Total No. of Questions – [04] Total No. of Printed Pages 02

G.R. No.	

paper Code: U359-123(T1)/U359-143(T1)

OCTOBER 2019/INSEM (T1)

T. Y. B. TECH. (COMPUTER ENGINEERING/INFORMATION TECHNOLOGY)

(SEMESTER -I)

COURSE NAME: DATABASE MANAGEMENT SYSTEM

COURSE CODE: CSUA31173/ITUA31173

SOLUTION (PATTERN 2017)

Time: [**1Hour**] [Max. Marks: **30**]

Q.1) a) **Drawbacks of using file systems to store data:** [6]

i)**Data redundancy and inconsistency**-Multiple file formats, duplication of information in different files.

ii)**Difficulty in accessing data**-Need to write a new program to carry out each new task.

iii)**Data isolation**- multiple files and formats.

iv)**Integrity problems**-Integrity constraints (e.g. account balance > 0) become part of program code. Hard to add new constraints or change existing ones.

v)**Atomicity of updates**-Failures may leave database in an inconsistent state with partial updates carried out.E.g. transfer of funds from one account to another should either complete or not happen at all.

vi)**Concurrent access by multiple users**-Concurrent accessed needed for performance. Uncontrolled concurrent accesses can lead to inconsistenciesE.g. two people reading a balance and updating it at the same time.

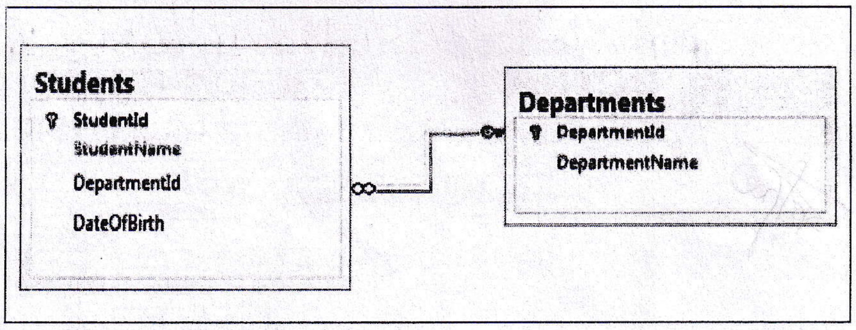
vii)**Security problems.**

b) Refer ER diagram concepts. [6]

c) Define following terms with suitable example: [4]

a) **Foreign key:**A Foreign Key is a column or a combination of columns whose values match a Primary Key in a different table. It is a key used to link two tables together. This is sometimes also called as a referencing key.The relationship between 2 tables matches the Primary Key in one of the tables with a Foreign Key in the second table.

Example:



```
CREATE TABLE [Departments] (
        [DepartmentId]  INTEGER   NOT  NULL  PRIMARY  KEY
AUTOINCREMENT,
        [DepartmentName] NVARCHAR(50)  NULL
);
CREATE TABLE [Students] (
        [StudentId] INTEGER  PRIMARY KEY AUTOINCREMENT
NOT NULL,
        [StudentName] NVARCHAR(50)  NULL,
        [DepartmentId] INTEGER  NOT NULL,
        [DateOfBirth] DATE  NULL,
        FOREIGNKEY(DepartmentId)                REFERENCES
Departments(DepartmentId)
);
```

b)Super key:A super key of an entity set is a set of one or more attributes whose values uniquely determine each entity.

Example:

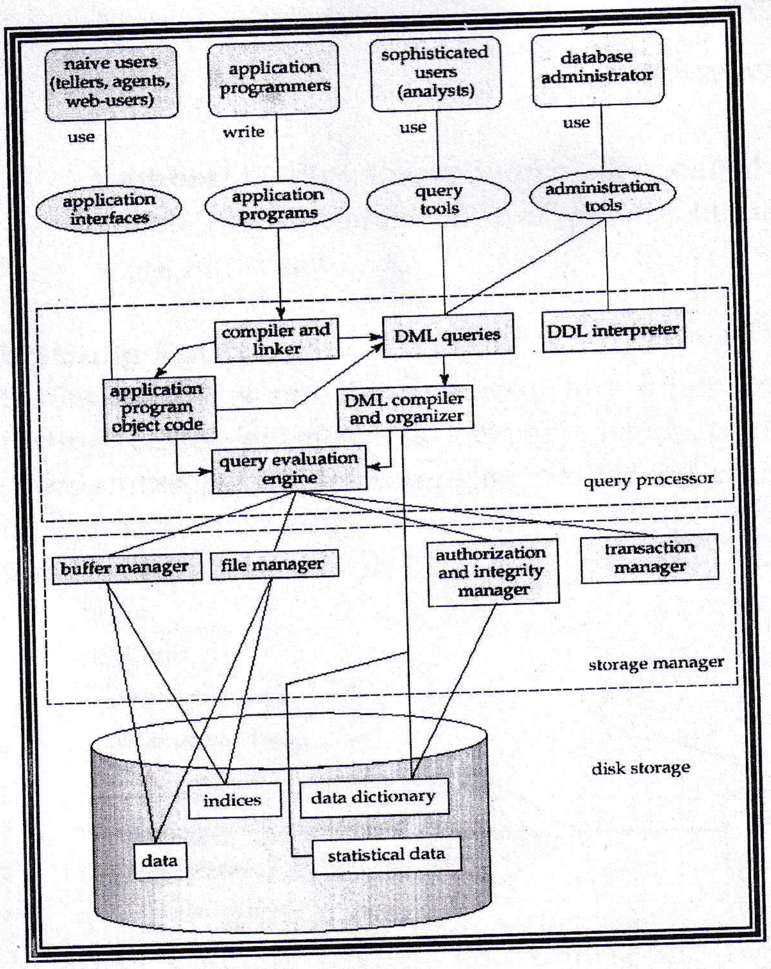
Employee(Emp_SSN,Emp_Number,Emp_Name)

