**Tecnológico Nacional de México**

**Instituto Tecnológico de Tijuana**

Subdirección Académica

Departamento de Sistemas y Computación

*Semestre Agosto-Diciembre 2015*

Materia: **Taller de Base de Datos**

**Trabajo Final**

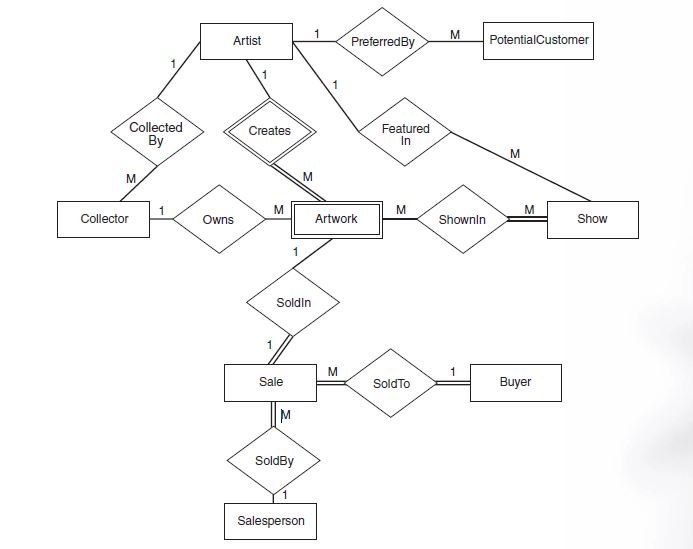
Navarro Mancillas Guillermo 13211447

Maestro:  
Doc. Margarita Ramírez Ramírez

*Tijuana B.C. a 09 de diciembre de 2015*

***Trabajo Final Taller de Base de Datos 2015-2.***

1. Utilizando la Base de Datos galería, del libro de Bases de Datos de Catherine Ricardo, realiza lo que a continuación se te solicita
2. O Utilizando una base de datos similar realiza las sentencias similares a las solicitadas.
3. Crea el diagrama entidad relación.



/\*2.Crea la base de Datos, con tablas y características definidas en el libro.\*/

create database trabajo\_final

use trabajo\_final

/\*32.Liga la regla al atributo ZIP de la tabla ZIPS. \*/

CREATE TABLE Zips (

zip varchar (5),

city varchar (15) not null,

state varchar (2) not null,

constraint Zips\_zip\_pk primary key (zip)

);

CREATE TABLE Artist (

ArtistId int,

firstName varchar (15) not null,

lastName varchar (20) not null,

interviewDate date,

interviewerName varchar (35),

areaCode varchar(3),

telephoneNumber varchar (20),

street varchar (50),

zip varchar (5),

salesLastYear decimal (8, 2),

salesYearToDate decimal (8, 2),

socialSecurityNumber varchar (10),

usualMedium varchar (15),

usualStyle varchar (15),

usualType varchar (20),

constraint Artist\_ArtistId\_pk primary key (ArtistId),

constraint Artist\_SSN\_uk unique (socialSecurityNumber),

constraint Artist\_fName\_lName\_uk unique (firstName, lastName),

constraint Artist\_zip\_fk foreign key (zip) references Zips (zip)

);

CREATE TABLE Collector (

socialSecurityNumber varchar (10),

firstName varchar (15) not null,

lastName varchar (20) not null,

interviewDate date,

interviewerName varchar (35),

areaCode varchar(3),

telephoneNumber varchar (20),

street varchar (50),

zip varchar (5),

salesLastYear decimal (8, 2),

salesYearToDate decimal (8, 2),

collectionArtistId int,

collectionMedium varchar (15),

collectionStyle varchar (15),

collectionType varchar (20),

constraint Collector\_SSN\_pk primary key (socialSecurityNumber),

constraint Collector\_collArtistid\_fk foreign key (collectionArtistId) references Artist (artistId),

constraint Collector\_zip\_fk foreign key (zip) references Zips (zip)

);

