|  |
| --- |
| **SELECCION DE CAMPOS** |
| *Edad, nombre, puesto de todos los empleados* |
| SELECT  e.edad, nombre, e.puesto  FROM  empleados e;  Una vez definido e, empleados no se puede usar como identificador |
| *Identificador, ciudad, región de las oficinas* |
| SELECT o.oficina, o.ciudad, region  FROM oficinas o; |
| **RENOMBRADO DE CAMPOS** |
| *Mostrar el nombre de los clientes como “Nombre del cliente” y su crédito como “Credito máximo”* |
| SELECT  c.nombre AS ‘Nombre del cliente’, limitecredito AS ‘Credito maximo’  FROM  clientes c;  Usar comillas simples |
| **SELECCIONAR TODOS LOS CAMPOS** |
| *Mostrar todos los datos de la tabla empleados* |
| SELECT  \*  FROM empleados; |
| **ORDENACION DE CONSULTAS** |
| *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* |
| 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** |
| *Listar los cuatro pedidos mas caros (con un importe mayor).* |
| SELECT \*  FROM pedidos  ORDER BY importe DESC  LIMIT 4; |
| **OPERADORES ARITMETICOS** |
| *Listar el precio de los productos y el precio con IVA (21%)* |
| SELECT precio, precio \* 1.21 AS ‘IVA’  FROM productos;  Los decimales utilizan punto |
| **FUNCIONES DE FECHA** |
| *Fecha de hoy* |
| SELECT CURRENT\_DATE();  Se puede ignorar los parentesis |
| *Mes actual* |
| SELECT MONTH(CURRENT\_DATE); |
| *Fecha de hoy* |
| SELECT CURRENT\_DATE();  Se puede ignorar los paréntesis de funciones sin entrada de parametros |
|  |
| *Listar de cada empleado su nombre, días trabajados y su año de nacimiento* |
| SELECT  nombre, DATEDIFF(CURRENT\_DATE, DATE(contrato)) AS 'Dias trabajados',  YEAR(CURRENT\_DATE) - edad AS 'Año de nacimiento'  FROM empleados; |
| *Obtener la fecha en formato “(<nombre\_dia>) <dia> de <nombre\_mes> del año <año>”* |
| SELECT  DATE\_FORMAT(CURRENT\_DATE, “(%W) %d de %M del año %Y”);  Valen comillas dobles y simples |
| *Obtener la hora en formato “Las <hora> horas y <minuto> minutos”* |
| SELECT  DATE\_FORMAT(NOW(), “Las %H horas y %i minutos”);  Valen comillas dobles y simples |
| **FUNCIONES DE CADENA** |
| *Longitud de un campo* |
| SELECT nombre, CHAR\_LENGTH(‘nombre’) AS ‘Longitud’  FROM empleados; |
| *Concatenar un campo* |
| SELECT CONCAT(‘Don ’, nombre)  FROM empleados; |
| *Reemplazar* |
| SELECT REPLACE(nombre, ‘e’, ‘@’)  FROM clientes; |
| *Mostrar los 10 caracteres por la derecha* |
| SELECT SUBSTRING(“Mi perro se llama Perico Perez”, -10); |
| **CONDICIONALES** |
| *Listar los empleados que no tienen oficina asignada mayores de 30* |
| SELECT numemp  FROM empleados  WHERE oficina IS NULL && edad>30; |
| *Listar los datos de las oficinas de las regiones del norte y del este* |
| SELECT \*  FROM oficinas  WHERE region=’norte’ OR region=’este’  ORDER BY region DESC; |
| *Mostrar los empleados que trabajan en las oficinas 11, 12, 23, 24* |
| SELECT \*  FROM empleados  WHERE oficina IN (11, 12, 23, 24); |
| *Mostrar empleados que tengan una ‘r’ en el nombre* |
| SELECT \*  FROM empleados  WHERE nombre LIKE ‘%r%’; |
| *Empleados que empiecen por ‘V’* |
| SELECT \*  FROM empleados  WHERE nombre LIKE ‘V%’; |
| **CONSULTAS MULTITABLAS** |
| *Mostrar los empleados junto a sus oficinas* |
| SELECT \*  FROM empleados e JOIN oficinas o  ON e.oficina = o.oficina; |
|  |
| *Una lista con las oficinas del este con sus empleados y otra lista con oficinas sin empleados* |
