Sql Server

Q. What is NewId()?

A. The following example uses NEWID() to assign a value to a variable declared as the uniqueidentifier data type. The value of the uniqueidentifier data type variable is printed before the value is tested.

```
-- Creating a local variable with DECLARE/SET syntax.
DECLARE @myiduniqueidentifier
SET @myid = NEWID()
PRINT 'Value of @myid is: '+ CONVERT(varchar(255), @myid)
```

Q. What is Scope_Identity()?

A. It returns the last IDENTITY value produced on a connection and by a statement in the same scope, regardless of the table that produced the value.

SCOPE_IDENTITY(), like @@IDENTITY, will return the last identity value created in the current session, but it will also limit it to your current scope as well. In other words, it will return the last identity value that you explicitly created, rather than any identity that was created by a trigger or a user defined function.

Q. What is difference between storeprocedures and functions?

A. Stored Procedure

- have to use EXEC or EXECUTE
- return output parameter
- can create table but won't return Table Variables
- you cannot join SP
- can be used to change server configuration
- can be used with XML FOR Clause
- can have transaction within SP

Functions

- can be used with Select statement
- Not returning output parameter but returns Table variables
- You can join UDF
- Cannot be used to change server configuration
- Cannot be used with XML FOR clause
- Cannot have transaction within function

Q. Can we call store procedure with in a function?

A. No, We cannot call store procedure with in a function.

Q. Can functions return table?

A. Yes a function can return table.

```
CREATE FUNCTION dbo.fnEmployeeList ()
RETURNS TABLE AS
RETURN (SELECT id, name, city FROM Employee)  GO
```

Q. How to create temporary table in SQL?

A. CREATE TABLE #MyTempTable (cola INT PRIMARY KEY) INSERT INTO #MyTempTable VALUES (1)
SELECT * FROM #MyTempTable

Q. What are various types of joins?

A. INNER JOIN: An inner join (sometimes called a simple join) is a join of two or more tables that returns only those rows that satisfy the join condition.

1) selectbookstudent.BookId,Book.BookName,student.sname from Book **inner join**BookStudent on Book.BookId=BookStudent.BookId**inner join** student on student.sid=BookStudent.studentid

SELF JOIN: When we join a table to itself it is called Self Join.

Select p1.iPageId, p1.sPageName as Parent, p2.sPageName as Sub from PageMgmt p1 join PageMgmt p2 on p1.iPageId=p2.iParentid

OUTER JOIN: An outer join extends the result of a simple join. An outer join returns all rows that satisfy the join condition and also returns non matching rows based on Outer join type.

- **Left Outer Join:** It brings all the records from the table on left hand side and matching records from table on right hand side and return null where no match found.

Select * from table1 **left outer join** table2 on table1.Id=table2.Id

- **Right Outer Join:** It brings all the records from the table on right hand side and matching records from table on left hand side and return null where no match found.

Select * from table1 **right outer join** table2 on table1.Id=table2.Id

- **Full Outer Join:** It brings all the records from both the table and return null where no match found.

Select * from table1 **full outer join** table2 on table1.Id=table2.Id

- **Cross Join:** A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table.

Q. What is a subquery? What are co-related sub queries?

A. SubQueries -A subquery is usually added in the WHERE Clause of the SQL statement. Most of the time, a subquery is used when you know how to search for a value using a SELECT statement, but do not know the exact value.

List of Customers that have placed atleast one Order:

Select iCustomerId,sCustomerName from tbCustomers cu where iCustomerId in(Select iCustomerId from tbOrder)

CO-RELATED SUB QUERY

First Outer Query is compiled, then inner according to output of outer and then again the Outer according to the Output Of inner.

RECORD OF CUSTOMER FROM ORDER TABLE WHEN THE CUSTOMER PLACED THE FIRST ORDER

```
Select o1.iCustomerId,o1.iOrderId,o1.dOrderDate from tbOrder o1
where o1.dOrderDate=(select min(o2.dOrderDate) from tbOrder o2
where o2.iCustomerID=o1.iCustomerId)
Order By iCustomerID
```

Q. What are views?

A. The view is a virtual table, which can have the multiple columns from the one or more table. It can be used like the normal table. Normally view cannot store the data permanently in the table. When we create the view it stores the view definition schema as object under the concern database.

