**SQL Server**

**Table Of Contents**

[***Assignment 1: Login to Sql Server Management Studio*** 2](#_Toc79134939)

[***Assignment 2: Create Database and Tables - Demo*** 4](#_Toc79134940)

[***Assignment 3: Identity and Constraints- Demo*** 5](#_Toc79134941)

[***Assignment 4: Alter Table to Add new Column - Demo*** 8](#_Toc79134942)

[***Assignment 5: Insert Records into tables - Demo*** 8](#_Toc79134943)

[***Assignment 6: Delete/Update specific Record from table - Demo*** 10](#_Toc79134944)

[***Assignment 7: Select Statement with Limiting Rows with Between Operator - Demo*** 10](#_Toc79134945)

[***Assignment 8: Select Statement with Limiting Rows using Like Operator - Demo*** 12](#_Toc79134946)

[***Assignment 9: Select Statement with Limiting Rows using Like Operator - Demo*** 12](#_Toc79134947)

[***Assignment 10: Select Statement with Limiting Rows using Like Operator - Demo*** 13](#_Toc79134948)

[***Assignment 11: Select Statement with Limiting Rows using in Operator - Demo*** 13](#_Toc79134949)

[***Assignment 12: Select Statement with Limiting Rows using > Operator - Demo*** 14](#_Toc79134950)

[***Assignment 13: Select Statement with Limiting Rows using not in Operator - Demo*** 15](#_Toc79134951)

[***Assignment 14: Inner join - Demo*** 15](#_Toc79134952)

[***Assignment 15: Left join - Demo*** 16](#_Toc79134953)

[***Assignment 16: Right join - Demo*** 17](#_Toc79134954)

[***Assignment 17: Full join - Demo*** 17](#_Toc79134955)

[***Assignment 18: Self join - Demo*** 18](#_Toc79134956)

[***Assignment 19: Creating View - Demo*** 19](#_Toc79134957)

[***Assignment 20: Inserting Records using View - Demo*** 19](#_Toc79134958)

[***Assignment 21: Updating Records using View - Demo*** 20](#_Toc79134959)

[***Assignment 22: Deleting Records using View - Demo*** 21](#_Toc79134960)

[***Assignment 23: Inserting new record using Stored Procedure - Demo*** 21](#_Toc79134961)

[***Assignment 24 :Updating record using Stored Procedure - Demo*** 22](#_Toc79134962)

[***Assignment 25 :Deleting record using Stored Procedure - Demo*** 23](#_Toc79134963)

[***Assignment 26:Selecting records using Stored Procedure - Demo*** 23](#_Toc79134964)

[***Assignment 27 :Deleting records using Stored Procedure and out parameter - Demo*** 24](#_Toc79134965)

[***Assignment 28: StoredProcedure with try catch - Demo*** 25](#_Toc79134966)

[***Assignment 29: Transaction - Demo*** 26](#_Toc79134967)

[***Assignment 30: Drop column- Demo*** 28](#_Toc79134968)

