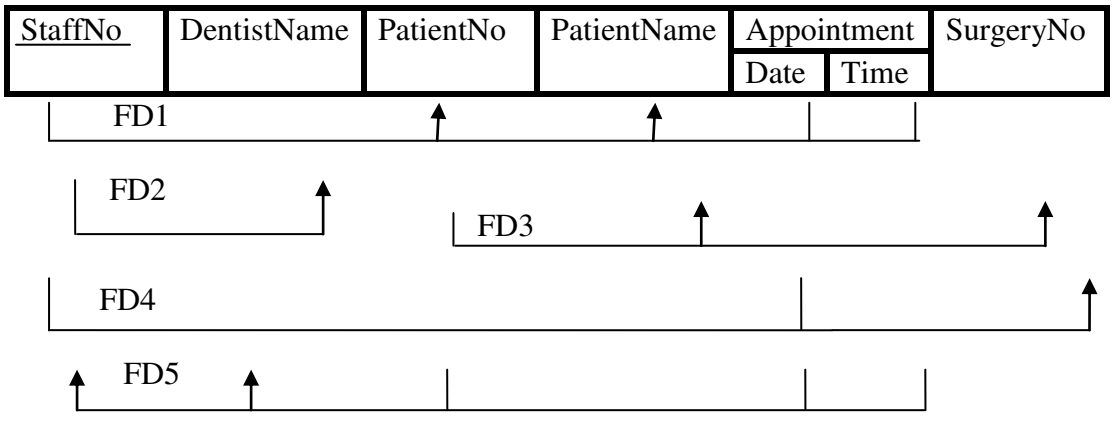
CMR
INSTITUTE OF
TECHNOLOGY

Internal Assessment Test II – November 2016

Sub:	Data Base Management Systems							Code:	10CS54
Date:	03/11/2016	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	ISE

	Note: Answer any five full questions	Marks	OBE CO	OBE RBT
1	**a)** How is a view created and dropped? What problems are associated with updating of views?	5M	CO5	L1
	b) With program segments, explain retrieving of tuples (single and multiple) using embedded SQL.	5M	CO5	L4
2	**a)** Describe how triggers and assertions are defined in SQL?Give examples.	5M	CO5	L2
	b) Explain the concept of stored procedures using example.	5M	CO5	L4
3	**a)** List the inference rules for functional dependencies.Write the algorithm to determine the closure of X(set of attributes) under F(set of functional dependencies) with an example.	6M	CO6	L1
	b) Define 2 NF and 3NF with suitable examples.	4M	CO6	L1
4	**a)** Describe the syntax of SELECT statement in SQL.	4M	CO5	L1
	b) What are insertion,deletion and modification anomalies.Describe with examples.	6M	CO6	L2
5	**a)** Consider R={A B C D E F};FDs{A →BC,B →E,CD →EF} Show that AD → F.	3M	CO6	L3
	b) Consider the following relation Book_title | Auth_Name | Book_Type | ListPrice | Affiliation | Publication FDs{Book_title→Book_Type,Publication Auth_Name →Affiliation Book_Type→ListPrice}**Find the Key and Normalise**.	4M	CO6	L4
	c) Given below are 2 sets of Functional Dependencies.Are they equivalent? i)A →B, AB →C,D→ AC, D →E ii)A →BC, D→ AE	3M	CO6	L3
6	Consider the following relations for a database. Supplier (Sno, Sname, Status, City) Product(Pno, Pname, Color, Weight, City) Shipments (Sno, Pno, Qty) **Solve the following questions using SQL**. i. Retrieve names of supplier who supply product P2. ii. Retrieve the names of suppliers who do not supply any product supplied by S2. iii. Retrieve product number for all products supplied by more than one supplier. iv. For each product supplied, get the product number, maximum quantity, minimum quantity supplied for that product. v. Retrieve supplier numbers for suppliers with status less than the current maximum in the supplier table.	10M	CO6	L3
7	State the informal guidelines for relational schema design. Illustrate how violation of these guidelines may be harmful.	10M	CO6	L3
8	 Normalise the above table to 3NF.	10M	CO6	L4