Q. If we insert or update view, will data be inserted into table also?

A. Yes

Q. What are triggers?

A. Triggers are the storedprocedures which fires automatically whenever an Insert, Update or Delete command fires.

DeleteTrigger

```
1)  Create trigger DeleteTrigger on Employee3 for delete
as
if((select EmpId from deleted)>5)
begin
print 'U Cannot delete Id greater than 5'
rollback transaction
end
```

2) Alter trigger DeleteTrigger on Employee3 for delete
as
begin
Insert into employee2 select * From deleted
end

CHECK USING TRUNCATE AND DELETE STATEMENT------TRUNCATED STATEMENT WILL NOT BE ROLLED BACK……..BUT DELETEd RECORDS CAN BE ROLLEDBACK

truncate table employee3
delete from employee3

UPDATE TRIGGER

TRIGGER FOR NOT UPDATING ANY RECORD IN TABLE

CREATE trigger UpdateTrigger on Employee2
for UPDATE as begin
if(select empId from inserted)>0
begin
print 'u Cant Update Id'
rollback transaction
end
end

if u hav 2 values then old value wll be availabe in deleted table and new value will be avaliabel in inserted table

4) CREATE trigger UpdateTrigger on Employee2 for UPDATE
as
begin
if(select empId from inserted)=17
begin
print 'u Cant Update Id'
rollback transaction
end
end

INSERT TRIGGER

```
CREATE trigger InsertTrigger
on Employee2
for Insert
as
begin
if(select empId from inserted)<1
or(select empId from inserted)>100
begin
print 'u Cant INsert.  Id must be between 1 to 100'
rollback transaction
end
end
```

Q. What are cursors?

A. Cursor is a database object used by applications to manipulate data in a set on a row-by-row basis, instead of the typical SQL commands that operate on all the rows in the set at one time.

In order to work with a cursor we need to perform some steps in the following order: Declare cursor, Open cursor, Fetch row from the cursor, Process fetched row, Close cursor, Deallocate cursor.

Q. What are indexes?

A. An index makes it easier for us to retrieval and presentation of the data. An Index is a system which provides faster access to rows and for enforcing constraints. If we don't create any indexes then the SQL engine searches every row in table (also called as table scan). As the table data grows to thousand, millions of rows and further then searching without indexing becomes much slower and becomes expensive.

Q. Difference between clustered and non clustered indexes?

A. **Clustered index** is a special type of index that reorders the way records in the table are physically stored. Therefore table can have only one clustered index. The leaf nodes of a clustered index contain the data pages.

Non Clustered Index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf node of a Non Clustered Index does not consist of the data pages. Instead, the leaf nodes contain index rows.

Q. Can store procedures return values?

A. Store procedure may or may not return value.

Q. What are various ranking methods?

A. Ranking Methods are:

1) Rank
2) Ntile
3) Dense Rank
4) Row Number

Q. How can we deal with Null values in SQL?

A. Null is a special marker used in Structured Query Language (SQL) to indicate that a data value does not exist in the database

DEALING WITH NULL

Keep age null in employee age column(not all)

1) select IsNULL(cast(Age as varchar),'Not Given') from employee

Q. How to create temporary tables in SQL?

A.

```
declare @TempPageMgmt table
(
PageNamevarchar(50),
ParentIdint
)
insert into @ TempPageMgmt
selectPageName,ParentId from pagemanagement where PageId between 1 and 5
select * from @ TempPageMgmt
```

Q. How to use select case statement in SQL?

A. It is used when we have multiple options available.

```
selectsName,iAge,Rank=
case
wheniAge>1 and iAge<33  then 'Third'
wheniAge>33 and iAge<60  then 'second'
wheniAge>60 and iAge<100  then 'First'
else 'SomethingElse'
end
fromtbStudent
```

