**Puntuación máxima 10, para superar la prueba es necesario obtener un 5/10**

**Rellenar en el mismo Word las imágenes y código SQL correspondientes a cada pregunta.**

1. Conectado a una base de datos se encuentran las siguientes tablas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabla001** | |  | **TablaABC** | |
| **ID** | **VALOR** |  | **COD** | **TEXTO** |
| A02 | F1B1 |  | A01 | 3T |
| B04 | X2C2 |  | D08 | 30G |
| C06 | B3D3 |  | B04 | 5G |
| D08 | D4E4 |  | C03 | 20Z |

* 1. Rellenar la tabla con los resultados de la siguiente consulta:

|  |  |  |  |
| --- | --- | --- | --- |
| **SELECT** valor, texto |  | **valor** | **texto** |
| **FROM** Tabla001 |  | X2C2 | 5G |
| **INNER JOIN** TablaABC |  | D4E4 | 30G |
| **ON** id = cod |  |  |  |
| **ORDER BY** valor, texto |  |  |  |
|  |  |  |  |

1. **punto**)
   1. Rellenar la tabla con los resultados de la siguiente consulta:

|  |  |  |  |
| --- | --- | --- | --- |
| **SELECT** valor, texto |  | **valor** | **texto** |
| **FROM** Tabla001 |  | F1B1 | NULL |
| **LEFT JOIN** TablaABC |  | X2C2 | 5G |
| **ON** id = cod |  | B3D3 | NULL |
| **ORDER BY** valor, texto |  | D4E4 | 30G |
|  |  |  |  |

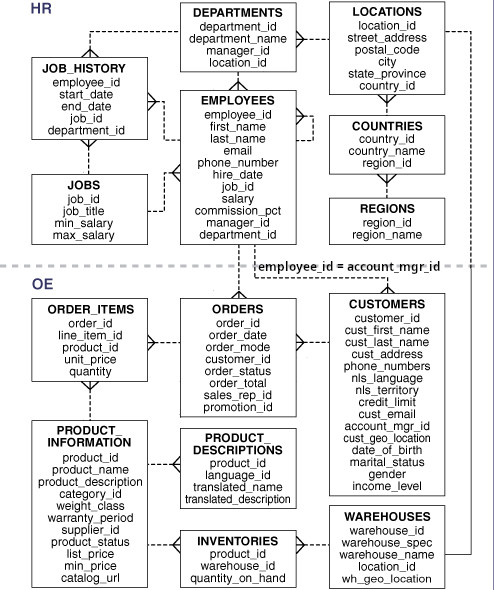
1. **punto**)
   1. Rellenar la tabla con los resultados de la siguiente consulta:

|  |  |  |  |
| --- | --- | --- | --- |
| **SELECT** valor, texto |  | **valor** | **texto** |
| **FROM** Tabla001 |  | NULL | 3T |
| **RIGHT JOIN** TablaABC |  | X2C2 | 5G |
| **ON** id = cod |  | NULL | 20Z |
| **ORDER BY** valor, texto |  | D4E4 | 30G |
|  |  |  |  |

(**1 punto**)

1. Dado esquema de base de datos HR ampliado:

(HR = Human Resources; OE = Order Entries)



La relación entre la tabla *employees* (empleados) y *customers* (clientes) se hace a través de la igualdad employee\_id = account\_mgr\_id cuya cardinalidad es 1:n.

* 1. Escribir la sentencia SQL de base de datos que obtenga el nombre y apellido de los clientes (CUSTOMERS) que hayan hecho algún pedido (ORDERS) usando alguna promoción (promotion\_id tiene valor, no es nulo) y el nombre y apellidos del empleado (EMPLOYEES) que se lo ha vendido.

La consulta debe estar ordenada por el apellido en orden descendente y luego por el nombre de manera ascendente.

**(1 punto)**

SELECT c.cust\_first\_name, c.cust\_last\_name, e.first\_name, e.last\_name

FROM customers c

JOIN orders o

ON c.customer\_id = o.customer\_id

JOIN employees e

ON c.account\_mgr\_id = e.employee\_id

WHERE o.promotion\_id != null

ORDER BY 4 DESC, 3 ASC;

* 1. Escribir la sentencia SQL de base de datos que obtenga todas las ciudades (LOCATIONS), países (COUNTRIES) y regiones (REGIONS) de la base de datos HR en un único listado y una única columna.

