



Gestión de Datos

#resumen

#resumen-parcial

SQL - Data Manipulating Language

Aprendizaje de ejercicios

⚠️ ¡Siempre probar todo! >

- Hacer pruebas a medida que se desarrolla el ejercicio
- Releer el enunciado antes de hacer las pruebas finales
- Probar aquello usado en `ORDER BY` y funciones de grupo

⚠️ `COUNT(*)` cuenta filas con columnas `NULL`

⚠️ Al hacer `JOIN` a una tabla con PK doble, saber que la **atomicidad** puede verse afectada >

Usar `DISTINCT` puede salvar la situación

⚠️ No olvidar el uso de `DISTINCT` en funciones de grupo

⚠️ Usar alias de tablas siempre en la que va "afuera"

🔗 ¿Cómo retornar solo la segunda fila de una tabla? >

```
SELECT
  <column(s)>
FROM
  <table(s)>
ORDER BY
  <sort column(s)>
```

```
OFFSET 1 ROWS    -- Skip this number of rows
FETCH NEXT 1 ROWS ONLY; -- Return this number of rows
```

🔗 ¿Cómo hacer una fecha literal? >

```
CAST('2012-01-01' AS DATETIME)
```

🔗 ¿Como fijarse si una fecha a cierta distancia de otra? >

```
@fecha_anterior >= DATEADD(day, -30, @fecha_posterior)
```

Especial atención al "-" antes del 30, siendo 30 los días de plazo.

🔗 La fecha actual puede obtenerse mediante `GETDATE()`

🔗 Al hacer `JOIN` y hacer un `WHERE` sobre una de esas tablas, la condición puede ponerse directamente en el `JOIN`

🔗 Al pedir "el que más (algo)" va obligatoriamente un `TOP 1`

Ejercicios especiales

Ejercicio	Peculiaridad
13	Doble join en <code>composición</code> , join en <code>producto</code>
15	Varios doble <i>join</i> , uso de <code><</code> en ID para evitar duplicados
18	Retornar 2º fila, trabajo sobre últimos 30 días
20	CASE

Transact-SQL

Aprendizaje de ejercicios

⚠️ Siempre revisar que estén bien las condiciones de los `if` >

Especialmente con los signos `<` y `>`.

⚠️ Probar la solución siempre >

Si no es posible probarla en su totalidad, hacerlo parcialmente o usando entradas específicas.

⚠️ Cuando se pide agrupar por mes, revisar si no se debería también hacer por año

⚠️ Al querer ahorrarse subselects, cuidado de no estar limitando el alcance a los registros `inserted`, ignorando los anteriores

⚠️ Sobre hacer `rollback` en tablas que representan conjuntos... >

Cuando en un ejercicio se dice que se desconoce algún procedimiento, o se tiene una tabla transaccional que conforma un conjunto de cosas, es necesario controlar, además de qué entra o no, la consistencia. Esto pasa en tablas donde haya claves compuestas o muchos componentes de un mismo algo (como `Composición` o `Item_Factura`). **Si un ítem de la factura no puede agregarse, deben eliminarse el resto y la factura en sí.**

❓ ¿Que objeto se debe utilizar? >

- Las funciones retornan un valor recibiendo argumentos, pero no pueden modificar tablas
- Los procedimientos sirven cuando se deben realizar cambios en datos existentes
- Los triggers se usan cuando es necesario automatizar el cumplimiento de alguna regla ya existente

❓ ¿Cómo se puede evaluar el último año en el que hubo ventas? >

```
WHERE YEAR(fact_fecha) = (SELECT TOP 1 YEAR(fact_fecha) FROM |
```

💡 Usar `INSTEAD OF` no es necesario a menos que el ejercicio aclare "los que están bien entran, los que están mal no"

ⓘ Usar un segundo cursor cuando no es requerido no va a afectar de manera significativamente negativa en la corrección ejercicio

Ejercicios especiales

Ejercicio	Peculiaridad
30	Agrupacion año/mes, transaccionalidad en tabla compuesta