Q. What are various aggregate functions available in SQL?

A. Aggregate Function

1) Sum
2) Avg
3) Min
4) Max

5) Count

Q. What are the various operators in SQL?

A. Operator

- =,<,>,<=,>=,<>,!=,!>,!<
- AND,OR ,NOT
- BETWEEN
- Like
- In
- Is
- All,Any,Some
- Exists

Q. What is the difference between All,Any and some?

A. ALL

ALL compares a single value against a set of data from a query. Each value from the query's results is combined with the scalar value to generate a single scalar expression. If all of the scalar expressions evaluate to true then the result of the ALL expression will be true. Otherwise the result will be false.

```
SELECT  iStudentId, sName, iAge FROM tbStudent
WHERE NOT iAge>= ALL (SELECT iRollNo FROM tbStudent)
```

SOME and ANY

The SOME and ANY operators provide equivalent functionality. The SOME and ANY operators return true if at least one of the generated expressions evaluates as true.

```
SELECT iStudentId, sName, iAge,iRollNo FROM tbStudent WHERE iAge < ANY
(SELECT iRollNo FROM tbStudent)
```

As the ANY and SOME operators are equivalent, you can interchange the two keywords without affecting the results.

Q. Where we can use "Is" Operator?

A. It is used to deal with NULL values.

```
Select * from tbEmployee where Salary is NULL
```

Q. What are Normalization and various forms of Normalization?

A. In the design of a relational database management system (RDBMS), the process of organizing data to minimize redundancy is called normalization. The goal of database normalization is to decompose relations with anomalies in order to produce smaller, well-structured relations. Normalization usually involves dividing large tables into smaller

(and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

Types of Normalization:

- 1 NF
- 2 NF
- 3 NF
- BCNF - Boyce Code Normal Form
- 5 NF

Q. What do you mean by Group By clause?

A. The Group by clause can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns**.** Remember when we use Group by every column in the select list has to either part of the group by or it must be an aggregate.

1) selectorderid,sum(quantity) total
from [order details] group by orderid

2) selectiOrderId,Avg(iQuantity) from tborderdetail where iorderId between 10248 and 10250 group by iorderId

Having Clause: It is used in a SELECT statement to filter the records that a GROUP BY returns.

Select iorderid,sum(iquantity) total
fromtborderdetail group by iorderid
having sum(iquantity)>300

Q. What is the difference between "Having" and "Where"?

A. The difference is that **WHERE** operates on individual rows, while **HAVING** operates on groups.

HAVING specifies a search condition for a group or an aggregate function used in SELECT statement.

Q. What is RollUp and Cube?

A. **ROLLUP** enables a SELECT statement to calculate multiple levels of subtotals across a specified group of dimensions. It also calculates a grand total. ROLLUP is a simple extension to the GROUP BY clause, so its syntax is extremely easy to use. The ROLLUP extension is highly efficient, adding minimal overhead to a query.

CUBE takes a specified set of grouping columns and creates subtotals for all of their possible combinations. In terms of multidimensional analysis, CUBE generates all the subtotals that could be calculated for a data cube with the specified dimensions. If you

have specified CUBE(time, region, department) the result set will include all the values that would be included in an equivalent ROLLUP statement plus additional combinations.

Q. Calculate three highest and three lowest salary using subQuery?

A. Maximum 3 Salaries

select * from Employee where Salary>=(select max(Salary) from Employee where salary <(select max(salary) from employee where salary <(select max(salary) from employee)))

Minimum 3 Salaries

select * from Employee where Salary<=(select min(Salary) from Employee where salary >(select min(salary) from employee where salary >(select min(salary) from employee)))

Q. What are Rules and Defaults?

A. Rules:Used to specify rules to be applied on particular columns.

a) create Rule Salary as @Salary>5000
b) sp_bindrule Salary,'Customers.Salary'
c) sp_unbindrule 'Customers.Salary'
d) drop rule Salary

Defaults: Used to assign default values to a column.

a) create default Age as 20
b) sp_bindefault Age,'Customers.Age'
c) sp_unbindefault 'Customers.Age'
d) drop default Age

Q. What are various constraints in SQL server?

A. NOT NULL**,** UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

Q. What are various keys available in SQL server?

A. A key allows us to identify a set of attributes andthus distinguishes entities from each other.

Different types of keys:

1) Super Key
2) Candidate Key
3) Primary Key
4) Unique Key
5) Foreign Key

Q. Can we have more than one primary key on single table?

A. Yes

Q. What are composite Keys?

A. If you have more than one primary key in a table, they are jointly called Composite Key**.**

Q. Difference between Set and Select?

A.

