For the submission of your work:

- Create a folder named RollNo_Name_DBI202_PaperNo, e.g. se01245_LongNT_DBI202_01. Do not create any subfolder in this folder. All file created will be located in the above folder.

- For each question, you are required to write a database script. Create a file with the name corresponding to the index of the question. For example, for question 1, we will create a file named Q1.sql and create a file Q2.sql for question 2. So, if you do 10 questions, your folder must contain only 10 files Q1.sql, Q2.sql, Q3.sql, Q4.sql, Q5.sql, Q6.sql, Q7.sql, Q8.sql, Q9.sql and Q10.sql.

- Do not use any commands having the database name such as create database, alter database, use [database name], *etc*.

- Your response must contain only necessary commands for answering the question. Do not include any other command. For example, if you are required to create a trigger/procedure, then your response should contain only commands for creating the corresponding trigger/procedure; all commands for testing the created trigger/procedure are forbidden.

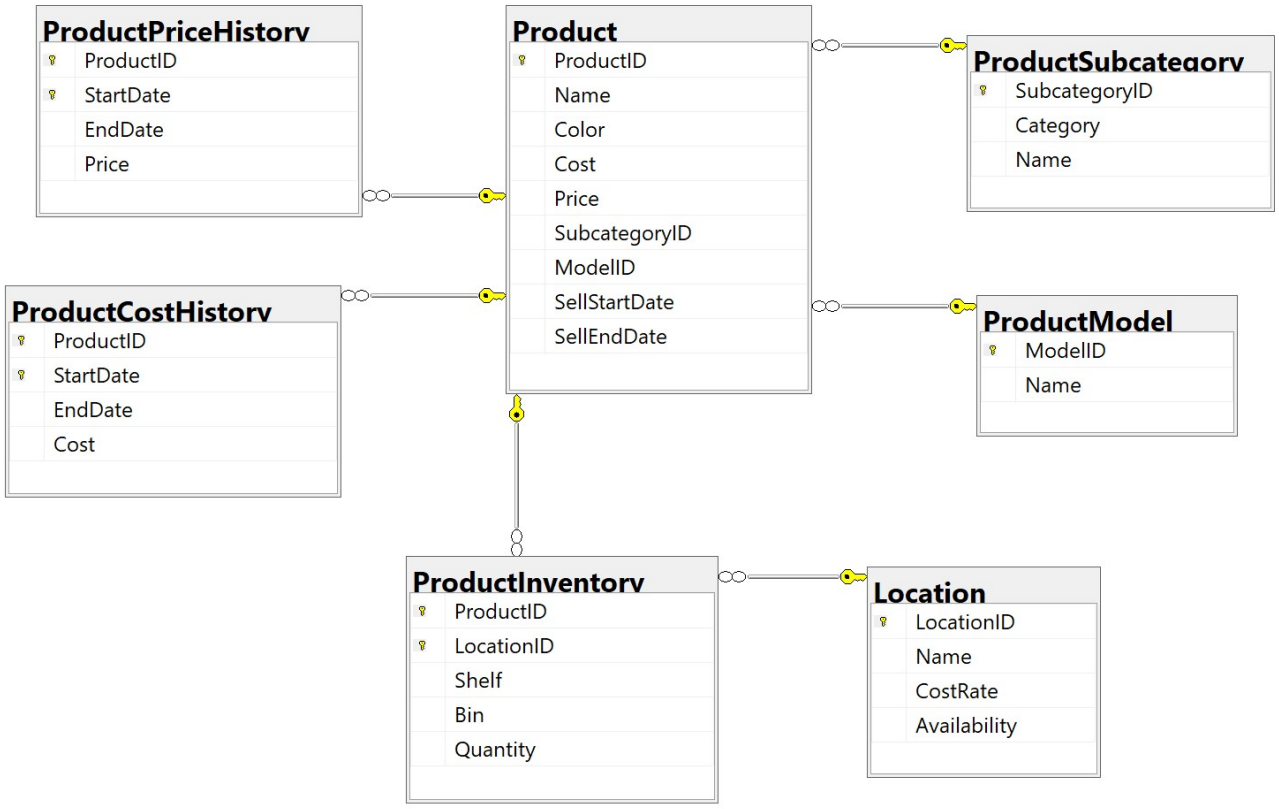
- On completion, import your work by browsing to the above folder.

