

REPASO COMBINACIONES, JOINS CON LAS TABLAS DEL USUARIO EMPLOYEES

1. Muestra cuántas unidades se han vendido de cada producto, visualizando código de producto y total de unidades.

```
SQL> select product_id, sum(quantity)
      from item
      group by product_id;
```

2. Realiza la consulta anterior pero mostrando de cada producto, su nombre y total de unidades vendidas.

```
SQL> select product.description, sum(item.quantity)
      from item, product
      where item.product_id=product.product_id
      group by product.description;
```

3. Nombre de empleados y nombre del depto donde trabajan de aquellos que cobran comisión.

```
SQL>select employee.first_name, employee.department_id,department.name,
      employee.commission
      from employee, department
      where employee.department_id=department.department_id
      and commission is not null and commission !=0;
```

4 Cuantos empleados hay en el depto de ventas.

```
SQL>select count(*)
      from employee, department
      where employee.department_id=department.department_id
      and department.name='SALES';
```

----- O TAMBIEN-----

```
SQL>select count(*)
      from employee
      where department_id in (select department_id
                             from department
                             where name='SALES');
```

5. Muestra de cada empleado, su nombre y nombre de función que realiza.

```
SQL>select employee.first_name, job.function
      from employee, job
      where employee.job_id=job.job_id;
```

6. Muestra el código y nombre de los departamentos donde trabaja algún empleado, puede que haya algún departamento que no tiene empleados.

```
SQL>select distinct(employee.department_id), department.name
      from employee, department
      where employee.department_id=department.department_id;
```

7. Muestra de cada departamento, su nombre y número de empleados que trabajan en él; considera los departamentos con el mismo nombre y en distintas localidades, como un único departamento.

```
SQL>select (department.name), count(*)
      from employee, department
      where employee.department_id=department.department_id
      group by department.name;
```

8. Visualiza el nombre de los clientes junto al nombre del empleado que es su representante de ventas.

```
SQL>select customer.name, employee.first_name
      from customer, employee
      where employee.employee_id=customer.salesperson_id;
```

9. Muestra el nombre de los clientes cuyo representante de ventas es WARD.

```
SQL>select customer.name
      from customer, employee
      where customer.salesperson_id=employee.employee_id
      and employee.last_name='WARD';
```

```
SQL>select c.name from customer c where
      c.salesperson_id IN(select e.employee_id from customer c, employee e where
      e.employee_id=c.salesperson_id and last_name ='WARD');
```

10. Muestra el nombre de los empleados que son vendedores (SALESPERSON)

```
SQL>select employee.first_name
      from employee, job
      where employee.job_id=job.job_id
      and job.function='SALESPERSON';
```

11. ¿Cuántos empleados son vendedores?

```
SQL>select count(*)
      from employee, job
      where employee.job_id=job.job_id
      and job.function='SALESPERSON';
```

12. Muestra de cada producto, su nombre y precio (LIST_PRICE) que se le aplica actualmente.

```
SQL>select product.description, price.list_price
      from product, price
      where product.product_id=price.product_id
      and price.end_date is null;
```

13. Muestra de cada cliente y sin repetidos, los códigos de los distintos productos que ha solicitado en sus pedidos.

```
SQL>select distinct(item.product_id)
      from customer, sales_order, item
      where customer.customer_id=sales_order.customer_id
      and sales_order.order_id=item.order_id;
```

14. ¿Cuál es el valor del pedido de mayor valor que ha realizado el cliente JOCKSPORTS?

```
SQL>select max(sales_order.total)
      from sales_order, customer
      where sales_order.customer_id=customer.customer_id
      and customer.name='JOCKSPORTS';
```

15. Visualiza de cada cliente, su nombre y el valor del pedido de mayor valor que ha realizado hasta el momento.

```
SQL>select customer.name, max(sales_order.total)
      from customer, sales_order
      where customer.customer_id=sales_order.customer_id
      group by customer.name;
```

16. Visualiza de cada cliente, su nombre y la suma de los totales de todos los pedidos que ha realizado.

```
SQL>select customer.name, sum(sales_order.total)
      from customer, sales_order
      where customer.customer_id=sales_order.customer_id
      group by customer.name;
```

COMBINACION NO COMUN

17. Mostrar de cada empleado su nombre y el grado salarial que le corresponde.

```
SQL>select employee.first_name, salary_grade.grade_id
```

```
from employee, salary_grade
where employee.salary between lower_bound and upper_bound;
```

AUTOCOMBINACION

18. Nombre de cada empleado y al lado el nombre de la persona que es su jefe.

```
SQL>select empleado.first_name, jefe.first_name
      from employee empleado, employee jefe
      where empleado.manager_id=jefe.employee_id;
```

