You can find this on my github: <https://github.com/princesscorn/assignments-antra.git>

# Write queries for following scenarios

Use Northwind database. All questions are based on assumptions described by the Database Diagram sent to you yesterday. When inserting, make up info if necessary. Write query for each step. Do not use IDE. BE CAREFUL WHEN DELETING DATA OR DROPPING TABLE.

## 1.      Create a view named “view\_product\_order\_[your\_last\_name]”, list all products and total ordered quantity for that product.

Solution:

CREATE VIEW view\_product\_order\_wang AS

SELECT p.ProductID, p.ProductName, SUM(od.Quantity) AS TotalQtyOfProductOrder

FROM Products p JOIN [Order Details] od ON p.ProductID = od.ProductID

GROUP BY p.ProductID, p.ProductName

## 2.      Create a stored procedure “sp\_product\_order\_quantity\_[your\_last\_name]” that accept product id as an input and total quantities of order as output parameter.

Solution:

CREATE OR ALTER PROC sp\_Product\_Order\_Quantity\_Wwk

@ProductID INT,

@TotalOrderQty INT OUT

AS

BEGIN

SELECT @TotalOrderQty = SUM(Quantity)

FROM [Order Details] OD JOIN Products P ON P.ProductID = OD.ProductID

WHERE P.ProductID = ProductID

END

DECLARE @Tot INT

EXEC sp\_Product\_Order\_Quantity\_Wwk 11, @Tot OUT

PRINT @To

## 3.      Create a stored procedure “sp\_product\_order\_city\_[your\_last\_name]” that accept product name as an input and top 5 cities that ordered most that product combined with the total quantity of that product ordered from that city as output.

Solution: (output NOT output parameter)

CREATE OR ALTER PROC sp\_Product\_Order\_City\_Wwk

@ProductName NVARCHAR(50)

AS

BEGIN

SELECT TOP 5 ShipCity,SUM(Quantity)

FROM [Order Details] OD JOIN Products P ON P.ProductID = OD.ProductID

JOIN Orders O ON O.OrderID = OD.OrderID

WHERE ProductName=@ProductName

GROUP BY ProductName,ShipCity

ORDER BY SUM(Quantity) DESC

END

EXEC sp\_Product\_Order\_City\_Wwk 'Queso Cabrales'

## 4.      Create 2 new tables “people\_your\_last\_name” “city\_your\_last\_name”. City table has two records: {Id:1, City: Seattle}, {Id:2, City: Green Bay}. People has three records: {id:1, Name: Aaron Rodgers, City: 2}, {id:2, Name: Russell Wilson, City:1}, {Id: 3, Name: Jody Nelson, City:2}. Remove city of Seattle. If there was anyone from Seattle, put them into a new city “Madison”. Create a view “Packers\_your\_name” lists all people from Green Bay. If any error occurred, no changes should be made to DB. (after test) Drop both tables and view.

Solution:

USE tempdb

CREATE TABLE City\_Wang

(

Id INT PRIMARY KEY,

City VARCHAR(30)

);

CREATE TABLE People\_Wang

(

Id INT PRIMARY KEY,

Name VARCHAR(30),

City INT FOREIGN KEY REFERENCES City\_Wang(Id)

);

BEGIN TRAN

INSERT City\_Wang VALUES(1, 'Seattle');

INSERT City\_Wang VALUES(2, 'Green Bay');

INSERT People\_Wang VALUES (1, 'Aaron Rodgers', 2);

INSERT People\_Wang VALUES (2, 'Russell Wilson', 1);

INSERT People\_Wang VALUES (3, 'Jody Nelson', 2);

IF EXISTS(SELECT Id FROM People\_Wang

WHERE City = (SELECT Id FROM City\_Wang WHERE City = 'Seattle' ))

BEGIN

INSERT City\_Wang VALUES (3, 'Madison');

UPDATE People\_Wang SET City = (SELECT ID

FROM City\_Wang WHERE City = 'Madison')

WHERE Id IN (SELECT Id FROM People\_Wang

WHERE City = (SELECT Id FROM City\_Wang

WHERE City = 'Seattle' ))

END

DELETE FROM City\_Wang WHERE City = 'Seattle'

CREATE VIEW Packers\_Wang AS

SELECT Name FROM People\_Wang

WHERE City = (SELECT ID FROM City\_Wang WHERE City = 'Green Bay')

SELECT \* FROM Packers\_Wang

COMMIT

DROP TABLE People\_Wang

DROP TABLE City\_Wang

DROP VIEW Packers\_Wang

## 5.       Create a stored procedure “sp\_birthday\_employees\_[you\_last\_name]” that creates a new table “birthday\_employees\_your\_last\_name” and fill it with all employees that have a birthday on Feb. (Make a screen shot) drop the table. Employee table should not be affected.