La consulta debe estar ordenada de manera ascendente.

**(1 punto)**

SELECT l.city AS Nombre

FROM locations l

JOIN countries c

ON l.country\_id = c.country\_id

JOIN regions r

ON c.region\_id = r.region\_id

UNION

SELECT c.country\_name AS Nombre

FROM locations l

JOIN countries c

ON l.country\_id = c.country\_id

JOIN regions r

ON c.region\_id = r.region\_id

UNION

SELECT r.region\_name AS Nombre

FROM locations l

JOIN countries c

ON l.country\_id = c.country\_id

JOIN regions r

ON c.region\_id = r.region\_id

ORDER BY ASC;

* 1. Escribir la sentencia SQL de base de datos que obtenga el salario anual de todos los empleados.

La consulta debe mostrar el nombre y apellidos del empleado y su salario, ordenado por el salario de modo ascendente.

El salario que hay en la tabla EMPLOYEES es mensual, por lo que hay que multiplicarlo por 12 mensualidades.

Hay que sumarle la comisión, que es anual: commission\_pct \* 300

Hay que considerar que hay empleados que no tienen comisión.

**(1 punto)**

SELECT e.first\_name AS 'Nombre',

e.last\_name AS 'Apellido',

((e.salary \* 12) + (commission\_pct \* 300)) AS 'Salario'

FROM employees e;

* 1. Escribir la sentencia SQL de base de datos que obtenga el nombre y apellido de todos los empleados, el nombre del departamento al que pertenecen actualmente y el nombre de los departamentos donde han estado. Debe añadirse una columna adicional que indique si el departamento mostrado es el actual o es pasado.

El resultado debe estar ordenado por el apellido del empleado, el nombre y la fecha en la que se incorporó a cada departamento por el que ha pasado.

**(1 punto)**

SELECT CONCAT(e.first\_name,'', e.last\_name) AS 'Empleado',

d.department\_name as departamento, "actual" AS 'Tipo'

FROM employees e

JOIN departments d

    ON e.department\_id = d.department\_id

UNION

SELECT CONCAT(e.first\_name,'', e.last\_name) AS 'Empleado',

d.department\_name AS departamento, "historico" AS 'Tipo'

FROM employees e

JOIN job\_history jh

    ON e.employee\_id = jh.employee\_id

JOIN departments d

    ON d.department\_id = jh.department\_id

ORDER BY 1;

* 1. Escribir la sentencia SQL de base de datos que obtenga el nombre y apellidos de todos los empleados cuyo salario sea mayor que el de Peter Vargas.

Mostrar en esta consulta el nombre del departamento al que pertenece cada empleado y ordenar la consulta en modo ascendente por el nombre del departamento.

**(1 punto)**

SELECT CONCAT(e.first\_name, ' ', e.last\_name) AS 'Empleado',

d.department\_name AS 'Departamento'

FROM employees e

JOIN departments d

ON e.department\_id = d.department\_id

WHERE e.salary >(

                    SELECT e.salary

                    FROM employees e

                    WHERE e.first\_name = 'Peter'

                    AND e.last\_name = 'Vargas'

                )

ORDER BY d.department\_name ASC;

* 1. Escribir la sentencia SQL de base de datos que muestre el nombre de todos los trabajos y los nombres y apellidos de los empleados que tengan esos trabajos.

La consulta debe estar ordenada por el nombre de los trabajos y por el apellido de los empleados.

**(1 punto)**

SELECT j.job\_title, e.first\_name, e.last\_name

FROM jobs j

JOIN employees e

ON j.job\_id = e.job\_id

ORDER BY 1, 2;

* 1. Escribir la sentencia SQL de base de datos que muestre el nombre de todos los departamentos, el nombre y apellidos del manager de cada departamento y el nombre del trabajo de cada encargado / manager.

El resultado debe estar ordenado por el nombre del trabajo y el nombre del departamento.

**(1 punto)**

SELECT e.first\_name, e.last\_name, j.job\_title, d.department\_name

FROM employees e

JOIN jobs j

ON e.job\_id = j.job\_id

JOIN departments d

ON e.manager\_id = d.manager\_id

ORDER BY 1, 4;