

## **UNIDAD 2 TALLER 4**

**Daniel Andrés Pinzón Jay**

**Id: 819793**

**Programa: Ingeniería de sistemas**

**Gestión de la información**

**Universidad cooperativa de Colombia**

**Sede: Bucaramanga**

**2024**

## CONSULTAS AEREOLINEA

1. Consultar el id, nombre y otros 2 atributos de los clientes junto con la información de cuántos vuelos ha realizado, el promedio, valor máximo y valor mínimo del costo de esos vuelos.

```
MariaDB [Aereolinea]> SELECT p.Cedula, pe.Nombre,  
-> COUNT(t.IDticket) AS TotalVuelos,  
-> AVG(t.Precio) AS PromedioCostoVuelos,  
-> MAX(t.Precio) AS MaximoCostoVuelo,  
-> MIN(t.Precio) AS MinimoCostoVuelo  
-> FROM Pasajero p  
-> JOIN Ticket t ON p.Cedula = t.CedulaPasajero  
-> JOIN Vuelos v ON t.IDvuelo = v.IDvuelo  
-> JOIN Persona pe ON p.Cedula = pe.Cedula  
-> GROUP BY pe.Nombre;
```

Cedula	Nombre	TotalVuelos	PromedioCostoVuelos	MaximoCostoVuelo	MinimoCostoVuelo
30	Carlos	1	300000.000000	300000.00	300000.00
7	Daniel	4	277500.000000	300000.00	260000.00
31	David	1	320000.000000	320000.00	320000.00
8	Juanita	5	302000.000000	320000.00	270000.00
6	Laura	5	276000.000000	290000.00	250000.00
32	Sebastian	1	280000.000000	280000.00	280000.00

6 rows in set (0.010 sec)

2. Consultar los atributos de los aviones junto con la información del piloto que lo maneja

```
MariaDB [Aereolinea]> SELECT a.*, per.Nombre NombrePiloto, p.*  
-> FROM Avion a  
-> JOIN Piloto p ON a.IDavion = p.IDavion  
-> JOIN Empleado e ON p.Cedula = e.Cedula  
-> JOIN Persona per ON e.Cedula = per.Cedula;
```

IDavion	Modelo	NumeroSerie	AltitudMaxima	VelocidadMaxima	NombrePiloto	Cedula	IDavion	AñosExperiencia	TipoAvion
1001	Boeing 737	12345	10000	900	Juan	1	1001	5	Comercial
1002	Airbus A320	54321	12000	850	Maria	2	1002	2	Comercial
1002	Airbus A320	54321	12000	850	Carlos	3	1002	8	Comercial

3 rows in set (0.001 sec)

3. Extraer la información de los pasajes (y/o tiquetes, boletos) junto con su itinerario (o en su defecto con la información del origen y destino)

```
MariaDB [Aereolinea]> SELECT t.*, v.Origem, v.Destino
-> FROM ticket t
-> JOIN vuelos v ON t.IDvuelo = v.IDvuelo;
```

IDticket	CedulaPasajero	IDvuelo	Precio	Origem	Destino
1	6	101	250000.00	Bucaramanga	Medellin
2	7	202	280000.00	Bucaramanga	Bogota
3	8	305	300000.00	Bucaramanga	Cali
4	6	403	270000.00	Medellin	Bucaramanga
5	7	508	260000.00	Bogota	Cucuta
6	8	609	270000.00	Cali	Cartagena
7	6	704	290000.00	Cartagena	Santa Marta
8	7	805	270000.00	Santa Marta	Pereira
9	8	907	310000.00	Pereira	Manizales
10	6	1102	280000.00	Medellin	Bogota
11	6	1103	290000.00	Bogota	Cali
12	7	1104	300000.00	Cali	Cartagena
13	8	1105	310000.00	Cartagena	Santa Marta
14	8	1106	320000.00	Santa Marta	Bucaramanga
15	30	1107	300000.00	Bogota	Medellin
16	31	1108	320000.00	Medellin	Cartagena
17	32	1109	280000.00	Cali	Santa Marta

17 rows in set (0.046 sec)

#### 4. Consultar solo vuelos con origen en Bucaramanga

```
MariaDB [Aereolinea]> SELECT *
-> FROM Vuelos
-> WHERE Origen = "Bucaramanga";
```

IDvuelo	Origem	Destino	Fecha	Hora	Duracion
101	Bucaramanga	Medellin	2024-04-26	09:00:00	120
202	Bucaramanga	Bogota	2024-04-27	10:30:00	90
305	Bucaramanga	Cali	2024-04-28	12:00:00	180