- Note that:

+ You could use only SQL Server, SQL Server Management Studio, and paper or offline document in your computer.

+ If any of the previous requirements is not respected, your mark will be 0.

From the 2nd question, you should use the database provided in the .sql file which has the following database diagram. Please, run the provided script to create tables and insert data into your database.



Question 1:

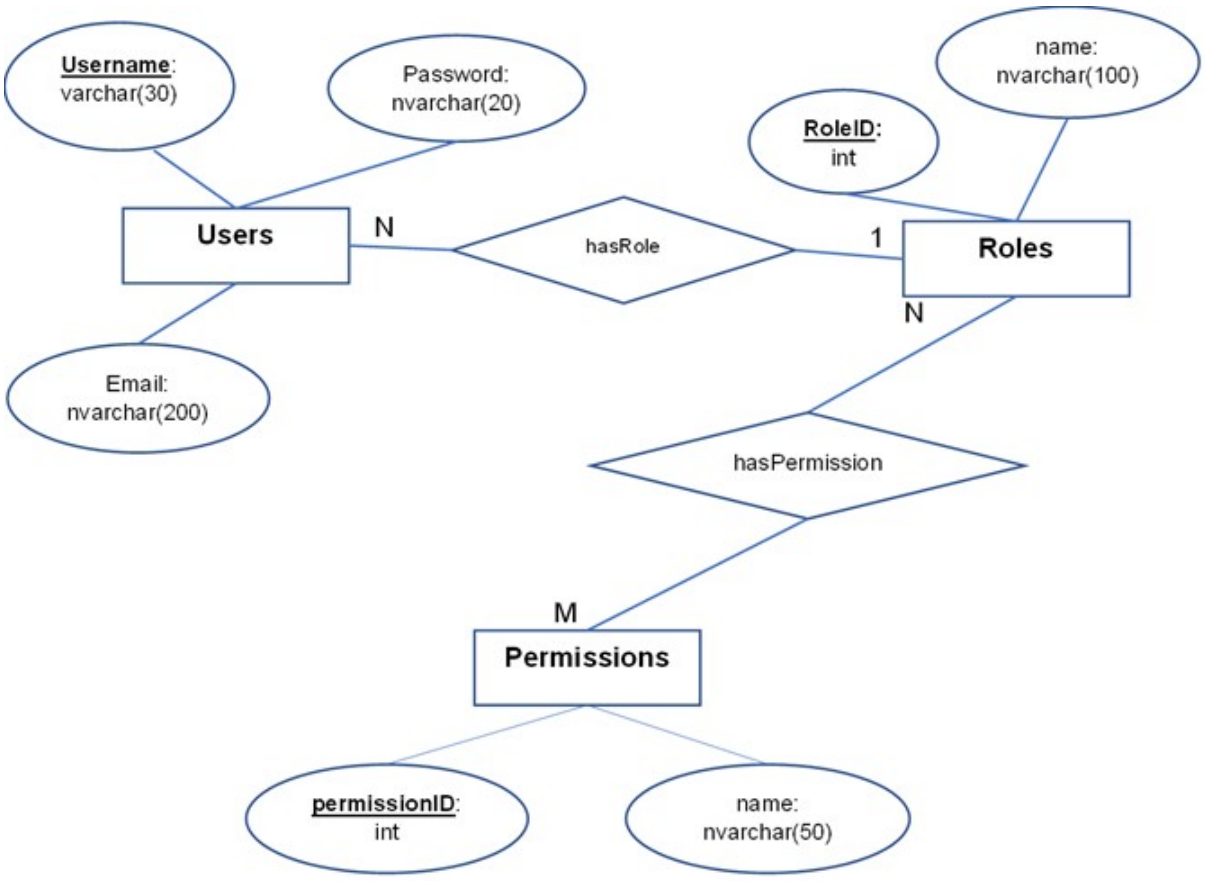
Create one database and then write SQL statements to create, in this database, all tables derived from the ERD given in the following picture with appropriate attributes, primary keys and foreign keys.

NOTE that when creating the SQL commands as request, you MUST keep the name of tables, relationship, attributes and data type of attributes as SAME as given in the given ERD.