CREATE TABLE PotentialCustomer (

potentialCustomerId int (6),

firstname varchar (15) not null,

lastName varchar (20) not null,

areaCode varchar(3),

telephoneNumber varchar (20),

street varchar(50),

zip varchar (5),

dateFilledIn date,

preferredArtistId int,

preferredMedium varchar (15),

preferredStyle varchar (15),

preferredType varchar (20),

constraint PotentialCustomer\_potCusId\_pk primary key (potentialCustomerId),

constraint PotentialCustomer\_zip\_fk foreign key (zip) references Zips (zip),

constraint Potential Customer\_prefAId\_fk foreign key (preferredArtistId) references Artist (artist\_Id)

);

CREATE TABLE Artwork (

artworkId int,

artistId int NOT NULL,

workTitle varchar (50) NOT NULL,

askingPrice decimal (8, 2),

dateListed date,

dateReturned date,

dateShown date,

status varchar (15),

workMedium varchar (15),

workSize varchar (15),

workStyle varchar (15),

workType varchar (20),

workYearCompeted int,

collectorSocialSecurityNumber int,

constraint Artwork\_artworkId\_pk primary key (artworkId),

constraint Artwork\_artId\_wTitle\_uk unique (artistId, workTitle),

constraint Artwork\_artId\_fk foreign key (artistId) references Artist (artistId),

constraint Artwork\_collSSN\_fk foreign key (collectorSocialSecurityNumber) references Collector (socialSecurityNumber)

);

CREATE TABLE Show (

showTitle varchar (50),

showFeaturedArtistId int,

showClosingDate date,

showTheme varchar (50),

showOpeningDate date,

constraint Show\_showTitle\_pk primary key (showTitle),

constraint Show\_showFeaturedArtId\_fk foreign key (showFeaturedArtistId) references Artist (artistId)

);

CREATE TABLE ShownIn (

artworkId int,

showTitle varchar (50),

constraint ShownIn\_artId\_showTitle\_pk primary key (artworkId, showTitle),

constraint ShownIn\_artId\_fk foreign key (artworkId) references Artwork (artworkId),

constraint ShownIn\_showTitle\_fk foreign key (showTitle) references Show (showTitle)

);

CREATE TABLE Buyer (

buyerId int,

firstName varchar (15) not null,

lastName varchar (20) not null,

street varchar (50),

zip varchar (5),

areaCode varchar (3),

telephoneNumber varchar (20),

purchasesLastYear decimal (8, 2),

purchasesYearToDate decimal (8, 2),

constraint Buyer\_buyerId\_pk primary key (buyerId),

constraint Buyer\_zip\_fk foreign key (zip) references Zips (zip)

);

CREATE TABLE Salesperson (

socialSecurityNumber varchar (10),

firstName varchar (15) not null,

lastName varchar (20) not null,

street varchar (50),

zip varchar (5),

constraint Salesperson\_SSN\_pk primary key (socialSecurityNumber),

constraint Salesperson\_fName\_lName\_uk unique (firstName, lastName),

constraint Salesperson\_zip\_fk foreign key (zip) references Zips (zip)

);

CREATE TABLE Sale (

invoiceNumber int,

artworkId int NOT NULL,

amountRemittedToOwner decimal (8, 2) default 0.00,

saleDate date,

salePrice decimal (8, 2),

saleTax decimal (6, 2),

buyerId int NOT NULL,

salespersonSocialSecurityNumber int,

constraint Sale\_invoiceNumber\_pk primary key (invoiceNumber),

constraint Sale\_artworkId\_uk unique (artworkId),

constraint Sale\_artworkId\_fk foreign key (artworkId) references Artwork (artworkId),

constraint Sale\_buyerId\_fk foreign key (buyerId) references Buyer (buyerId)

);

create unique index Artist\_lastName\_firstName on Artist(lastName, firstName);

create index Artist\_zip on Artist(zip);

create index Collector\_collectionArtistId on Collector(collectionArtistId);

create index Collector\_zip on Collector(zip);

create index Collector\_lastName\_firstName on Collector(lastName, firstName);

create index PotentialCustomer\_zip on PotentialCustomer(zip);

create index PotentialCustomer\_lastName\_firstName on PotentialCustomer(lastName, firstName);

create unique index Artwork\_artistId\_workTitle on Artwork (artistId, workTitle);

create index Artwork\_artistId on Artwork(artistId);

create index Artwork\_collectorSocialSecurityNumber on Artwork (collectorSocialSecurityNumber);

create index Show\_showFeaturedArtistId on Show (showFeaturedArtistId);

create index Shownin\_artworkId on Shownin (artworkId);

create index Shownin\_show Title on ShownIn (showTitle);

create index Buyer\_zip on Buyer(zip);

create index Buyer\_lastName\_firstName on Buyer (lastName, firstName);

create unique index Salesperson\_lastName\_firstName on Salesperson (lastName, firstName);

create index Salesperson\_zip on Salespeson (zip);

create index Sale\_buyerId on Sale (buyerId);