Course Outcomes		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1:	Describe features, classifications and characteristics of database systems.	1	2	1	1	1	1	1	-		-	-	1
CO2:	Design an information model for given requirements expressed in the form of an entity relation diagram.	1	3	3	1	1	2	-	-	1	-	-	1
CO3:	Design a relational data model for a given information model.	1	3	3	1	1	2	-	-	1	-	-	1
CO4:	Write relational algebra query for a given problem.	1	3	3	1	1	2	-	-	1	-	-	1
CO5:	Write SQL for CRUD to fulfill given requirement.	1	3	3	1	1	2	-	-	1	-	-	1
CO6:	Apply normalization techniques for a given relational model.	1	3	3	1	1	2	-	-	1	-	-	1

Cognitive level	KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PO1 - *Engineering knowledge*; PO2 - *Problem analysis*; PO3 - *Design/development of solutions*; PO4 - *Conduct investigations of complex problems*; PO5 - *Modern tool usage*; PO6 - *The Engineer and society*; PO7- *Environment and sustainability*; PO8 – *Ethics*; PO9 - *Individual and team work*; PO10 - *Communication*; PO11 - *Project management and finance*; PO12 - *Life-long learning*

CMR
INSTITUTE OF
TECHNOLOGY

Internal Assessment Test II – November 2016

Sub:	Data Base Management Systems							Code:	10CS54
Date:	03/11/2016	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	ISE

Note: Answer any five full questions — **Marks**

1 **a)** How is a view created and dropped? What problems are associated with updating of views?

```
CREATE VIEW   WORKS_ON1
AS SELECT     Fname, Lname, Pname, Hours
    FROM      EMPLOYEE, PROJECT, WORKS_ON
    WHERE     Ssn=Essn AND Pno=Pnumber;
```

```
DROP VIEW WORKS_ON1;
```

Updating of views is complicated and can be ambiguous. In general, an update on a view defined on a single table without any aggregate functions can be mapped to an update on the underlying base table under certain conditions. For a view involving joins, an update operation may be mapped to update operations on the underlying base relations in multiple ways. Hence, it is often not possible for the DBMS to determine which of the updates is intended.

```
UPDATE WORKS_ON1
SET       Pname = 'ProductY'
WHERE     Lname='Smith' AND Fname='John'
          AND Pname='ProductX';
```

Here are two possible updates, (a) and (b), on the base relations corresponding to the view
update operation

```
UPDATE WORKS_ON
SET       Pno =   ( SELECT   Pnumber
                    FROM     PROJECT
                    WHERE    Pname='ProductY' )
WHERE     Essn IN ( SELECT   Ssn
                    FROM     EMPLOYEE
                    WHERE    Lname='Smith' AND Fname='John' )
          AND
          Pno =   ( SELECT   Pnumber
                    FROM     PROJECT
                    WHERE    Pname='ProductX' );

UPDATE PROJECT   SET    Pname = 'ProductY'
WHERE     Pname = 'ProductX';
```

A view with a single defining table is updatable if the view attributes contain the primary key of the base relation, as well as all attributes with the NOT NULL constraint that do not have default values specified.

- Views defined on multiple tables using joins are generally not updatable.
- Views defined using grouping and aggregate functions are not updatable.

5M

b) With program segments, explain retrieving of tuples (single and multiple) using embedded SQL.

