**ASSIGNMENT-1**

**Drive link: https://drive.google.com/file/d/19PTE\_CnGoA-m01FTo4OxXQcY0nXRpvwI/view?usp=sharing**

-- create

CREATE TABLE customer (

custid int primary key,

custname varchar(20),

age tinyint,

caddress varchar(40)

);

CREATE table [order] (

orderid tinyint primary key,

custid int,

orderdate date,

product varchar(20),

price int,

quantity int

foreign key(custid) references customer(custid)

);

-- inserting rows into customer table

INSERT INTO customer(custid,custname,age,caddress) VALUES (1, 'alien',12, 'hyd');

INSERT INTO customer(custid,custname,age,caddress) VALUES (2, 'sache', 22,'delhi');

INSERT INTO customer(custid,custname,age,caddress) VALUES (3, null,55, 'j&k');

--how to insert a record with multiple rows

INSERT into customer VALUES(200,'jay',24,'blr'),(300,'jayaa',24,'hyd'),(400,'ajay',24,'che'),(4,'potti',23,'blr'),(500,'bravo',23,'blr'),(5,'jay',24,'hyd'),(6,'raj\_kumar',24,'hyd');

--how to insert a record to specific column

insert into customer VALUES(600,'giri',null,null);

--(or)

insert into customer(custid,custname) VALUES (700,'mani');

--hoe to update the record

update customer set age=33,caddress='jaipur' where custid=2

--if dont specify where custid then for all the custid the age and caddress will set to 33 and jaipur

--update customer set age=33,caddress='jaipur'

--how to delete the record

delete FROM customer where custid=300

--inserting rowa into ordertable

insert into [order] values (10,1,'2000-01-01','phone',30000,5),

(11,1,'2002-02-01','watch',90000,800),

(12,2,'2000-03-01','rings',30,1556),

(13,4,'2000-04-01','headset',900,85),

(14,3,'2025-05-01','bottle',600,1400),

(15,5,'2025-05-01','bottle',null,1400),

(16,5,'1990-12-01','pens',4500,14);

SELECT \* from [order];

-- fetch

SELECT \* FROM customer;

SELECT custid from customer;

GO

--1. Display the list of customers who resides in Bangalore

SELECT \* from customer where caddress='blr';

--2. Display the list of customers who does not resides in Bangalore or chennai

SELECT \* from customer where caddress!='blr' and caddress!='che';

SELECT \* from customer WHERE caddress NOT IN ('blr', 'che');

--3. Display the list of customers who’s age is greater then 50 and does not resides in Bangalore

SELECT \* from customer where age>50 and caddress!='blr';

--4. Display the list of customers who’s name starts with A

SELECT \* from customer where custname like 'A%';

--5.Display the list of customers who’s name contains a word Br

SELECT \* from customer where custname like 'br%'

--6.Display the list of customer who’s name start between a to k

SELECT \* FROM customer

where custname like '[A-k]%';

--7.Display the list of customers who’s name is 5 character long

SELECT \* FROM customer where custname like '\_\_\_\_\_%';

--(or)

SELECT \* from customer where len(custname)=5;

--8.Display the list of customer who’s name a. Start with s b. Third character is c c. Ends with e

SELECT \* FROM customer where custname like 's%' and custname like '\_\_c%' and custname like '%e';

--(or)

SELECT \* from customer where custname like 's\_c%e';

--9.Display unique customer names from customers table

SELECT distinct custname from customer;

--10.List orders details where qty falling in the range 100-200 and 700-1200

SELECT \*

FROM [order]

WHERE quantity BETWEEN 100 AND 200

OR quantity BETWEEN 700 AND 1200;

--11. List customer details where custname beginning with AL and ending with N

SELECT \* FROM customer where custname like 'al%' and custname like '%n';

--12.Display what each price would be if a 20% price increase were to take

--place. Show the custid , old price and new price ,using meaningful headings(use orders table)

SELECT custid,product,price as old\_price ,(0.2)\*price as new\_price

from [order]

where price is not null;

--13.Display top 3 highest qty from orders table

SELECT top 3 \*

from [order]

where price is not null

order by quantity desc;