The above schema has following super keys.

- {Emp_SSN}
- {Emp_Number}
- {Emp_SSN, Emp_Number}
- {Emp_SSN, Emp_Name}
- {Emp_SSN, Emp_Number, Emp_Name}
- {Emp_Number, Emp_Name}

OR

Q.2) a Overall structure of Database System. [6]



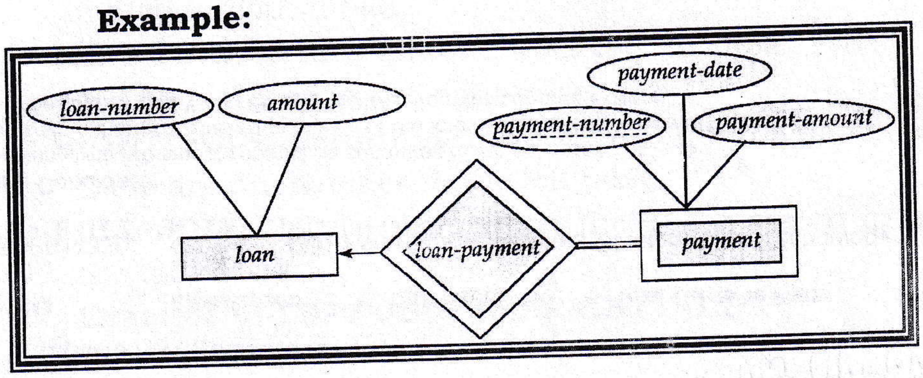
Give explanation of components like Storage Manager, Transaction Manager, Buffer Manager, Query Processor, File Manager, Database Users.

b)Refer ER diagram concepts

c) Define following terms with suitable example : [4]

a) Weak entity : An entity set that does not have a primary key is referred to as a weak entity set.We depict a weak entity set by double rectangles.

Example:

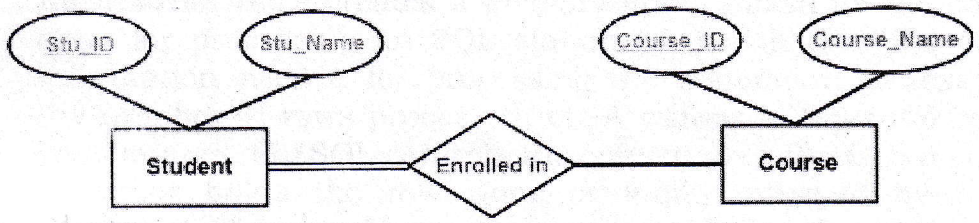


Here Payment is the weak entity set. The **payment-**

3

number is the discriminator also called as partial key which distinguishes entities among all the entities of a weak entity set.

b)Strong Entity:
Strong entity is not dependent of any other entity in schema. Strong entity always has primary key. Strong entity set is represented by single rectangle. Two strong entity's relationship is represented by single diamond.
Example:



In above example student and Course are two strong entities present with Stu_ID and Course_ID primary keys respectively.

Q.3) a) 1) Select patient_name from patient where city='Pune'. [6]
2) Select p.physician _name from physician as p, visit as v where p. reg_no=v. reg_no and v.fee<500.
3) Select physician_name, reg_no from physician where reg_no = (select reg_nofrom visit where date_of_visit='01-06-2019').

b) **Join Operations:**-Different SQL JOIN operations are mentioned below: [4]

- **JOIN/INNER join**: Return rows when there is at least one match in both tables.
- **LEFT JOIN**: Return all rows from the left table, even if there are no matches in the right table.
- **RIGHT JOIN**: Return all rows from the right table, even if there are no matches in the left table.
- **FULL JOIN**: Return rows when there is a match in one of the tables.