- Set is a ANSI standard for variable assignment
- Select is a Non-ANSI standard when assigning variables
- Set can assign only one variable at a time
- Select can assign multiple variable at a time
- When assigning from a query that returns more than one value, SET will fail with an error
- When assigning from a query that returns more than one value, SELECT will assign the last value returned by the query and hide the fact that the query returned

Q. What is the difference between Varchar and NVarchar?

A. An nvarchar column can store any Unicode data. A varchar column is restricted to an 8-bit codepage.

Q. What are Sparse columns in SQL 2008?

A. Sparse column is a tool that helps to reduce amount of physical storage used in a database.

Q. What is Order BY Clause?

A. It **s**pecifies the sort order used on columns returned in a SELECT statement.

Q. What is difference between Caste and Convert Method?

A. Cast and Convert perform datatype conversion.Convert also does some date formatting conversions that cast doesn't offer.

Q. What are Store Procedures and its Uses?

A. Stored procedures are compiled objects. You can execute multiple statements in stored procedures.

Q. What are cursors?

A. Cursors help us to do an operation on a set of data one row at a time. For example: If we have duplicate records in a table we can remove it by declaring a cursor which would

check the records during retrieval one by one and remove rows which have duplicate values.

Q. When do we use the UPDATE_STATISTICS command?

A. This command is basically used when we do a large processing of data. If we do a large amount of deletions any modification or Bulk Copy into the tables, we need to basically update the indexes to take these changes into account. UPDATE_STATISTICS updates the indexes on these tables accordingly.

Q. Which TCP/IP port does SQL Server run on?

A. SQL Server runs on port 1433 but we can also change it for better security.

Q. From where can you change the default port?

A. From the Network Utility TCP/IP properties –> Port number, both on client and the server.

Q. Can you tell me the difference between DELETE & TRUNCATE commands?

A. Deleted data can be rolled back, but data one truncated cannot be rolled back.

Q. Can we use truncate command on a table which is referenced by FOREIGN KEY?

A. No. We cannot use truncate command on a table with Foreign Key because of referential integrity.

Q. What is the use of DBCC commands?

A. DBCC stands for database consistency checker. We use these commands to check the consistency of the databases, i.e., maintenance, validation task and status checks.

Q. What command do we use to rename a db?

A. sp_renamedb ‘oldname’, ‘newname’

Q. What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?

A. Having Clause is basically used only with the GROUP BY function in a query and WHERE clause is applied to each row before they are part of the GROUP BY function in a query.

Q. What do you mean by COLLATION?

A. Collation is basically the sort order. There are three types of sort order Dictionary case sensitive, Dictionary - case insensitive and binary.

Q. What is a Linked Server?

A. Linked Servers is a concept in SQL Server by which we can add other SQL Server to a Group and query both the SQL Server database using T-SQL Statements.

Q. Which stored procedure will be running to add a linked server?

A. sp_addlinkedserver, sp_addlinkedsrvlogin

Q. What are the authentication modes in SQL Server?

A. Windows mode and mixed mode (SQL & Windows).

Q. Where do you think the user's names and passwords will be stored in SQL server?

A. They get stored in master db in the sysxlogins table.

Q. What is log shipping?

A. In logshipping the transactional log file from one server is automatically updated into the backup database on the other server. If one server fails, the other server will have the same db and we can use this as the DR (disaster recovery) plan.

Q. What is BCP? When do we use it?

A. BulkCopy is a tool used to copy huge amount of data from tables and views. But it won't copy the structures of the same.