Solution:

USE tempdb

ALTER PROC sp\_birthday\_employees\_wang

AS

BEGIN

SELECT \* INTO birthday\_employees\_wang FROM Northwind.dbo.Employees

WHERE MONTH(BirthDate) = 02

SELECT \* FROM birthday\_employees\_wang

DROP TABLE birthday\_employees\_wang

END

Graphical user interface, text, application

Description automatically generated

## 6.      How do you make sure two tables have the same data?

Solution:

SELECT \* FROM TABLE1

EXCEPT

SELECT \* FROM TABLE2

# Test

## 1. What is View? What are the benefits of using views?

View is a virtual table, whose contents are defined by a query. View consists of a set of named columns and rows of data.

Benefits:

1) Simplify Data Manipulation

2) Allows backward compatible interface

3) Customize Data: use views to give relevant information with customization to the code based on the login ID of user.

## 2. Can data be modified through views?

Yes, we can perform CRUD operations in a basic view (single table view), but the complex view is not allowed CRUD operations.

## 3. What is stored procedure and what are the benefits of using it?

A stored procedure groups one or more Transact-SQL statements into a logical unit, stored as an object in a SQL Server database. So the code can be reused multiple times as required.

Benefits:

1) Increase database security.

2) Faster execution.

3) Help centralize transact.

4) Help reduce network traffic for larger queries.

5) Encourage code reusability.

## 4. What is the difference between view and stored procedure?

1) VIEW can only read data, while procedure can alter the data.

2) View does not accept parameters, stored procedure does.

3) View can contain only Select query, a stored procedure can contain several statements like if, else, loop, etc.

## 5. What is the difference between stored procedure and functions?

1) Usage: SP is for DML, functions are for calculations.

2) Call method: SP uses exec, Functions must use SELECT

3) Input: SP may or may not have any inputs, Functions do require one.

4) Output: SP may or may not have any output, but functions must have output parameters.

5) SP can call functions, but functions cannot call SP.

## 6. Can stored procedure return multiple result sets?

Yes. Most stored procedures return multiple result sets. Such a stored procedure usually includes one or more select statements.

## 7. Can stored procedure be executed as part of SELECT Statement? Why?

No. Because stored procedure may or may not return a value, the Stored procedure cannot be used in the select statements. You can re-write this SP in Function and call this function in the Select statement as you need.

## 8. What is Trigger? What types of Triggers are there?

Trigger is a special type of SP that automatically fired when a specific event occurs.

There are 3 types of Trigger:

DML Triggers: Modify Data

DDL Triggers: Create alter Drop

Logon Triggers: Fires on authentication

## 9. What is the difference between Trigger and Stored Procedure?

1) Trigger is a stored procedure that runs automatically when various events happen (eg update, insert, delete), Stored Procedures can be invoked explicitly by the user.

2) Trigger cannot take input as a parameter, but the stored procedure can.

3) We can't use transaction statements inside a trigger, but We can use transaction statements like begin transaction, commit transaction, and rollback inside a stored procedure

4) Triggers cannot return values, stored procedures can.

## 10. What's the difference between primary key and unique constraints?

Primary Key and Unique Key ensure they don’t have duplicate values.

1) **Nulls**. In Primary, we cannot put NULL, but in unique key, we can.

2) **Number**. You can have many unique keys, but only one Primary Key.

3) **Index**: Primary Key automatically attaches a clustered index to the datatype. Only one.

Unique automatically attaches a non-clustered index to the datatype. One or more.

## 11. What is Check constraint?

The CHECK constraint is used to limit the value range that can be placed in a column.

# Review

## 1. What is index; types of indices; pros and cons

Index is a on desk object, contains one or more columns in the table or view, allows the SQL Server to find data without scanning the whole table, helps to make queries go faster.

There are two types of indices:

Clustered Index and Non-Clustered Index

Pros: Helps to make queries go faster.

Cons:

1) Too many indexes can cause slowdown. A good number of indexes per table is 3-4.

