Movies DDL

create table Movies(

id int Primary Key,

Title varchar(100),

Director varchar(100),

Year int,

Length\_Minutes int)

GO

Create Table BoxOffice(

Movie\_id int,

Rating float,

Domestic\_sales bigint,

International\_sales bigint)

GO

ALTER TABLE Movies ALTER COLUMN Title varchar(100) NOT NULL

GO

ALTER TABLE Movies ALTER COLUMN Director varchar(100) NOT NULL

GO

ALTER TABLE Movies ALTER COLUMN Year int NOT NULL

GO

ALTER TABLE Movies ALTER COLUMN Length\_Minutes int NOT NULL

GO

ALTER TABLE Movies ADD CONSTRAINT UQ\_Title UNIQUE(Title)

GO

ALTER TABLE BoxOffice

ADD CONSTRAINT FK\_\_Boxoffice\_Movies

FOREIGN KEY (Movie\_id)

REFERENCES Movies (id)

GO

**Lodging DRL**

select\*from room\_type

select\*from rate

select\*from room

select r.description as "Room Type",count(r1.room\_type\_id) as "No of Booking"

from room\_type r

inner join room r1

on(r.room\_type\_id=r1.room\_id)

group by r1.room\_type\_id,r.description

order by r1.room\_type\_id desc

go

**Lodging procedure**

create procedure usp\_GetBookingCost

as

begin

declare @booking\_date datetime

select a.room\_id as [Room Number],b.room\_type\_id as [Room Type],sum(c.amount) as [Total Booking Cost]

from room a,room\_type b,rate c

join rate c on

b.room\_type\_id=c.room\_type\_id

join room a on

a.room\_type\_id=b.room\_type\_id

where @booking\_date=booking.booking\_date

group by room\_type\_id

end

GO

////////-------employee count per department------/////

Create procedure EmployeeCount

(

@deptno char(3),

@total\_employees int output

)

As

Begin

select @total\_employees = count(empno) from employee where workdept = @deptno

Return @total\_employees

End

Go

////////////-----------cricket player club- sql server------///////

select p.PlayerName as Player,pr.RoleName as rolename from player p join player\_role pr on

p.RoleId=pr.RoleId

go

select PlayerName as players\_with\_no\_roles from player where RoleId is null

go

select pr.RoleName as role\_with\_no\_roles from player\_role pr

where pr.RoleId not in (select p.RoleId from player as p join player\_role as pr

on pr.RoleId=p.RoleId)

go

///////////-------display the department name and total salary as sum of salary where employ count greater than 5...--------//

Select sum(e.salary) as "TOTALSALARY" ,d.deptname from employee e

INNER JOIN department d ON e.workdept = d.deptno

group by d.deptname having count (d.deptno) > 5

order by sum(e.salary);

/////////--------retrieve employee details-----/////

create procedure RetrieveEmpDetails (@EmpNo char(6), @Name varchar(15) out, @Job char(9) out, @Sal decimal(9,2) out)

as

Begin

select @Name=lastname, @Job=job,

@Sal=salary from Employee where empno = @EmpNo

End

GO

//////-----Extract the employees and their department salaries----//

select emp.firstname,emp.salary,tbl.avg\_sal from employee e join

(select workdept,avg(salary) as avg\_sal from employee group by workdept)

as tbl on emp.workdept=tbl.workdept and emp.salary>tbl.avg\_sal

go

//////////----------nirav bank--------////////////////

//////--------Display employees who earn more than the avg of their dept---///

Select emp.firstname, emp.salary, tbl.avg\_sal from employee e join

(Select workdept, avg(salary) as avg\_sal from employee group by workdept) as tbl on emp.workdept=tbl.workdept and emp.salary>tbl.avg\_sal

GO

/////--------PROCEDURE TO RETRIEVE THE PLAYER DATA----///////

create PROC DisplayPlayer @rolename varchar(30) as

Begin

select playername from Player join Player\_Role on

(Player\_Role.RoleID=Player.RoleID and @rolename=rolename)

End

go

//////////---------- Extract the employee and salary grades with department names-------/////

select e.firstname,d.deptname, e.salary,

case

when e.salary>50000 then 'HIGH'

when e.salary between 25000 and 50000 then 'MEDIUM'

else 'LOW'

end as sal\_grade

from Employee e, Department d

where e.workdept = d.deptno

GO

//////////////--------------Display the first-name,lastname and deptname of employees who have worked more than 50yrs------////////

/////////--------ACMEKart LOGISTICS please---------////////

create table Customer(

CustomerID int not null,

LastName varchar(50) not null,

FirstName varchar(50) not null,

Email varchar(100) not null,

Phone varchar(10) not null,

AddressLine1 varchar(50) not null,

AddressLine2 varchar(50) not null,

City varchar(50) not null,

State varchar(50) not null,

PostalCode varchar(10) not null,

Country varchar(50) not null,

EmailVerified char(1) not null

)

create table Product(

ProductID int not null,

ProductDesc varchar(50) not null,

VendorDesc varchar(50) not null,

Rating int not null,

SellingPrice decimal(9,2) not null,

Discount int not null

)

create table Orders(

OrderID int not null,

CustomerID int not null,

OrderDate date not null,

ShipmentDate date not null,

Status char(1) not null,

Comments varchar(50)

)

create table ProdOrdCount(

ProdOrdID int not null,

OrderID int not null,

ProductID int not null,

Quantity int not null

)

Alter table Customer

add constraint pk\_Customer primary key(CustomerID)

alter table Product

add constraint pk\_Product primary key(ProductID)

alter table Product

add constraint ck\_Rating Check(Rating >=0 and Rating <=5)

alter table Orders

add constraint pk\_Orders primary key(OrderID)

alter table Orders

add constraint fk\_Orders\_Customer foreign key (CustomerID) references Customer(CustomerID)

alter table ProdOrdCount

add constraint pk\_ProdOrdCount primary key(ProdOrdID)

alter table ProdOrdCount

add constraint fk\_ProdOrdCount\_Product foreign key(ProductID) references Product(ProductID)

alter table ProdOrdCount

add constraint fk\_ProdOrdCount\_Orders foreign key(OrderID) references Orders(OrderID)

go

create proc usp\_PetProcedureDetails(@ProcedureType varchar(255)) as

begin

select p.PetID as petid, p.Name as [Pet Name],o.Name as [Owner Name],h.Description as description FROM Pet p join PetOwner o on

p.OwnerID=o.OwnerID join ProcedureHistory h ON p.PetID=h.PetID where

h.ProcedureType = @ProcedureType

end

GO

select a.name as [Pet Name],a.kind,b.name as [Owner Name]

from Pet a join PetOwner b on a.ownerID=b.ownerID

order by a.Name ASC

GO

select a.PetID as petid,a.name as NAME,a.kind as kind,

a.Gender as gender from Pet a join ProcedureHistory b

ON a.PetID=b.PetID where 1=(Select MONTH(b.ProcedureDate))

Order by a.PetID ASC

GO

CREATE TABLE Categories(category\_id int primary key,category\_name varchar(255) not null)

GO

CREATE TABLE Brands(brand\_id int primary key,brand\_name varchar(255) not null)

GO

insert into Categories VALUES(1,"Children Bicycles"),(2,"Comfort Bicycles"),(3,"Cruisers Bicycles"),(4,"Cyclocross Bicycles")

GO

INSERT into Brands VALUES(1,"Electra"),(2,"Haro"),(3,"Heller"),(4,"Pure Cycles"),(5,"Ritchey")

GO