

Punjab Technical University

B.Tech. – Computer Science & Engineering (Sem. – 6th)

RDBMS

Subject Code: CS-302

2 Mark Questions:-

Q:-1) What is Data Manipulation Language? (May 2012)

Ans:-1) **Data Manipulation Language** (DML) statements are used for managing data within schema objects. Some examples:

- SELECT - Retrieve data from the database
- INSERT - Insert data into a table
- UPDATE - Updates existing data within a table
- DELETE - Deletes all records from a table, the space for the records remain

Q:-2) What are Triggers? (May 2012)

Ans:-2) In a DBMS, a **Trigger** is a SQL procedure that initiates an action (i.e., *fires* an action) when an event (INSERT, DELETE or UPDATE) occurs. Since

triggers are event-driven specialized procedures, they are stored in and managed by the DBMS.

A trigger cannot be called or executed; the DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion.

Q:-3) What do you mean by Query by Example? (May 2012)

Ans:-3) Query by Example (QBE) is a method of query creation that allows the user to search for documents based on an example in the form of a selected text string or in the form of a document name or a list of documents.

Q:-4) What is Referential Integrity? (May 2012)

Ans:-4) A feature provided by **Relational Database Management Systems (RDBMS's)** that prevents users or applications from entering inconsistent data. Most RDBMS's have various referential integrity rules that you can apply when you create a relationship between two tables.

For example, suppose Table B has a *foreign key* that points to a field in Table A. Referential integrity would prevent you from adding a record to Table B that cannot be linked to Table A.

Q:-5) What is Foreign Key Constraints? (May 2012)

Ans:-5) A **Foreign Key** is a column or combination of columns that is used to establish and enforce a link between the data in two tables. You can create a foreign key by defining a FOREIGN KEY constraint when you create or modify a table.

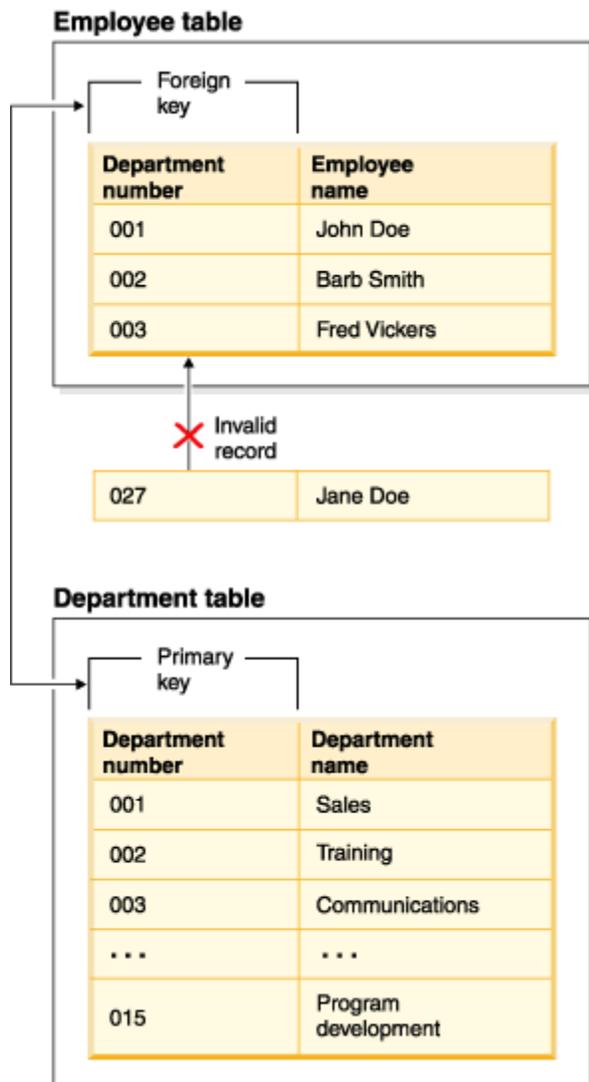
For example, a typical foreign key constraint might state that every employee in the EMPLOYEE table must be a member of an existing department, as defined in the DEPARTMENT table.