2) DML statements can slow down Insert/Delete/Update statement

3) Need extra Space

## 2. What's the difference between Primary key and Unique constraint?

Primary Key and Unique Key ensure they don’t have duplicate values.

1) **Nulls**. In Primary, we cannot put NULL, but in unique key, we can.

2) **Number**. You can have many unique keys, but only one Primary Key.

3) **Index**: Primary Key automatically attaches a clustered index to the datatype. Only one.

Unique automatically attaches a non-clustered index to the datatype. One or more.

## 3. Tell me about check constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

## 4. Difference between temp table and table variable

Solution:

1) Storage: Both Table variables(@tableVar) and Temp tables(#tmpTable) are stored in tempdb.

2) Scope: Table Variables scoped for current batch,

Temp Tables scoped local/global.

3) Scenario: Table variables for small data (like rows < 1000)

Temp tables for big data.

4) Usage: Table variables can be passed as a parameter to functions and stored procedures,

while the same cannot be done with Temp tables.

5) Structure: Temp tables are allowed CREATE INDEX except FOREIGN KEY.

Table variables can have index by using Primary Key/Unique Constraint.

6) Performance: Table variables are faster than temp tables.

## 5. Difference between WHERE and HAVING

1) WHERE applies to before aggregates.

HAVING applies to after aggregates.

2) WHERE could only work on columns that already exist in original queries,

HAVING only filters on aggregation functions.

3) WHERE after FROM and before key words like “GROUP BY”.

HAVING shall be written after GROUP BY (all in specific sequence).

4) WHERE can be used with SELECT, UPDATE, and DELETE,

HAVING can only be used with SELECT

## 6. Difference between RANK() and DenseRank() — value gap

RANK() and DENSE\_RANK() are similar to ROW\_NUMBER(), but when there are ties, they will give the same value to the tied values, then

RANK() will keep the ranking, so the number may go like 1, 2, 2, 4

DENSE\_RANK() will never give any gaps, so the number may go like 1, 2, 2, 3

## 7. COUNT(\*) vs. COUNT(colName)

COUNT(\*) will include NULL value, but COUNT(colName) will not.

## 8. What's the difference between left join and inner join? JOIN and Subquery, which one has a better performance, why?

**INNER JOIN:** Returns matched records from both tables

**LEFT JOIN:** Returns all the records from the left table, and matched records from the right table, for the non-matching records in the right table, the result set will return null values

JOIN has better performance, because there is an optimizer for joins. However you could try using a subquery when there is a lot of columns to group by. Correlated subqueries can also improve performance if used in the select clause.

## 9. What is correlated subquery

A correlated subquery is a subquery that uses values from the outer query.

## 10. What is a CTE, why do we need CTE?

CTE stands for Common Table Expression, which is a temporary named result set that you can reference within SELECT, INSERT, UPDATE, and DELETE. CTE can also be used in a View.

CTE allows you to create and use a query just before your main query.

Benefits:

1) Common Table Expressions better organize long queries. Multiple subqueries often look messy.

2) CTEs also make a query more readable, as you have a name for each of the Common Table Expressions used in a query.

3) CTEs organize the query so that it better reflects human logic. With CTEs, you start by defining the temporary result...

## 11. What does SQL Profiler do?

Microsoft SQL Server Profiler is a graphical user interface to SQL Trace for **monitoring an instance of the Database Engine or Analysis Services**. You can capture and save data about each event to a file or table to analyze later.

## 12. What is SQL injection, how to avoid SQL injection?

Some hackers inject malicious code in our SQL Queries, more or less to cause harm or destruction to the database.

The only sure way to prevent SQL Injection attacks is input validation and parametrized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms.

## 13. Difference between SP and user defined function? When to use SP when to use function?

Difference:

1) Usage: SP is for DML, functions are for calculations.

2) Call method: SP uses exec, Functions must use SELECT

3) Input: SP may or may not have any inputs, Functions do require one.

4) Output: SP may or may not have any output, but functions must have output parameters.

5) SP can call functions, but functions cannot call SP.

We can use SP when no input and output are required.

## 14. Criteria of Union and Union all? Difference between UNION and UNION ALL

Criteria:

1) column numbers must match,

2) column types must match

3) Column names are chosen from the first select statement.

Difference:

1) Union will automatically remove the duplicate records, Union All will not.