Q. What should we do to copy the tables, schema and views from one SQL Server to another?

A. We have to write some DTS packages for it**.**

Q. What is referential integrity? What are the advantages of it?

A. Referential integrity is a database constraint that ensures that references between data are indeed valid and intact.

Advantages of Referential integrity are:

- **Referential integrity** is usually enforced by the combination of a primary key and a foreign key. For referential integrity to hold, any field in a table that is declared a foreign key can contain only values from a parent table's primary key field.
- **Referential integrity** is a feature provided by relational database management systems that prevents users or applications from entering inconsistent data.
- **Referential integrity** is a database management safeguard that ensures every foreign key match a primary key.

Q. What is the difference between a local and a global variable?

A. Difference between a local and a global variable

- A **local temporary table** exists only for the duration of a connection or, if defined inside a compound statement, for the duration of the compound statement.
- A **global temporary table** remains in the database permanently, but the rows exist only within a given connection. When connections are closed, the data in the global temporary table disappears. However, the table definition remains with the database for access when database is opened next time.

Q. Can a stored procedure call another stored procedure? If yes at what level it will be controlled?

A. Yes. When one stored procedure calls another stored procedure (SP) you have what is called stored procedure nesting. We can have recursive stored procedures upto 32 nest levels.

Q. Can a stored procedure call itself(recursive)?

A. Yes

Q. Explain DBMS, RDBMS?

A. DBMS Stands for "Database Management System." In short, a DBMS is a database program. Technically speaking, it is a software system that uses a standard method of cataloging, retrieving, and running queries on data. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs.

Relational Database Management System (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd. A short definition of an RDBMS is: a DBMS in which data is stored in tables and the relationships among the data are also stored in tables.

Q. What are primary keys and foreign keys?

A. **Primary Key**: The primary key of a relational table uniquely identifies each record in the table

Foreign Key:A foreign key is a field (or fields) that points to the primary key of another table

Q. How would you update the rows which are divisible by 10, given a set of numbers in column?

A. UPDATE table_name
SET column1=value WHERE column1= value/10

Q. How many to many relationships are implemented?

A. We have to create a third table that contains foreign keys of primary tables.

Q. How can you get @@error and @@rowcount at the same time?

A. If @@Rowcount is checked after Error checking statement then it will have 0 as the value of @@Recordcount as it would have been reset.
And if @@Recordcount is checked before the error-checking statement then @@Error would get reset. To get @@error and @@rowcount at the same time do both in same statement and store them in local variable. SELECT @RC = @@ROWCOUNT, @ER = @@ERROR

Q. What is SQL Injection?

A. SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.

Q. What is Cascading?

A. Cascading referential integrity constraints are foreign key constraints that tell SQL Server to perform certain actions when a primary key field in a primary key-foreign key relationship is updated or deleted.There are two types of Cascades
1) Delete Cascade
2) Update Cascade

Q. What are different types of Collation Sensitivity?

A.

- **Case** sensitivity A and a, B and b, etc.
- **Accent** sensitivity a and á, o and ó, etc.
- **Kana** Sensitivity When Japanese kana characters Hiragana and Katakana are treated differently, it is called Kana sensitive.
- **Width** sensitivity when a single-byte character (half-width) and the same character when represented as a double-byte character (full-width) are treated differently than it is width sensitive.

Q. What's the difference between a primary key and a unique key?

A. Both primary key and unique key enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column; where as unique key creates a nonclustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

Q. How to implement one-to-one, one-to-many and many-to-many relationships while designing tables?

A.

- **One-to-One relationship** can be implemented as a single table and rarely as two tables with primary and foreign key relationships.
- **One-to-Many relationships** are implemented by splitting the data into two tables with primary key and foreign key relationships.
- **Many-to-Many relationships** are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table.

Q. What is a NOLOCK?

A. Using the **NOLOCK** query optimizer hint is generally considered good practice in order to improve concurrency on a busy system. When the NOLOCK hint is included in a SELECT statement, no locks are taken when data is read. The result is a Dirty Read, which means that another process could be updating the data at the exact time you are reading it. There are no guarantees that your query will retrieve the most recent data. The advantage to performance is that your reading of data will not block updates from taking place, and updates will not block your reading of data.