Attributes with underline belong to the Primary Key of each entity.

Attributes which reference to the primary key of another table must have the same name as the attributes in the primary key of the referencing table.

When submitting the responses for this question, submit only SQL statements for creating tables with corresponding keys and foreign keys. Do not use “create database”, “Alter database”, “use database_name” or any statements using database’s name in your submission.



Picture 1.1

Question 2:

Select all product subcategories of the category 'Accessories' as follows:

	SubcategoryID	Category	Name
1	26	Accessories	Bike Racks
2	27	Accessories	Bike Stands
3	28	Accessories	Bottles and Cages
4	29	Accessories	Cleaners
5	30	Accessories	Fenders
6	31	Accessories	Helmets
7	32	Accessories	Hydration Packs
8	33	Accessories	Lights
9	34	Accessories	Locks
10	35	Accessories	Panniers
11	36	Accessories	Pumps
12	37	Accessories	Tires and Tubes

Picture 2.1

Question 3:

Write a query to select ProductID, LocationID, Quantity of all product inventory corresponding to the location 7 and the quantity greater than 250; display the results in descending order of Quantity as follows:

	ProductID	LocationID	Quantity
1	708	7	324
2	845	7	324
3	872	7	324
4	871	7	288
5	707	7	288
6	844	7	288
7	843	7	252
8	870	7	252

Picture 3.1

Question 4:

Write a query to display the product inventory information for all products having the 'Yellow' color and having the Cost smaller than 400. Display the results with the following attributes: ProductID, ProductName, Color, Cost, Price, LocationID, LocationName, Shelf, Bin, Quantity. Note that LocationID, LocationName, Shelf, Bin, Quantity are NULL if we have no information about the product inventory of the corresponding product.

	ProductID	ProductName	Color	Cost	Price	LocationID	LocationName	Shelf	Bin	Quantity
1	822	ML Road Frame-W - Yellow, 38	Yellow	360,9428	594,83	10	Frame Forming	N/A	0	115
2	822	ML Road Frame-W - Yellow, 38	Yellow	360,9428	594,83	20	Frame Welding	N/A	0	155
3	822	ML Road Frame-W - Yellow, 38	Yellow	360,9428	594,83	30	Debur and Polish	N/A	0	142
4	822	ML Road Frame-W - Yellow, 38	Yellow	360,9428	594,83	40	Paint	N/A	0	100
5	822	ML Road Frame-W - Yellow, 38	Yellow	360,9428	594,83	50	Subassembly	N/A	0	153
6	822	ML Road Frame-W - Yellow, 38	Yellow	360,9428	594,83	60	Final Assembly	N/A	0	131
7	833	ML Road Frame-W - Yellow, 40	Yellow	360,9428	594,83	NULL	NULL	NULL	NULL	NULL
8	834	ML Road Frame-W - Yellow, 42	Yellow	360,9428	594,83	NULL	NULL	NULL	NULL	NULL
9	835	ML Road Frame-W - Yellow, 44	Yellow	360,9428	594,83	NULL	NULL	NULL	NULL	NULL
10	836	ML Road Frame-W - Yellow, 48	Yellow	360,9428	594,83	NULL	NULL	NULL	NULL	NULL
11	886	LL Touring Frame - Yellow, 62	Yellow	199,8519	333,42	NULL	NULL	NULL	NULL	NULL
12	899	LL Touring Frame - Yellow, 44	Yellow	199,8519	333,42	NULL	NULL	NULL	NULL	NULL
13	900	LL Touring Frame - Yellow, 50	Yellow	199,8519	333,42	NULL	NULL	NULL	NULL	NULL
14	901	LL Touring Frame - Yellow, 54	Yellow	199,8519	333,42	NULL	NULL	NULL	NULL	NULL
15	902	LL Touring Frame - Yellow, 58	Yellow	199,8519	333,42	NULL	NULL	NULL	NULL	NULL

Picture 4.1

Question 5:

Write a query to display SubcategoryID, SubCategoryName, Category, NumberOfProducts corresponding to each subcategory where NumberOfProducts is the count of distinct products of the corresponding subcategory; display the results in ascending order of Category, then in descending order of NumberOfProducts for rows of the same category and finally in ascending order of SubCategoryName of rows having the same Category and NumberOfProducts as follows:

	SubcategoryID	SubCategoryName	Category	NumberOfProducts
1	37	Tires and Tubes	Accessories	11
2	28	Bottles and Cages	Accessories	3
3	31	Helmets	Accessories	3
4	33	Lights	Accessories	3
5	36	Pumps	Accessories	2
6	26	Bike Racks	Accessories	1
7	27	Bike Stands	Accessories	1
8	29	Cleaners	Accessories	1
9	30	Fenders	Accessories	1
10	32	Hydration Packs	Accessories	1
11	34	Locks	Accessories	1
12	35	Panniers	Accessories	1
13	2	Road Bikes	Bikes	43
14	1	Mountain Bikes	Bikes	32
15	3	Touring Bikes	Bikes	22
16	14	Road Frames	Components	33
17	12	Mountain Frames	Components	28
18	16	Touring Frames	Components	18
19	17	Wheels	Components	14
20	15	Saddles	Components	9
21	4	Handlebars	Components	8
22	13	Pedals	Components	7
23	5	Bottom Brackets	Components	3
24	8	Cranksets	Components	3
25	10	Forks	Components	3
26	11	Headsets	Components	3
27	6	Brakes	Components	2
28	9	Derailleurs	Components	2
29	7	Chains	Components	1

Picture 5.1

Question 6:

Write a query to display ProductID, Name, TotalQuantity of the products having the highest TotalQuantity; where TotalQuantity is the total quantity in all locations of the corresponding product.

	ProductID	Name	TotalQuantity
1	379	Hex Nut 7	1911

Picture 6.1

Question 7:

Write a query to display, for each location, the information of the products having the highest Quantity. Display the information with the following attributes: LocationID, LocationName, ProductID, ProductName, Quantity; where ProductID, ProductName, Quantity is the information of the products having the highest quantity in the given location. Display the result in ascending order of LocationName, then in descending order of ProductName with the products of the same location as follows:

	LocationID	LocationName	ProductID	ProductName	Quantity
1	30	Debur and Polish	747	HL Mountain Frame - Black, 38	148
2	60	Final Assembly	528	Seat Lug	729
3	7	Finished Goods Storage	708	Sport-100 Helmet, Black	324
4	7	Finished Goods Storage	872	Road Bottle Cage	324
5	7	Finished Goods Storage	845	Mountain Pump	324
6	10	Frame Forming	478	Metal Bar 2	622
7	20	Frame Welding	398	Handlebar Tube	550
8	5	Metal Storage	350	Fork Crown	622
9	6	Miscellaneous Storage	528	Seat Lug	924
10	40	Paint	747	HL Mountain Frame - Black, 38	107
11	3	Paint Shop	494	Paint - Silver	49
12	3	Paint Shop	495	Paint - Blue	49
13	4	Paint Storage	495	Paint - Blue	35
14	2	Sheet Metal Racks	483	Metal Sheet 3	691
15	45	Specialized Paint	944	LL Mountain Frame - Silver, 40	136
16	50	Subassembly	397	Hex Nut 19	763
17	1	Tool Crib	389	Hex Nut 2	657

Picture 7.1

Question 8:

Create a stored procedure named proc_product_model to calculate the count of distinct products belonging to a given model where @modelID int is an input parameter and @numberOfProducts int is an output parameter of the procedure.

For example, when we execute the procedure proc_product_model by using the following statements, the result should be as in the following figure:

```
declare @x int

exec proc_product_model 9, @x output

select @x as NumberOfProducts
```

	NumberOfProducts
1	12

Picture 8.1

Question 9:

Create a trigger named tr_insert_Product for the insert statement on table Product so that when we insert one or more products into the table Product, the system will display ProductID, ProductName, ModelID, ModelName corresponding to the products that have been inserted.

for example, when you execute the following statement, the system will display the results as in the following figure:

insert into Product(ProductID, Name, Cost, Price, ModelID, SellStartDate)

values (1000, 'Product Test', 12.5, 15.5, 1, '2021-10-25')

	ProductID	ProductName	ModelID	ModelName
1	1000	Product Test	1	Classic Vest

Picture 9.1

Question 10:

Write a query to delete from the table ProductInventory all rows corresponding to products belonging to the Model having ModelID = 33.