/\*3.Inserta los datos a cada tabla\*/

INSERT INTO Zips vALUES (‘10101’, ‘New York’, ‘NY’);

INSERT INTO Zips vALUES (‘10801’, ‘New Rochelle’, ‘NY’);

INSERT INTO Zips vALUES (‘92101’, ‘San Diego’, ‘CA’);

INSERT INTO Zips vALUES (‘33010’, ‘Miami’, ‘FL’);

INSERT INTO Zips vALUES (‘60601’, ‘Chicago’, ‘IL’);

CREATE SEQUENCE artistId\_sequence;

INSERT INTO Artist vALUES(artistId\_sequence.NEXTvAL, ‘Leonardo’, ‘vincenti’, ‘1999-10-10’, ‘Hughes,’ ‘212’, ‘5559999’, ‘10 Main

Street’, ‘10101’, 9000, 4500, ‘099999876’, ‘oil’, ‘realism’, ‘painting’);

INSERT INTO Artist vALUES(artistId\_sequence.NEXTvAL, ‘vincent’, ‘Gogh’, ‘2004-06-15’, ‘Hughes’, ‘914,’ ‘5551234,’ ‘55

West 18 Street’, ‘10801’, 9500, 5500, ‘099999877’, ‘oil’, ‘impressionism’, ‘painting’);

INSERT INTO Artist vALUES(artistId\_sequence.NEXTvAL, ‘Winslow’, ‘Homes’, ‘2004-01-05’, ‘Hughes’, ‘619’, ‘1234567’, ‘100

Water Street’, ‘92101’, 14000, 4000, ‘083999876’, ‘watercolor’, ‘realism’, ‘painting’);

INSERT INTO Artist vALUES(artistId\_sequence.NEXTvAL, ‘Alexander’, ‘Calderone’, ‘1999-02-10’, ‘Hughes’, ‘212’, ‘5559999’,

‘10 Main Street’, ‘10101’, 20000, 20000, ‘123999876’, ‘steel’, ‘cubism’, ‘sculpture’);

INSERT INTO Artist vALUES(artistId\_sequence.NEXTvAL, ‘Georgia’, ‘Keefe’, ‘2004-10-05’, ‘Hughes’, ‘305’, ‘1239999’, ‘5

Chestnut Street’, ‘33010’, 19000, 14500, ‘987999876’, ‘oil’, ‘realism’, ‘painting’);

INSERT INTO Collector vALUES(‘102345678’, ‘John’, ‘Jackson’, ‘2004-02-01’, ‘Hughes’, ‘917’, ‘7771234’, ‘24 Pine Avenue’,

‘10101’, 4000, 3000, 1, ‘oil’, ‘realism’, ‘collage’);

INSERT INTO Collector vALUES (‘987654321’, ‘Mary’, ‘Lee’, ‘2003-03-01’, ‘Jones’, ‘305’, ‘5551234’, ‘10 Ash Street’, 33010,

‘2000’, 3000, 2, ‘watercolor’, ‘realism’, ‘painting’);

INSERT INTO Collector vALUES(‘034345678’, ‘Ramon’, ‘Perez’, ‘2003-04-15’, ‘Hughes’, ‘619’, ‘8881234’, ‘15 Poplar Avenue’,

‘92101’, 4500, 3500, 3, ‘oil’, ‘realism’, ‘painting’);

INSERT INTO Collector vALUES(‘888881234’, ‘Rick’, ‘Lee’, ‘2004-06-20’, ‘Hughes’, ‘212’, ‘9991234’, ‘24 Pine Avenue’, ‘10101’,

