

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 7 - Ciclo 2022 (fecha de entrega 01/11/22)

SQL *(Structured Query Language - Lenguaje de Consulta Estructurado)*

1 CONSULTAS – FUNCIONES AGREGADAS

Ejercicio 1:

- a) Muestre el valor promedio del precio de los artículos.

```
SELECT AVG(precio) FROM articulo;
```

-- Filas devueltas: 1

- b) Obtenga el sueldo más alto de los empleados.

```
SELECT MAX(sueldo) FROM empleado;
```

-- Filas devueltas: 1

- c) Liste el valor más económico de las membresías que no sea la membresía 'basico', el campo debe llamarse "membresía más barata".

```
SELECT MIN(precio) AS Membresía_mas_barata FROM membresia
```

```
WHERE membresia <> 'basico';
```

-- Filas devueltas: 1

- d) Obtenga la cantidad de préstamos de los artículos 101, 169, 182, 218, 368 y 392. Llamar el campo como "cantidad de préstamos".

```
SELECT id_articulo, COUNT(id_articulo) AS cantidad_de_prestamos FROM prestamo
```

```
WHERE id_articulo IN (101, 169, 182, 218, 368, 392)
```

```
ORDER BY id_articulo;
```

-- Filas devueltas: 6

- e) Muestre la cantidad de los préstamos que aún no fueron devueltos, llamar el campo como "no devueltos".

```
SELECT COUNT(*) AS No_devueltos FROM prestamo
```

```
WHERE fecha_devolucion IS NULL;
```

-- Filas devueltas: 1

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 7 - Ciclo 2022 (fecha de entrega 01/11/22)

- f) Obtenga el monto total de todas las ventas realizadas entre el '10-05-2019' y '31-07-2020'.
`SELECT SUM(total) FROM venta`
`WHERE fecha BETWEEN '2019-05-10' AND '2020-07-31';`
-- Filas devueltas: 1
- g) Muestre la cantidad de socios proveniente de cada ciudad. Debe mostrar el nombre de la ciudad, la provincia y la cantidad de socios, ordenados alfabéticamente por provincia y luego por ciudad.
`SELECT provincia, ciudad, COUNT(id_socio) FROM socio`
`INNER JOIN ciudad USING(id_ciudad)`
`INNER JOIN provincia USING(id_provincia)`
`GROUP BY provincia, ciudad`
`ORDER BY provincia, ciudad;`
-- Filas devueltas: 819
- h) Obtenga el listado de los empleados que tienen familiares, mostrando el nombre del empleado y la cantidad de familiares que tiene.
`SELECT p.nombre, COUNT(*) FROM empleado`
`INNER JOIN persona p ON id_empleado = id_persona`
`INNER JOIN familiar USING(id_empleado)`
`GROUP BY id_empleado;`
-- Filas devueltas: 603
- i) Muestre el monto total vendido por cada empleado, indicando el id, nombre y función del mismo. El monto debe llamarse total de ventas. Ordenado por nombre alfabéticamente.
`SELECT id_empleado, nombre, funcion, SUM(total) AS total_ventas FROM venta`
`INNER JOIN empleado USING(id_empleado)`
`INNER JOIN persona ON id_empleado = id_persona`
`INNER JOIN funcion USING(id_funcion)`
`GROUP BY id_empleado`
`ORDER BY nombre;`
-- Filas devueltas: 610
- j) Liste la cantidad de préstamos solicitados por los socios que no sean de "Córdoba" ni "Buenos Aires". Debe mostrar el nombre del socio y la ciudad a la que pertenece, la cantidad debe llamarse "cant_prest". Ordenado alfabéticamente por ciudad, luego de forma descendente por cantidad de mayor a menor y por último por nombre del socio.

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 7 - Ciclo 2022 (fecha de entrega 01/11/22)

```
SELECT nombre, ciudad, COUNT(inicio_prestamo) AS cant_prest, provincia FROM prestamo
INNER JOIN socio USING(id_socio)
INNER JOIN persona ON id_socio = id_persona
INNER JOIN ciudad USING(id_ciudad)
INNER JOIN provincia USING(id_provincia)
WHERE provincia NOT IN ('Córdoba', 'Buenos Aires')
GROUP BY nombre, ciudad
ORDER BY ciudad, cant_prest DESC, nombre;
-- Filas devueltas: 2229
```

- k) Muestre el código, título y el precio de los 20 artículos más vendidos en los años 2019 y 2020, donde la "cantidad_total" sea menor a 25.

```
SELECT id_articulo, titulo, ar.precio, SUM(cantidad) AS cantidad_total FROM articulo ar
INNER JOIN detalle_venta dv USING(id_articulo)
INNER JOIN venta USING(id_venta)
WHERE YEAR(venta.fecha) BETWEEN 2019 AND 2020
GROUP BY id_articulo
HAVING cantidad_total < 25
ORDER BY cantidad_total DESC
LIMIT 20;
-- Filas devueltas: 20
```

