

TED (10)-4070

Reg. No.

(REVISION—2010)

Signature.....

FOURTH SEMESTER DIPLOMA EXAMINATION IN ENGINEERING/
TECHNOLOGY—OCT,2012

DATABASE MANAGEMENT SYSTEMS

(Common for CT and IF)

[Time : 3 hours

(Maximum marks : 100)

PART—A

- I. Answer the following questions in one or two sentences. Each question carries 2 marks.

1. Define an entity.

Entity: A distinguishable object that has an independent existence in the real world

2. Draw the symbol used to represent a primary key attribute in an ER-diagram



3. Define Functional dependency.

- Constraints on the set of legal relations.
- Require that the value for a certain set of attributes determines uniquely the value for another set of attributes.
- A functional dependency is a generalization of the notion of a *key*

4. Name the command used to identify embedded SQL request.

EXEC SQL

5. Give the command to make a transaction permanent.

COMMIT command

PART B

II. Answer *any five* of the following. Each question carries 6 marks.

1. List the advantages of DBMS.

- Controlling redundancy in data storage and in development and maintenance efforts.
- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing Inferences and Actions using rules

2. Logical Data independence is very difficult to implement. Justify the statement.

Logical Data Independence – the ability to modify the logical schema without changing the application programs

Applications depend on the logical schema via the Views

Can be supported on a limited basis only (if view is not affected)

3. List any 4 keys in a database environment. Explain the use of keys in general.

Superkey: A set of one or more attributes, when taken together, helps in uniquely identifying each entity

Candidate key: A minimized set of attributes that can be used to uniquely identify a single entity instance

Primary key: The candidate key which is chosen by the db designer to uniquely identify entities

Composite key: A primary key formed by the combination of two or more attributes

4. Distinguish between specialization and generalization.

Specialization	generalization.
<ul style="list-style-type: none">● Is the process of defining a set of subclasses of a superclass● The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass● Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon <i>job type</i>.<ul style="list-style-type: none">– May have several specializations of the same superclass	<ul style="list-style-type: none">● The reverse of the specialization process● Several classes with common features are generalized into a superclass; original classes become its subclasses● Example: CAR, TRUCK generalized into VEHICLE; both CAR, TRUCK become subclasses of the superclass VEHICLE.

5. Define normalization. Explain 1NF & 2NF.

Database normalization is the process of removing redundant data from your tables in to improve storage efficiency, data integrity, and scalability.

In the relational model, methods exist for quantifying how efficient a database is.

These classifications are called normal forms (or NF), and there are algorithms for converting a given database between them.

Normalization generally involves splitting existing tables into multiple ones, which must be re-joined or linked each time a query is issued.

• **First Normal Form (1NF)**

A table is considered to be in 1NF if all the fields contain

only scalar values (as opposed to list of values).

- **Second Normal Form (2NF)**

For a table to be in 2NF, there are two requirements

- The database is in first normal form
- All nonkey attributes in the table must be functionally dependent on the entire primary key

Note: Remember that we are dealing with non-key attributes

6. **Table 'SALES' contain the attributes date, product code and number of items sold. Write suitable command to find total number of each item sold.**

SALES

DATE	PRODUCT CODE	NO.OF ITEMS SOLD
------	--------------	------------------

SQL>Select COUNT(*) from SALES;

7. **Define a trigger and list the commands that invoke the execution of the trigger.**

For any event that causes a change in the contents of a table, a user can specify an associated action that the DBMS should carry out. The three events that can trigger an action are attempts to INSERT, DELETE, or UPDATE rows of the table. The action triggered by an event is specified by a sequence of SQL statements. To understand how a trigger works, let's examine a concrete example. When a new order is added to the ORDERS table, these

Two changes to the database should also take place:

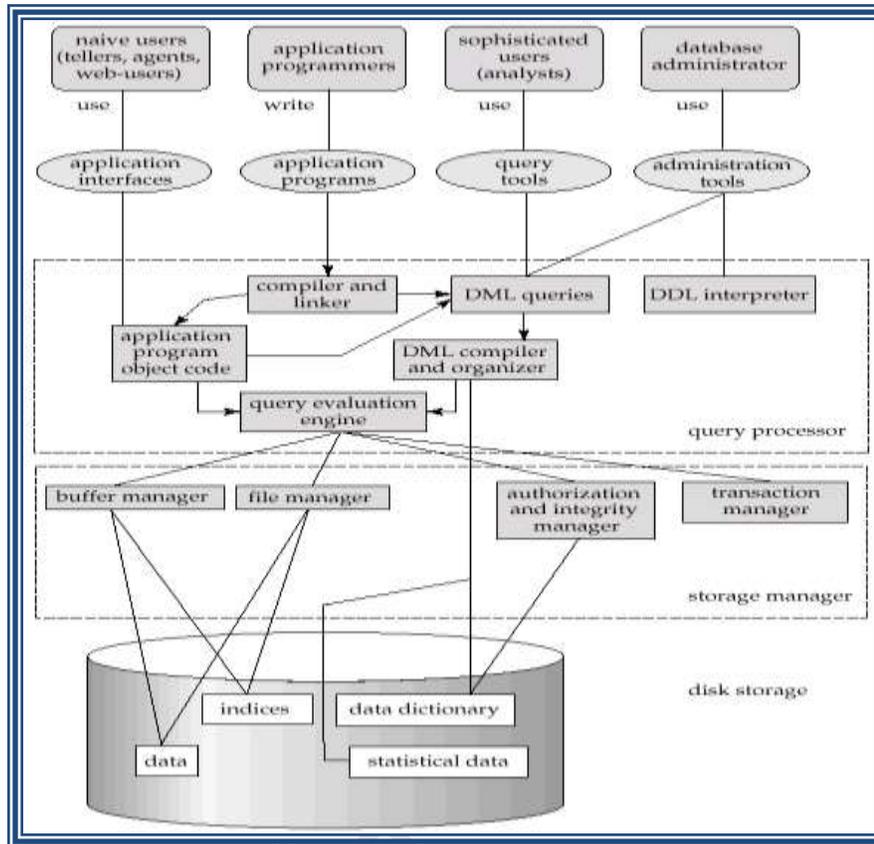
- The SALES column for the salesperson who took the order should be increased by the amount of the order.
- The QTY_ON_HAND amount for the product being ordered should be decreased by the quantity ordered.

PART—C

(Answer one full question from each unit. Each question carries 15 marks.)

UNIT--I

III. (a) With neat sketch Explain DBMS architecture.



- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
 - interaction with the file manager
 - efficient storing, retrieving and updating of data

Concurrency Control

- Concurrent execution of user programs is essential for good DBMS performance.
 - Because disk accesses are frequent, and relatively slow, it is important to keep the cpu humming by working on several user programs concurrently.

- Interleaving actions of different user programs can lead to inconsistency: e.g., check is cleared while account balance is being computed.
- DBMS ensures such problems don't arise: users can pretend they are using a single-user system.

Transaction Management

- A *transaction* is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

(b). Distinguish centralized and distributed database architecture.

- **A distributed database** is a database in which portions of the database are stored on multiple computers within a network. Users have access to the portion of the database at their location so that they can access the data relevant to their tasks without interfering with the work of others. A centralized distributed database management system (DDBMS) manages the database as if it were all stored on the same computer. The DDBMS synchronizes all the data periodically and, in cases where multiple users must access the same data, ensures that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere.

Centralized Systems:

- Run on a single computer system and do not interact with other computer systems.
- General-purpose computer system: one to a few CPUs and a number of device controllers that are connected through a common bus that provide access to shared memory.
- Single-user system (e.g., personal computer or workstation): desk-top unit, single user, usually has only one CPU and one or two hard disks; the OS may support only one user.

- Multi-user system: more disks, more memory, multiple CPUs, and a multi-user OS. Serve a large number of users who are connected to the system via terminals. Often called *server* systems.

Or

IV. (a) Compare hierarchical, network and relational model of a database with example.

hierarchical	network	relational
<p>In this model, each record in the database represented a specific part. The records had parent/child relationships, linking each part to its subpart, and so on.</p> <p>Simple structure</p> <p>Parent/child organization</p> <p>Performance:</p>	<p>Flexibility: Multiple parent/child relationships allowed a network database to represent data that did not have a simple hierarchical structure.</p> <p>• Standardization: The CODASYL standard boosted the popularity of the network model, and minicomputer vendors such as Digital Equipment Corporation and Data General implemented network databases.</p> <p>• Performance: Despite their greater complexity, network databases boasted performance approaching that of hierarchical</p>	<p>■ Data manipulation is supported by theory of sets, relational algebra and calculus</p> <p>■ Basic constructs:</p> <ul style="list-style-type: none"> ■ A relation ■ A record ■ A field (or a column) <p>■ Integrity constraints</p> <p>■ Supports all types of relationships between two records which may change over time:</p> <ul style="list-style-type: none"> ■ 1:1 ■ 1:N ■ N:M

	<p>databases. Sets were represented by pointers to physical data records, and on some systems, the database administrator could specify data clustering based on a set relationship.</p>	
--	--	--

(b) State the use of DML commands.