Referential integrity is the state of a database in which all values of all foreign keys are valid. A foreign key is a column or a set of columns in a table whose values are required to match at least one primary key or unique key value of a row in its parent table. A referential constraint is the rule that the values of the foreign key are valid only if one of the following conditions is true:

- They appear as values of a parent key.
- Some component of the foreign key is null.

To establish this relationship, you would define the department number in the EMPLOYEE table as the foreign key, and the department number in the DEPARTMENT table as the primary key.

Figure 1 shows how a record with an invalid key is prevented from being added to a table when a foreign key constraint exists between two tables:



Q:-6) List DDL Commands? (May 2012)

Ans:-e) Data Definition Language (DDL) statements are used to define the database structure or schema. Some examples:

- CREATE - To Create objects in the database
- ALTER - Alters the structure of the database
- DROP - Delete objects from the database

- TRUNCATE - Remove all records from a table, including all spaces allocated for the records are removed
- RENAME - Rename an object

Q:-7) What are the advantages of PL/SQL? (May 2012)

Ans:-7) Advantages of PL/SQL:-

- Support SQL data manipulation.
- Provide facilities like conditional checking, branching and looping.
- Provide fast code execution since it sends SQL statement as a block to the oracle engine.

Q:-8) What are Cursors? (May 2012)

Ans:-8) Cursors in SQL procedures

In SQL procedures, a cursor make it possible to define a result set (a set of data rows) and performs complex logic on a row by row basis. By using the same mechanics, an SQL procedure can also define a result set and return it directly to the caller of the SQL procedure or to a client application.

A cursor can be viewed as a pointer to one row in a set of rows. The cursor can only reference one row at a time, but can move to other rows of the result set as needed.

To use cursors in SQL procedures, you need to do the following:

1. Declare a cursor that defines a result set.
2. Open the cursor to establish the result set.
3. Fetch the data into local variables as needed from the cursor, one row at a time.
4. Close the cursor when done

To work with cursors you must use the following SQL statements:

- DECLARE CURSOR
- OPEN
- FETCH
- CLOSE

Q:-9) Define the Term Functional Dependency? (May 2012)

Ans:-9) A Functional Dependency occurs when one attribute in a relation uniquely determines another attribute. This can be written $A \rightarrow B$ which would be the same as stating "B is functionally dependent upon A."

Q:-10) Define Parameterized Cursor? (Dec. 2012)

Ans:-10) Parameterized cursors (PL/SQL)

Parameterized cursors are static cursors that can accept passed-in parameter values when they are opened.

The following **example** includes a parameterized cursor. The cursor displays the name and salary of each employee in the EMP table whose salary is less than that specified by a passed-in parameter value.

```
DECLARE
    my_record          emp%ROWTYPE;
    CURSOR c1 (max_wage NUMBER) IS
        SELECT * FROM emp WHERE sal < max_wage;
BEGIN
    OPEN c1(2000);
    LOOP
        FETCH c1 INTO my_record;
        EXIT WHEN c1%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Name = ' || my_record.ename || ',
salary = '
        || my_record.sal);
    END LOOP;
    CLOSE c1;
END;
```

If 2000 is passed in as the value of *max_wage*, only the name and salary data for those employees whose salary is less than 2000 is returned:

```
Name = SMITH, salary = 800.00
Name = ALLEN, salary = 1600.00
```

Q:-11) What are Stored Procedures? (Dec. 2012)

Ans:-11) A **Stored Procedure** is nothing more than prepared SQL code that you save so you can reuse the code over and over again. So if you think about a query that you write over and over again, instead of having to write that query each time you would save it as a stored procedure and then just call the stored procedure to execute the SQL code that you saved as part of the stored procedure.

In addition to running the same SQL code over and over again you also have the ability to pass parameters to the stored procedure, so depending on what the need is the stored procedure can act accordingly based on the parameter values that were passed.

