

## Assignment Day 5–SQL: Comprehensive practice

Amber Han

### Answer following questions

1. What is an object in SQL?

A database object is any defined object in a database that is used to store or reference data. Like tables, views, clusters, etc.

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

Indexes are database objects based on table column for faster retrieval of data.

Advantage:

- a. Quickly find data that satisfy conditions in the where clause.
- b. To find matching rows in join clause.
- c. To maintain uniqueness of key column during insert and update.
- d. To sort, aggregate and group data.

Disadvantage:

- a. Additional disk space.
- b. Insert, update, delete statement become slow.
- c. A cluster index always cover a query.

3. What are the types of Indexes?

Clustered, Non-clustered, Unique, Full-text, Spatial, Filtered.

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

Yes, unique constraints.

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

No, one table can have only one clustered index. A clustered index sorts and stores the data rows in the table based on the index key values. Therefore, only one clustered index can be created on each table because the data rows themselves can only be sorted in one order.

6. 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.

Yes. It matters.

7. Can indexes be created on views?

Yes.

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

Data normalization is a process of organizing data to minimize data duplication, and ensure data dependency. Normalization has a series of steps called 'Forms', the more steps you take the more normalized your table are. There are three steps: first normal form, second normal form, third normal form.

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

Denormalization is about deliberately adding redundancy to improve performance. Denormalization is used when there is a lot of tables involved in retrieving data.

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

There are 2 types of data integrity: System/Pre defined integrity, user-defined integrity. To enforce data integrity, you can constrain or restrict the data values that users can insert, delete, or update in the database.

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

**NOT NULL** - Ensures that a column cannot have a NULL value

**UNIQUE** - Ensures that all values in a column are different

**PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

**FOREIGN KEY** - Prevents actions that would destroy links between tables

**CHECK** - Ensures that the values in a column satisfies a specific condition

**DEFAULT** - Sets a default value for a column if no value is specified

**CREATE INDEX** - Used to create and retrieve data from the database very quickly

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

Primary Key is a column that is used to uniquely identify each tuple of the table.

It is used to add integrity constraints to the table. Only one primary key is allowed to be used in a table.

Unique key is a constraint that is used to uniquely identify a tuple in a table.

Multiple unique keys can present in a table. NULL values are allowed in case of a unique key.

13. What is foreign key?

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

14. Can a table have multiple foreign keys?

Yes. A table may have multiple foreign keys, and each foreign key can have a different parent table.

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

It does not. It can be NULL or duplicate.

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

No. Indexes cannot be created on table variables, but can be created on temporary tables.

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

A transaction is a logical unit of work that contains one or more SQL statements. A transaction is an atomic unit.

There are five type of Isolation level in MS SQL Server

1. Read Committed (The Default Isolation Level of MS SQL Server)
2. Read Uncommitted
3. Repeatable Read
4. Serializable
5. Snapshot

## 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)

```
CREATE TABLE customer(customer_id INT, iname VARCHAR(50))
CREATE TABLE [order](order_id INT, customer_id INT, amount MONEY, order_date
SMALLDATETIME)
SELECT c.iname, SUM(o.amount) total FROM customer c
INNER JOIN [order] o
ON c.customer_id = o.customer_id
WHERE YEAR(order_date) = 2002
```

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".

```
CREATE TABLE person(id INT, firstname VARCHAR(100), lastname VARCHAR(100))
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.

```
CREATE TABLE person(person_id INT PRIMARY KEY, manager_id INT NULL, name VARCHAR(100) NOT NULL)
SELECT x.name, COUNT(*) FROM person p
LEFT JOIN
(SELECT * FROM person p WHERE manager_id IS NULL) x
ON p.person_id = x.manager_id
GROUP BY x.name
```

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

Insert, delete, update.

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:

- Each Company can have one or more Divisions.
- Each record in the Company table represents a unique combination
- Physical locations are associated with Divisions.
- Some Company Divisions are collocated at the same physical of Company Name and Division Name.
- 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(companyid INT PRIMARY KEY, companyname VARCHAR(100) NOT NULL)
```

```
CREATE TABLE Division(divisionid INT PRIMARY KEY, divisionname VARCHAR(100) NOT NULL)
```

```
CREATE TABLE Contacts(companyid INT PRIMARY KEY, divisionid INT FOREIGN KEY,
locationid VARCHAR(100) NOT NULL FOREIGN KEY, suitid INT FOREIGN KEY, mail VARCHAR(100),
PRIMARY KEY (companyid, divisionid, locationid, suitid))
```

```
CREATE TABLE Physical_location(locationid INT PRIMARY KEY, address VARCHAR(100) NOT NULL)
```

```
CREATE TABLE Contacts_adress(suitid INT PRIMARY KEY, mail VARCHAR(100))
```

-

GOOD LUCK.