19. Nombre de cada empleado y al lado el nombre de la persona que es su jefe, y si no tiene que aparezca "No tiene jefe".

```
SQL>select empleado.first_name, nvl (jefe.first_name,'no tiene jefe')
      from employee empleado, employee jefe
      where empleado.manager_id=jefe.employee_id(+);
```

FIRST_NAME	NVL(J.FIRST_NAM
PAUL	JEAN
FRANCIS	No tiene jefe
MARY	MARION
DIANE	DONALD

20. Mostrar de cada depto su código, su nombre y el nº de empleados que tiene, pero si no tiene empleados que salga un 0.

```
SQL>select department.department_id, department.name,
count(employee.employee_id)
      from employee, department
      where employee.department_id(+) = department.department_id
      group by department.department_id, department.name;
```

21. Mostrar el nombre de cada empleado y nombre de la localidad donde trabaja.

```
SQL>select employee.first_name, location.regional_group
      from employee, department, location
      where employee.department_id=department.department_id
      and department.location_id=location.location_id;
```

22. Muestra de cada cliente y sin repetidos, los nombres de los productos que ha solicitado en sus pedidos.

```
SQL>select customer.name, (product.description)
      from customer, sales_order, item, product
      where customer.customer_id=sales_order.customer_id
      and sales_order.order_id=item.order_id
      and item.product_id=product.product_id
      group by customer.name, product.description;
```

23. Calcula cuántos empleados trabajan en el departamento de ventas (SALES) de CHICAGO.

```
SQL>select count(*)
      from employee, department, location
      where employee.department_id=department.department_id
      and department.location_id=location.location_id
      and department.name='SALES'
      and location.regional_group='CHICAGO';
```

24. Muestra de cada departamento y de cada localidad, cuántos trabajadores tienen; muestra nombre de localidad, nombre de departamento y número de empleados.

```
SQL>select department.name, location.regional_group, count(*)
      from employee, department, location
      where employee.department_id=department.department_id
      and department.location_id=location.location_id
      group by location.regional_group, department.name;
```

25. Muestra el nombre del vendedor con más clientes.

```
SQL>select employee.first_name
      from employee, customer
      where employee.employee_id=customer.salesperson_id
      group by employee.first_name
      having count(*)= (select max(count(*))
                        from customer
                        group by salesperson_id);
```

26. Muestra el nombre del departamento cuya suma de salarios de empleados es la mayor de todas las sumas.

```
SQL>select department.department_id,department.name
      from employee, department
      where employee.department_id=department.department_id
      group by department.name, department.department_id
      having sum(employee.salary)= (select max(sum(salary))
                                    from employee
                                    group by department_id);
```

27. Muestra el nombre del cliente que ha realizado el mayor número de pedidos.

```
SQL>select customer.name
      from customer, sales_order
      where customer.customer_id=sales_order.customer_id
      group by customer.name
      having count(*) = ( select max(count(*))
                        from sales_order
                        group by customer_id );
```

28. Muestra el nombre del cliente que realizo el pedido de mayor valor.

```
SQL>select customer.name
      from customer, sales_order
      where customer.customer_id=sales_order.customer_id
      and sales_order.total = ( select max(total)
                              from sales_order);
```

29. Muestra el nombre del producto más caro.

```
SQL>select product.description
      from product, price
      where product.product_id=price.product_id
      and price.list_price = ( select max(list_price)
                              from price);
```

30. Muestra el nombre del producto más vendido.

```
SQL>select distinct(product.description), sum(item.quantity)
      from product, item
      where product.product_id=item.product_id
      group by product.description
      having sum(item.quantity) = ( select max(sum(quantity))
                                   from item
                                   group by product_id);
```

31. Muestra un listado con el nombre de los vendedores y la comisión que obtendrían si se les pagara un 25% del total facturado a sus clientes.

```
SQL>select employee.first_name, sum(sales_order.total)*0.25
      from employee, customer,sales_order
      where employee.employee_id=customer.salesperson_id
      and customer.customer_id=sales_order.customer_id
      group by employee.first_name;
```

32. Muestra un listado con el nombre de los vendedores y la comisión que obtendrían si se les pagara un 25% del total facturado a sus clientes. Para aquellos vendedores que no han vendido, se debe visualizar cero como comisión.

```
SQL>select employee.first_name, nvl(sum(sales_order.total)*0.25,0)
```

```
from employee, customer,sales_order
where employee.employee_id=customer.salesperson_id and
customer.customer_id=sales_order.customer_id (+)
group by employee.first_name;
```