Q:-12) What is Embedded SQL? (Dec. 2012)

Ans:-12) **Embedded SQL** is a method of inserting inline SQL statements or queries into the code of a programming language, which is known as a host language. Because the host language cannot parse SQL, the inserted SQL is parsed by an embedded SQL preprocessor.

Embedded SQL is a robust and convenient method of combining the computing power of a programming language with SQL's specialized data management and manipulation capabilities.

Q:-13) What is difference between Procedural and Non-Procedural Language? (Dec. 2012)

Ans:-13) Procedural Language

1. **Direction:** - Procedural language directs the computer what to and how to do.
2. **Learning:** - It is difficult to learn.
3. **Debugging:**-It is difficult to debug.
4. **Requirement:**-It requires large number of procedural instruction.
5. **Uses:**-It is normally used by professional programmers.
6. **Orientation:**-It is typically file-oriented.
7. **Capabilities:**-Procedural language provides many programming capabilities.

Non-Procedural Language

1. **Direction:** - Non-Procedural language directs the computer what to do and not how to do.
2. **Learning:**-It is easy to learn.
3. **Debugging:** - It is easy to debug.
4. **Requirement:**-It requires a few non-procedural instructions.
5. **Uses:**-It can be used professional and non-technical users.
6. **Orientation:**-It is typically database-oriented.
7. **Capabilities:**-Non-procedural language provides less programming capabilities.

Q:-14) What is Nested Query? (Dec. 2012)

Ans:-14) A SQL nested query is a **SELECT** query that is nested inside a **SELECT**, **UPDATE**, **INSERT**, or **DELETE** SQL query. Here is a simple example of SQL nested query:

```
SELECT Model FROM Product
WHERE ManufacturerID IN (SELECT ManufacturerID FROM Manufacturer
WHERE Manufacturer = 'Dell')
```

The nested query above will select all models from the Product table manufactured by Dell:

```
Model
Inspiron B120
Inspiron B130
Inspiron E1705
```

You can have more than one level of nesting in one single query.

Q:-15) How Views are created in SQL? (Dec. 2012)

Ans:-15) The **SQL view** is, in essence, a virtual table. It does not physically exist. Rather, it is created by a query joining one or more tables.

The syntax for creating a VIEW in SQL is:

```
CREATE VIEW view_name AS
  SELECT columns
  FROM table
  WHERE predicates;
```

Q:-16) What is Normalization? (Dec. 2011)

Ans:-16) Normalization is the process of organizing data into a related table; it also eliminates redundancy and increases the integrity which improves performance of the query. To normalize a database, we divide the database into tables and establish relationships between the tables.

Normalization Avoids:

- **Duplication of Data** - The same data is listed in multiple lines of the database
- **Insert Anomaly** - A record about an entity cannot be inserted into the table without first inserting information about another entity - Cannot enter a customer without a sales order
- **Delete Anomaly** - A record cannot be deleted without deleting a record about a related entity. Cannot delete a sales order without deleting all of the customer's information.
- **Update Anomaly** - Cannot update information without changing information in many places. To update customer information, it must be updated for each sales order the customer has placed

Q:-17) List two advantages of SQL? (Dec. 2011)

Ans:-17) Advantages of SQL:

*** High Speed:**

SQL Queries can be used to retrieve large amounts of records from a database quickly and efficiently.

*** Well Defined Standards Exist:**

SQL databases use long-established standard, which is being adopted by ANSI & ISO. Non-SQL databases do not adhere to any clear standard.

*** No Coding Required:**

Using standard SQL it is easier to manage database systems without having to write substantial amount of code.

*** Emergence of ORDBMS:**

Previously SQL databases were synonymous with relational database. With the emergence of Object Oriented DBMS, object storage capabilities are extended to relational databases.

Q:-18) What is Data Independence? (May 2010)

Ans:-18) Data independence is a form of database management that keeps data separated from all programs that make use of the data. As a cornerstone for the idea of a DBMS or database management system, data independence ensures that the data cannot be redefined or reorganized by any of the programs that make use of the data. In this manner, the data remains accessible, but is also stable and cannot be corrupted by the applications using it.