--14.Display how many times customers have purchased a product (display count and customerid from orders table)

SELECT custid,count(custid) as purchase\_often from [order] group by custid;

--15.Display the list of orders who’s orders made earlier then 5 years from now

SELECT \* from [order] where orderdate < dateadd(year,-5,getdate());

SELECT \* from [order] where year(getdate()) - year(orderdate) > 5;

--16.Select \* from customers where custname is null

Select \* from customer where custname is null;

--17. Display orderdetails in following format

--OrderID-Date Total(price\*qty)

--100-1/1/2000 500

Select cast(orderid as varchar)+ '-' +cast(orderdate as varchar) as orderid\_date,(price\*quantity) as total from [order] where price is not null;

Select concat(orderid,'-',orderdate),price\*quantity as total from [order];

--18. Update orders table by decreasing price by 20% for qty > 50

update [order] set price = price - (0.2\*price) where price is not null and quantity>50;

Select \* from [order] where price is not null;

--display the record which cotains underscore(\_) in their name

Select \* from customer where custname like '%[\_]%';

--19.You want to retrieve data for all the orders who made order '1-12-90' having price 4000 – 6000 and sort the column in descending order on price

Select \*

from [order]

where orderdate='1990-12-01' and price BETWEEN 4000 and 6000 and price is not null

order by price desc;

--20.Display order details in following format

--Custid Price (sum of price) Count (count of qty)

--1 5000 3

--2 4000 9

--3 6700 6

Select custid,sum(price) as price\_sumofprice,count(quantity) as count\_qty

from [order]

group by custid;

--21.Display above details only for price > 4000

Select custid,sum(price) as price\_sumofprice,count(quantity) as count\_qty

from [order]

group by custid

having sum(price)>4000;

--22.Write a query to create duplicate table of customer , and name it as custhistory

--a. Delete all the records of custhistory

--b. Copy records of customers to custhistory where age > 30

Select \* into custhistory from customer ;

Select \* from custhistory;

--a. Delete all the records of custhistory

Delete from custhistory;

Select \* from custhistory;

--b Copy records of customers to custhistory where age > 30

insert into custhistory

Select \* from customer where age>30;

Select \* from custhistory;

**ASSIGNMENT-2**

Drive link: [SubQueries and joins.pdf - Google Drive](https://drive.google.com/file/d/19iAsUqkyTEHHAU1qvLKpVYipTtiMuiuY/view)

-- create

CREATE TABLE customer (

custid int primary key,

custname varchar(20),

age tinyint,

caddress varchar(40)

);

CREATE table [order] (

orderid tinyint primary key,

custid int,

orderdate date,

product varchar(20),

price int,

quantity int

foreign key(custid) references customer(custid)

);

-- inserting rows into customer table

INSERT INTO customer(custid,custname,age,caddress) VALUES (1, 'alien',12, 'hyd');

INSERT INTO customer(custid,custname,age,caddress) VALUES (2, 'sache', 22,'delhi');

INSERT INTO customer(custid,custname,age,caddress) VALUES (3, 'sony',55, 'j&k');

--how to insert a record with multiple rows

INSERT into customer VALUES(200,'jay',24,'blr'),(300,'jayaa',24,'hyd'),(400,'ajay',24,'che'),(4,'potti',23,'blr'),(500,'bravo',23,'blr'),(5,'jay',24,'hyd'),(6,'raj\_kumar',24,'hyd');

--how to insert a record to specific column

insert into customer VALUES(600,'giri',null,null);

--(or)

insert into customer(custid,custname) VALUES (700,'mani');

--hoe to update the record

update customer set age=33,caddress='jaipur' where custid=2

--if dont specify where custid then for all the custid the age and caddress will set to 33 and jaipur

--update customer set age=33,caddress='jaipur'