5M

```
0) prompt("Enter the Department Name: ", dname) ;
1) EXEC SQL
2)    select Dnumber into :dnumber
3)    from DEPARTMENT where Dname = :dname ;
4) EXEC SQL DECLARE EMP CURSOR FOR
5)    select Ssn, Fname, Minit, Lname, Salary
6)    from EMPLOYEE where Dno = :dnumber
7)    FOR UPDATE OF Salary ;
8) EXEC SQL OPEN EMP ;
9) EXEC SQL FETCH from EMP into :ssn, :fname, :minit, :lname, :salary ;
10) while (SQLCODE == 0) {
11)    printf("Employee name is:", Fname, Minit, Lname) ;
12)    prompt("Enter the raise amount: ", raise) ;
13)    EXEC SQL
14)      update EMPLOYEE
15)      set Salary = Salary + :raise
16)      where CURRENT OF EMP ;
17)    EXEC SQL FETCH from EMP into :ssn, :fname, :minit, :lname, :salary ;
18)    }
19) EXEC SQL CLOSE EMP ;
```

2 **a)** Describe how triggers and assertions are defined in SQL?Give examples.

A trigger is Event-Condition-Action (ECA) model.

1. The event(s) that triggers the rule: These events are usually database update operations that are explicitly applied to the database.However, in the general model, they could also be temporal events2 or other kinds of external events.

2. The condition that determines whether the rule action should be executed: Once the triggering event has occurred, an optional condition may be evaluated.If no condition is specified, the action will be executed once the event occurs. If a condition is specified, it is first evaluated, and only if it evaluates to true will the rule action be executed.

3. The action to be taken: The action is usually a sequence of SQL statements,but it could also be a database transaction or an external program that will be automatically executed.

```
CREATE TRIGGER Total_sal1
AFTER INSERT ON EMPLOYEE
FOR EACH ROW
WHEN ( NEW.Dno IS NOT NULL )
UPDATE DEPARTMENT
SET Total_sal = Total_sal + NEW.Salary
WHERE Dno = NEW.Dno;
```

In SQL, users can specify general constraints via declarative assertions, using the CREATE ASSERTION statement of the DDL.

```
CREATE ASSERTION SALARY_CONSTRAINT
CHECK ( NOT EXISTS ( SELECT *
FROM EMPLOYEE E, EMPLOYEE M,
DEPARTMENT D
WHERE E.Salary>M.Salary
AND E.Dno=D.Dnumber
AND D.Mgr_ssn=M.Ssn ) );
```

5M

b) Explain the concept of stored procedures using example.

The term used in the SQL standard for stored procedures is persistent stored modules because these programs are stored persistently by the DBMS, similarly to the persistent data stored by the DBMS.

Stored procedures are useful in the following circumstances:

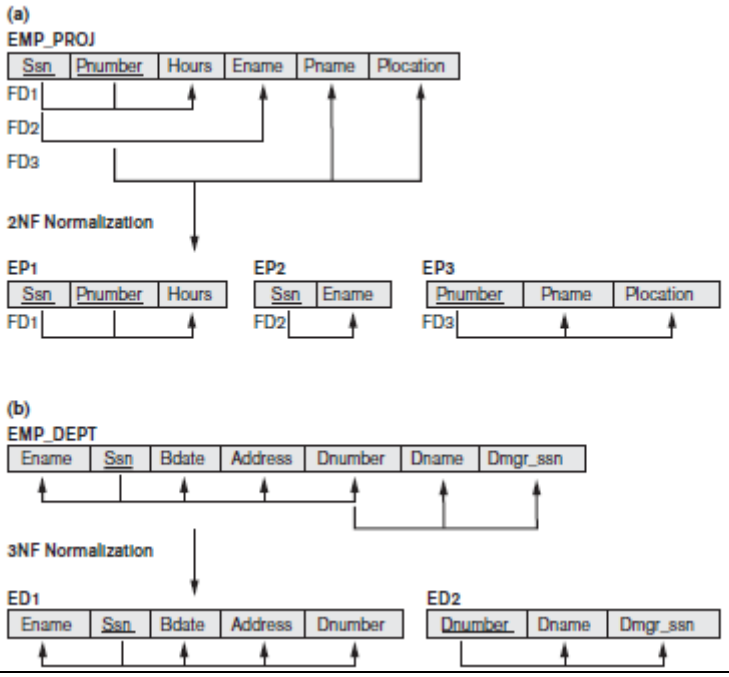
■ If a database program is needed by several applications, it can be stored at the server and invoked by any of the application programs. This reduces duplication of effort and improves software modularity.