3 rows in set (0.020 sec)

#### 5. Identificar el id del pasajero y el promedio de vuelos que ha hecho, pero solo mostrar los que sean mayores a 2 vuelos.

```
MariaDB [Aereolinea]> SELECT CedulaPasajero IDPasajero, COUNT(*) AS TotalVuelos,
-> (SELECT COUNT(*) FROM ticket) / COUNT(*) AS PromedioVuelos
-> FROM ticket
-> GROUP BY CedulaPasajero
-> HAVING TotalVuelos > 2;
```

IDPasajero	TotalVuelos	PromedioVuelos
6	5	3.4000
7	4	4.2500
8	5	3.4000

3 rows in set (0.001 sec)

```
MariaDB [Aereolinea]> |
```

## CONSULTAS ABOGADOS

### 1. Consultar los casos (todos sus atributos) que and concluido en veredicto inocente.

```
MariaDB [Abogados]> SELECT *
-> FROM Veredicto
-> WHERE Conclusion = "Inocente";
```

IDveredicto	CedulaJuez	Evidencia	Conclusion	IDcaso
110	10	Testigo no confiable	Inocente	10
111	10	Falta de pruebas contundentes	Inocente	11
112	11	Cámara de seguridad lo exonera	Inocente	12
113	10	Testimonio verificado	Inocente	13
114	11	Análisis forense demuestran lo contrario a la acusasion	Inocente	14

5 rows in set (0.004 sec)

### 2. Consultar los casos (todos sus atributos) que and concluido en veredicto culpable.

```
MariaDB [Abogados]> SELECT *
-> FROM Veredicto
-> WHERE Conclusion = "Culpable";
```

IDveredicto	CedulaJuez	Evidencia	Conclusion	IDcaso
101	10	Huellas dactilares encontradas en la escena del crimen	Culpable	1
102	11	ADN coincidente con el del acusado	Culpable	2
103	10	Testigos presenciales identifican al acusado en el lugar del crimen	Culpable	3
104	11	Vídeo de vigilancia muestra al acusado cometiendo el delito	Culpable	4
105	10	Arma del crimen encontrada en posesión del acusado	Culpable	5
106	11	Confesión del acusado durante el interrogatorio policial	Culpable	6
107	10	Pruebas forenses indican la participación del acusado	Culpable	7
108	11	Testimonio de cómplice que incrimina al acusado	Culpable	8
109	10	Móvil del crimen relacionado con el acusado	Culpable	9
777	10	Fotografías	Culpable	15

10 rows in set (0.000 sec)

### 3. Consultar la información del abogado (defensor y/o acusador (fiscal)) junto con la cantidad de casos en los que ha trabajado, el promedio de casos ganados y el promedio de casos perdidos

```
MariaDB [Abogados]> SELECT
-> p.nombre,
-> a.Cedula,
-> SUM(a.CasosGanados + a.CasosPerdidos) AS TotalCasosTrabajados,
-> AVG(a.CasosGanados) AS PromedioCasosGanados,
-> AVG(a.CasosPerdidos) AS PromedioCasosPerdidos,
-> a.Tipo
-> FROM abogado a
-> JOIN Persona p ON a.Cedula = p.Cedula
-> LEFT JOIN Veredicto v ON a.Cedula = v.CedulaJuez
-> GROUP BY a.Cedula;
```

nombre	Cedula	TotalCasosTrabajados	PromedioCasosGanados	PromedioCasosPerdidos	Tipo
Paco Gonzalez	1	8	5.0000	3.0000	Fiscal
Ana maria	2	11	7.0000	4.0000	Defensor
Luisa Serrano	3	10	6.0000	4.0000	Defensor
Steven Acevedo	4	8	8.0000	0.0000	Fiscal
Nicolas Velasco	5	10	7.0000	3.0000	Defensor
Sirly Catherine	6	8	6.0000	2.0000	Fiscal
Daniel Pinzon	7	10	9.0000	1.0000	Fiscal
Valentina Fiayo	8	12	8.0000	4.0000	Defensor
Mariana Acuña	9	14	10.0000	4.0000	Fiscal
Juan Pérez	12	14	10.0000	4.0000	Fiscal

10 rows in set (0.001 sec)