--how to delete the record

delete FROM customer where custid=300

--inserting rowa into ordertable

insert into [order] values (10,1,'2000-01-01','phone',30000,5),

(11,1,'2002-02-01','watch',90000,800),

(12,2,'2000-03-01','rings',30,1556),

(13,4,'2000-04-01','headset',900,85),

(14,3,'2025-05-01','bottle',600,1400),

(15,5,'2025-05-01','bottle',100,1400),

(16,5,'1990-12-01','books',600,14),

(17,5,'1990-12-01','books',200,14),

(18,6,'1990-12-01',null,4500,14),

(19,6,'1990-02-12',null,4500,14);

SELECT \* from [order];

-- fetch

SELECT \* FROM customer;

SELECT custid from customer;

GO

--Subqueries

--1.display all the records from customers who made a purchase of books

SELECT \*

from customer

where custid in(SELECT custid

from [order]

where product='books');

--2.display all the records from customer who made a purchase of books ,phone,headset

SELECT \*

from customer

where custid in(SELECT custid

from [order]

where product in('books','phone','headset'));

--3.display the list of customers who never made any purchase

SELECT \*

from customer

where custid in(SELECT custid

FROM [order]

where product is null);

--4. display the second highest age from customers (do not use top keyword)

SELECT max(age) as second\_highest\_age

from customer

where age<(SELECT max(age)

from customer);

--5. display the list of customer who's age is greater then ajay's age ( but we dont know ajay's age)

SELECT custname

from customer

where age > (SELECT age

from customer

where custname='ajay');

--6.update customer table where custid =100's age = custid=200's age

UPDATE customer

SET age = (

SELECT age

FROM customer

WHERE custid = 200

)

WHERE custid = 500;

SELECT \* FROM customer WHERE custid IN (500, 200);

--7. Display those orders who made orders in December of any year

SELECT \*

from [order]

WHERE orderdate like '\_\_\_\_-12-\_\_';

-- (or)

SELECT \*

from [order]

WHERE month(orderdate)=12;

--8. Show all orders made before first half of the month (before the 16th of the

--month who does not reside in bangalore).

SELECT \*

from [order]

WHERE day(orderdate) between 1 and 15 and custid not in (SELECT custid

from customer

WHERE caddress in ('blr'));

--9. Display list of customers from delhi and Bangalore who made purchase of less than 3 product

SELECT \*

from customer

where caddress in('hyd','blr') and custid in (SELECT custid

from [order]

group by custid

having count(custid)<3);

--10.Display list of orders where price is greater than average price

SELECT \*

from [order]

where price >(SELECT avg(price)

from [order] );

--11.Write a query to display custid , custname who have purchased > 2 products with sum of price < 1000

SELECT custid,custname

from customer

WHERE custid in (SELECT custid

from [order]

group by custid

having sum(price)<1000 and count(custid)>2);

--12.Write a query to display list of customers, where month of orderdate=month of dob

-- SELECT \*

-- from customer

-- where month(dob) = month(orderdate)

-- and custid in (SELECT custid

-- from [order]);

--13.update table increasing price by 10% for customers residing in Bangalore

UPDATE [order]

set price = (0.1)\*price

where custid in (SELECT custid

FROM customer

where caddress='blr');

--joins

--1.Write a query to display the custid ,name, age, orderid,orderdate for

--product books or cd, sort data by name in descending order

MS.NET Framework Fundamentals

• What is Microsoft .NET?