Example of LEFT JOIN and RIGHT JOIN.

c) **i)Stored Procedure-** [4]

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

Example:-

```
CREATE OR REPLACE PROCEDURE greetings

AS

BEGIN

    dbms_output.put_line('Hello World!');

END;
```

ii)Cursors-Oracle creates a memory area, known as the context area, for processing an SQL statement, which contains all the information needed for processing the statement; for example, the number of rows processed, etc.A **cursor** is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the **active set**. We can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors –

- **Implicit cursors-** the most recent implicit cursor as the SQL cursor, which always has attributes such as %FOUND, %ISOPEN, %NOTFOUND, and %ROWCOUNT.
- **Explicit cursors-**Working with an explicit cursor includes the following steps:
  - **Declaring the cursor for initializing the memory-**
    ```
    CURSOR c_customers IS
    SELECT id, name, address FROM customers;
    ```
  - **Opening the cursor for allocating the memory-**
    ```
    OPEN c_customers;
    ```
  - **Fetching the cursor for retrieving the data**
    ```
    FETCH c_customers INTO c_id, c_name, c_addr;
    ```
  - **Closing the cursor to release the allocated memory**
    ```
    CLOSE c_customers;
    ```

OR

Q.4) a) [6]

1) Select Name from persons where city='Mumbai'.
2) Select product_Namefrom Product where reg_no = (select reg_no from visit where date_of_visit= '01-06-2019').
3) Select Persons.LastName,Persons.FirstName,Orders.OrderNo

from Persons Full Join Orders on
Persons.P_Id=Orders.P_Id.

b) Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. The Relational operations are mentioned below: [4]

i)Select Operation (σ)

It selects tuples that satisfy the given predicate from a relation.

Notation – $\sigma_p(r)$

Where **σ** stands for selection predicate and **r** stands for relation. *p* is prepositional logic formula which may use connectors like **and, or,** and **not**. These terms may use relational operators like – =, ≠, ≥, < , >, ≤.

For example –

$\sigma_{subject = "database"}(\text{Books})$.

ii)Project Operation (∏)

It projects column(s) that satisfy a given predicate.

Notation – $\prod_{A1, A2, An}(r)$

Where A_1, A_2 , A_n are attribute names of relation **r**.

Duplicate rows are automatically eliminated, as relation is a set.

For example –

$\prod_{subject,\ author}(\text{Books})$.

Likewise remaining operations Union, Set difference, Cartesian product, Rename can also be explain.

c) **i)Functions:-**A function is a named PL/SQL Block which is similar to a procedure. The major difference between a procedure and a function is, a function must always return a value, but a procedure may or may not return a value. [4]

Example:-

```
CREATE OR REPLACE FUNCTION totalCustomers RETURN
number
IS total number(2) := 0;
BEGIN
     SELECT count(*) into total FROM customers;
RETURN total;
END;
/
```

ii)Triggers:- Triggers are stored programs that are fired automatically when some event occurs. The code to be fired can be defined as per the requirement.Oracle has also provided the facility to mention the event upon which the trigger needs to be fire and the timing of the execution.

- Triggers are, in fact, written to be executed in response to any of the following events –
- A **database manipulation (DML)** statement (DELETE, INSERT, or UPDATE)
- A **database definition (DDL)** statement (CREATE, ALTER, or DROP).
- A **database operation** (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Example:

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON customers
FOR EACH ROW WHEN (NEW.ID > 0)
DECLARE
    sal_diff number;
BEGIN
    sal_diff:=:NEW.salary-:OLD.salary;
    dbms_output.put_line('Old salary: ' || :OLD.salary);
    dbms_output.put_line('New salary: ' || :NEW.salary);
    dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
```

Total No. of Questions – [04] Total No. of Printed Pages 02

G.R. No.	

Paper Code: U3S9-123 (T1) / U3S9-143 (T1)

OCTOBER 2019/INSEM (T1)

T. Y. B. TECH. (COMPUTER ENGINEERING/INFORMATION TECHNOLOGY)

(SEMESTER -I)

COURSE NAME: DATABASE MANAGEMENT SYSTEM

COURSE CODE: CSUA31173/ITUA31173

MARKING SCHEME (PATTERN 2017)

Time: [**1Hour**] [Max. Marks: **30**]

Q.1)	a)	Each drawback carries 1 M.	[6]
	b)	Each ER feature carries 1 M.	[6]
	c)	Definition 1 M and example 1 M.	[4]
		OR	
Q.2)	a)	Each database component carries 1 M.	[6]
	b)	Each ER feature carries 1 M.	[6]
	c)	Definition 1 M and example 1 M.	[4]
Q.3)	a)	Each SQL Query carries 2 M.	[6]
	b)	Each Join operation with example carries 2 M.	[4]
	c)	Each concept carries 2 M.	[4]
		OR	
Q.4)	a)	Each SQL Query carries 2 M.	[6]
	b)	Each Relational Algebra operation with example carries 2 M.	[4]
	c)	Each concept carries 2 M.	[4]