#### 4. Consultar todos los imputados (todos sus atributos) junto con información del caso (fecha, tiempo, veredicto), también con la información de su sentencia (todos sus atributos).

```
MariaDB [Abogados]> SELECT i.*, v.IDveredicto, ca.*
-> FROM Imputado i
-> JOIN Condena c ON i.IDcondena = c.IDcondena
-> JOIN Veredicto v ON c.IDveredicto = v.IDveredicto
-> JOIN Caso ca ON v.IDcaso = ca.IDcaso;
```

Cedula	IDcondena	IDveredicto	IDcaso	MotivoDenuncia	Fecha_Caso	Tiempo_Caso
15	111	101	1	Huellas dactilares encontradas en la escena del crimen	2024-03-01	12:00:00
16	222	102	2	ADN coincidente con el del acusado	2024-03-02	13:00:00
17	333	103	3	Testigos presenciales identifican al acusado en el lugar del crimen	2024-03-03	14:00:00
18	444	104	4	Vídeo de vigilancia muestra al acusado cometiendo el delito	2024-03-04	15:00:00
19	555	105	5	Arma del crimen encontrada en posesión del acusado	2024-03-05	16:00:00
20	666	106	6	Confesión del acusado durante el interrogatorio policial	2024-03-06	17:00:00
21	777	107	7	Pruebas forenses indican la participación del acusado	2024-03-07	18:00:00
22	888	108	8	Testimonio de cómplice que incrimina al acusado	2024-03-08	19:00:00

8 rows in set (0.001 sec)

#### 5. Consultar los tipos de condena (intramural/domiciliaria,etc) junto con la información de: conteo, cantidad de meses/años, cantidad de imputados.

```
MariaDB [Abogados]> WITH ClasificacionCondena AS (
-> SELECT
-> IDcondena,
-> CASE
-> WHEN IDcondena IN (SELECT IDcondena FROM economica) THEN "Economica"
-> WHEN IDcondena IN (SELECT IDcondena FROM intramural) THEN "Intramural"
-> WHEN IDcondena IN (SELECT IDcondena FROM domiciliaria) THEN "Domiciliaria"
-> WHEN IDcondena IN (SELECT IDcondena FROM trabajocomunitario) THEN "TrabajoComunitario"
-> WHEN IDcondena IN (SELECT IDcondena FROM libertadcondicional) THEN "Liberta condicional"
-> END AS tipoCondena,
-> DATEDIFF(FechaFin, FechaInicio) AS DuracionDias
-> FROM Condena
-> )
-> SELECT
-> tipoCondena,
-> COUNT(DISTINCT i.Cedula) AS TotalImputados,
-> FLOOR(SUM(DuracionDias) / 30) AS TotalMeses,
-> FLOOR(SUM(DuracionDias) / 365) AS TotalAños
-> FROM ClasificacionCondena cc
-> JOIN Imputado i ON cc.IDcondena = i.IDcondena
-> GROUP BY tipoCondena;
```

tipoCondena	TotalImputados	TotalMeses	TotalAños
Domiciliaria	2	24	2
Economica	2	12	1
Intramural	2	9	0
Liberta condicional	1	6	0
TrabajoComunitario	1	8	0

5 rows in set (0.001 sec)

### CONSULTAS JUGUETERIA:

#### 1. Hacer una consulta de los empleados agrupándolos por tipo (Arquitecto, Administrativo e Ingeniero); mostrar la cantidad de cada grupo.

```
MariaDB [Jugueteria]> SELECT "Arquitecto" AS Tipo_Trabajador, COUNT(*) AS Total_Trabajadores
-> FROM Arquitecto
-> UNION
-> SELECT "Administrativo" Tipo_Trabajador, COUNT(*) AS Total_Trabajadores
-> FROM Administrativo
-> UNION
-> SELECT "Ingeniero" Tipo_Trabajador, COUNT(*) AS Total_Trabajadores
-> FROM Ingeniero;
```

Tipo_Trabajador	Total_Trabajadores
Arquitecto	3
Administrativo	2
Ingeniero	3

3 rows in set (0.001 sec)

2. Consultar solo los ingenieros agrupándolos por especialidad e identificando de los años de experiencia: el promedio, máximo, mínimo y sumatoria

```
MariaDB [Jugueteria]> SELECT
-> Especialidad,
-> AVG(AñosExperiencia) PromedioExperiencia,
-> MAX(AñosExperiencia) MaxExperiencia,
-> MIN(AñosExperiencia) MinExperiencia,
-> SUM(AñosExperiencia) SumatoriaExperiencia
-> FROM Ingeniero
-> GROUP BY Especialidad;
```