***Assignment 1: Login to Sql Server Management Studio***

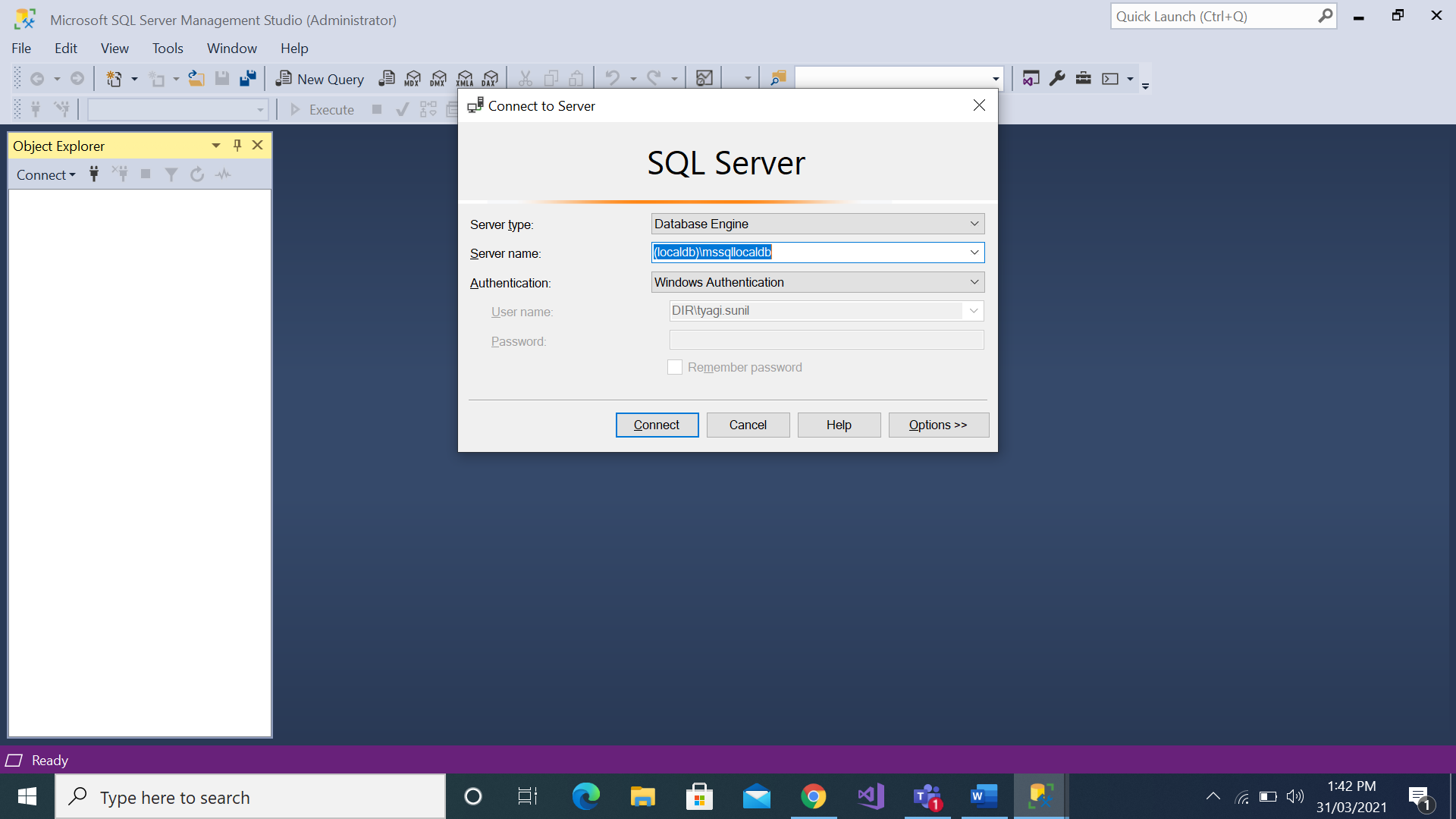
**NOTE: Below is an example of solved assignment**