Q:-19) What is a View? (May 2010)

Ans:-19) View: - A database *view* displays one or more database records on the same page. A view can display some or all of the database fields. Views have *filters* to determine which records they show. Views can be *sorted* to control the record order and *grouped* to display records in related sets.

Q:-20) What is Relational Calculus? (May 2010)

Ans:-20)

- Relational calculus is Non-procedural
- It has the same expressive power as relational algebra, i.e. it is relationally complete
- It is a formal language based upon a branch of mathematical logic called "predicate calculus"

- There are two approaches: Tuple relational calculus and Domain relational calculus

Q:-21) What is 1NF (Normal Form)? (May 2010)

Ans:-21) First normal form (1NF) sets the very basic rules for an organized database:

- Define the data items required, because they become the columns in a table. Place related data items in a table.
- Ensure that there are no repeating groups of data.
- Ensure that there is a primary key.

Q:-22) What is BCNF (Boyce-Codd Normal Form)? (May 2010)

Ans:-22) Boyce - Codd Normal Form (BCNF) :- A normal form is said to be a Boyce - Codd Normal Form if it is in 3NF and there is not a possibility of a key attribute is determined by a Non Key attribute.

Q:-23) What is Static SQL? How does it differ from Dynamic SQL? (May 2010)

Ans:-23) Static SQL is SQL statements in an application that do not change at runtime and, therefore, can be hard-coded into the application. Static SQL

is usually targeted at a specific database and in many cases gets stored in stored procedures.

Dynamic SQL is SQL statements that are constructed at runtime. Dynamic SQL generation is also needed in applications where the databases structure itself is dynamic. Many Enterprise Applications allow users to customize the way data is stored and displayed.

for example, the application may allow users to enter their own queries. Thus, the SQL statements cannot be hard-coded into the application.

Static SQL provides performance advantages over **dynamic SQL** because static SQL is preprocessed, which means the statements are parsed, validated, and optimized only once.

Q:-24) Difference between Implicit and Explicit Cursor? (May 2010)

Ans:-24) Implicit Cursor are declared and used by the oracle internally. Whereas the **Explicit Cursors** are declared and used by the user. More over Implicit cursors are no need to declare oracle creates and process and closes automatically. The Explicit cursor should be declared and closed by the user.

Q:-25) What is Database Security? (May 2010)

Ans:-25) Database security is the form of a security which deals with the information security so that the database and the other objects contained in

it can be protected. The data in the database is found to be of confidential in nature and along with that the data is also of reliable and integrated too. These are found to be the main features of the database security as it all ensures that the data is consistent to be used. The database may include the information about the storage of the database applications along with the data in it.

Q:-26) Write the Functions of a DBA? (Dec. 2010)

Ans:-26) Functions of a Database Administrator

One of the main reasons for using DBMS is to have a central control of both data and the programs accessing those data. A person who has such control over the system is called a Database Administrator (DBA). The following are the functions of a Database administrator.

- [Schema Definition](#)
- [Storage structure and access method definition](#)
- [Schema and physical organization modification.](#)
- [Granting authorization for data access.](#)
- [Routine Maintenance](#)

Schema Definition

The Database Administrator creates the database schema by executing DDL statements. Schema includes the logical structure of database table (Relation) like data types of attributes, length of attributes, integrity constraints etc.

Storage structure and access method definition

Database tables or indexes are stored in the following ways: Flat files, Heaps, B+ Tree etc..

Schema and physical organization modification

The DBA carries out changes to the existing schema and physical organization.

Granting authorization for data access

The DBA provides different access rights to the users according to their level. Ordinary users might have highly restricted access to data, while you go up in the hierarchy to the administrator, you will get more access rights.

Routine Maintenance

Some of the routine maintenance activities of a DBA is given below.

Taking backup of database periodically

ensuring enough disk space is available all the time.

Monitoring jobs running on the database.

Ensure that performance is not degraded by some expensive task submitted by some users.

Performance Tuning