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

Ans:

**ER Diagram** stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

## Components of the ER Diagram

This model is based on three basic concepts:

* Entities
* Attributes
* Relationships

**ER Diagram Examples**

For example, in a University database, we might have entities for Students, Courses, and Lecturers. Students entity can have attributes like Rollno, Name, and DeptID. They might have relationships with Courses and Lecturers.

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

A transaction in SQL Server is a single unit of work in a database. We perform many transactions daily. In the real world, consider a banking transaction. Suppose you withdraw money from your bank account, you expect that it should get successful once you get money in your account. In case, you try to withdraw money, but after you are done with all formalities, due to a technical error, the amount got deducted from your account, but it did not reach out to you. In this case, the transaction should be rolled back, and the amount should reappear in your account.

Similar to the above scenario, consider you are executing a script on the production database that updates data in the existing table. Your developer missed putting a WHERE clause and it might mess your data. In this article, we try to find answers to the following questions.

* Do we have an option to rollback changes in SQL Server?
* What precautions can we take before executing queries in a production environment?

Before we move further, you can refer to [SQL Server Transaction Overview](https://www.sqlshack.com/sql-server-transaction-overview/) article, and go through ACID properties and transaction states.

### **Explicit SQL Server transaction**

In this mode, each code block starts with a BEGIN TRANSACTION statement, and it requires an explicit COMMIT or ROLLBACK statement. It gives the flexibility to decide whether you want to save changes performed by query or not.

* **BEGIN TRANSACTION** – It indicates the starting point of a transaction
* **ROLLBACK TRANSACTION** -It starts the rollback process and reverts any changes performed by the transaction. It might take a long time depending upon the changes performed earlier
* **COMMIT TRANSACTION** -It commits the changes in the database. Once we issued a commit transaction, it cannot be rolled back

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

Ans:

An aggregate function allows you to perform a calculation on a set of values to return a single scalar value. We often use aggregate functions with the [GROUP BY](https://www.zentut.com/sql-tutorial/sql-group-by/)and [HAVING](https://www.zentut.com/sql-tutorial/sql-having/) clauses of the [SELECT](https://www.zentut.com/sql-tutorial/sql-select/)statement.

The following are the most commonly used SQL aggregate functions:

* [AVG](http://www.sqltutorial.org/sql-avg.aspx)– calculates the average of a set of values.
* [COUNT](http://www.sqltutorial.org/sql-count.aspx)– counts rows in a specified table or view.
* [MIN](http://www.sqltutorial.org/sql-min-max.aspx)– gets the minimum value in a set of values.
* [MAX](https://www.zentut.com/sql-tutorial/sql-max/)– gets the maximum value in a set of values.
* [SUM](http://www.sqltutorial.org/sql-sum.aspx)– calculates the sum of values.

4.In SQL, create a pivot query.

Ans:

SQL Server PIVOT operator rotates a table-valued expression. It turns the unique values in one column into multiple columns in the output and performs aggregations on any remaining column values.

You follow these steps to make a query a pivot table:

* First, select a base dataset for pivoting.
* Second, create a temporary result by using a derived table or [common table expression](https://www.sqlservertutorial.net/sql-server-basics/sql-server-cte/) (CTE)
* Third, apply the PIVOT operator.

Let’s apply these steps in the following example.

First, select category name and product id from the production.products and production.categories tables as the base data for pivoting:

SELECT

category\_name,

product\_id

FROM

production.products p

INNER JOIN production.categories c

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

Code language: SQL (Structured Query Language) (sql)

Second, create a temporary result set using a derived table:

SELECT \* FROM (

SELECT

category\_name,

product\_id

FROM

production.products p

INNER JOIN production.categories c

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

) t

Code language: SQL (Structured Query Language) (sql)

Third, apply the PIVOT operator:

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;

Code language: SQL (Structured Query Language) (sql)

This query generates the following output:



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

Ans:

JOINS in SQL are commands which are used to combine rows from two or more tables, based on a related column between those tables.  There are predominantly used when a user is trying to extract data from tables which have one-to-many or many-to-many relationships between them.

Now, that you know what joins mean, let us next learn the different types of joins.