33. Muestra el nombre del cliente que ha comprado más de 100 unidades del producto 'ACE TENNIS RACKET II'.

```
SQL>select customer.name
      from product, customer, item, sales_order
      where customer.customer_id=sales_order.customer_id
      and sales_order.order_id=item.order_id
      and item.product_id=product.product_id
      and product.description='ACE TENNIS RACKET II'
      group by customer.name
      having sum(item.quantity)>100;
```

34. ¿Cuántos clientes hay que sólo han realizado un pedido?

```
SQL>select count(count(*))
      from customer, sales_order
      where customer.customer_id=sales_order.customer_id
      group by customer.customer_id
      having count(*)=1;
```

35. ¿Cuánto dinero se ha recaudado con la venta de 'ACE TENNIS RACKET I'?

```
SQL>select sum(item.total)
      from product, item
      where product.product_id=item.product_id
      and product.description='ACE TENNIS RACKET I';
```

36. ¿Qué cliente o clientes realizaron el primer pedido a la empresa?

```
SQL>select customer.name
      from customer, sales_order
      where customer.customer_id = sales_order.customer_id
      and sales_order.order_date = ( select min(order_date)
                                     from sales_order);
```

NAME

THE POWER FORWARD

37. Muestra el nombre de cada cliente y el código del pedido (ORDER_ID) que, entre todos los pedidos realizados por el cliente, es el más caro (el de mayor valor en el campo TOTAL).

```
SQL>select customer.name, sales_order.order_id
from customer, sales_order
where customer.customer_id=sales_order.customer_id
and sales_order.total = ( select max(s2.total)
                        from sales_order s2
                        where sales_order.customer_id=s2.customer_id);
```

38. Muestra de cada departamento, su código, nombre, localidad y nombre del empleado más antiguo en ese departamento.

```
SQL>select department.department_id, department.name,
location.regional_group,
employee.first_name
from employee, department, location
where employee.department_id=department.department_id
and department.location_id=location.location_id
and employee.hire_date = ( select min(e2.hire_date)
                        from employee e2
                        where
employee.department_id=e2.department_id);
```

```
select department.department_id, department.name, location.regional_group,
employee.first_name
from employee, department, location
where employee.department_id=department.department_id
and department.location_id=location.location_id
group by department.department_id, department.name, location.regional_group,
employee.first_name
having (min(employee.hire_date),department.department_id) in (select min(hire_date),
department_id
from employee
group by department_id);
```

39. Muestra de cada producto su descripción y el nombre del cliente que más unidades ha comprado de dicho producto.

```
SQL>select product.description, customer.name
from product, customer, item, sales_order
where item.order_id=sales_order.order_id
and product.product_id=item.product_id
and customer.customer_id=sales_order.customer_id
group by product.description,customer.name
```



```

having sum(item.quantity) = ( select max(sum(i2.quantity))
                             from product p2, customer c2, sales_order s2,
item i2
                             where i2.order_id=s2.order_id
                             and c2.customer_id=s2.customer_id
                             and p2.product_id=i2.product_id
                             group by p2.description, c2.name
                             having product.description=p2.description);

```

40. Muestra el nombre de cada vendedor junto al nombre del cliente que, de todos los que representa, es el que más capital ha gastado en compras.

```

SQL>select employee.first_name, customer.name
      from employee, customer, sales_order
      where employee.employee_id=customer.salesperson_id
      and customer.customer_id=sales_order.customer_id
      group by employee.first_name, customer.name
      having sum(sales_order.total) = ( select max(sum(s2.total))
                                       from employee e2, customer c2,
sales_order s2
                                       where e2.employee_id=c2.salesperson_id
                                       and c2.customer_id=s2.customer_id
                                       group by e2.first_name, c2.name
                                       having employee.first_name=e2.first_name);

```

41. Muestra de cada localidad su nombre y nombre del empleado que más cobra de los que trabajan en esa localidad (debes tener en cuenta salario y comisión).

```

SQL>select location.regional_group, employee.first_name
      from employee, department, location
      where employee.department_id=department.department_id
      and department.location_id=location.location_id
      group by location.regional_group, employee.first_name

```

```
having sum(employee.salary + nvl(employee.commission,0))= ( select max(sum(e2.salary +  
nvl(e2.commission,0)))  
  
from employee e2, department d2, location l2  
where e2.department_id=d2.department_id  
and d2.location_id=l2.location_id  
group by l2.regional_group, e2.first_name  
having location.regional_group=l2.regional_group);
```

42. Seleccionar para cada empleado el nombre, número, número de jefe, departamento y nombre del jefe.

(AUTOCOMBINACIÓN)

```
SQL>select empleado.employee_id, empleado.first_name, jefe.employee_id,  
jefe.first_name, department.department_id, department.name  
from employee empleado, employee jefe, department  
where empleado.manager_id=jefe.employee_id  
and department.department_id=empleado.department_id;
```