**Objective:** login to sql server management studio to work with sql server database.

**Problem description:**  Check server name and credential details to login to sql server management studio.

**Step 1-** Open Sql server management studio

**Step 2-** It will ask for server name to connect to database engine and login credentials



**Step -3** Select Server Type – **Database Engine**

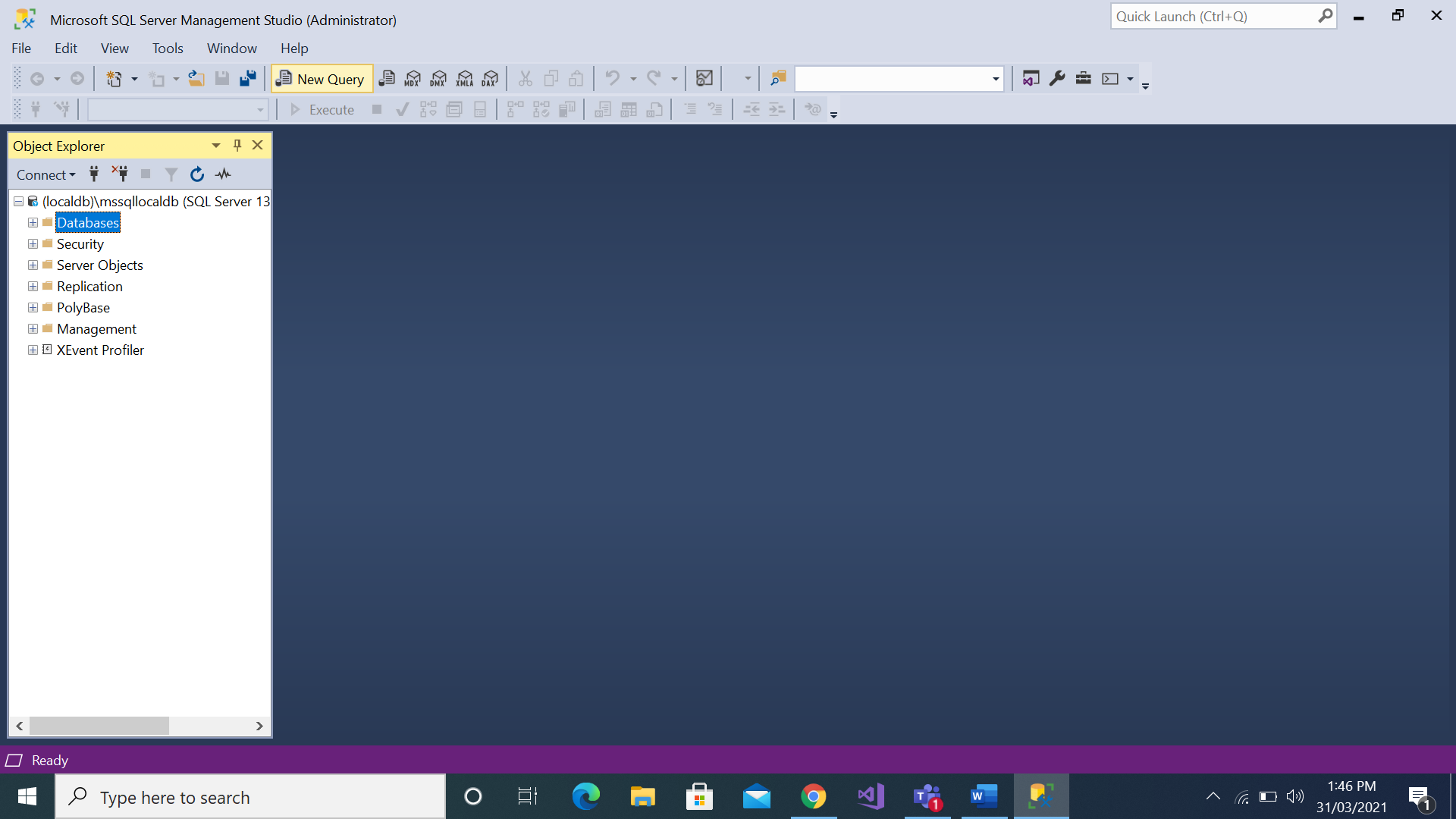
Server Name = **Your system name**

If using **MsSqlLocaldb** then ServerName=**Your System Name\MsSqlLocaldb**

then choose Authentication as windows Authentication.

**Step 4-** Click on connect.

**Step 5-** Click on New Query button to open query window as below.



***Assignment 2: Create Database and Tables - Demo***

**NOTE: Below is an example of solved assignment**

**Objective:**  Create Database and Tables.

**Problem description:**

We need to develop **Product Management System** .Net application for which we need to create one database and two tables Products and Categories.

**Products Table-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Data type** | **Constraint** | **Description** |
| ProductID | INT | Primary key | ID of the Product |
| ProductName | VARCHAR(100) |  | Name of the Product |
| Price | DECIMAL |  | Price of the Product |
| CategoryId | INT | Foreign Key | Category Id of the Product |

**Categories Table-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Data type** | **Constraint** | **Description** |
| CategoryID | INT | Primary key | ID of the Category |
| CategoryName | VARCHAR(50) |  | Name of the Category |

**Step 1**- Createdatabase **ProductDb**

**Step 2-** Create table **Categories**

(

CategoryID INT Primary key,

CategoryName VARCHAR(50)

)

**Step 3-** Create table **Products**

(

ProductID INT Primary key,

ProductName VARCHAR(100),

Price DECIMAL,

CategoryID INT Foreign key References Category(CategoryID)

)

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt to use Create table DDL operations.

***Assignment 3: Identity and Constraints- Demo***

**NOTE: Below is an example of solved assignment**

**Objective:**  Create a table as Customer with identity and constraints (not null,default,primary key,unique and check)using following schema.