• .Net Framework

• .Net Core Framework

• .Net Framework vs .Net Core

• Common Language Runtime

• Metadata

• Common Type System

• Framework Class Library

• Language Interoperability

C# First Program

• First .Net Application using Visual Studio 2022

• Command Line Arguments

• Return Value of Main

• Using Command Line Compiler

• Using VS Code for building .Net Core Applications

• Creating .Net Core Applications using VSCode

C# Language Syntax • C# Introduction and Evolution

Classes and Structures

• Data Types

• Value Types and Reference Types

• Implicit and Explicit Casting

• Programming Constructs

C# Language Syntax

• Boxing & Unboxing

• Operators

• Control Statements

• Working with Arrays

• Multi-Dimensional Arrays

• Jagged Arrays

• Nullable Types

• Ref and Out Parameters

• Unsafe Code

Overview of Gang of Four

(GoF) Design Patterns

• Creational Patterns

• Structural Patterns

• Behavioral Patterns

• Implementing GoF Patterns in C#

• Understanding the implementation of GoF Patterns in

.Net Library / APIs

• Consequences of using Design Patterns

Getting Started withOOP

• Writing Classes & Initializing Objects

• Access specifiers

• Writing methods in Classes

• Working with Properties in Class

• Constructors and Destructors

• Parameterized Constructors

• Copy Constructors in C#

• Mutable & Immutable types

• Singleton Pattern in C#

Static keyword

• Static Classes

• Static Constructors

• Static variables

• Static Members

• Static vs non static

Implementing Inheritance

in C#

• Protected Keyword and Constructors In Inheritance

• Casting Between Reference Types

• Static And Dynamic Binding

• Abstract Class & Methods

• Object Class As Parent

• Single Inheritance

• Multi vs Multi level Inheritance

• Var and dynamic keyword

• Stopping Inheritance using sealed keyword

• Sealed Classes

• Abstract Factory Pattern

• Factory Method Pattern

6 Implementing Interfaces

& Polymorphism

• Polymorphism And Syntax Of Interface

• Explicit Implementation & Casting

• Types Of Interfaces

• Method overloading

• Method Overriding

• Virtual keyword

• Late binding vs early binding

• Runtime polymorphism

• Façade Pattern

C# Programming

Constructs

• Partial Classes

• Extension methods

• Collection Initializers

• Object Initializers

• Nullable Types

• Enums

• Tuples

• Const keyword

• Readonly keyword

• Anonymous Types

Collections and Generics

• Introduction To Collection Classes

• ArrayList

• HashTable

• Dictionary

• Stack

• Queue

• LinkedList

• BinaryTree

• IEnumerable, IComparable And IComparer Interface

• Indexers

• Writing Generic Classes & Methods

• Generic Constraints

• Generic Delegates

• Generic Interfaces

• Generic Collection Classes

Assemblies and GAC

• Assemblies

• Public and Private Assemblies

• Class Library

• Shared Assemblies And GAC

LINQ

• LINQ

• Linq Operators

• Query Expressions

• Lambda Expressions

• IQueryable interface

• PLinq

Course Design Document

Exception Handling

• What are Exceptions?

• Try & Catch Blocks

• Throw And Finally Keywords

• Writing Custom Exceptions

• Global Exceptions

Memory Management

• Garbage Collection

• Mark-Sweep Algorithm

• Finalizers

• IDisposable Interface

• Dispose method

String Operations

• Handling Strings

• String Operations

• String Builder

• Builder Design Pattern

Regex

• RegularExpressions

• Regex Class

• Match method

Windows Forms

• Creating Windows Forms

• Working with Controls like Textboxes, Buttons,

Listboxes,Menus etc.

IO Streams

• What are Streams & Types Of Streams

• Standard IO Streams

• Dealing With FileStreams

• Binary Reader & Binary Writer

• TextReader, TextWriter classes

• Working With File System

• Directory

• Path

• MemoryStream

Reflection and Attributes

• Reflection

• Reflection And Attributes

• Pre-Defined Attributes

• Custom Attributes Include Invoking members using

reflection with binding options

Delegates & Events

• Declaring And Using Delegates

• Singlecast & Multicast Delegates

• Anonymous Delegates

• Covariance and Contravariance

• Async Callbacks

• Declaring & Handling Custom Events

Multi-Threading

• Multithreading Overview

• Programming Threads

• Thread Priority

• Suspend Resume Interrupt and Cross Thread

Operations

• Background & foreground threads

Course Design Document

• Thread Pool

• Sync Using Monitor

• Sync Using Mutex

• Lock statement

• Sync Using Semaphore & Events

Task Parallel Library (TPL)

• Creating and working with Tasks

• async-await

• Task Continuation

• Parallel.For and Parallel.ForEach

• Task.Factory and TaskCreationOptions

• TaskSchedulers

• Handling Exceptions in TPL

• CancellationToken

Serialization and

Deserialization

• Need for Serialization

• XML Serialization

• Binary Serialization

• JSON Serialization

• Controlling Serialization

• Memento Design Pattern