

SELECCION DE CAMPOS
<i>Edad, nombre, puesto de todos los empleados</i>
SELECT e.edad, nombre, e.puesto FROM empleados e; Una vez definido e, empleados no se puede usar como identificador
<i>Identificador, ciudad, región de las oficinas</i>
SELECT o.oficina, o.ciudad, region FROM oficinas o;
RENOMBRADO DE CAMPOS
<i>Mostrar el nombre de los clientes como “Nombre del cliente” y su crédito como “Credito máximo”</i>
SELECT c.nombre AS ‘Nombre del cliente’, limitecredito AS ‘Credito maximo’ FROM clientes c; Usar comillas simples
SELECCIONAR TODOS LOS CAMPOS
<i>Mostrar todos los datos de la tabla empleados</i>
SELECT * FROM empleados;
ORDENACION DE CONSULTAS
<i>Obtener las oficinas ordenadas por región y dentro de cada región por ciudad, si hay mas de una oficina en la misma ciudad, ordenar por numero de oficina mayor</i>
SELECT región, ciudad, oficina FROM oficinas ORDER BY región, ciudad, oficina DESC; Si no se especifica ordenación ASC   DESC, por defecto usa ASC
LIMITACION DEL NUMERO DE REGISTROS
<i>Listar los cuatro pedidos mas caros (con un importe mayor).</i>
SELECT * FROM pedidos ORDER BY importe DESC LIMIT 4;
OPERADORES ARITMETICOS
<i>Listar el precio de los productos y el precio con IVA (21%)</i>
SELECT precio, precio * 1.21 AS ‘IVA’ FROM productos; Los decimales utilizan punto
FUNCIONES DE FECHA
<i>Fecha de hoy</i>
SELECT CURRENT_DATE(); Se puede ignorar los parentesis
<i>Mes actual</i>
SELECT MONTH(CURRENT_DATE);
<i>Fecha de hoy</i>
SELECT CURRENT_DATE(); Se puede ignorar los paréntesis de funciones sin entrada de parametros

<i>Listar de cada empleado su nombre, días trabajados y su año de nacimiento</i>
<pre> SELECT     nombre, DATEDIFF(CURRENT_DATE, DATE(contrato)) AS 'Días trabajados',     YEAR(CURRENT_DATE) - edad AS 'Año de nacimiento' FROM empleados;</pre>
<i>Obtener la fecha en formato “(&lt;nombre_día&gt; &lt;día&gt; de &lt;nombre_mes&gt; del año &lt;año&gt;”</i>
<pre> SELECT     DATE_FORMAT(CURRENT_DATE, “(%W) %d de %M del año %Y”);</pre> <p>Valen comillas dobles y simples</p>
<i>Obtener la hora en formato “Las &lt;hora&gt; horas y &lt;minuto&gt; minutos”</i>
<pre> SELECT     DATE_FORMAT(NOW(), “Las %H horas y %i minutos”);</pre> <p>Valen comillas dobles y simples</p>
<b>FUNCIONES DE CADENA</b>
<i>Longitud de un campo</i>
<pre> SELECT nombre, CHAR_LENGTH('nombre') AS 'Longitud' FROM empleados;</pre>
<i>Concatenar un campo</i>
<pre> SELECT CONCAT('Don ', nombre) FROM empleados;</pre>
<i>Reemplazar</i>
<pre> SELECT REPLACE(nombre, 'e', '@') FROM clientes;</pre>
<i>Mostrar los 10 caracteres por la derecha</i>
<pre> SELECT SUBSTRING(“Mi perro se llama Perico Perez”, -10);</pre>
<b>CONDICIONALES</b>
<i>Listar los empleados que no tienen oficina asignada mayores de 30</i>
<pre> SELECT numemp FROM empleados WHERE oficina IS NULL &amp;&amp; edad&gt;30;</pre>
<i>Listar los datos de las oficinas de las regiones del norte y del este</i>
<pre> SELECT * FROM oficinas WHERE region='norte' OR region='este' ORDER BY region DESC;</pre>
<i>Mostrar los empleados que trabajan en las oficinas 11, 12, 23, 24</i>
<pre> SELECT * FROM empleados WHERE oficina IN (11, 12, 23, 24);</pre>
<i>Mostrar empleados que tengan una ‘r’ en el nombre</i>
<pre> SELECT * FROM empleados WHERE nombre LIKE ‘%r%’;</pre>
<i>Empleados que empiecen por ‘V’</i>
<pre> SELECT * FROM empleados WHERE nombre LIKE ‘V%’;</pre>
<b>CONSULTAS MULTITABLAS</b>
<i>Mostrar los empleados junto a sus oficinas</i>
<pre> SELECT * FROM empleados e JOIN oficinas o ON e.oficina = o.oficina;</pre>