2) Union will sort the result by the order of the first column by default, Union All will not.

3) Union All’s performance is better than Union.

4) Union All can be used in recursive CTEs, Union cannot.

## 15. Steps you take to improve SQL Queries

1) Favor set - based logic over procedural or cursor logic

2) Test query variations for performance

3) Avoid query hints

4) DO NOT Use correlated subqueries to improve performance

5) Avoid using a scalar user- defined function in the WHERE clause

6) Use table-valued user-defined functions as derived tables.

7) Avoid unnecessary GROUP BY columns, Use a Subquery instead.

8) Use CASE expressions to include variable logic in a query

9) Divide joins into temporary tables when you query very large tables

10) Refactoring Cursors into Queries

## 16. concurrency problem in transaction

Occurs when two or more users are trying to access the same data or information.

--Dirty Reads: If t1 allows t2 to read uncommitted data and then t1 rolls back. This causes possible misread.

Happens in read uncommitted isolation level, solved by read committed isolation level.

--Lost Updates: t1 and t2 read and update the same data, but t2 finishes its work earlier than t1, then t2 will lose its updates.

Happens in read committed isolation level, solved with repeatable read isolation level.

--Non-Repeatable Reads: t1 read the same data twice while t2 is updating the data.

happens in read committed isolation level, solved with repeatable read isolation level.

--Phantom Reads: t1 reads the same data twice while t2 is inserting records

Caused by repeatable reads isolation level. Updates happens between t1 read causing two different queries. Solved by Serializable Isolation level and causes t2 wait for t1 to finish.

## 17. what is deadlock, how to prevent

Deadlocks occur when two processes want to access resources that are mutually being locked by each other. This locked situation can continue forever if nobody stops it.

To prevent:

* Access the resources in the same order
* Write the shortest transactions as much as possible and lock the resource for the minimum duration
* Consider using READ COMMITTED SNAPSHOT ISOLATION and SNAPSHOT ISOLATION levels
* Limiting the usage of the cursors
* Design more normalized databases
* Avoid poorly-optimized queries

## 18. what is normalization, 1NF - BCNF, benefits using normalization

Normalization is a database design technique to remove/minimize redundant data.

Normalization is implemented by splitting tables into two: one having reference data (master table), another having transaction data.

1NF-BCNF:

Data in each column should be atomic, no multiple values separated by comma

Table does not contain any repeating column group.

Identify each record using primary key

Entity integrity.

(Atomic means when you want to insert/update/delete, you have to write zero code.)

## 19. what are the system defined databases?

There is a total of 5 System Databases, out of these you can see 4 under the System Databases folder. The 5th database is Resource which does not appear on SSMS.

Master: Records all the system-level information for an instance of SQL Server.

Model: Is used as the template for all databases created on the instance of SQL Server.

Msdb: Is used by SQL Server Agent for scheduling alerts and jobs.

Tempdb: holding temporary objects or intermediate result sets.

Resource: Is a read-only database that contains system objects,which physically persisted in the Resource database, but they logically appear in the sys schema of every database.

## 20. composite key

A composite key is a combination of two or more columns in a table that can be used to uniquely identify each row in the table. Uniqueness is only guaranteed when the columns are combined, when taken individually the columns do not guarantee uniqueness.

This is usually seen in Joint tables.

## 21. candidate key

A key that is not a PK but eligible to be a PK.

Mostly it is a unique key without null value.

## 22. DDL vs. DML

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database. It’s common statements:

CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

## 23. ACID property

A: Atomicity - All or nothing. Either all of the work is committed or not

C: Consistency - The records will not change unless all is committed or not, there will not be any half-records. Begin Trans -Commit/Rollback.

Whatever happens in the middle of the transaction, this property will never leave the database in a half-completed state. (Does not alther or change the structure of the database. Referring to the “Correctness” of the db)

I: Isolation: two transactions will be isolated from each other by locking the resource.

D: Durability: Once the transactions are completed, the changes to the

database will be permanent.

## 24. table scan vs. index scan

Table scan means iterate over all table rows.

Index scan means iterate over all index items, when item index meets search condition, table row is retrieved through index.

Usually index scan is less expensive than a table scan because index is more flatter than a table.

## 25. Difference between Union and JOIN

**UNION** increases the number of records(rows), **JOIN** increases the number of columns.

**UNION** must have same column number, same data types of columns selected from each table.