Part 2: In an MS Word document, describe how you would apply the concepts covered above to the database you are creating for the MRP project. Explain why you believe applying specific schemas beyond the default "dbo" could be of a benefit to the physical implementation of a database. What are some roles (potential types of users) you might define for your MRP database? Explain why? Explain the permissions each role would have. Then on your own, conduct research using our text or Google to answer the following:

Based on the concepts about admin permission, I would apply it to the MRP project by adding a different schema rather than the default. As what I consume from the material, using the default schema mean that the user does not have full access or permission. Whereas if I apply specific schemas, I could use it to divide between the sections.

Possible users that I would define would be divided based on the section. For example, employees, inventory and customers would all have different user. Through that, the person who is in charge in inventory and raw materials can check and have access to storage. Those who are in charge of orders can have access to a customer's quantity and address. By having separate users, the permission that they can access will benefit them better. Such as having a section of people to have the only access to change if there is any changes in the orders might be helpful for them. So those who don't have any authority does not change it.

1. For an existing table, how would you change its schema using SQL?

   To change the schema of existing table, I can right-click on the table and then click design. Press f4 to open the properties window. In the schema box, select a new schema.

2. Can you Revoke a Deny?
   Apparently, in sql, I can't revoke a deny permission directly. Once I have rejected or denied a permission to a user a role. A way to undo it is to grant the same permission.

3. How is it possible for a user to create or drop tables among multiple schemas? Provide an example using SQL commands.

```
DECLARE @schema_name NVARCHAR(255);
DECLARE @table_name NVARCHAR(255);
DECLARE @sql NVARCHAR(MAX);

DECLARE cur CURSOR FOR
SELECT s.name, t.name
FROM sys.tables t
INNER JOIN sys.schemas s ON t.schema_id = s.schema_id

OPEN cur

FETCH NEXT FROM cur INTO @schema_name, @table_name

WHILE @@FETCH_STATUS = 0
BEGIN
```

```
    SET @sql = 'DROP TABLE ' + QUOTENAME(@schema_name) + '.' +
QUOTENAME(@table_name)
    EXEC sp_executesql @sql
    FETCH NEXT FROM cur INTO @schema_name, @table_name
END

CLOSE cur;
DEALLOCATE cur;
```