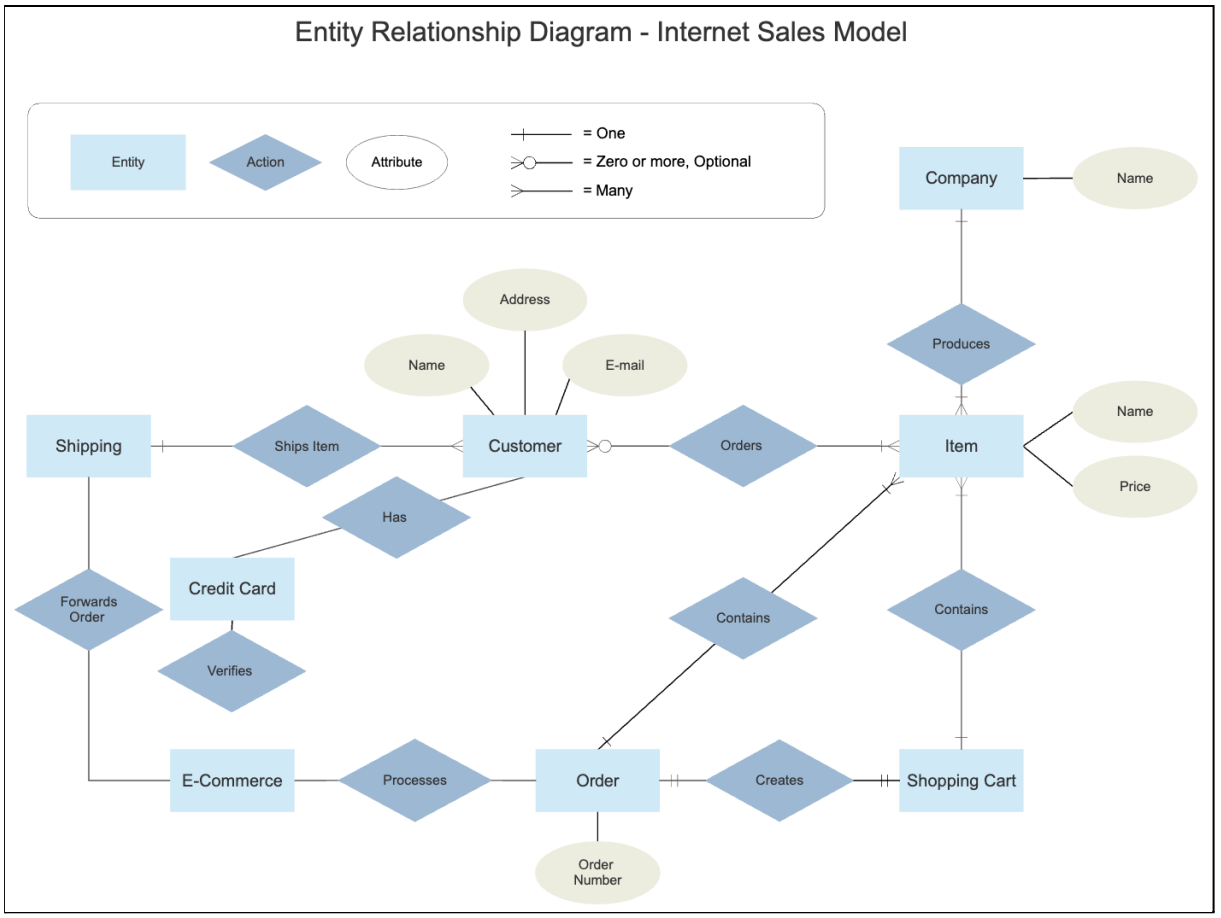
SQL Assignment 2

---

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

   - An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define their properties.
   - By defining the entities, and their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.
   - ER diagrams are used to sketch out the design of a database.
   - Let us understand from the example of ERD



   - Database object schema from the above ER diagrams

Customer

Customer ID **PK**	Name	Address	Email	Orders ID **FK**	Shipping ID **FK**	Credit card No **FK**

Item

Orders **ID PK**	Name	Price	Company ID **FK**	Shopping cart ID **FK**

SQL Assignment 2

Order

Orders **ID PK**	Order Number

Credit card

Varifies	Credit card No **PK**

Company

Name	Company ID **PK**

E-commerce

Oredr ID **PK**	Shipping ID **FK**

Shipping

Shipping ID **PK**	Order ID **PK**

Shopping cart

Orders **ID PK**	Shopping cart ID

Note:- Please check above ER diagrams schema and provide me the correct answer. if I'm wrong, because it may be wrong somehow.

2. **Create a SQL store process to register the use of the database, complete it with proper validation and transaction rollback and commit.**
   Ans- I didn't get the question.

