**SQL Assignment 2**

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

A-

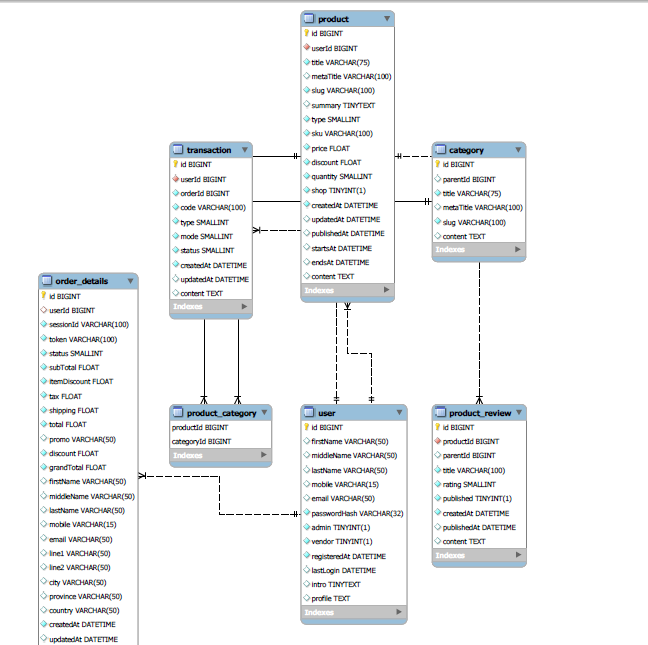
An Entity Relationship Diagram (ERD) is a snapshot of data structures. An Entity Relationship Diagram shows entities (tables) in a database and relationships between tables within that database. For a good database design it is essential to have an Entity Relationship Diagram.

There are three basic elements in ER-Diagrams:

Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.

Attributes are the data we want to collect for an entity.

Relationships describe the relations between the entities.



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

A-

These statements provide control over use of transactions:

* START TRANSACTION or BEGIN start a new transaction.
* COMMIT commits the current transaction, making its changes permanent.
* ROLLBACK rolls back the current transaction, cancelling its changes.
* SET auto commit disables or enables the default auto commit mode for the current session.

Scenario 1-with ‘commit’ on:

CREATE TABLE customer (a INT, b CHAR (20), INDEX (a));

Query OK, 0 rows affected (0.00 sec)

-- Do a transaction with autocommit turned on.

START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

INSERT INTO customer VALUES (30, 'Kiran');

Query OK, 1 row affected (0.00 sec)

COMMIT;

Query OK, 0 rows affected (0.00 sec)

Scenario 2-with ‘commit’ off:

mysql> SET autocommit=0;

Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO customer VALUES (15, 'Sam');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO customer VALUES (20, 'Phani');

Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM customer WHERE b = 'Kiran';

Query OK, 1 row affected (0.00 sec)

mysql> -- Now we undo those last 2 inserts and the delete.

mysql> ROLLBACK;

Query OK, 0 rows affected (0.00 sec)

1. List the SQL aggregate function and demonstrate how to utilize it.
2. An aggregate function allows you to perform a calculation on a set of values to return a single scalar value.

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 – calculates the sum of values.

1-AVG- SELECT

AVG(sales)

FROM

products;

2-SUM- SELECT

categoryid, SUM(sales)

FROM

products

GROUP BY categoryid;

3-COUNT- SELECT

COUNT(\*)

FROM

products;

4-MIN- SELECT

MIN(sales)

FROM

products;

5-MAX- SELECT

MAX(sales)

FROM

products;

1. In SQL, create a pivot query.

A-In SQL, pivot is used to convert rows to columns.

SELECT (ColumnNames)

FROM (TableName)

PIVOT

(

AggregateFunction(ColumnToBeAggregated)

FOR PivotColumn IN (PivotColumnValues)

) AS (Alias)

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

A-There are mainly three types of joins:

* Inner Join-Only the common elements are included.
* Outer Join-
* Full
* Left
* Right
* Cross Join-Cartesian join of the two tables.

Ex: Inner Join:

*select c.CustomerID,c.CustomerName,c.Country,o.OrderID,o.OrderDate from Customers c inner join Orders o*

*on c.CustomerID=o.CustomerID;*

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

*select customerid,totalcharges from customer\_churn\_telco\_cp*

*order by TotalCharges desc limit 3,1;*