

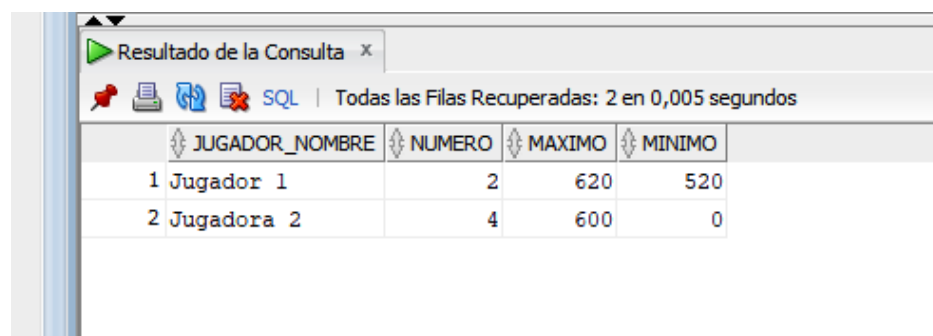
## Sesión 2 - Ejercicios

### SENTENCIA WITH, VALORES NULL, ALTER TABLE, IDENTITY, COMMIT, ROLLBACK, y SAVEPOINT.

Vamos a continuar utilizando el mismo modelo que la sesión anterior. Resuelve los siguientes requerimientos:

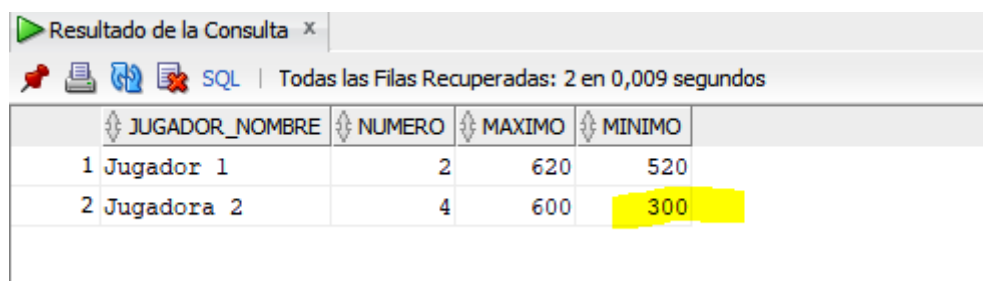
1. Obtener de cada jugador su nombre, el número de partidas jugadas, el máximo y el mínimo de puntos alcanzado en alguna de sus partidas, pero sólo de los jugadores que hayan jugado al menos 2 partidas.

Deberás implementar la sentencia con la cláusula WITH y hacer lo mismo, pero sin la cláusula WITH y controlando los valores nulos con la función NVL. Obtendrás el resultado:



JUGADOR_NOMBRE	NUMERO	MAXIMO	MINIMO
1 Jugador 1	2	620	520
2 Jugadora 2	4	600	0

Repite lo mismo, pero sin el control de valores nulos. En ese caso obtendrás este resultado:



JUGADOR_NOMBRE	NUMERO	MAXIMO	MINIMO
1 Jugador 1	2	620	520
2 Jugadora 2	4	600	300

Explica el porqué de esta diferencia.

**2.-** Queremos enriquecer nuestro modelo de datos, incorporando una nueva tabla de datos. En ella incluiremos un registro de entradas y salidas al sistema. Será una tabla con muchas filas, y necesitamos identificarlas una a una. Llamaremos a la tabla “Registro\_entradas”, y tendrá la siguiente estructura.

Nombre de la columna	Tipo de datos	Restricciones
<b>Jugador_ID</b>	Varchar2(50)	Acepta nulos
<b>Registro_fecha_hora</b>	Date	No acepta nulos
<b>Registro_valido</b>	Int	Sólo podrá valer 0 ó 1, y no acepta nulos. Por defecto valdrá 1

¿Cómo implementarías la clave primaria de esta tabla? Sugerencia: Incluye una nueva columna a la tabla, llamada Registro\_SK de tipo INT, que sea IDENTITY y se incremente de uno en uno de manera automática.

**3.-** Realiza varias inserciones en la tabla Registro\_entradas, para comprobar que la columna IDENTITY se comporta de la manera esperada.

**4.-** ¿Cuál es la diferencia entre la sentencia ALTER TABLE y la sentencia UPDATE?

**5.-** El ejercicio 2 de la sesión anterior decía:

“Nos hemos dado cuenta de que se nos ha olvidado incluir información acerca de la puntuación conseguida en una partida. Borra la tabla y vuélvela a crear con esa nueva columna, a la que debes llamar ‘Partida\_puntuacion’. Debe aceptar nulos, y ser de tipo INT. En la próxima clase veremos una sentencia PL/SQL que nos evitará tener que borrar la tabla, y nos permitirá cambiar su estructura fácilmente.”

Aplicando lo que hemos aprendido en esta sesión, ¿Cómo lo solucionarías ahora, en lugar de borrando la tabla y volviéndola a crear?

**6.-** Borra todas las filas de la tabla Registro\_entradas. A continuación, modifica la estructura de la tabla Registro\_entradas, e indica que la columna Jugador\_ID es una clave ajena hacia la tabla de jugadores. Con otra sentencia, indica que la columna Registro\_fecha\_hora valdrá por defecto la fecha del sistema (SYSDATE).

**7.-** Inserta en la tabla Registro\_entradas una entrada para el jugador Jugador1@dbdm.es, con fecha por defecto del sistema, y con entrada válida a 1. Haz una consulta select para mostrar todos los datos de esa tabla. ¿Aparece la nueva fila? Ejecuta la sentencia Rollback. Haz de nuevo la select sobre la tabla Registro\_entradas ¿Qué ha ocurrido?

**8.-** Elige a otra persona del turno, y dale permiso para que vea tu tabla Registro\_entradas. Esto ya lo veremos más adelante, y se hace mediante la siguiente instrucción:

Grant select on Registro\_entradas to dbdm\_xxxx;

Donde dbdm\_xxxx es el usuario de la otra persona.

Una vez realizado esto (y no antes) vuelve a registrar la entrada de Jugador1@dbdm.es, con fecha por defecto del sistema, y con entrada válida a 1 usando el comando insert del ejercicio anterior. Haz tú una select de la tabla. ¿Te aparece? Ahora pídele a la otra persona que haga una consulta select sobre tu tabla. Deberá ejecutar:

```
Select * from dbdm_yyyy.Registro_entrada
```

Donde dbdm\_yyyy es tu usuario

¿Le aparece la fila nueva? ¿Por qué crees que ocurre eso? Ejecuta la sentencia "Commit" ahora, y que vuelva a ejecutar la select. ¿Le aparece ahora?