INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENERIA Y TECNOLOGIAS
BASES DE DATOS DISTRIBUIDAS

upiita-ipn

SQL Server 2016

USO DE JOINS

Nombre: Sanchez Ramirez Noel Adan

Objetivo

La resolución de cada ejercicio, se debe escribir el código utilizado. Al terminar los ejercicios subirlo al google classroom (pdf)

Ejercicios:

Dada las siguientes tablas.

TBL_NAVEGANTES				
ID_NAVEGANTES	NOMBRE	RATING	EDAD	CIUDAD
22	Pedro	7	45	Toluca
23	Andrés	1	35	Puebla
33	Loreto	8	31	Guanajuato
29	Natalia	7	40	Ciudad de México
30	Esteban	4	50	Oaxaca
31	Susana	2	24	Tamaulipas
35	Oscar	1	42	Mexicali

TBL_RSERVA		
ID_RESERVA	ID_BOTES	FECHA_RESERVA
23	102	10.11.2020
22	102	10.11.2020
33	102	05.11.2020
30	104	05.11.2020

CAT_BOTES		
ID_BOTES	NOMBRE_BOTES	COLOR
101	MARINO	AZUL
102	INTER-CONTINENTAL	ROJO
103	CLIPPER	MORADO
104	RAY	VERDE

Actividades

1. Crear las siguientes operaciones por cada tabla, solo si aplican:
    - Inner join
    - Left join
    - Right join

- Outer join
- Self join
- Cross join
- Left join (navegantes-reserva)
- Right join (reserva-botes)
- Full outer join (reserva-botes)

2. Poner el código

```
CREATE TABLE TBL_NAVEGANTES(
		ID_NAVEGANTES INT PRIMARY KEY NOT NULL,
		NOMBRE NVARCHAR(20) NOT NULL,
		RATING INT DEFAULT 0,
		EDAD INT DEFAULT 18,
		CIUDAD NVARCHAR(50)
);
SELECT* FROM TBL_NAVEGANTES;
CREATE TABLE CAT_BOTES(
		ID_BOTES INTEGER PRIMARY KEY NOT NULL,
		NOMBRE_BOTES NVARCHAR(25) NOT NULL,
		COLOR NVARCHAR(20) NOT NULL
);
SELECT* FROM CAT_BOTES;
CREATE TABLE TBL_RESERVA(
		ID_RESERVA INTEGER PRIMARY KEY NOT NULL,
		ID_BOTES INT NOT NULL,
		FECHA_RESERVA DATE NOT NULL,
		CONSTRAINT FK_BOTES FOREIGN KEY (ID_BOTES) REFERENCES CAT_BOTES(ID_BOTES),
		CONSTRAINT FK_NAVEG FOREIGN KEY (ID_RESERVA) REFERENCES TBL_NAVEGANTES(ID_NAVEGANTES)
);
SELECT* FROM TBL_RESERVA;
-- Inner join
SELECT ID_RESERVA,TR.ID_BOTES,FECHA_RESERVA,ID_NAVEGANTES,NOMBRE,RATING,EDAD,CIUDAD,NOMBRE_BOTES,COLOR
 FROM TBL_RESERVA TR
INNER JOIN TBL_NAVEGANTES TN ON TN.ID_NAVEGANTES = TR.ID_RESERVA
INNER JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES;
--left join
SELECT * FROM TBL_RESERVA TR
LEFT JOIN TBL_NAVEGANTES TN ON TN.ID_NAVEGANTES = TR.ID_RESERVA
LEFT JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES;
--right join
SELECT * FROM TBL_RESERVA TR
RIGHT JOIN TBL_NAVEGANTES TN ON TN.ID_NAVEGANTES = TR.ID_RESERVA
RIGHT JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES;
-- outer join
SELECT * FROM TBL_RESERVA TR
RIGHT OUTER JOIN TBL_NAVEGANTES TN ON TN.ID_NAVEGANTES = TR.ID_RESERVA
RIGHT OUTER JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES;
SELECT * FROM TBL_RESERVA TR
LEFT OUTER JOIN TBL_NAVEGANTES TN ON TN.ID_NAVEGANTES = TR.ID_RESERVA
LEFT OUTER JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES;
-- SELF JOIN
SELECT * FROM TBL_NAVEGANTES TN, TBL_NAVEGANTES TNB WHERE TN.RATING > TNB.RATING;
-- CROSS JOIN
SELECT * FROM TBL_RESERVA TR
CROSS JOIN TBL_NAVEGANTES TN
CROSS JOIN CAT_BOTES CB;
-- Left join (navegantes-reserva)
SELECT * FROM TBL_NAVEGANTES TN LEFT JOIN TBL_RESERVA TR ON TN.ID_NAVEGANTES = TR.ID_RESERVA;
-- Right join (reserva-botes)
SELECT * FROM TBL_RESERVA TR RIGHT JOIN CAT_BOTES CB ON TR.ID_BOTES = CB.ID_BOTES;
-- FULL OUTER JOIN
SELECT * FROM TBL_RESERVA TR
FULL OUTER JOIN TBL_NAVEGANTES TN ON TN.ID_NAVEGANTES = TR.ID_RESERVA
FULL OUTER JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES;
-- Tabla de llenado, Solo se insertarán datos de obtenidos... Esta tabla no cumple ninguna regla de normalización
CREATE TABLE TODO_XD(
		ID_RESERVA INTEGER,
		ID_BOTES INTEGER,
		FECHA_RESERVA DATE,
		ID_NAVEGANTES INTEGER,
		NOMBRE INTEGER
```

3. Crear la nueva tabla resultante

Preguntas

1. Mostrar el Navegante el más joven

   SELECT * FROM TBL_NAVEGANTES TN WHERE TN.EDAD IN(SELECT MIN(TNB.EDAD) FROM TBL_NAVEGANTES TNB);
   O
   SELECT TOP 1 * FROM TBL_NAVEGANTES TN ORDER BY TN.EDAD ASC;

2. Mostrar el bote más solicitado.

   SELECT TOP 1 TR.ID_BOTES, CB.NOMBRE_BOTES, CB.COLOR, COUNT(TR.ID_BOTES) CUENTA FROM TBL_RESERVA TR INNER JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES GROUP BY TR.ID_BOTES,CB.NOMBRE_BOTES, CB.COLOR;

3. Quienes son los navegantes que no tienen reserva.

   SELECT * FROM TBL_NAVEGANTES TN WHERE TN.ID_NAVEGANTES NOT IN(SELECT TR.ID_RESERVA FROM TBL_RESERVA TR);

4. Agrupar los botes y mostrar desde el más solicitado hasta el que no es solicitado.

   SELECT TR.ID_BOTES, CB.NOMBRE_BOTES, CB.COLOR, COUNT(TR.ID_BOTES) CUENTA FROM TBL_RESERVA TR INNER JOIN CAT_BOTES CB ON CB.ID_BOTES = TR.ID_BOTES GROUP BY TR.ID_BOTES,CB.NOMBRE_BOTES, CB.COLOR;

5. Agrupar los navegantes mediante ratings y mostrar quienes son

   SELECT * FROM TBL_NAVEGANTES TN ORDER BY TN.RATING, TN.ID_NAVEGANTES;

   No hay necesidad de agrupar, ya que no hay alguna función de agregado.