Assignment Day 5–SQL: Comprehensive practice

# Answer following questions

1. What is an object in SQL?

An object is any SQL Server resource, such as a SQL Server lock or Windows process.

1. What is Index? What are the advantages and disadvantages of using Indexes?

Index is an on-disk structure associated with a table or view that speeds retrieval of rows from the table or view. The advantages of using indexes include speed up SELECT query, helps to make a row unique or without duplicates, and if the index is set to fill-text index, then we can search against large string values. The disadvantages of using indexes include additional disk space used and insert, update and delete statements become slow.

1. What are the types of Indexes?

Clustered Index, Non-Clustered Index, Column Store Index, Filtered Index, Hash Index, and Unique Index.

1. Does SQL Server automatically create indexes when a table is created? If yes, under which constraints?

Yes, a unique index is automatically created when you define a primary key or unique constraint.

1. Can a table have multiple clustered index? Why?

No, a table cannot have multiple clustered indexes, because the data rows themselves can be stored in only one order.

1. Can an index be created on multiple columns? Is yes, is the order of columns matter?

Yes, an index can be created on multiple columns and the order of columns matter.

1. Can indexes be created on views?

Yes, indexes can be created on views.

1. What is normalization? What are the steps (normal forms) to achieve normalization?

Normalization is a process of organizing data to minimize redundancy, which in turn ensures data consistency. When the first rule is applied, the data is in “first normal form”. Then, the second rule is applied and the data is in “second normal form”. The third rule is then applied and the data is in “third normal form”, which achieves normalization.

1. What is denormalization and under which scenarios can it be preferable?

Denormalization is a database optimization technique in which we add redundant data to one or more tables. Denormalization minimizes the need for joins and improves performance.

1. How do you achieve Data Integrity in SQL Server?

Data Integrity is used to maintain accuracy and consistency of data in a table. It is achieved by using constraints or triggers.

1. What are the different kinds of constraint do SQL Server have?

Not Null Constraint, Check Constraint, Default Constraint, Unique Constraint, Primary Constraint, and Foreign Constraint.

1. What is the difference between Primary Key and Unique Key?

Primary key cannot accept NULL values, whereas unique key can accept NULL values. One table can only have one primary key, whereas it can have more than one unique key. Primary key is in clustered index, whereas unique Key is in non-clustered unique index.

1. What is foreign key?

Foreign key is a column or combination of columns that is used to establish and enforce a link between the data in two tables.

1. Can a table have multiple foreign keys?

Yes, a table can have multiple foreign keys.

1. Does a foreign key have to be unique? Can it be null?

Yes, a foreign key has to be unique and it can be null.

1. Can we create indexes on Table Variables or Temporary Tables?

Yes, we can create indexes on Table Variables or Temporary Tables.

1. What is Transaction? What types of transaction levels are there in SQL Server?

Transaction is a logical unit of work that contains one or more SQL statements. The four key properties of a transaction are atomicity, consistency, isolation, and durability. And the four transaction isolation levels include Read Uncommitted, Read Committed, Repeatable Read, and Serializable.

# Write queries for following scenarios

1. Write an sql statement that will display the name of each customer and the sum of order totals placed by that customer during the year 2002

Create table customer(cust\_id int, iname varchar (50)) create table order(order\_id int,cust\_id int,amount money,order\_date smalldatetime)

SELECT c.iname, SUM(o.amount)

FROM customer c inner join order o ON c.cust\_id = o.cust\_id

WHERE YEAR(order\_date) = ‘2002’

GROUP BY c.iname

2. The following table is used to store information about company’s personnel:

Create table person (id int, firstname varchar(100), lastname varchar(100)) write a query that returns all employees whose last names start with “A”.

SELECT \* FROM person WHERE lastname like ‘A%’

3. The information about company’s personnel is stored in the following table:

Create table person(person\_id int primary key, manager\_id int null, name varchar(100)not null) The filed managed\_id contains the person\_id of the employee’s manager.

Please write a query that would return the names of all top managers(an employee who does not have a manger, and the number of people that report directly to this manager.

SELECT dt.person\_id, COUNT(p.person\_id)

FROM person p inner join (SELECT person\_id FROM person WHERE manager is NULL) dt

ON p.person\_id = dt.person\_id

GROUP BY dt.person\_id

4. List all events that can cause a trigger to be executed.

DML Events - INSERT , UPDATE , or DELETE

DDL Events - CREATE , ALTER , or DROP

LOGON Event - when a user session is being established

5. Generate a destination schema in 3rd Normal Form. Include all necessary fact, join, and dictionary tables, and all Primary and Foreign Key relationships. The following assumptions can be made:

a. Each Company can have one or more Divisions.

b. Each record in the Company table represents a unique combination

c. Physical locations are associated with Divisions.

d. Some Company Divisions are collocated at the same physical of Company Name and Division Name.

e. Contacts can be associated with one or more divisions and the address, but are differentiated by suite/mail drop records.status of each association should be separately maintained and audited.

Create Table Company (CompID int PRIMARY KEY, CompanyName varchar(30) NOT NULL, DivID int FOREIGN KEY REFERENCES Divisions(DivID))

Create Table Divisions (DivID int PRIMARY KEY, DivisionName varchar(30) NOT NULL, LocID int FOREIGN KEY REFERENCES Locations(LocID))

Create Table Locations (LocID int PRIMARY KEY, address varchar(30) NOT NULL, sid int FOREIGN KEY REFERENCES Suites(sid))

Create Table Suites (sid int PRIMARY KEY, suite varchar (20) NOT NULL)

GOOD LUCK.