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 SQL Server lock or windows process

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

Index is database object based on table column for faster retrieval of data

Pro: quickly find data, find matching rows in join clause, maintain uniqueness of key column during insert and update, to sort, aggregate and group data

Con: additional disk space, insert update, delete statement become slow, clustered index always cover a query

1. What are the types of Indexes?

Clustered indexes and non clustered indexes

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

Yes, unique constraints

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

No. One table 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.

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. Yes . It mattersCan indexes be created on views?

1. 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 and third normal form.

1. 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 retreiving data.

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

We can apply Entity **integrity** to the Table by specifying a primary key, unique key, and not null. Referential **integrity** ensures the relationship between the Tables. We can apply this using a Foreign Key constraint.What are the different kinds of constraint do SQL Server have?

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

1. Primary key does not accept null value but unique constraint accepts one null value

2. one table can have only one primary key but a table can have multiple unique constraints

3. primary key will sort the data in asc order by default but unique constraint can not do that

4. primary key by default creates clustered index but unique constrain creates non clustered index

1. 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. The table with the **foreign key** is called the child table, and the table with the primary **key** is called the referenced or parent table.

1. Can a table have multiple foreign keys?

Yes

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

it can be null or duplicate

1. Can we create indexes on Table Variables or Temporary Tables?
2. What is Transaction? What types of transaction levels are there in SQL Server?

A **transaction** is the propagation of one or more changes to the database.

Read Uncommitted, Read Committed, Repeatable Read, 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) total from customer c

join [order] o

on c.cust\_id = o.cust\_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, firstanme 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.

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:

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(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 foreign key, divisionid int foreign key, locationid varchar(100) 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\_address(suitid int primary key,mail varchar(100))