4000, 3000, 3, ‘oil’, ‘realism’, ‘sculpture’);

INSERT INTO Collector vALUES(‘777345678’, ‘Samantha’, ‘Torno’, ‘2004-05-05’, ‘Jones’, ‘305’, ‘5551234’, ‘10 Ash Street’,

‘33010’, 40000, 30000, 1, ‘acrylic’, ‘realism’, ‘painting’);

CREATE SEQUENCE potentialCustomerId-sequence;

INSERT INTO PotentialCustomer vALUES(potentialCustomerId\_sequence.NEXTvAL, ‘Adam’, ‘Burns’, ‘917’, ‘3456789’, ‘1

Spruce Street’, ‘10101’, ‘2003-12-12’, 1, ‘watercolor’, ‘impressionism’, ‘painting’);

INSERT INTO PotentialCustomer vALUES(potentialCustomerId sequence.NEXTvAL, ‘Carole’, ‘Burns’, ‘917’, ‘3456789’, ‘1 Spruce

Street’, ‘10101’, ‘2003-12-12’, 2, ‘watercolor’, ‘realism’, sculpture’);

INSERT INTO PotentialCustomer vALUES(potentialCustomerId\_sequence.NEXTvAL, ‘David’, ‘Engel’, ‘914’, ‘7777777’, ‘715

North Avenue’, ‘10801’, ‘2003-08-08’, 3, ‘watercolor’, ‘realism’, ‘painting’);

INSERT INTO PotentialCustomer vALUES(potentialCustomerId\_sequence.NEXTvAL, ‘Frances’, ‘Hughes’, ‘619’, ‘3216789’, ‘10

Pacific Avenue’, ‘92101’, ‘2004-01-05’, 2, ‘oil’, ‘impressionism’, ‘painting’);

INSERT INTO PotentialCustomer vALUES(potentialCustomerId\_sequence.NEXTvAL, ‘Irene’, ‘Jacobs’, ‘312’, ‘1239876’, ‘1

Windswept Place’, ‘60601’, ‘2003-09-21’, 5, ‘watercolor’, ‘abstract expressionism’, ‘painting’);

CREATE SEQUENCE artworkId\_sequence;

INSERT INTO Artwork vALUES(artworkid\_sequence.NEXTvAL, 1, ‘Flight’, 15000.00, ‘2003-09-08’,NULL ,NULL, ‘for sale’,’oil’, ‘36 in X 48 in’, ‘realism’, ‘painting’,’2001’,NULL );

INSERT INTO Artwork vALUES(artworkid\_sequence.NEXTvAL, 3, ‘Bermuda Sunset’, 8000.00, ‘2004-03-15’,NULL

,’2004-04-01’ , ‘sold’,’watercolor’, ‘22 in X 28 in’, ‘realism’, ‘painting’, 2003’,NULL );

INSERT INTO Artwork vALUES(artworkid\_sequence.NEXTvAL, 3, ‘Mediterranean Coast’, 4000.00, ‘10-18-2003’,NULL

,’2004-04-01’, ‘for sale’,’watercolor’, ‘22 in X 28 in’, ‘realism’, ‘painting’,’2000’,’102345678’);

INSERT INTO Artwork vALUES(artworkid\_sequence.NEXTvAL, 5, ‘Ghost orchid’, 18000.00’,

2003-06-01’,NULL ,NULL , ‘sold’,’oil’, ‘36 in X 48 in’, ‘realism’, ‘painting’,’2001’,’034345678’ );

INSERT INTO Artwork vALUES(artworkid\_sequence.NEXTvAL, 4, ‘Five Planes’, 15000.00,

’2004-01-10’,NULL ,’03-10-2004’ , ‘for sale’,’steel’, ‘36inX30inX60in’, ‘cubism’ ‘sculpture’ ’2003’,’034345678’ );

INSERT INTO Show vALUES(‘The Sea in Watercolor’,3, ‘2004-04-30’, ‘seascapes’, ‘2004-01-04’);

INSERT INTO Show vALUES(‘Calderone: Mastery of Space’,4,’2004-03-20’,’mobiles’, ‘2004-03-04’);

INSERT INTO ShownIn vALUES(2,’The Sea in Watercolor’);