There are mainly four types of joins that you need to understand. They are:

* [INNER JOIN](https://www.edureka.co/blog/sql-joins-types#INNER%20JOIN)
* [FULL JOIN](https://www.edureka.co/blog/sql-joins-types#FULL%20JOIN)
* [LEFT JOIN](https://www.edureka.co/blog/sql-joins-types#LEFT%20JOIN)
* [RIGHT JOIN](https://www.edureka.co/blog/sql-joins-types#RIGHT%20JOIN)

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/table1-3.png)

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

Ans:

To find the nth highest value of a column, you need to use ORDER BY DESC with LIMIT clause. If you want the second highest value of a column, use the below syntax:

SELECT \*FROM yourTableName ORDER BY DESC yourColumnName LIMIT 1,1;

If you want the fourth highest value of a column, use the below syntax:

SELECT \*FROM yourTableName ORDER BY DESC yourColumnName LIMIT 3,1;

If you want the first highest value of a column, use the below syntax:

SELECT \*FROM yourTableName ORDER BY DESC yourColumnName LIMIT 1;

As discussed in the above syntax, you need to change only in LIMIT clause. To understand the above syntax, let us create a table. The query to create a table is as follows:

mysql> create table NthSalaryDemo

   -> (

   -> Id int NOT NULL AUTO\_INCREMENT,

   -> Name varchar(10),

   -> Salary int,

   -> PRIMARY KEY(Id)

   -> );

Query OK, 0 rows affected (1.03 sec)

Insert some records in the table using insert command. The query is as follows:

mysql> insert into NthSalaryDemo(Name,Salary) values('Larry',5700);

Query OK, 1 row affected (0.41 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('Sam',6000);

Query OK, 1 row affected (0.16 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('Mike',5800);

Query OK, 1 row affected (0.16 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('Carol',4500);

Query OK, 1 row affected (0.17 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('Bob',4900);

Query OK, 1 row affected (0.20 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('David',5400);

Query OK, 1 row affected (0.27 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('Maxwell',5300);

Query OK, 1 row affected (0.21 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('James',4000);

Query OK, 1 row affected (0.19 sec)

mysql> insert into NthSalaryDemo(Name,Salary) values('Robert',4600);

Query OK, 1 row affected (0.19 sec)

Display all records from the table using select statement. The query is as follows:

mysql> select \*from NthSalaryDemo;

The following is the output:

+----+---------+--------+

| Id | Name    | Salary |

+----+---------+--------+

|  1 | Larry   |   5700 |

|  2 | Sam     |   6000 |

|  3 | Mike    |   5800 |

|  4 | Carol   |   4500 |

|  5 | Bob     |   4900 |

|  6 | David   |   5400 |

|  7 | Maxwell | 5300 |

|  8 | James   | 4000 |

|  9 | Robert  | 4600 |

+----+---------+--------+

9 rows in set (0.00 sec)

**Case 1:** Here is the query to get the nth highest value of a column.

The below query will give the fourth highest value of a column ‘Salary’:

mysql> select \*from NthSalaryDemo order by Salary desc limit 3,1;

The following is the output:

+----+-------+--------+

| Id | Name | Salary |

+----+-------+--------+

| 6 | David | 5400 |

+----+-------+--------+

1 row in set (0.00 sec)

**Case 2:** Here is the query to get the second highest value of a column ‘Salary’:

mysql> select \*from NthSalaryDemo order by Salary desc limit 1,1;

The following is the output:

+----+------+--------+

| Id | Name | Salary |

+----+------+--------+

| 3 | Mike | 5800 |

+----+------+--------+

1 row in set (0.00 sec)

**Case 3:**Here is the query to get the first highest value of a column:

mysql> select \*from NthSalaryDemo order by Salary desc limit 1;

The following is the output:

+----+------+--------+

| Id | Name | Salary |

+----+------+--------+

| 2 | Sam | 6000 |

+----+------+--------+

1 row in set (0.00 sec)

**Case 4:** If you want to get the 8th highest value of a column ‘Salary’, use the following query:

mysql> select \*from NthSalaryDemo order by Salary desc limit 7,1;

The following is the output:

+----+-------+--------+

| Id | Name  | Salary |

+----+-------+--------+

|  4 | Carol | 4500 |

+----+-------+--------+

1 row in set (0.00 sec)