Q. Difference between Function and Stored Procedure?

A.

- UDF can be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section where as Stored procedures cannot be.
- UDFs that return tables can be treated as another rowset. This can be used in JOINs with other tables.
- Inline UDF's can be thought of as views that take parameters and can be used in JOINs and other Rowset operations.

Q. What kind of User-Defined Functions can be created?

A. There are three types of User-Defined functions in SQL Server 2000 and they are Scalar, Inline Table-Valued and Multi-statement Table-valued.

Q. What is the difference between a local and a global variable?

A.

- A **local temporary** table exists only for the duration of a connection or, if defined inside a compound statement, for the duration of the compound statement.
- A **global temporary** table remains in the database permanently, but the rows exist only within a given connection. When connection is closed, the data in the

global temporary table disappears. However, the table definition remains with the database for access when database is opened next time.

Q. What are three SQL keywords used to change or set someone's permissions?

A. GRANT, DENY, and REVOKE.

Q. What is the STUFF function and how does it differ from the REPLACE function?

A.

- **STUFF** function to overwrite existing characters. Using this syntax, STUFF(string_expression, start, length, replacement_characters), string_expression is the string that will have characters substituted, start is the starting position, length is the number of characters in the string that are substituted, and replacement_characters are the new characters interjected into the string.
- **REPLACE** function to replace existing characters of all occurrences. Using this syntax REPLACE(string_expression, search_string, replacement_string), where every incidence of search_string found in the string_expression will be replaced with replacement_string.

Q. Using query analyzer, name 3 ways to get an accurate count of the number of records in a table?

A. → select sum(1) from tbRanking
→ select count(*) from tbRanking
→ update tbRanking set Id = Id; select @@rowcount

Q. What is data integrity? Explain constraints?

A. Data integrity is an important feature in SQL Server. When used properly, it ensures that data is accurate, correct, and valid. It also acts as a trap for otherwise undetectable bugs within applications.

- A **PRIMARY KEY** constraint is a unique identifier for a row within a database table.
- A **UNIQUE** constraint enforces the uniqueness of the values in a set of columns, so no duplicate values are entered
- A **FOREIGN KEY** constraint prevents any actions that would destroy links between tables with the corresponding data values. A foreign key in one table points to a primary key in another table.
- A **CHECK** constraint is used to limit the values that can be placed in a column. The check constraints are used to enforce domain integrity.

- A **NOT NULL** constraint enforces that the column will not accept null values. The not null constraints are used to enforce domain integrity, as the check constraints.

Q. What are the properties of the Relational tables?

A. Relational tables have six properties:

Values are atomic.
Column values are of the same kind.
Each row is unique.
The sequence of columns is insignificant.
The sequence of rows is insignificant.
Each column must have a unique name.

Q. What is Identity?

A. Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1.

Q. Which virtual table does a trigger use?

A. Inserted and Deleted.

Q. List few advantages of Stored Procedure.

A.

- **Stored procedure** can reduced network traffic and latency, boosting application performance.
- **Stored procedure** execution plans can be reused, staying cached in SQL Server's memory, reducing server overhead.
- **Stored procedures** help promote code reuse.
- **Stored procedures** can encapsulate logic. You can change stored procedure code without affecting clients.
- **Stored procedures** provide better security to your data.

Q. What is OLTP(OnLine Transaction Processing)?

A. In OLTP (online transaction processing) systems relational database design use the discipline of data modeling and generally follow the Codd rules of data normalization in order to ensure absolute data integrity. Using these rules complex information is broken down into its most simple structures (a table) where all of the individual atomic level elements relate to each other and satisfy the normalization rules.

Q. What are Transactions?

A. A transaction is a group of operations combined into a logical unit of work. Developers use transactions to control and maintain the consistency and integrity of each action in a transaction, despite errors that might occur in the system.

Q. What are ACID Properties of a Transaction?

A. 1) Atomicity
2) Consistency
3) Isolation
4) Durability