

Assignment Day 5–SQL: Comprehensive practice

Answer following questions

1. What is an object in SQL?

SQL objects are schemas, journals, catalogs, tables, aliases, views, indexes, constraints, triggers, sequences, stored procedures, user-defined functions, user-defined types, global variables, and SQL packages. SQL creates and maintains these objects as system objects.

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.

Query optimizer depends on indexed columns to function. Indexes can separate structure attached to a table. They contain pointers to the physical data. Server can quickly find data that satisfy conditions in the WHERE clause. Indexes also save time in finding matching rows in the JOIN clause and maintaining uniqueness of key column during INSERT and UPDATE.

Indexes take up additional disk space. Redundant indexes will slower Insert, Update, Delete Statements.

3. What are the types of Indexes?

Clustered index and non-clustered index.

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

Yes, it will create clustered index under the primary key and a non-clustered index under the unique key.

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

No. One table only has one clustered index. Because it sorts and stores the records by key value in the disk space. Every table's data can be sorted by only one order so that only one clustered index is allowed for each table.

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

Yes it can. And yes it matters. The database checks the list from left to right, and has to find a corresponding column reference matching the order defined.

7. Can indexes be created on views?

Yes, but indexes can only be created on views which have the same owner as the referenced table or tables.

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

Database Normalization is a process of organizing data to minimize redundancy (data duplication), which in turn ensures data consistency.

Normalization has a series of steps called "Forms", the more steps you take the more normalized your tables are. There are 1st NF, 2ND NF, 3RD NF and BCNF.

9. 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. This can help us avoid costly joins in a relational database.
10. How do you achieve Data Integrity in SQL Server?
Data Integrity is used to maintain accuracy and consistency of data in a table. We can use constraints to achieve System/Pre Defined Integrity and use triggers to achieve User-Defined Integrity.
11. What are the different kinds of constraint do SQL Server have?
Not Null Constraint, Check Constraint, Default Constraint, Unique Constraint, Primary Constraint, Foreign Constraint
12. What is the difference between Primary Key and Unique Key?
The primary key does not accept null value. Unique key does, it can accept only one null value.
13. What is foreign key?
A foreign key (FK) 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.
14. Can a table have multiple foreign keys?
Yes. A table can have a composite key that contains no foreign key columns, and can have multiple foreign keys that are not a part of any composite key.
15. Does a foreign key have to be unique? Can it be null?
It can be NULL or duplicate.
16. Can we create indexes on Table Variables or Temporary Tables?
Yes.
17. What is Transaction? What types of transaction levels are there in SQL Server?
Transaction is a logical unit of work.
Read Uncommitted (Lowest level)
Read Committed
Repeatable Read
Serializable (Highest Level)
Snapshot Isolation

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)

```

;with cte
as
(
select c.iname, count(distinct o.order_id ) count_orders from customer c
join order o on o.cust_id = c.cust_id
where year(order_date ) = 2002
group by c.iname
)
select c.iname, isnull(cte.count_orders, 0) from customer c
left join cte on c.iname = cte.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 id 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 p1.name,
(
select manager_table.emp_qty from
(select manager_id mgr, count(person_id) emp_qty from person group by manager_id)
manager_table
where p1.person_id = manager_table.mgr
)
from person p1
where p1.manager_id is null

```

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

Database-level events, such as creating, altering, or dropping objects; DML Statements like Insert , Delete or Update; Server level events like creation of databases, changes to logins, log on.

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 Location(location_id int primary key, address varchar(50) not null)
create table Company(company_id int primary key, company_name varchar(50) not null,
location_id int foreign key references Location(location_id) )
create table Division(ddivision_id int primary key, division_name varchar(50) not null,
location_id int foreign key references Location(location_id))

```



```
create table Contacts(concact_name varchar(50) not null, suite_mail_id int primary
key,suite_mail_address varchar(50) not null)
create table Contacts_Association(
company_id int foreign key references Company(company_id),
division_id int foreign key references Division(division_id),
suite_mail_id int foreign key references Contacts(suite_mail_id),
primary key (company_id, division_id, suite_mail_id))
create table Company_With_Divison(company_id int foreign references Company(company_id),
division_id int foreign key references Division(division_id), primary key(company_id,
division_id) )
```

GOOD LUCK.