Especialidad	PromedioExperiencia	MaxExperiencia	MinExperiencia	SumatoriaExperiencia
Bioingenieria	5.3333	7	3	16
Mecatronica	7.3333	10	4	22
Sistemas	4.0000	5	2	12

3 rows in set (0.020 sec)

3. Consultar de los arquitectos cuál es el promedio y sumatorio total del valor de las comisiones y de la cantidad de proyectos.

```
MariaDB [Jugueteria]> SELECT
-> SUM(CantidadProyectos) SumaProyectos,
-> AVG(CantidadProyectos) PromedioProyectos,
-> SUM(a.Comisiones) SumaComisiones,
-> AVG(a.Comisiones) PromedioComisiones
-> FROM (
-> SELECT ccArquitecto, COUNT(*) AS CantidadProyectos
-> FROM Proyectos
-> GROUP BY ccArquitecto
-> ) AS pa
-> JOIN Arquitecto a ON pa.ccArquitecto = a.Cedula;
```

SumaProyectos	PromedioProyectos	SumaComisiones	PromedioComisiones
15	3.0000	9000000	1800000.0000

1 row in set (0.001 sec)

#### 4. Consultar los juguetes, su precio y peso, así como la información del proveedor o proveedores relacionados.

```
MariaDB [Jugueteria]> SELECT j.IDjuguete, j.Precio, j.Peso, pe.Nombre NombreProveedor, p.Especialidad EspecialidadProveedor
-> FROM Juguete j
-> LEFT JOIN Proveedor p ON j.IDproveedor = p.Cedula
-> LEFT JOIN Persona pe ON pe.Cedula = p.Cedula;
```

IDjuguete	Precio	Peso	NombreProveedor	EspecialidadProveedor
1	20000	200	Ana López	Juguetes
2	15000	150	Juan Martínez	Electrónica
3	18000	180	Sofía Pérez	Materiales
4	22000	220	Ana López	Juguetes
5	19000	190	Juan Martínez	Electrónica
6	21000	210	Sofía Pérez	Materiales
7	24000	240	Ana López	Juguetes
8	17000	170	Juan Martínez	Electrónica
9	23000	230	Sofía Pérez	Materiales

9 rows in set (0.001 sec)

#### 5. Agrupar los tipos de juguetes e identificar la cantidad, promedio de peso y precio, peso máximo y precio mínimo.

```
MariaDB [Jugueteria]> SELECT
-> TipoJuguete,
-> COUNT(*) AS Cantidad,
-> AVG(Peso) AS PromedioPesoJuguetes,
-> AVG(Precio) AS PromedioPrecioJuguetes,
-> MAX(Peso) AS PesoMaximoJuguetes,
-> MIN(Precio) AS PrecioMinimoJuguetes
-> FROM (
-> SELECT
-> j.IDjuguete, j.Peso, j.Precio,
-> CASE
-> WHEN m.IDjuguete IS NOT NULL THEN "Mecanico"
-> WHEN d.IDjuguete IS NOT NULL THEN "Digital"
-> WHEN dd.IDjuguete IS NOT NULL THEN "Didactico"
-> END AS TipoJuguete
-> FROM juguete j
-> LEFT JOIN mecanico m ON j.IDjuguete = m.IDjuguete
-> LEFT JOIN digital d ON j.IDjuguete = d.IDjuguete
-> LEFT JOIN didactico dd ON j.IDjuguete = dd.IDjuguete
-> ) AS TipoJuguetes
-> GROUP BY TipoJuguete;
```

TipoJuguete	Cantidad	PromedioPesoJuguetes	PromedioPrecioJuguetes	PesoMaximoJuguetes	PrecioMinimoJuguetes
Didactico	3	206.6667	20666.6667	220	19000
Digital	3	213.3333	21333.3333	240	17000
Mecanico	3	176.6667	17666.6667	200	15000

3 rows in set (0.001 sec)

### CONSULTAS CASA SOFTWARE:

#### 1. Consultar cada empleado con cedula, nombres, dirección y tipo; junto con la información de la cantidad de productos desarrollados, la sumatoria de horas dedicadas.

```
MariaDB [Casasoftware]> SELECT e.Cedula, e.Nombre, e.Direccion, e.Tipo, COUNT(efp.Codigo) AS Total_Productos, SUM(efp.num_horas) AS Horas_Dedicadas
-> FROM Empleado e
-> LEFT JOIN Empleado_Fase_Producto efp ON e.Cedula = efp.Cedula
-> GROUP BY e.Cedula;
```

