**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)

1. 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,

RATING INTEGER,

EDAD INTEGER,

CIUDAD NVARCHAR(20),

NOMBRE\_BOTES NVARCHAR(25),

COLOR NVARCHAR(20)

);

1. 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;**

1. 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;**

1. 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);**

1. 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;**

1. 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.**