

EVALUACIÓN	Parcial 1	GRUPO	M3B	FECHA	Octubre 2023
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	<ul style="list-style-type: none"> - Puntos: <u>Máximo</u>: 25. - Sin Material - Duración: 2 horas 				

A continuación se detalla una parte del modelo relacional de la plataforma de venta de videojuegos “Éxitos en el Parcial” donde se gestiona una tienda online de videojuegos.

Usuarios(idUsuario, nombre, email, saldo,pais)

Almacena información sobre los usuarios registrados en la plataforma.

Juegos(idJuego, nombre, precio, desarrollador)

Contiene información sobre los juegos disponibles en la plataforma.

Compras(idUsuario, idJuego, fecha)

FK: idUsuario -> Usuarios.IdUsuario

FK: idJuego -> Juegos.IdJuego

Registra los juegos comprados por los usuarios.

Valoraciones(idValoracion, idUsuario, idJuego, puntuacion, comentario,FechaValoracion)

FK: idUsuario -> Usuarios.IdUsuario

FK: idJuego -> Juegos.IdJuego

Permite a los usuarios dejar valoraciones y comentarios sobre los juegos.

Géneros(idGenero, nombre)

Contiene los géneros de los juegos (p.ej. acción, aventura, estrategia).

Juegos_Generos(idJuego, idGenero)

FK: idJuego -> Juegos.IdJuego

FK: idGenero-> Generos.IdGenero

Relaciona los juegos con sus géneros correspondientes.

Se pide:

1. Crear un script que permita en la tabla valoración restringir el atributo puntuación solo a valores entre 0 y 10. **(3 puntos)**

ALTER TABLE Valoraciones

ADD CONSTRAINT chk_Puntuacion

CHECK (puntuacion >= 0 AND puntuacion <= 10);

2. Crear los índices que considere necesarios para optimizar la siguiente consulta: **(3 puntos)**

```
SELECT
    U.nombre AS 'Nombre de Usuario',
    J.nombre AS 'Nombre del Juego',
    C.fecha AS 'Fecha de Compra',
    V.puntuacion AS 'Puntuacion',
    V.comentario AS 'Comentario'
FROM
    Usuarios U
JOIN
    Compras C ON U.idUsuario = C.idUsuario
JOIN
    Juegos J ON C.idJuego = J.idJuego
LEFT JOIN
    Valoraciones V ON C.idUsuario = V.idUsuario AND C.idJuego = V.idJuego
WHERE v.puntuación >5
```

CREATE INDEX idx_Compras_idUsuario ON Compras (idUsuario);

CREATE INDEX idx_Compras_idJuego ON Compras (idJuego);

CREATE INDEX idx_Valoraciones_idUsuario ON Valoraciones (idUsuario);

CREATE INDEX idx_Valoraciones_idJuego ON Valoraciones (idJuego);

CREATE INDEX idx_Valoraciones_puntuacion ON Valoraciones (puntuacion);

3. Dado que la tabla valoraciones ha crecido demasiado se creó la siguiente tabla:

ValoracionesHistóricas (idValoracion, idUsuario, idJuego, puntuacion,
comentario, FechaValoracion, FechaVolcado)

FK: idUsuario -> Usuarios.IdUsuario

FK: idJuego -> Juegos.IdJuego

Se le pide:

- a. Insertar en la nueva tabla todos los registros de valoraciones creados antes del 2018. **(4 puntos)**

INSERT INTO ValoracionesHistóricas (idValoracion, idUsuario, idJuego, puntuacion, comentario, FechaValoracion, FechaVolcado)

SELECT idValoracion, idUsuario, idJuego, puntuacion, comentario, FechaValoracion, GETDATE()

FROM Valoraciones

WHERE YEAR(FechaValoracion) < 2018;

- b. Eliminar de la tabla valoraciones los registros volcados (insertados en la parte a). **(3 puntos)**

```
DELETE FROM Valoraciones
WHERE idValoracion IN (
    SELECT idValoracion
    FROM ValoracionesHistóricas
);
```

4. Resolver las siguientes consultas en SQL

- a. Escribe una consulta SQL que muestre los nombres de los usuarios que han realizado compras y, si lo han hecho, la puntuación promedio que han dado a los juegos comprados. Ordenar los resultados por el nombre del usuario de forma ascendente. **(4 puntos)**

```
SELECT
    U.nombre AS 'Nombre del Usuario',
    AVG(V.puntuacion) AS 'Puntuacion Promedio'
FROM    Usuarios U
LEFT JOIN  Compras C ON U.idUsuario = C.idUsuario
LEFT JOIN  Valoraciones V ON C.idJuego = V.idJuego AND C.idUsuario = V.idUsuario
GROUP BY   U.nombre
ORDER BY   U.nombre ASC;
```

- b. Escribir una consulta SQL que muestre los juegos que tienen una puntuación promedio superior a 8 y que hayan sido comprados al menos una vez.

```
SELECT    J.nombre AS 'Nombre del Juego',  AVG(V.puntuacion) AS 'Puntuacion Promedio'
FROM      Juegos J
INNER JOIN Valoraciones V ON J.idJuego = V.idJuego
WHERE     J.idJuego IN (SELECT DISTINCT C.idJuego
                        FROM Compras C)

GROUP BY  J.nombre
HAVING    AVG(V.puntuacion) > 8;
```

- c. Escribe una consulta SQL que muestre los usuarios que han realizado más compras que cada usuario de Uruguay. **(4 puntos)**

```
SELECT U.nombre AS 'Nombre del Usuario', COUNT(C.idCompra) AS 'Total de Compras'
FROM Usuarios U
JOIN Compras C ON U.idUsuario = C.idUsuario
GROUP BY U.nombre
HAVING COUNT(C.idCompra) > ALL (
    SELECT COUNT(C2.idCompra) AS 'Total de Compras'
    FROM Usuarios U2
    JOIN Compras C2 ON U2.idUsuario = C2.idUsuario
    WHERE U2.pais = 'Uruguay'
    GROUP BY U2.idUsuario)
```