**Customers Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **Data type** | **Constraint** | **Description** |
| ID | INT | Primary key | ID of the Customer |
| Name | VARCHAR(100) | Not Null | Name of the Customer |
| City | VARCHAR(100) | Default | City of The Customer |
| PanNo | VARCHAR(10) | Unique | PanNo Of Customer |

**Note**:Here Identity of Customer is Autoincrement which starts from 100 and incremented by 1 automatically as you insert new customer record in the table.

create table Customers

(

Id int identity(100,1) primary key,

Name varchar(100) not null,

City varchar(100) default 'Delhi',

PanNo varchar(10) unique

)

**Step 2:**

**1-Write the following Insert command to insert the records in Customer table**

insert into Customers(Name,PanNo) values('raj','AlIpt890C')

insert into Customers(Name,City,PanNo)values('raj','Noida','AlIpt890D')

**2. Write the following insert command to check not null constraint**

insert into Customers(name,City,PanNo) values(null,'chennai','alibt7690v')

**Note:**after executing above query,you will see following error:

Msg 515, Level 16, State 2, Line 3

Cannot insert the value NULL into column 'Name', table 'dbo.Customers'; column does not allow nulls. INSERT fails.

The statement has been terminated.

**3. Write the following insert command to check default constraint**

insert into Customers values('kalyan','alibt7690t')

**Note:**In this case, default constraint will not work,so in case of default constraint,change above query as follows:

insert into Customers(Name,PanNo) values('kalyan','alibt7690w')

**4. Write the following insert command to check unique constraint**

insert into Customers(Name,PanNo) values('kalyan','alibt7690w')

**Note:**after executing above query,you will see following error:

Msg 2627, Level 14, State 1, Line 4

Violation of UNIQUE KEY constraint 'UQ\_\_Customers\_\_F255F583F5560053'. Cannot insert duplicate key in object 'dbo.Customers'. The duplicate key value is (alibt7690w).

The statement has been terminated.

**5. Write the following insert command to check check constraint**

**5.1)**First add the check constraint for city “delhi” only.

alter table [CustomerDb].[dbo].[Customers]add constraint cust\_chk check(City='Delhi')

**5.2)**Write the following insert command :

insert into Customers(Name,City,PanNo)values('pratap','Noida','Alipt890A')

**Note:**After executing above query,you will see following error:

Msg 547, Level 16, State 0, Line 9

The INSERT statement conflicted with the CHECK constraint "cust\_chk". The conflict occurred in database "CustomerDb", table "dbo.Customers", column 'City'.

The statement has been terminated.

Completion time: 2021-08-03T18:07:26.9441615+05:30

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt to use Identity,primary key,not null,default ,unique and check constraints.

***Assignment 4: Alter Table to Add new Column - Demo***

**NOTE: Below is an example of solved assignment**

**Objective:**  Add new Column in the Product Table using Alter command.

alter table Products add ProductDescription varchar(500)

**Note:**after executing above command,a new column **ProductDescription** will be added into Product table.

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt to use Alter table DDL operations.

***Assignment 5: Insert Records into tables - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** We need to insert records in above created tables Product and Category using sql queries. Insert 2 related records in each table satisfying Primary key and foreign key relation and one unrelated record.

**Step 1-** Insert 3 records in Category table

INSERT INTO Categories

(CategoryID,CategoryName)

VALUES

(1,'Mobile'),(2,'Laptop'), (3,'Tv')

**Step 2-** Insert 2 related records in Products table

INSERT INTO Products

(ProductID,ProductName, Price,CategoryID )

VALUES

(101,'Samsung Mobile',8000,1),(102,'Hp Laptop',20000,2)

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to use Insert DML operation.

***Assignment 6: Delete/Update specific Record from table - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** How to delete the record with ProductId 101 and Update the CategoryName to Stationary where CategoryId is 1

**Step 1-** delete 1 record from product table

delete from Products where ProductID=101

**Step 2-** update 1 record and 1 column in Categories table

update Categories set CategoryName='Stationery' where CategoryID=1

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to use delete/update DML operation.

***Assignment 7: Select Statement with Limiting Rows with Between Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductId,ProductName and Price from Products table whose Product id is between 102 and 104 and records should be in descending order. So Insert some more records to apply Between operator.