INSERT INTO ShownIn vALUES(3, ‘The Sea in Watercolor’);

INSERT INTO ShownIn vALUES(5, ‘Calderone: Mastery of Space’);

CREATE SEQUENCE buyerId\_sequence;

INSERT INTO Buyer vALUES (BuyerId\_sequence.NEXTvAL, ‘valerie’, ‘Smiley’, ‘15 Hudson Street’, ‘10101’, ‘718’,’5551234’,

5000, 7500);

INSERT INTO Buyer vALUES (BuyerId\_sequence.NEXTvAL, ‘Winston’, ‘Lee’, ‘20 Liffey Avenue’, ‘60601’, ‘312’,’7654321’,

3000, 0);

INSERT INTO Buyer vALUES (BuyerId\_sequence.NEXTvAL, ‘Samantha’, ‘Babson’, ‘25 Thames Lane’, ‘92101’, ‘619’,’4329876’,

15000, 0);

INSERT INTO Buyer vALUES (BuyerId\_sequence.NEXTvAL, ‘John’, ‘Flagg’, ‘22 Amazon Street’, ‘10101’, ‘212’,’7659876’,

3000, 0);

INSERT INTO Buyer vALUES (BuyerId\_sequence.NEXTvAL, ‘Terrence’, ‘Smallshaw’, ‘5 Nile Street’, ‘33010’, ‘305’,’2323456’,

15000, 17000);

INSERT INTO Salesperson vALUES(‘102445566’, ‘John’,’Smith’, ‘10 Sapphire Row’, ‘10801’);

INSERT INTO Salesperson vALUES(‘121344321’, ‘Alan’,’Hughes’, ‘10 Diamond Street’, ‘10101’);

INSERT INTO Salesperson vALUES(‘101889988’, ‘Mary’,’Brady’, ‘10 Pearl Avenue’, ‘10801’);

INSERT INTO Salesperson vALUES(‘111223344’, ‘Jill’,’Fleming’, ‘10 Ruby Row’, ‘10101’);

INSERT INTO Salesperson vALUES(‘123123123’, ‘Terrence’,’DeSimone’, ‘10 Emerald Lane’, ‘10101’);

INSERT INTO Sale vALUES(1234, 2,NULL ,’2004-04-05’,7500,600, 1, ‘102445566’);

INSERT INTO Sale vALUES(1235, 6,NULL ,’2004-04-06’,17000,1360, 5, ‘121344321’);

/\*4.Crea una consulta que despliegue los datos de los artistas cuyo estilo sea realismo.\*/

select \* from artist where usualStyle = 'realism';

/\*5.Crea una consulta que despliegue los datos de los artistas cuyas ventas del último año este entre 10,000 y 20,000.\*/

select \* from artist where salesLastYear => 10000 and salesLastYear <= 20000

/\*6.Despliega los datos de la tabla Buyer cuyo zip sea de la ciudad de ‘San Diego’.\*/

select \* from buyer where zip = '92101'

/\*7.Crea una vista que despliegue los datos de ArtWork y del artista que los ha creado cuyo status sea ‘sold’\*/

create view dataArtwork as

select status from Artwork where titles = 'sold'

select \* from dataArtwork

/\*8.Crea una consulta que despliegue los montos acumulados de ventas por año (saleslastYear, SalesYearToDate) agrupados por collector.\*/

select saleslastYear, SalesYearToDate from Collector

/\*9.Crea una consulta en la que se desplieguen los datos de la tabla Artwork ordenados alfabéticamente por workTitle. \*/

select \* from Artwork order by workTitle

/\*10.Despliega los datos de las obras Artwork que se encuentran en status "for sale", si no hubiese obras con este status despliega el mensaje las obras están vendidas\*/

select status from Artwork where titles = 'for sale'

/\*11.Crea una consulta en el que se desplieguen los datos de las tablas Artist y Collector unidas utiliza la sentencia Inner Join. \*/

select \* from Artist inner join Collector on Artist.ArtistId = Collector.collectionArtistId

/\*12.Crea una consulta en la que se desplieguen los datos de los artistas cuyo Artwork tenga como showTheme ‘seascapes’.\*/

select \* from Artwork where showTheme = 'seascapes'