Cedula	Nombre	Direccion	Tipo	Total_Productos	Horas_Dedicadas
1005	Daniel Jay	Calle 123	Programador	6	230
1006	Yeimy Corzo	Avenida 52	Programador	3	125
1007	Pedro Rodriguez	Carrera 456	Programador	2	75
1008	Juan Lopez	Calle 234	Analista	3	130
1009	Julio Gonzalez	Carrera 48	Analista	1	50
1010	Valentina fiayo	Calle 13u	Analista	2	98
1011	Katherin Pinzon	Calle 52	Analista	1	23
1012	Oscar Pinto	Carrera 15	Programador	1	46
1013	William Reyes	Carrera 34	Analista	2	81

9 rows in set (0.001 sec)

## 2. Consultar los proyectos (todos sus atributos) junto con la cantidad promedio de fases, cantidad máxima de empleados, conteo de empleados (programadores y analistas)

```
MariaDB [CasaSoftware]> SELECT efp.Proyecto, AVG(sf.Suma_Fases) AS Promedio_Fases, MAX(sf.TotalEmpleados) AS Maximo_Empleados,
-> SUM(CASE WHEN e.tipo = "Programador" THEN 1 ELSE 0 END) AS NumeroProgramadores,
-> SUM(CASE WHEN e.tipo = "Analista" THEN 1 ELSE 0 END) AS NumeroAnalistas
-> FROM Empleado_Fase_Producto efp
-> JOIN Empleado e ON efp.Cedula = e.Cedula
-> JOIN
-> (SELECT Proyecto, COUNT(Cedula) AS TotalEmpleados, COUNT(numFase) AS Suma_Fases
-> FROM Empleado_Fase_Producto
-> GROUP BY Proyecto) AS sf ON efp.Proyecto = sf.Proyecto
-> GROUP BY efp.Proyecto;
```

Proyecto	Promedio_Fases	Maximo_Empleados	NumeroProgramadores	NumeroAnalistas
1	6.0000	6	4	2
2	4.0000	4	2	2
3	4.0000	4	2	2
4	5.0000	5	3	2
5	2.0000	2	1	1

5 rows in set (0.001 sec)

## 3. Consultar por año la cantidad de hora de duración de los proyectos

```
MariaDB [CasaSoftware]> SELECT YEAR(fechaInicio) AS Año, SUM(numHoras) AS TotalHoras
-> FROM Proyecto
-> GROUP BY Año;
```

Año	TotalHoras
2019	790
2021	250
2024	700

3 rows in set (0.003 sec)



**4. Consultar los productos (nombre, descripción, tipo) cuyo estado esté “en curso” (ó pendiente) solo de los proyectos en ejecución desde enero de 2024 hasta la fecha**

```
MariaDB [CasaSoftware]> SELECT Nombre, Descripcion, Tipo
-> FROM Producto
-> WHERE Estado = "En curso"
-> AND Proyecto IN (
-> SELECT Proyecto
-> FROM Fase
-> WHERE fechaInicio >= '2024-01-01');
+-----+-----+-----+
| Nombre | Descripcion | Tipo |
+-----+-----+-----+
| Software Pruebas Agropecuario | Plataforma digital para pruebas y testeo en el sector agropecuario | software |
| Prototipo Desarrollo B.D. Financiera | Prototipo en desarrollo para el manejo de bases de datos financieras | prototipos |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

**5. Consultar por cada empleado (cedula y tipo) la cantidad de proyectos en los que trabaja**

```
MariaDB [CasaSoftware]> SELECT e.Cedula, e.Tipo, e.nombre, COUNT(pe.codProyecto) AS Numero_Proyectos_Asignados
-> FROM Empleado e
-> JOIN proyecto_empleado pe ON e.Cedula = pe.Cedula
-> GROUP BY Cedula;
+-----+-----+-----+-----+
| Cedula | Tipo | nombre | Numero_Proyectos_Asignados |
+-----+-----+-----+-----+
| 1005 | Programador | Daniel Jay | 3 |
| 1006 | Programador | Yeimy Corzo | 2 |
| 1007 | Programador | Pedro Rodríguez | 2 |
| 1008 | Analista | Juan Lopez | 1 |
| 1009 | Analista | Julio Gonzalez | 2 |
+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```