<i>Una lista con las oficinas del este con sus empleados y otra lista con oficinas sin empleados</i>
<pre> SELECT oficinas.oficina, ciudad, numemp, nombre FROM oficinas JOIN empleados ON oficinas.oficina=empleados.oficina WHERE region='este'; SELECT oficinas.oficina, ciudad, numemp, nombre FROM oficinas LEFT JOIN empleados ON oficinas.oficina=empleados.oficina WHERE region='este'; </pre>
<i>Listar los pedidos superiores a 2.500, incluyendo el nombre del responsable del pedido y el nombre del cliente. Ordenar por nombre del cliente</i>
<pre> SELECT * FROM (pedidos JOIN clientes ON pedidos.clie=clientes.numclie) JOIN empleados ON pedidos.resp=empleados.numemp WHERE importe&gt;2500 ORDER BY clientes.nombre; </pre> <p>Union de pedidos con clientes con empleados</p>
<i>Listar ordenados por el nombre los empleados que han realizado algún pedido</i>
<pre> SELECT DISTINCT nombre FROM empleados JOIN pedidos ON empleados.numemp=pedidos.resp ORDER BY empleados.nombre; </pre> <p>Distinct solo te muestra valores unicos</p>
<i>Listar los empleados con una cuota superior a la de su jefe</i>
<pre> SELECT empleados.*, jefe.numemp, jefe.nombre, jefe.cuota FROM empleados JOIN empleados AS jefe ON empleados.jefe = jefe.nuemp WHERE empleados.cuota &gt; jefe.cuota; </pre>
<i>Mostrar las oficinas que no tienen director o están en la región sur</i>
<pre> SELECT * FROM oficinas WHERE oficinas.dir IS NULL UNION SELECT * FROM oficinas WHERE oficinas.region='sur'; </pre>
<b>CONSULTAS DE AGREGADOS</b>
<i>Cuota media y ventas medias de empleados</i>
<pre> SELECT AVG(cuota) AS "Media cuota", AVG(ventas) AS "Media ventas" FROM empleados; </pre>
<i>Empleado mas joven y viejo</i>
<pre> SELECT MIN(edad) AS "Edad minima", MAX(edad) AS "Edad maxima" FROM empleados; </pre>
<i>Importe medio de pedidos, importe total de pedidos y el precio medio de venta</i>
<pre> SELECT AVG(importe) AS 'Importe medio', SUM(importe) AS 'Importe total', AVG(importe/cant) AS 'Media precio unitario' FROM pedidos; </pre>

<i>Importe medio de pedidos, importe total de pedidos y el precio medio de venta</i>
SELECT AVG(importe) AS 'Importe medio', SUM(importe) AS 'Importe total', AVG(importe/cant) AS 'Media precio unitario' FROM pedidos;
<i>Para cada empleado cuyo pedido suma mas de 3000, hallar su importe medio</i>
SELECT resp AS 'Numero de empleado', COUNT(resp) AS 'Numero de pedidos', AVG(importe) AS 'Media de los pedidos' FROM pedidos GROUP BY resp HAVING SUM(importe) > 3000;
<i>Listar de cada producto, su descripción, precio y cantidad total pedida, incluyendo sólo los productos cuya cantidad total pedida sea superior al 75% del stock; y ordenado por cantidad total pedida.</i>
SELECT descripcion, precio, SUM(cant) FROM productos JOIN pedidos ON pedidos.fab=productos.idfab && pedidos.producto=productos.idproducto GROUP BY idfab, idproducto, descripcion, existencias HAVING SUM(cant) > (075*existencias) ORDER BY SUM(cant);
<b>SUBCONSULTAS</b>
<i>Nombres de clientes que tienen asignado como responsable Alvaro Aluminio</i>
SELECT * FROM clientes WHERE resp = (SELECT numemp FROM empleados WHERE nombre = 'Alvaro Aluminio');
<i>Mostrar información de los productos cuyas existencias estén por debajo de la existencia media de los productos.</i>
SELECT * FROM productos WHERE existencias < (SELECT AVG(existencias) FROM productos);
<i>Listar los empleados (numemp, nombre, y no de oficina) que trabajan en oficinas “buenas” (las que tienen ventas superiores a su objetivo).</i>
SELECT numemp,nombre, oficina FROM empleados WHERE oficina in (SELECT oficina FROM oficinas WHERE ventas>objetivo);
<i>Mostrar empleados que no son directores de ninguna oficina</i>
SELECT * FROM empleados WHERE nombre NOT IN (SELECT dir FROM oficinas WHERE dir IS NOT NULL);
<i>Escribir una consulta que muestre los empleados cuyo primer nombre coincide con el primer nombre de algún cliente.</i>
SELECT LEFT(nombre, LOCATE(' ', nombre)-1) FROM empleados WHERE LEFT(nombre, LOCATE(' ', nombre)-1) IN (SELECT LEFT(nombre, LOCATE(' ', nombre)-1) FROM clientes);

SINTAXIS	
SELECT [<predicado>] <campos> FROM <expresión_tabla>  [WHERE . . .]  [GROUP BY . . .] [HAVING . . .]  [ORDER BY . . .]  [LIMIT . . .]	<predicado>: ALL   DISTINCT   DISTINCTROW <campos>: [tabla].*   [tabla].campo1 [AS alias1] [, . . .] <expresión_tabla>: Nombre_tabla [JOIN . . .]
FUNCIONES DE FECHA	
DATE_FORMAT (<fecha>, <formato>)	<formato>: %y, %Y (Año) %m, %M (Mes) %d, %D (Día) %h, %H (Hora) %i, %I (Minuto) %w, %W (Día de la semana)
CURRENT_DATE ()	YEAR (<fecha>)
NOW ()	MONTH (<fecha>)
CURTIME ()	DAYOFMONTH (<fecha>)
DAYNAME (<fecha>)	MONTHNAME (<fecha>)
FUNCIONES DE CADENA	
[NOT] LIKE	CHAR_LENGTH (<cadena>)
CONCAT (<listaCadenas>)	CONCAT_WS (<separador>, <listaCadenas>)
FIELD (<cadena>, <listaCadenas>)	REPEAT (<cadena>, <numero>)
REPLACE (<cadena>, origen, destino)	RIGHT (<cadena>, <long>)
COMPARADORES	
= (igual, devuelve NULL) < = > (igual, devuelve 1) <>, != (distinto) <, <=, >, >=	IS [NOT] <valor> IS [NOT] NULL <expr> [NOT] BETWEEN <min> AND <max> <expr> [NOT] IN (<value>, . . .)