■ Executing a program at the server can reduce data transfer and communication cost between the client and server in certain situations.

5M

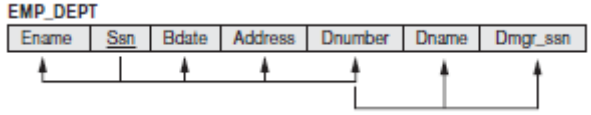
	■ These procedures can enhance the modeling power provided by views by allowing more complex types of derived data to be made available to the database users. Additionally, they can be used to check for complex constraints that are beyond the specification power of assertions and triggers. The general form of declaring stored procedures is as follows: CREATE PROCEDURE <procedure name> (<parameters>) <local declarations> < procedure body> ;	

3	**a)** List the inference rules for functional dependencies.Write the algorithm to determine the closure of X(set of attributes) under F(set of functional dependencies) with an example.
	a) IR1 (reflexive rule)1: If X ⊇ Y, then X→Y.
	b) IR2 (augmentation rule)2: {X→Y} |=XZ→YZ.
	c) IR3 (transitive rule): {X→Y, Y→Z} |=X→Z.
	d) IR4 (decomposition, or projective, rule): {X→YZ} |=X→Y.
	e) IR5 (union, or additive, rule): {X→Y, X→Z} |=X→YZ.
	f) IR6 (pseudotransitive rule): {X→Y,WY→Z} |=WX→Z.
	g) Define 2 NF and 3NF with suitable examples.
	
	Determining X+, the Closure of X under F
	Input: A set F of FDs on a relation schema R, and a set of attributes X, which is a subset of R.
	X+ := X;
	repeat
	oldX+ := X+;
	for each functional dependency Y→Z in F do
	if X+ ⊇ Y then X+ := X+ ∪ Z;
	until (X+ = oldX+);
		6M

	b) Define 2 NF and 3NF with suitable examples. A relation schema *R* is in 2NF if every nonprime attribute *A* in *R* is *fully functionally dependent* on the primary key of *R*. **Definition.** According to Codd's original definition, a relation schema *R* is in **3NF** if it satisfies 2NF *and* no nonprime attribute of *R* is transitively dependent on the primary key.  	4M
4	**a)** Describe the syntax of SELECT statement in SQL. The basic form of the SELECT statement, sometimes called a mapping or a select-from-where block, is formed of the three clauses SELECT, FROM, and WHERE and has the following	4M

form:
SELECT <attribute list>
FROM <table list>
WHERE <condition>;
where
■ <attribute list> is a list of attribute names whose values are to be retrieved by the query.
■ <table list> is a list of the relation names required to process the query.
■ <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query.
Retrieve the birth date and address of the employee(s) whose name is 'John B. Smith'.
SELECT Bdate, Address
FROM EMPLOYEE WHERE Fname='John' AND Minit='B' AND Lname='Smith';

b) What are insertion,deletion and modification anomalies.Explain with examples.



Insertion Anomalies. Insertion anomalies can be differentiated into two types,illustrated by the following examples based on the EMP_DEPT relation:
■ To insert a new employee tuple into EMP_DEPT, we must include either the attribute values for the department that the employee works for, or NULLs (if the employee does not work for a department as yet). For example, to insert a new tuple for an employee who works in department number 5, we must enter all the attribute values of department 5 correctly so that they are consistent with the corresponding values for department 5 in other tuples in
EMP_DEPT.
■ It is difficult to insert a new department that has no employees as yet in the EMP_DEPT relation. The only way to do this is to place NULL values in the attributes for employee. This violates the entity integrity for EMP_DEPT because Ssn is its primary key.Moreover, when the first employee is assigned to that department, we do not need this tuple with NULL values any more

Deletion Anomalies. The problem of deletion anomalies is related to the second insertion anomaly situation just discussed. If we delete from EMP_DEPT an employee tuple that happens to represent the last employee working for a particular department, the information concerning that department is lost from the database.
Modification Anomalies. In EMP_DEPT, if we change the value of one of the attributes of a particular department—say, the manager of department 5—we must update the tuples of all employees who work in that department; otherwise, the database will become inconsistent. If we fail to update some tuples, the same department will be shown to have two different values for manager in different employee tuples, which would be wrong.