DML (data manipulation language)

- ❖ Select: Command description: select command is used to display the table & table values.
- ❖ Insert: Command description: insert command is used to insert the values to the table.
- ❖ Update: command is used to update the values
- ❖ Delete: command is used to delete the constraint from the table

UNIT—II

V. (a) Explain E-R diagram with an example.

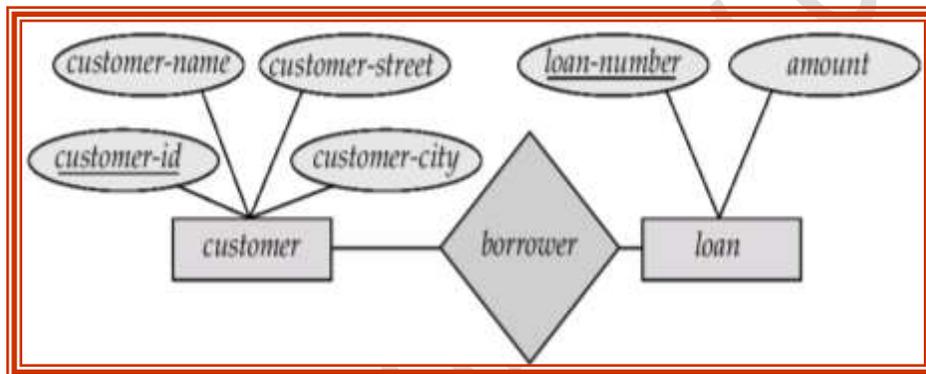
ER diagram is widely used in database design

- Represent conceptual level of a database system
- Describe things and their relationships in high level

■ Basic Concepts

- Entity set – an abstraction of similar things, e.g. cars, students

- An entity set contains many entities
- Attributes: common properties of the entities in a entity sets
- Relationship – specify the relations among entities from two or more entity sets
- A relationship may be thought as a set as well
 - For binary relationship, it enumerates the pairs of entities that relate to each other
 - For example, entity set $M = \{Mike, Jack, Tom\}$ entity set $F = \{Mary, Kate\}$. The relationship set *married* between M and F may be $\{<Mike, Mary>, <Tom, Kate>\}$



(b) Consider the relation loan (loan-number, branch-name, amount). Identify and write suitable relational operations for the following :

(i) List tuples of the relation where the branch is “XYZ”.

SQL> select * from loan where branch='XYZ';

(ii) List loan_number and amount where amount is >10000.

SQL>select loan_number,amount from loan where amount>10000;

Or

VI. (a) Explain Union, Intersection and Minus operation on sets with example.

The set operations union, intersect, and except operate on relations and correspond to the relational algebra operations \cup , \cap , $-$.

Each of the above operations automatically eliminates duplicates; to retain all duplicates use the corresponding multiset versions union all, intersect all and except all.

Suppose a tuple occurs m times in r and n times in s , then, it occurs:

- $m + n$ times in r union all s
- $\min(m,n)$ times in r intersect all s
- $\max(0, m - n)$ times in r except all s
- ❖ Find all customers who have a loan, an account, or both:

(select *customer_name* from *depositor*)

union

(select *customer_name* from *borrower*)

- ❖ Find all customers who have both a loan and an account.

(select *customer_name* from *depositor*)

intersect

(select *customer_name* from *borrower*)

- ❖ Find all customers who have an account but no loan.

(select *customer_name* from *depositor*)

except

(select *customer_name* from *borrower*)

(b). Distinguish weak entity set and strong entity set with example.

Weak Entity set	Strong Entity set
<ul style="list-style-type: none"> ■ Some entity sets in real world naturally depend on some other entity set <ul style="list-style-type: none"> ■ They can be uniquely identified only if combined with 	<ul style="list-style-type: none"> ■ Some entity sets in real world naturally independent on from other entity set <ul style="list-style-type: none"> ■ They can be uniquely identified ■ Example: <ul style="list-style-type: none"> ■ Vehicles, student.etc

<p style="text-align: center;">another entity set</p> <ul style="list-style-type: none"> ■ Example: <ul style="list-style-type: none"> ■ section1, section2, ... <p style="text-align: center;">become unique only if you put them into a context, e.g. csce4350</p>	
---	--

UNIT—III

VII. (a) Define constraints and gives its uses. Explain any 2 domain constraints.

Constraints are *conditions* that must hold on *all* valid relation instances. There are three main types of constraints:

Key constraints

Entity integrity constraints

(b) List and explain properties of a transaction.

- **Key of R:** A "minimal" superkey; that is, a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey.