- l) Obtenga la cantidad de membresías vendidas hasta el '01-07-2021', mostrando lo recaudado por cada membresía en cada mes y año, donde la recaudación sea mayor a 40000.

```
SELECT membresia, fecha_pago, COUNT(id_membresia) AS cantidad, SUM(precio) AS
recaudacion FROM membresia
INNER JOIN pago USING(id_membresia)
WHERE fecha_pago < '2021-07-01'
GROUP BY id_membresia, anio, mes
HAVING recaudacion > 40000;
-- Filas devueltas: 205
```

- m) Muestre el total de artículos comprados por los socios, indicando el nombre y la provincia del socio, la cantidad de artículos comprados y el monto gastado por dichas compras. De los socios que sean de "Córdoba" o "Buenos Aires". Donde las cantidades sean mayores o iguales a 10. Ordenado por cantidad de mayor a menor y luego por el total de manera descendente.

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 7 - Ciclo 2022 (fecha de entrega 01/11/22)

```
SELECT nombre, provincia, SUM(cantidad) AS cantidad, SUM(dv.precio * cantidad) AS  
monto_gastado  
FROM venta  
INNER JOIN detalle_venta dv USING(id_venta)  
INNER JOIN socio USING(id_socio)  
INNER JOIN persona ON id_socio = id_persona  
INNER JOIN ciudad USING(id_ciudad)  
INNER JOIN provincia USING(id_provincia)  
WHERE provincia IN ('Córdoba', 'Buenos Aires')  
GROUP BY id_socio  
HAVING cantidad >= 10  
ORDER BY cantidad DESC, monto_gastado DESC;  
-- Filas devueltas: 44
```

- n) Muestre el nombre, la función, el sueldo y la antigüedad de los empleados en la empresa.

```
SELECT nombre, funcion, sueldo, TIMESTAMPDIFF(YEAR, fecha_alta, CURDATE()) AS antigüedad  
FROM empleado  
INNER JOIN funcion USING(id_funcion)  
INNER JOIN persona ON id_empleado = id_persona;  
-- Filas devueltas: 886
```

- o) Muestre el nombre, el domicilio, la ciudad y la provincia de cada socio. El domicilio, la ciudad y la provincia deben estar en una sola columna separados por coma (domicilio, ciudad, provincia). Provincia debe estar en mayúsculas.

```
SELECT nombre, CONCAT(domicilio, ', ', ciudad, ', ', UPPER(provincia)) AS direccion FROM socio  
INNER JOIN persona ON id_socio = id_persona  
INNER JOIN ciudad USING(id_ciudad)  
INNER JOIN provincia USING(id_provincia);  
-- Filas devueltas: 4659
```

- p) Muestre la cantidad de empleados que tienen 1, 2 y 3 años de antigüedad.

```
SELECT TIMESTAMPDIFF(YEAR, fecha_alta, CURDATE()) AS antigüedad, COUNT(id_empleado) AS  
cantidad FROM empleado  
GROUP BY antigüedad  
HAVING antigüedad BETWEEN 1 AND 3;  
-- Filas devueltas: 3
```

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 7 - Ciclo 2022 (fecha de entrega 01/11/22)

- q) Liste todos los empleados indicando el nombre, la fecha de alta y la función (la misma debe estar abreviada indicando solo las 3 primeras letras). Ordene alfabéticamente por la función.
`SELECT nombre, fecha_alta, SUBSTRING(funcion, 1, 3) AS funcion FROM empleado
INNER JOIN funcion USING(id_funcion)
INNER JOIN persona ON id_empleado = id_persona
ORDER BY funcion`
-- Filas devueltas: 886
- r) Muestre el nombre más largo de entre todas las personas.
`SELECT nombre, LENGTH(nombre) AS largo FROM persona
ORDER BY largo DESC
LIMIT 1;`
-- Filas devueltas: 1
- s) Muestre el promedio de sueldos (debe estar redondeado a 2 decimales).
`SELECT ROUND(AVG(sueldo), 2) FROM empleado`
-- Filas devueltas: 1
- t) Muestre el total a pagar de los préstamos que tienen multas (monto + multa) donde la multa sea mayor al monto del préstamo.
`SELECT inicio_prestamo, monto, multa, (monto + multa) AS total_a_pagar
FROM prestamo
WHERE multa IS NOT NULL AND multa > monto`
-- Filas devueltas: 470
- u) Muestre el nombre del empleado, la fecha de alta y la antigüedad, teniendo en cuenta que, para este caso, la antigüedad se computa por año completo, es decir, realice el cálculo desde el 1 de enero del año siguiente a la fecha de alta.
`SELECT nombre, fecha_alta, TIMESTAMPDIFF(YEAR, CONCAT(YEAR(fecha_alta) + 1, '-01-01'),
CURDATE()) AS antigüedad
FROM empleado
INNER JOIN persona ON id_empleado = id_persona`
-- Filas devueltas: 886