**TAREA 1**

* **TÍTULO**

SUBCONSULTAS

* **RESUELVE:**

1. Rellena la tabla adjunta realizando un resumen del contenido de la unidad del tema 5 “Realización de consultas avanzadas”, incluyendo un ejemplo diferente al mostrado en la teoría. A modo de ejemplo, se incorpora el primer contenido de dicha sección:

|  |  |  |
| --- | --- | --- |
| Sección | Título | Resumen |
| 1 | Consulta de varias tablas | Hace referencia a un tipo de consulta en la cual intervienen 2 o más tablas, por ejemplo las tablas Empresa, Ciudad y Contacto, relacionadas entre sí por campos comunes. |
| 2 | Composiciones con criterios de selección de fila | La condición de búsqueda que especifica las columnas de emparejamiento en  una consulta multitabla puede combinarse con otras condiciones de búsqueda para  restringir aún más los contenidos de los resultados. |
| 3 | Múltiples columnas de emparejamiento | Las composiciones multicolumna son menos habituales y se encuentran  generalmente en consultas que afectan a claves foráneas compuestas. Para  componer las tablas basándose en la relación padre/hijo, deben especificarse ambos  pares de columnas de emparejamiento. |
| 4 | Consultas de tres o más tablas | SQL puede combinar datos de tres o más tablas utilizando las mismas técnicas  básicas utilizadas para las consultas de dos tablas. |
| 5 | Otras equicomposiciones | La mayoría de las consultas multitabla se basan en relaciones padre/hijo, pero  SQL no exige que las columnas de emparejamiento estén relacionadas como clave  primaria y clave foránea. Cualquier par de columnas de dos tablas pueden servir de  columnas de emparejamiento, siempre que tengan tipos de datos comparables. |
| 6 | Composiciones basadas en desigualdad | El término composición (join) se aplica a cualquier consulta que combina datos  de dos tablas mediante comparación de los valores en una pareja de columnas de las  tablas. Aunque las composiciones basadas en la igualdad entre columnas  correspondientes (equicomposiciones) son con mucho las composiciones más  habituales, SQL también permite componer tablas basándose en otros operadores de  comparación. |
| 7 | Consideraciones SQL para consultas de varias tablas | Las palabras reservadas de SQL aparecen en mayúsculas.  • Los nombres de objetos (tablas, columnas, etcétera) suelen aparecer en  minúsculas.  • La notación lista\_de\_elementos especifica una lista de elementos separados por  comas.  • La barra vertical ( | ) indica la elección entre dos elementos.  • Las llaves ({ }) indican la elección obligatoria entre varios elementos.  • Los corchetes ( [ ] ) encierran un elemento opcional.  • El punto y coma ( ; ) que aparece al final de cada comando, en realidad no forma  parte de la sintaxis del lenguaje SQL pero suele ser un elemento requerido por  las herramientas de cliente para determinar el final del comando SQL y enviar la  órden (sin el ;) al servidor. |
| 8 | La estructura de una composición | La sintaxis es la siguiente:  FROM tabla 1 INNER JOIN table 2 ON tabla1.col1 = tabla2.col2  Ejemplo:  SELECT \*  FROM pedidos INNER JOIN clientes ON pedidos.clie = clientes.numclie |
| 9 | Composiciones externa, interna y completa | Externa: La reunión externa puede verse en este caso como una reunión interna donde no es necesario que el registro hijo tenga informada la clave foránea para ser mostrado, por ejemplo, cuando se mostraban los cursos junto a los profesores que los imparten, como uno de los cursos no tiene padre, es decir, no tiene un profesor asignado, o lo que es lo mismo, el campo ID\_PROFE de la tabla CURSOS está a nulo, este curso no se muestra dado que no satisface la cláusula ON. Bien, este recurso nos ofrece la posibilidad de mostrar estos registros con los campos del registro padre a nulo.  Interna: Esta cláusula está diseñada precisamente para reunir registros de varias tablas, en ella intervienen las claves primarias y foráneas, y no intervienen, o lo hacen en la cláusula WHERE, los filtros propiamente dichos. Veamos una de las consultas que se expuso en la lección anterior usando esta sintaxis.  Completa: Esta operación presenta los resultados de tabla izquierda y tabla derecha aunque alguna no tengan correspondencia en la otra tabla. La tabla combinada contendrá, entonces, todos los registros de ambas tablas y presentará valores nulos NULLs para registros sin pareja. |
| 10 | Subconsultas | La característica de subconsulta de SQL permite utilizar los resultados de una consulta como parte de otra. La capacidad de utilizar una consulta dentro de otra fue la razón original para la palabra “estructurada” en el nombre Lenguaje de Consultas Estructuradas (Structured Query Language-SQL).t |
| 11 | Subconsultas de la cláusula WHERE | Las subconsultas suelen ser utilizadas principalmente en la cláusula WHERE de  una sentencia SQL. Cuando aparece una subconsulta en la cláusula WHERE, ésta  funciona como parte del proceso de selección de filas. |
| 12 | Subconsultas en la cláusula HAVING | Al igual que la cláusula WHERE puede ser utilizada para seleccionar y rechazar  filas individuales que participan en una consulta, la cláusula HAVING puede ser  utilizada para seleccionar y rechazar grupos de filas. El formato de la cláusula HAVING  es análogo al de la cláusula WHERE, consistiendo en la palabra clave HAVING seguida de la condición de búsqueda. Por tanto, la cláusula HAVING especifica una condición  de búsqueda por grupos. |

Conclusión: Se ha visto cómo se pueden hacer varias cosas con las tablas y la base de datos de sql, también se ha visto que se pueden hacer de dos formas por lo menos en general, la primera sería con los join, inner join, etc, y la otra sería la que se enseñó en la teoría que en mi opinión me gusta más esta última, ya que para mí es más sencilla y entendible.

Bibliografía

[Consideraciones](https://www.cartagena99.com/recursos/alumnos/apuntes/Manual-SQL1.pdf)

[Estructura de composición](https://www.aulaclic.es/sql/t_3_4.htm)

[Composiciones externas e internas](http://deletesql.com/viewtopic.php?f=5&t=21)

[Composición completa](https://es.wikipedia.org/wiki/Sentencia_JOIN_en_SQL)

[El resto](http://informatica.uv.es/estguia/ATD/apuntes/teoria/documentos/SQL-I.pdf)