**Step 1- Insert 3 records in Products table**

insert into Products values(103,'Nokia mobile',7000,1),

(104,'Samsung Laptop',30000,2),

(105,'Lenovo Laptop',28000,2)

**Step 2 -Select statement**

select ProductID, ProductName,Price from Products where ProductID between 102 And 104 order by ProductID Desc

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause and between operator and sorting.

***Assignment 8: Select Statement with Limiting Rows using Like Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductID,ProductName ,Price from Products table whose Product Names contains letter “H”

**Select query-**

select \* from Products

where ProductName like '%H%'

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause with LIKE keyword.

***Assignment 9: Select Statement with Limiting Rows using Like Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductName ,Price from Products table whose Product Names contains second letter as “o”

**Select query-**

select ProductName, Price from products where productName like '\_o%'

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause with LIKE keyword.

***Assignment 10: Select Statement with Limiting Rows using Like Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductName ,Price from Products table whose Product Names contains ‘mobile’ word.

**Select query-**

select ProductName, Price from products where productName like '%mobile%'

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause with LIKE Opearator.

***Assignment 11: Select Statement with Limiting Rows using in Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductName ,Price from Products table whose product price is 2000,3000,and 4000

**Select query-**

select ProductName, Price from products where Price in(2000,3000,4000)

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause with in operator.

***Assignment 12: Select Statement with Limiting Rows using > Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductName ,Price from Products table whose product price is greater than 30000.

**Select query-**

select ProductName, Price from product where Price >30000

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause with > operator.

***Assignment 13: Select Statement with Limiting Rows using not in Operator - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Select the ProductName ,Price from Products table whose product price is not equal to 30000,40000 and 50000.

**Select query-**

select ProductName, Price from products where Price Not in (30000,40000,50000)

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to select specific columns from existing table with limiting rows by Where clause with not in operator.

***Assignment 14: Inner join - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:**Write a query to display the Product Names and their respective Category names.

select p.ProductName,c.CategoryName

from Products p

inner join

Categories c

ON p.CategoryID=c.CategoryID

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to use inner join and find matching records from 2 tables using primary and foreign key relation.

***Assignment 15: Left join - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:**Write a query to display the Category Name for those you do not have products in product table.

select \* from categories c

left join products p

on c.categoryid=p.categoryid

where p.categoryid is null

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to use left join and found matching records from right table and all records from left table.

***Assignment 16: Right join - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:**Write a query to display the Products for those you do not have Category Id in Categories table.

select \* from categories c

right join products p

on c.categoryid=p.categoryid

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to use right join and found matching records from left table and all records from right table.

***Assignment 17: Full join - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:**Write a query to display the All Products with and without category.

select \* from categories c

full join products p

on c.categoryid=p.categoryid

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to use full join and found all matching and nonmatching records from both the tables.

***Assignment 18: Self join - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective**:Write a query to display the Employee,EmployeeId,ManagerId and Manager using Self Join.

**Step1:** Create the following EmployeeDB database and Employees table.

* 1. Create database EmployeeDb

Use EmployeeDB

* 1. create table Employees(Id int primary key,Name varchar(100),ManagerId varchar(100))

**Step 2**:Insert following records in Employees Table.

insert into Employees values

(101,'Ram',null),

(102,'Shyam',103),

(103,'Mohan',105),

(104,'Kapil',101),

(105,'Raj',102),

(106,'Anil',103)

**Step 3**:Write following self inner join query which will display Employee Id,Employee Name with corresponding ManagerId ,Manager Name.

select e.Id as EmployeeId,e.Name as Employee,m.id as ManagerId,m.Name as Manager from Employees e

inner join Employees m

on e.ManagerId=m.Id

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to use self join and found all Employees with their Managers.

***Assignment 19: Creating View - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a view to fetch Product Names and the Price.

**Step 1-**

create view viewProductPrice

as select productName,Price from

products

**Step 2-**

select \* from viewProductPrice

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create view to display products and their respective prices.

***Assignment 20: Inserting Records using View - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a view to insert a new record into product table.

**Step 1-**

create view viewProducts

as select \* from products

**Step 2-**

insert into viewProducts values(107,'blackberry mobile',12000,1)

**Step 3-**

select \* from viewProducts

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create view to insert a new record into product table.

***Assignment 21: Updating Records using View - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Update Price of a product on the basis of product Id using viewProducts view.

**Step 1-**

update viewProducts set Price=12500 where productId=107

**Step 2-**

select \* from viewProducts

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create view to update a record in product table.

