1. DDL PET CLINIC

CREATE TABLE PetOwner(

OwnerId int PRIMARY KEY,Name char(50),Surname char(50),

StreetAddresschar(100), City char(100),

State char(10), ZipCode int)

GO

CREATE TABLE Pet(

PetIdchar(10) PRIMARY KEY,Name char(20),

Kind char(10),Gender char(6), Age int,

OwnerId int)

GO

CREATE TABLE ProcedureHistory(

PetIdchar(10),ProcedureDate date,

ProcedureTypechar(50),Description char(100))

GO

ALTER TABLE PetOwner

ADD CONSTRAINT state\_michi

DEFAULT 'Michigan' FOR State

GO

ALTER TABLE Pet

ADD CONSTRAINT fh FOREIGN KEY(OwnerId) REFERENCES

PetOwner(OwnerId)

GO

ALTER TABLE ProcedureHistory

ADD CONSTRAINT fh1 FOREIGN KEY(PetId) REFERENCES

Pet(PetId)

GO

Pet Clinic DRL

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

Pet Stored Store Procedure

CREATE PROCEDURE usp\_PetProcedureDetails @ProcedureType

AS

SELECT p.PetID,p.Name,po.Name,ph.Description

from Pet p join PetOwner po on Pet.OwnerID=PetOwner.OwnerID

join ProcedureHistory ph on p.PetID =ph.PetId

go

1. DRL CLOTHING STORE

select FirstName+","+SurName as CustomerName

from Customers

where CustomerId in

(select CustomerIdfrom(select Customers.CustomerId, count(Customers.CustomerId) count from Customers

join Transactions

on Customers.CustomerId= Transactions.CustomerId

group by Customers.CustomerIe) a

where a.count>1)

go

select Description, (RetailPrice - (RetailPrice\*Discount)) as Cost\_Price

from Clothing

go

Clothing Store DDL

Create Table Customers(CustomerId int primary key,

firstName varchar(20),

surName varchar(20),

customerState varchar(20))

go

create table Clothing(ItemCode bigint primary key,

Description varchar(100),

RetailPrice decimal(6,2),

Discount decimal (5,2))

go

create table Transactions (TransactionId int primary key,

TransactionDate datetime,

CustomerId int,

ItemCode bigint)

go

alter table Clothing

add constraint unq unique(Description)

go

alter table Transactions

add constraint fk foreign key(CustomerId) references Customers(CustomerId)

go

alter table Transactions

add constraint fgk foreign key(ItemCode) references Clothing(ItemCode)

go

Clothing Stored Store Procedure

Create proc usp\_TransactionDetails

as

begin

select a.FirstName,c.RetailPrice as TotalPrice from customers a join Transactions b

on a.CustomerId=b.CustomerId

join clothing c

on b.ItemCode=c.ItemCode

group by a.FirstName,c.RetailPrice

end

go

1. PROCEDURE BIKE STORE

CREAT proc usp\_MaxNoOfProducts as

Begin

SELECT b.brand\_name as Brand\_Name, c.category\_name as Category\_Name, p.product\_name as Product\_Name

FROM Brands b, Categories c, Products p

WHERE p.brand\_id IN( SELECT p.brand\_id,count(p.brand\_id)

FROM Products p

GROUP BY p.brand\_id

ORDER BY count(p.brand\_id) desc,Limit 1)

End

GO

Bike Stored DDL

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')

insert into Categories values(2,'Comfort Bicycles')

insert into Categories values(3,'Cruisers Bicycles')

insert into Categories values(4,'Cyclocross Bicycles')

Go

insert into Brands values(1,'Electra')

insert into Brands values(2,'Haro')

insert into Brands values(3,'Heller')

insert into Brands values(4,'Pure Cycles')

insert into Brands values(5,'Ritchey')

Go

Bike Stored DRL

SELECT TOP 1 B.brand\_name AS Brand\_name From Brands B

JOIN Products P ON B.brand\_id =P.brand\_id

Order BY P.list\_Price desc

go

SELECT P.product\_name AS Product\_name,P.model\_year AS Model\_year FROM Products P

JOIN Categories c ON p.Category\_id=C.Category\_id

WHERE C.Category\_name Like '%children%'

Go

MOVIE 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

MOVIE DRL

select m.Title, m.Director, m.Year, b.Rating from

Movies m

join BoxOffice b

on m.Id = b.Movie\_id

where m.Year between 2000 and 2005 and b.Rating>5

go

select cast(concat(Title, ' in the Year ', Year) as varchar(117)) as [Movie and Release Year]

from Movies

go

Lodging DRL

SELECT room\_id "Room Number", room\_type\_id "Room Type"

from room

where room\_id NOT IN

( SELECT r.room\_id from room r join booking b

on r.room\_id = b.room\_no) AND room\_type\_id != "single"

GO

select r2.room\_type\_id as "Room Type",

count(r1.room\_no) as "No Of Bookings"

from room\_type r2 left outer join room r on

(r.room\_type\_id = r2.room\_type\_id)

left outer join booking r1

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

group by r2.room\_type\_id

order by count(r1.room\_no) desc

GO

Lodging DDL

CREATE TABLE room\_type (room\_type\_id varchar(6) primary key,description varchar(100))

go

CREATE TABLE rate (room\_type\_id varchar(6) not null,occupancy int not null,amount decimal(10,2) default 0)

go

CREATE TABLE room (room\_id int primary key,room\_type\_id varchar(6),max\_occupancy int)

go

alter table room add foreign key(room\_type\_id) references room\_type(room\_type\_id)

go

alter table rate add constraint pk\_name primary key (room\_type\_id,occupancy)

go

alter table rate add foreign key(room\_type\_id) references room\_type(room\_type\_id)

go