6M

5	**a)**	Consider R={A B C D E F};FDs{A →BC,B →E,CD →EF} Show that AD → F.	3M

1. A → BC (given)
2. B → E (given)
3. CD → EF (given)
4. AD → BCD (1, aug)
5. AD → CD (4, decomp)
6. AD → EF (5,3 trans)
7. AD → F (6, decomp)

b) Consider the following relation — 4M

Book_title	Auth_Name	Book_Type	ListPrice	Affiliation	Publication

FDs{Book_title→Book_Type,Publication
Auth_Name → Affiliation
Book_Type→ListPrice}**Find the Key and Normalise**.

c) Given below are 2 sets of Functional Dependencies.Are they equivalent? — 3M
i)A→B, AB →C,D→ AC, D →E ii)A →BC, D→ AE

6 Consider the following relations for a database. — 10M

Supplier (Sno, Sname, Status, City)
Product(Pno, Pname, Color, Weight, City)
Shipments (Sno, Pno, Qty) **Solve the following questions using SQL**.

i. Retrieve names of supplier who supply product P2.

```
select sname
  from Supplier
  where Sno in (select Sno
          from Shipments
          where Pno = 'p2');
```

ii. Retrieve the names of suppliers who do not supply any product supplied by S2.

```
select sname
  from Supplier
  where Sno in (select Sno
          from Shipments
          where Pno = 'p2');
```

iii. Retrieve product number for all products supplied by more than one supplier.

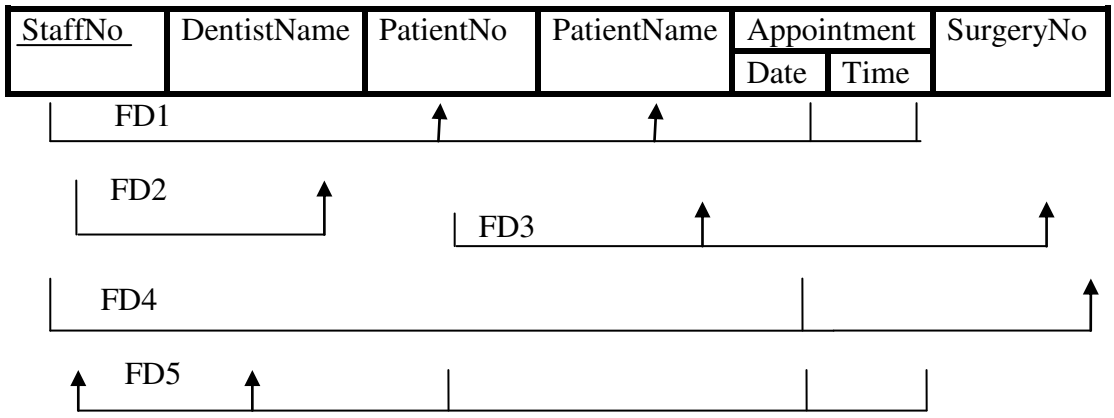
```
select pno
from Shipments
group by pno
having count(*) > 1;
```

iv. For each product supplied, get the product number, maximum quantity, minimum quantity supplied for that product.

```
select pno, max(qty), min(qty)
from Shipments
group by pno
```

v. Retrieve supplier numbers for suppliers with status less than the current maximum in the supplier table.

```
select sname
   from s
   where status < (select max(status)
            from s
            where status is not null);
```

7	State the informal guidelines for relational schema design. Illustrate how violation of these guidelines may be harmful. • Making sure that the semantics of the attributes is clear in the schema • Reducing the redundant information in tuples • Reducing the NULL values in tuples • Disallowing the possibility of generating spurious tuples	10M
8	 Normalise the above table to 3NF.	10M