***Assignment 22: Deleting Records using View - Demo***

**NOTE: Below is an example of solved assignment.**

Objective: Create a view to delete a product on the basis of product Id .

**Step 1-**

delete from viewProducts where productId=107

**Step 2-**

select \* from viewProducts

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create view to delete a record from product table.

***Assignment 23: Inserting new record using Stored Procedure - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective**: Create a stored procedure to insert a new product into products table.

**Step 1-**

create proc spInsertProduct @productId int,@productName varchar(100),@price decimal,@categoryId int

as

begin

insert into products(productID,productName,Price,categoryid) values

(@productId,@productName,@price,@categoryId)

end

**Step 2-**

exec spInsertProduct 110,'lg mobile',5000,1

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure to insert a new product in products table.

***Assignment 24 :Updating record using Stored Procedure - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a stored procedure to update product price on the basis of product ID.

**Step 1-**

create proc spUpdatePrice @productId int,@price decimal

as

begin

update products set Price=@price where productId=@productId

print 'Price Updated Successfully'

end

**Step 2-**

exec spUpdatePrice 107,'14000'

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure to update product price in products table.

***Assignment 25 :Deleting record using Stored Procedure - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a stored procedure to delete a product on the basis of productId.

**Step 1-**

create proc spDeleteProduct @productId int

as

begin

delete from products where productId=@productId

print 'Product Deleted Successfully'

end

**Step 2-**

exec spDeleteProduct 108

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure to delete product from products table.

***Assignment 26:Selecting records using Stored Procedure - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a stored procedure to select a product on the basis of product ID.

**Step 1-**

create proc SpSelectProduct(@productId int)

as

begin

select \* from products where productid=@productId

end

**Step 2-**

exec SpSelectProduct 101

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure to select a product from products table.

***Assignment 27 :Deleting records using Stored Procedure and out parameter - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a stored procedure to delete a product on the basis of product ID using out parameter.

**Step 1-**

Create proc SpDeleteProductOut(@productId int,@message varchar(500) out)

as

begin

if exists(select productid from products where productId=@productId)

begin

delete from products where productid=@productId

set @message='Product Deleted!'

return 1

end

else

begin

set @message='Product Id Does Not Exist'

return -1

end

end

**Step 2-**

declare @message varchar(500)

exec spDeleteProduct 104,@message out

print @message

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure using out parameter to delete a product from products table.

***Assignment 28: StoredProcedure with try catch - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a stored procedure to insert a new product into products table using try catch for exception handling

**Step 1-**

create proc spInsertProduct(@productId int,@productName varchar(100),@price decimal,

@categoryId int)

as

begin

begin try

insert into products(productID,productName,productPrice,categoryid) values

(@productId,@productName,@price,@categoryId)

print 'Product Inserted SuccessFully'

return 1

end try

begin catch

select ERROR\_MESSAGE() as ErrorMessage

return -1

end catch

end

**Step 2-**

DECLARE @return\_value int

EXEC @return\_value = spInsertProduct

@productId = 108,

@productName = 'android mobile',

@price = 12000,

@categoryId = 1

SELECT 'Return Value' = @return\_value

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure with exception handling to insert a new product in products table.

***Assignment 29: Transaction - Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** Create a stored procedure to insert a new product into products table as a transaction which is committed only if the price of mobile is greater than INR 2000 otherwise rollback the transaction.

**Step 1:**

create proc SpTransaction(@productId int,@productName varchar(100),

@price decimal,@categoryId int)

as

begin

begin transaction

if(@price>=2000 and @categoryId=1)

begin

insert into products(productID,productName,Price,categoryid) values

(@productId,@productName,@price,@categoryId)

print 'Product Inserted Successfully'

commit transaction

end

else

begin

print 'Product price can not be below 2000'

rollback transaction

end

end

**Step 2:**

exec SpTransaction 110,'lg mobile',5000,1

**Estimated time:** 10 mins

**Summary of this assignment:** You have learnt how to create stored procedure with transaction to insert a new product in products table.

***Assignment 30: Drop column- Demo***

**NOTE: Below is an example of solved assignment.**

**Objective:** How to drop column from Products table.

alter table Products drop column Price

**Estimated time:** 5 mins

**Summary of this assignment:** You have learnt how to drop column in existing table.