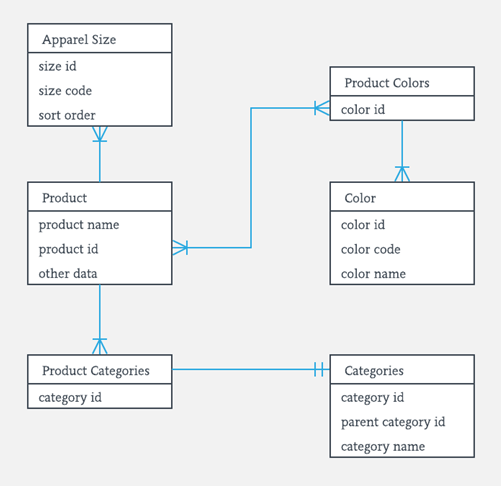
**SQL Assignment 2**

1. For an online purchasing database, create entity relationship diagrams. Create a database object from your entity diagram.



1. Create a SQL store process to register the use of the database, complete it with proper validation and transaction rollback and commit.

DECLARE @TransactionName VARCHAR(20)= 'Demotran1';

BEGIN TRAN @TransactionName;

INSERT INTO Demo

VALUES(1), (2);

ROLLBACK TRAN @TransactionName;

SELECT \* FROM demo;

DECLARE @TransactionName1 VARCHAR(20)= 'Demotran2';

BEGIN TRAN @TransactionName;

INSERT INTO Demo VALUES(1), (2);

COMMIT TRAN @TransactionName1;

SELECT \* FROM demo;

1. List the SQL aggregate function and demonstrate how to utilize it.

AVG – calculates the average of a set of values.

Select avg(column\_name) from table.

COUNT – counts rows in a specified table or view.

Select count(column\_name) from table.

MIN – gets the minimum value in a set of values.

Select max(column\_name) from table.

MAX – gets the maximum value in a set of values.

Select max(column\_name) from table.

SUM – calculates the sum of values.

Select sum(column\_name) from table.

1. In SQL, create a pivot query.

SELECT \* FROM

(

SELECT

category\_name,

product\_id

FROM

production.products p

INNER JOIN production.categories c

ON c.category\_id = p.category\_id

) t

PIVOT(

COUNT(product\_id)

FOR category\_name

IN (

[Children Bicycles],

[Comfort Bicycles],

[Cruisers Bicycles],

[Cyclocross Bicycles],

[Electric Bikes],

[Mountain Bikes],

[Road Bikes])

) AS pivot\_table;

1. With an example, describe how to join in SQL.

**SQL Join** statement is used to combine data or rows from two or more tables based

on a common field between them. Different types of Joins are as follows

* INNER JOIN

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

INNER JOIN table2

ON table1.matching\_column = table2.matching\_column;

table1: First table.

table2: Second table

matching column: Column common to both the tables.

* LEFT JOIN

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

LEFT JOIN table2

ON table1.matching\_column = table2.matching\_column;

table1: First table.

table2: Second table

matching column: Column common to both the tables.

* RIGHT JOIN

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

RIGHT JOIN table2

ON table1.matching\_column = table2.matching\_column;

table1: First table.

table2: Second table

matching\_column: Column common to both the tables.

* FULL JOIN

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

FULL JOIN table2

ON table1.matching\_column = table2.matching\_column;

table1: First table.

table2: Second table

matching\_column: Column common to both the tables.

1. How to locate the 4th highest value in a column in a row. Create your table.

select \* from (

select \*, row\_number() over(partition by column\_name order by column\_name) as ranking from table ) a

where a.ranking=4