/\*13.Crea una consulta con los datos de las tablas Artist, Artwork, show, show In, cuando no haya correspondencia pueden aparecer en NULL\*/

select \* from Show inner join ShowIn on Show.Showtitle = ShowIn.Showtitle

inner join Artwork

on Artwork.ArtworkId = ShowIn.ArtworkId

inner join Artist on Artist.ArtistId = Artwork.ArtistId

/\*14.Despliega los datos de la tabla buyer, ordenados por firstName calcula el promedio de venta de cada comprador. \*/

select AVG(purchasesYearToDate) as ['Average of sales'], buyerId, firstName, lastName from Buyer order by firstName

/\*15.Crea una consulta que despliegue todos los datos del artista cuya venta (salesYearToDate) sea la mayor.\*/

select MAX(salesYearToDate) from Artist

/\*16.Crea una tabla que contenga el areacode denomínala código postal, evita duplicados.\*/

createtable codigo\_postal(

areacode char(3)

)

/\*17.Despliega si la venta del último año de Vincent Van Gogh es mayor o menor que el promedio de ventas. \*/

select Artist.salesLastYear, Artist.salesYearToDate

categoria =

case

when Artist.salesLastYear < Artist.salesYearToDate then 'menor'

when Artist.salesLastYear > Artist.salesYearToDate then 'mayor'

from Artist where firstName = 'Vincent'

/\*18.Despliega los datos de los artistas y clasifica si las ventas son regulares, buenas o malas, si el valor es menor que 5000 son malas, si son entre 5001 y 9000 son regulares, si son mayores de 9000 son buenas\*/

select Artist.salesYearToDate

categoria =

case

when Artist.salesYearToDate < 5000 then 'malas'

when Artist.salesYearToDate > 5000 and Artist.salesYearToDate <= 9000 then 'regulares'

when Artist.salesYearToDate > 9000 then 'buenas'

/\*19.Despliega los nombres de los artistas, que se entrevistaron después del 1ero de Enero del 2004, pero que no tengan obras de arte en lista. \*/

SELECT firstName, lastName

FROM Artist

WHERE interviewDate > '2004-01-01' AND NOT EXISTS

(SELECT \*

FROM Artwork

WHERE artistId = Artist.artistId);

/\*20.Crea una procedimiento que despliegue los datos de Collector y despliega un mensaje con el encabezado tipo de acuerdo a coleccionType, el mensaje solo debe desplegar el tipo de colección en español painting pintura, utiliza Case\*/

create procedure Collector1

as

begin

select collectionStyle,coleccionType =

case collectionType

when collectionType ='painting' then 'pintura'

when collectionType ='collage'

end

from Collector;

/\*21.Crea una consulta que despliegue los datos de la tabla Artist, Collector, Artwork, usa Join\*/

select \* from Artist inner join Collector on Artist.socialSecurityNumber = Collector.socialSecurityNumber

inner join Artwork.socialSecurityNumber = Collector.socialSecurityNumber

/\*22.Diseña una consulta que despliegue los datos de los artistas cuya primer letra sea una variable, a la cual le asignas valor mediante comando set.\*/

declare @a char

set @a =

select firstName,lastName,interviewDate

from artist

where firstName like('%@a')

select\*from Artist

/\*23.Encuentre el precio de venta promedio de las obras de la artista Georgia Keefe.\*/

create view promedio1

as

selectAVG (salePrice) as promedio

from Sale

where artworkId in (select artworkId

from Artwork

where artworkId = (select artistId from artist where lastName ='KeefeE'and firstName ='Georgia')

)

/\*24.Despliega los datos de la tabla artista\*/

select \* from Artist;

/\*25.Despliega los datos de la tabla artwork.\*/

select \* from Artwork;

/\*26.Despliega los datos de la tabla buyer\*/

select \* from Buyer;

/\*27.Despliega los datos de la tabla collector\*/

select \* from Collector;

/\*28.Despliega los datos de la tabla potentialcustomer\*/

select \* from PotentialCustomer;

/\*29.Despliega los datos de la tabla salesperson\*/

select \* from Salesperson;