| 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'; |
| *Listar los pedidos superiores a 2.500, incluyendo el nombre del responsable del pedido y el nombre del cliente. Ordenar por nombre del cliente* |
| SELECT \*  FROM (pedidos JOIN clientes  ON pedidos.clie=clientes.numclie)  JOIN empleados  ON pedidos.resp=empleados.numemp  WHERE importe>2500  ORDER BY clientes.nombre;  Union de pedidos con clientes con empleados |
| *Listar ordenados por el nombre los empleados que han realizado algún pedido* |
| SELECT DISTINCT nombre  FROM empleados JOIN pedidos  ON empleados.numemp=pedidos.resp  ORDER BY empleados.nombre;  Distinct solo te muestra valores unicos |
| *Listar los empleados con una cuota superior a la de su jefe* |
| SELECT empleados.\*, jefe.numemp, jefe.nombre, jefe.cuota  FROM empleados JOIN empleados AS jefe  ON empleados.jefe = jefe.nuemp  WHERE empleados.cuota > jefe.cuota; |
| *Mostrar las oficinas que no tienen director o están en la región sur* |
| SELECT \*  FROM oficinas  WHERE oficinas.dir IS NULL  UNION  SELECT \*  FROM oficinas  WHERE oficinas.region=’sur’; |
| **CONSULTAS DE AGREGADOS** |
| *Cuota media y ventas medias de empleados* |
| SELECT AVG(cuota) AS “Media cuota”,  AVG(ventas) AS “Media ventas”  FROM empleados; |
| *Empleado mas joven y viejo* |
| SELECT MIN(edad) AS “Edad minima”,  MAX(edad) AS “Edad maxima”  FROM empleados; |
| *Importe medio de pedidos, importe total de pedidos y el precio medio de venta* |
| SELECT AVG(importe) AS 'Importe medio',  SUM(importe) AS 'Importe total',  AVG(importe/cant) AS 'Media precio unitario'  FROM pedidos; |
|  |
| *Importe medio de pedidos, importe total de pedidos y el precio medio de venta* |
| SELECT AVG(importe) AS 'Importe medio',  SUM(importe) AS 'Importe total',  AVG(importe/cant) AS 'Media precio unitario'  FROM pedidos; |
| *Para cada empleado cuyo pedido suma mas de 3000, hallar su importe medio* |
| 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; |
| *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.* |
| 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); |
| **SUBCONSULTAS** |
| *Nombres de clientes que tienen asignado como responsable Alvaro Aluminio* |
| SELECT \*  FROM clientes  WHERE resp = (SELECT numemp  FROM empleados  WHERE nombre = ‘Alvaro Aluminio’); |
| *Mostrar información de los productos cuyas existencias estén por debajo de la existencia media de los productos.* |
| SELECT \*  FROM productos  WHERE existencias < (SELECT AVG(existencias)  FROM productos); |
| *Listar los empleados (numemp, nombre, y no de oficina) que trabajan en oficinas “buenas” (las que tienen ventas superiores a su objetivo).* |
| SELECT numemp,nombre, oficina  FROM empleados  WHERE oficina in (SELECT oficina  FROM oficinas  WHERE ventas>objetivo); |
| *Mostrar empleados que no son directores de ninguna oficina* |
| SELECT \*  FROM empleados  WHERE nombre NOT IN (SELECT dir  FROM oficinas  WHERE dir IS NOT NULL); |
| *Escribir una consulta que muestre los empleados cuyo primer nombre coincide con el primer*  *nombre de algún cliente.* |
| 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 (Dia)  %h, %H (Hora)  %i, %I (Minuto)  %w, %W (Dia 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>, . . .) |