Example: The CAR relation schema:

CAR(State, Reg#, SerialNo, Make, Model, Year)

has two keys Key1 = {State, Reg#}, Key2 = {SerialNo}, which are also superkeys.
{SerialNo, Make} is a superkey but *not* a key.

- If a relation has *several* candidate **keys**, one is chosen arbitrarily to be the **primary key**. The primary key attributes are *underlined*.

- **Entity Integrity:** The *primary key attributes* PK of each relation schema R in S cannot have null values in any tuple of r(R). This is because primary key values are used to *identify* the individual tuples.

$t[\text{PK}] \neq \text{null}$ for any tuple t in r(R)

- Note: Other attributes of R may be similarly constrained to disallow null values, even though they are not members of the primary key.

Or

VIII. (a) List and explain the advantages of concurrent execution of transactions.

These are mechanisms to achieve isolation

- To control the interaction among the concurrent transactions in order to prevent them from destroying the consistency of the db
- Schedules: sequences that indicate the order in which instructions of concurrent transactions are executed.
- A schedule for a set of transactions must consist of all instructions of those transactions
- Must preserve the order in which instructions appear in each individual transaction
- Executing transactions simultaneously has the following merits:
 - Increased processor and disk utilization, leading to better throughput
 - One transaction can be using the CPU while another is reading from or writing to the disk
 - Reduced average response time for transactions: short transactions need not wait behind long ones

(b) Explain referential integrity constraints with example.

- A constraint involving *two* relations (the previous constraints involve a *single* relation).

- Used to specify a *relationship* among tuples in two relations: the referencing relation and the referenced relation.
- Tuples in the *referencing relation* R_1 have attributes FK (called foreign key attributes) that reference the primary key attributes PK of the *referenced relation* R_2 . A tuple t_1 in R_1 is said to reference a tuple t_2 in R_2 if $t_1[\text{FK}] = t_2[\text{PK}]$.
- A referential integrity constraint can be displayed in a relational database schema as a directed arc from $R_1.\text{FK}$ to R_2 .

Statement of the constraint

- The value in the foreign key column (or columns) FK of the the referencing relation R_1 can be either:
 - (1) a value of an existing primary key value of the corresponding primary key PK in the referenced relation R_2 , or..
 - (2) a null.
- In case (2), the FK in R_1 should not be a part of its own primary key.

UNIT—IV

IX. (a) Define cursors. List the steps to create and use them.

Cursors enable us to examine, in the host language program, a collection of JWS computed by an Embedded SQL statement:

1) We usually need to open a cursor if the embedded statement is SELECT (i.e.) a query).

However, we can avoid opening a cursor if the answer contains a single row, as we see shortly.

2) NSERT, DELETE, and UPDATE statements typically require no cursor, although some variants of DELETE and UPDATE use a cursor.

As an example, we can find the name and age of a sailor, specified by assigning a value to the host variable `c_sid`, declared earlier, as follows:

```
EXEC SQL SELECT S.sname, S.age
        INTO   :c_sname, :c_age
        FROM   Sailors S
        WHERE  S.sid = :c_sid;
```

This query returns a collection of rows, not just one row. When executed interactively, the answers are printed on the screen

(b) Consider two tables *employee* (code, name, basic pay and department no.) and *department*(department no, department name). Write suitable commands for the following :

(i) Add a column date of joining to table employee.

SQL>alter table employee add (DOJ);

(ii) List employee details along with department name.

SQL>select name, department no. from employee;

(iii) Calculate and print their allowance which is 40% of basic pay.

SQL>select basic pay from employee where allowance=40%;

(iv) List employees in the descending order of their basic pay.

SQL>select name from employee order by basic pay DESC;

Or

X. (a) Explain the following commands with syntax :

(i) Create index

Creating Indexes

To create an index, you need to use the CREATE INDEX command. In addition, you can use the UNIQUE keyword to specify that an index contains only unique values.

Syntax:

```
CREATE INDEX index_name
ON table_name (column_name1 [ASC | DESC] [, column_name2
[ASC | DESC]].....)
```

(ii) Create unique index

```
CREATE [UNIQUE] INDEX index_name
```

ON table_name (column_name1 [ASC | DESC] [, column_name2
[ASC | DESC]].....)

(iii) Drop index.

To remove or delete an index, we need to use the DROP INDEX
command.

Syntax:

DROP INDEX <index name>

(b) Create a trigger restrict users from using the table 'student' on Sundays.

CREATE TRIGGER

NoCreateStudentTable ON DATABASE

FOR CREATE_TABLE

AS

PRINT 'You may not use the table on Sundays'

ROLLBACK TRANSACTION