|  |  |
| --- | --- |
| **Construcción de consultas a partir de una especificación**   Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.   Recuerda que la fecha puede indicarse como '01-JAN-2000' o '01/01/00'.   **Importante:** Recuerda que cuando vayas a trabajar con fechas, antes de que realices tus consultas debes ejecutar la instrucción "set dateformat dmy". Basta con que la ejecutes una sola vez para que el manejador sepa que vas a trabajar con ese formato de fechas.   **¿Por qué aparecen varias veces algunas descripciones de material?** | Por que fueron entregados más de una vez el mismo material |
| **Uso del calificador distinct**   En el resultado anterior, observamos que una misma descripción de material aparece varias veces.   Agrega la palabra distinct inmediatamente después de la palabra select a la consulta que planteaste antes.   ¿Qué resultado obtienes en esta ocasión? | Solo se seleccionan las duplas con descripción que no se repite. |
| **Operadores de cadena**   El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.   También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.   SELECT \* FROM productos where Descripcion LIKE 'Si%'   ¿Qué resultado obtienes?  Explica que hace el símbolo '%'.  ¿Qué sucede si la consulta fuera : LIKE 'Si' ?  ¿Qué resultado obtienes?  Explica a qué se debe este comportamiento. | % es un indicador para que no tome en cuenta la comparación después del símbolo.  Cuando le quitas en porcentaje ya no encuentra nada, porque no hay ninguna restricción que sea si |
| DECLARE @foo varchar(40);  DECLARE @bar varchar(40);  SET @foo = '¿Que resultado';  SET @bar = ' ¿¿¿??? '  SET @foo += ' obtienes?';  PRINT @foo + @bar;   **¿Qué resultado obtienes de ejecutar el siguiente código?  ¿Para qué sirve DECLARE?  ¿Cuál es la función de @foo?  ¿Que realiza el operador SET?** | Muestra el texto: “¿Que resultado ¿¿¿??? obtienes?”   Para declarar variables, la función de foo sirve para declarar el tamaño y tipo de string, darle el valor a la variable |
| SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%'; | Muestra los RFC’s que van de A - D |
| SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%'; | Muestra todos los RFC’s excepto los que comienzan con ‘A’ |
| SELECT Numero FROM Entregan WHERE Numero LIKE '\_\_\_6'; | Muestra los números en que en digito 4 (después de 3 guiones bajos) coincida con el 6 |
| ¿Cómo filtrarías rangos de fechas?  USANDO BETWEEN | SELECT Clave,RFC,Numero,Fecha,Cantidad  FROM Entregan  WHERE Fecha Between '2000/01/01' and '2000/12/31'  Para Materiales entregados en el 2000 |
| SELECT RFC,Cantidad, Fecha,Numero  FROM [Entregan]  WHERE [Numero] Between 5000 and 5010 AND  Exists ( SELECT [RFC]  FROM [Proveedores]  WHERE RazonSocial LIKE 'La%' and [Entregan].[RFC] = [Proveedores].[RFC] )   ¿Qué hace la consulta?  ¿Qué función tiene el paréntesis ( ) después de EXISTS? | Muestra el *RFC,Cantidad, Fecha,Numero*donde numero va de 5000 – 5010, en los cuales la razón social empieza por “La”. |
| ¿Qué hace la siguiente sentencia? Explica por qué.   SELECT TOP 2 \* FROM Proyectos | Muestra las columnas de los primeros 2 de la tabla ordenado por el numero ascendente |
| ¿Qué sucede con la siguiente consulta? Explica por qué.   SELECT TOP Numero FROM Proyectos | Muestra la columna Numero del primer valor de la tabla |
|  |  |
|  |  |