3. **List the SQL aggregate function and demonstrate how to utilize it.**
   Ans-
   - In database management, an aggregate function is a function where the values of multiple rows are grouped together as input on certain criteria to form a single value of more significant meaning.
   - Here is a list of most use aggregate function:
     - 1) Count (SELECT COUNT(Name) FROM Customer)
     - 2) Sum(SELECT SUM(Total Price) FROM Sales)
     - 3) Avg(SELECT AVG(Volume) FROM Sales)
     - 4) Min(SELECT MIN(Volume) FROM Sales)
     - 5) Max(SELECT MAX(Volume) FROM Sales)

4. **In SQL, create a pivot query.**
   Ans-
   - **PIVOT** relational operator converts data from row-level to column level.
   - **PIVOT** rotates a table-valued expression by turning the unique values from one column in the expression into multiple columns in the output.

SQL Assignment 2

- Using the PIVOT operator, we can perform an aggregate operation where we need them.
- **Syntax of Pivot query**

```
SELECT <non-pivoted column>,
          <list of pivoted column>
FROM
(<SELECT query  to produces the data>)
    AS <alias name>
PIVOT
(
<aggregation function>(<column name>)
FOR
[<column name that  become column headers>]
    IN ( [list of  pivoted columns])

) AS <alias name  for  pivot table>
```

- **Example Query01**

```
SELECT [Year], Pankaj,Rahul,Sandeep FROM
(SELECT Name, [Year] , Sales FROM Employee )Tab1  PIVOT
(
SUM(Sales) FOR Name IN (Pankaj,Rahul,Sandeep)) AS Tab2
ORDER BY [Tab2].[Year]
```

Ans-

Results | Messages

	Year	Pankaj	Rahul	Sandeep
1	2010	93000	60500	84000
2	2011	45000	236100	82500
3	2012	32500	NULL	NULL

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

Ans-

- A 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:
  - INNER JOIN
  - LEFT JOIN
  - RIGHT JOIN
  - FULL JOIN

Let's consider two tables:-

SQL Assignment 2

Table Student

Roll_No	Name	Address	Age
1	Amit	Dehli	16
2	Shivam	Bihar	17
4	Shubham	Punjab	20

Table SCourse

Course_ID	Roll_No
1	1
2	2
3	9

- **INNER JOIN** keyword selects all rows from both the tables as long as the condition satisfies, value of the common field will be the same.
  Query
  - ```
    SELECT SCourse.Course_ID, Student.Name,
    Student.Age FROM Student
    INNER JOIN SCourse
    ON Student.Roll_No = SCourse.Roll_No;
    ```

  Output

Course_ID	Name	Age
1	Amit	16
2	Shivam	17

- **LEFT JOIN** returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join
  Query:
  - ```
    SELECT Student.Name, Student.Age
    SCourse.Course_ID,  FROM Student
    LEFT JOIN SCourse
    ON SCourse.Roll_No = Student.Roll_No;
    ```

  Output:

Name	Age	Course_ID
Amit	16	1
Shivam	17	2
Shubham	20	Null

SQL Assignment 2

- **RIGHT JOIN** is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join

  Query:

  - ```
    SELECT Student.Name, SCourse.Course_ID, FROM
    Student
    RIGHT JOIN SCourse
    ON SCourse.Roll_No = Student.Roll_No;
    ```

  Output:

Name	Course_ID
Amit	1
Shivam	2
Null	9

- **FULL JOIN** FULL JOIN creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables.

  Query:

  - ```
    SELECT Student.Name,Student.Roll_No
    SCourse.Course_ID, FROM Student
    RIGHT JOIN SCourse
    ON SCourse.Roll_No = Student.Roll_No;
    ```

  Output:

Name	Roll_No	Course_ID
Amit	1	1
Shivam	2	2
Shubham	4	Null
Null	Null	9

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

   Ans-

SQL Assignment 2

Query:

- ```
  CREATE TABLE STUDENT(
  Stu_ID INT NOT NULL,
  Stu_NAME VARCHAR(25),
  Stu_ROLL NO NVARCHAR,
  Stu_MARKS INT,
  PRIMARY KEY STUDENT(Stu_id)
  );
  INSERT INTO STUDENT VALUE(1, 'AMIT', 'A101', 567)
  INSERT INTO STUDENT VALUE(1, 'AMIT', 'A101', 450)
  INSERT INTO STUDENT VALUE(1, 'AMIT', 'A101', 467)
  INSERT INTO STUDENT VALUE(1, 'AMIT', 'A101', 589)
  INSERT INTO STUDENT VALUE(1, 'AMIT', 'A101', 347)
  INSERT INTO STUDENT VALUE(1, 'AMIT', 'A101', 290)
  ;
  ```

Query for getting 4th highest value.

```
1ST METHOD
SELECT * FROM (NAME, MARKS,
DENSE_RANK()OVER(ORDER BY MARKS DESC)R FROM
STUDENT)
WHERE R=&4;
2ND METHOD
SELECT NAME, MARKS FROM STUDEND S1 WHERE
4-1 = (SELECT COUNT(DISTINCT MARKS) FROM STUDENT
S2 WHERE S2.MARKS > S1.MARKS)
3RD METHOD
SELECT TOP 1 MARKS FROM ( SELECT DISTINCT TOP 4
MARKS FROM STUDENT ORDER BY MARKS DESC ) AS temp
ORDER BY salary
4RT METHOD
SELECT MARKS FROM STUDENT ORDER BY MARKS DESC
LIMIT 3,1
```
```