/\*30.Despliega los datos de la tabla zips\*/

select \* from Zips;

/\*31.Crea una regla en la tabla ZIPS, que restrinja la entrada de datos a solo números de 0 a 9.\*/

create rule reglaZips like ('0,1,2,3,4,5,6,7,8,9')

exec sp\_bindrule 'reglaZips', 'Zips.\*';

/\*33.Crea una regla y asóciala al atributo areacode, asígnale el valor 619.\*/

create rule reglaareacode as @zip like ('619[0-9][0-9]')

exec sp\_bindrule 'reglaZips', Zips.\*

/\*34.Crea una regla y asóciala al atributo interviewdate en el que la fecha sea mayor o igual a la fecha actual\*/

create rule reglaintdate where @interviewdate > = Convert(date, getdate());

/\*36.Utilizando la clausula compute, Despliega los datos de la tabla salesperson y encuentra quien tiene el socialsecurity\_number mayor\*/

select \* from Salesperson

order by socialSecurityNumber

compute max(socialSecurityNumber);

/\*37.Despliega la sumatoria de saleprice de la tabla sale\*/

select sum(salePrice) from Sale

/\*38.Despliega el area\_code y la cantidad de artistas que haya por cada area\_code\*/

select areaCode, count(\*) from Artist group by areaCode

/\*39.Despliega los datos de la tabla Zips cuya ciudad inicie con las palabras ne\*/

select \* from Zips where city like 'ne%'

/\*40.Despliega los datos de la tabla Collector en donde el primer nombre sea Rick \*/

select \* from Collector where firstName like 'Rick%'

/\*41.Despliega los datos de la tabla potentialcustomer, cuyo potential\_customerid este entre 2 y 4.\*/

select \* from PotentialCustomer where potentialCustomerId between 2 and 4

/\*42.Despliega los datos de la tabla salesperson ordenadas por socialsecurity\_num.\*/

select socialSecurityNumber from Salesperson.socialSecurityNumber order by socialSecurityNumber

/\*43.Crea una vista de la tabla SALE en donde se desplieguen los datos Invoice\_num, artworkid, saleprice, y el precio de venta con un incremento del 12 %, agregue la palabra iva.\*/

create view Sale1 as

select invoiceNumber, artworkId, salePrice, (count(salePrice) \* .12) from Sale

group by SalePrice

/\*44.Crea una vista que despliegue los datos de los artistas cuyo zip sea el de SanDiego.\*/

create view Artistzip as

ArtistId, firstName, lastName, zip from Artist

where zip = '92101'

/\*45.Crea un Procedimiento almacenado en el que se capture el nombre de un collector y te despliegue su información\*/

create procedure CollectorName

as

Begin

select \* from Collector where firstName = not null and lastName = not null

end

exec CollectorName

select \* from Collector

/\*46.Despliega los datos zip, city, artista\_id, firstname, lastname de aquellos cuyo ZIP sea < =25000, utiliza Join\*/

select zip, ArtistId, firstName, lastName from Artist inner join Zips on Artist.zip = Zips.zip

where zip <= '25000'

/\*47.Encuentra la comisión total para el vendedor John Smith obtenida entre las fechas 1 de Abril y 15 de Junio del 2004\*/

SELECT .05 \* SUM(salePrice)

FROM Sale

WHERE saleDate > = '2004-04-01' AND

saleDate < = '2004-04-15' AND

salespersonSocialSecurityNumber = (SELECT socialSecurityNumber

FROM Salesperson

WHERE firstName= 'John' AND lastName

= 'Smith');

/\*49.Crea un triger que almacene en una tabla denominada Movimientos, cada vez que se realice un movimiento (ABC) y afecte una tabla, de alta, bajas o cambios a un registro en cualquier tabla, debe almacenar los siguientes datos (Fecha de la Movimiento, nombre de la tabla, descripción de la acción (alta, baja, cambio))\*/

CREATE OR REPLACE TRIGGER Movimientos

BEFORE UPDATE OF askingPrice ON Artwork

FOR EACH ROW

BEGIN

INSERT INTO ArtworkPriceAudit

VALUES(SYSDATE, USER, :OLD.artworkId,:OLD.askingPrice,

:NEW.askingPrice);

END;