**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**Escuela de Tecnología**

**OBLIGATORIO BASE DE DATOS 2**

**DOCUMENTO**

Hombre con barba y bigote

Descripción generada automáticamente

**Nicolas Giménez – 291950**

Un hombre con una playera de color azul

Descripción generada automáticamente

**Cristian García – 317010**

**Grupo N3D**

**Docente: Daniel Capone**

**Analista en tecnologías de la información**

**25/11/2024**

​​**Índice**

[1. Crear índices. 3](#_Toc183387850)

[2. Datos de prueba 3](#_Toc183387851)

[2.1 Propietarios. 3](#_Toc183387852)

[2.2 Gatos. 3](#_Toc183387853)

[2.3 Habitaciones. 4](#_Toc183387854)

[2.4 Reservas. 5](#_Toc183387855)

[2.5 Servicios. 7](#_Toc183387856)

[2.6 Reservas servicios. 7](#_Toc183387857)

[3. Consultas. 10](#_Toc183387858)

[3.1 Consulta a. 10](#_Toc183387859)

[3.2 Consulta b. 10](#_Toc183387860)

[3.3 Consulta c. 11](#_Toc183387861)

[3.4 Consulta c. 12](#_Toc183387862)

[3.5 Consulta e. 13](#_Toc183387863)

[3.6 Consulta f. 14](#_Toc183387864)

[3.7 Consulta g. 15](#_Toc183387865)

[4. T-QSL 16](#_Toc183387866)

[4.1 Ejercicio a. 16](#_Toc183387867)

[4.2 Ejercicio b. 20](#_Toc183387868)

[5. Disparadores. 22](#_Toc183387869)

[5.1 Ejercicio a. 22](#_Toc183387870)

[5.2 Ejercicio b. 24](#_Toc183387871)

[6. Vista 26](#_Toc183387872)

[7. MongoDB 29](#_Toc183387873)

[7.1 Ejercicio a. 29](#_Toc183387874)

[7.2 Ejercicio b. 33](#_Toc183387875)

[7.3 Ejercicio c. 34](#_Toc183387876)

[7.4 Ejercicio d. 34](#_Toc183387877)

[7.5 Ejercicio e. 35](#_Toc183387878)

# Crear índices.

Se deciden crear los siguientes índices:

* propietarioDocumento en la tabla Gato.
* gatoID en la tabla GatoReserva.
* HabitaciónNombre en la tabla Reserva.

# Datos de prueba

Se presentan los siguientes datos de prueba.

## 2.1 Propietarios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Documento** | **Nombre** | **Telefono** | **Email** |
| 12345672 | Alejandro Torres | +598 91 234 567 | alejandro.torres@gmail.com |
| 51953448 | Beatriz Mendoza | +598 92 345 678 | beatriz.mendoza@hotmail.com |
| 14406662 | Camila Rodríguez | +598 93 456 789 | camila.rodriguez@outlook.com |
| 35569956 | Daniel Herrera | +598 94 567 890 | daniel.herrera@gmail.com |
| 30240527 | Elena Jiménez | NULL | elena.jimenez@hotmail.com |
| 53468374 | Francisco Morales | +598 96 789 012 | francisco.morales@outlook.com |
| 56779790 | Gabriela Castro | +598 97 890 123 | gabriela.castro@gmail.com |
| 68047090 | Héctor Vargas | +598 98 901 234 | NULL |
| 55525512 | Inés Romero | +598 99 012 345 | ines.romero@outlook.com |
| 75984413 | Jorge Salazar | +598 2612 3456 | jorge.salazar@gmail.com |
| 60150754 | Karina Paredes | +598 2623 4567 | NULL |
| 15450103 | Luis Ortega | +598 2201 2345 | luis.ortega@outlook.com |

## 2.2 Gatos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Raza** | **Edad** | **Peso** | **DocPropietario** |
| Darius | Persa | 11 | 8,2 | 55525512 |
| Miau | Persa | 12 | 5,2 | 30240527 |
| Garfield | Exótico | 5 | 8 | 30240527 |
| Mish | Siames | 2 | 4,5 | 14406662 |
| Ginger | Mestizo | 3 | 4,6 | 53468374 |
| Missy | Exótico | 3 | 6,1 | 35569956 |
| Kitty | Manx | 3 | 4,9 | 56779790 |
| Nala | Siames | 2 | 4,2 | 30240527 |
| Leo | Abyssinian | 4 | 5,3 | 68047090 |
| Nina | British Shorthair | 2 | 5,8 | 53468374 |
| Loki | Esfinge | 2 | 4 | 55525512 |
| Oliver | Persa | 4 | 5,8 | 56779790 |
| Luna | Ragdoll | 3 | 5,7 | 60150754 |
| Oscar | Scottish Fold | 5 | 7 | 68047090 |
| Yuna | Maine Coon | 1 | 6,8 | 75984413 |
| Pelusa | Persa | 3 | 5,2 | 55525512 |
| Max | Exótico | 2 | 5,9 | 15450103 |
| Pepper | Cornish Rex | 2 | 4,1 | 75984413 |
| Maya | Somalí | 2 | 4,7 | 12345672 |
| Rocky | Siames | 5 | 7,5 | 60150754 |
| Milo | Maine Coon | 1 | 7,2 | 51953448 |
| Salem | Bombay | 4 | 5,5 | 15450103 |
| Shadow | Bombay | 3 | 6,2 | 12345672 |
| Simba | Bengalí | 2 | 6 | 14406662 |
| Alberto | Bengalí | 4 | 7,1 | 51953448 |
| Toby | Angora | 4 | 4,8 | 35569956 |
| Tom | Oriental | 4 | 5,4 | 30240527 |
| Zoe | Ragdoll | 1 | 4,5 | 53468374 |
| Bella | Birmano | 1 | 4,4 | 12345672 |
| Chloe | Siberiano | 5 | 6,9 | 51953448 |
| Cleo | Siames | 3 | 4,3 | 14406662 |
| Felix | Bengalí | 3 | 6,5 | 35569956 |

## 2.3 Habitaciones.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Capacidad** | **Precio** | **Estado** |
| El Rascador | 3 | 80 | DISPONIBLE |
| La Almohada | 2 | 100 | DISPONIBLE |
| El Mirador | 4 | 70 | DISPONIBLE |
| La Cueva | 2 | 100 | DISPONIBLE |
| El Jardín | 6 | 40 | DISPONIBLE |
| El Solarium | 1 | 120 | DISPONIBLE |
| El Árbol | 5 | 50 | DISPONIBLE |
| La Estantería | 7 | 30 | DISPONIBLE |
| La Caja | 4 | 70 | DISPONIBLE |

## 2.4 Reservas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GatoID** | **Nombre Habitacion** | **Fecha Inicio** | **Fecha Fin** | **Monto** |
| 24 | La Caja | 7/7/2023 | 18/7/2023 | 840 |
| 7 | El Rascador | 18/7/2023 | 29/7/2023 | 960 |
| 16 | El Solarium | 22/8/2023 | 28/8/2023 | 840 |
| 23 | El Solarium | 27/8/2023 | 3/9/2023 | 960 |
| 24 | La Cueva | 31/8/2023 | 10/9/2023 | 1100 |
| 2 | El Solarium | 3/9/2023 | 9/9/2023 | 840 |
| 15 | La Caja | 21/9/2023 | 23/9/2023 | 210 |
| 2 | La Cueva | 12/10/2023 | 22/10/2023 | 1100 |
| 23 | El Solarium | 8/11/2023 | 12/11/2023 | 600 |
| 7 | La Almohada | 21/11/2023 | 22/11/2023 | 200 |
| 13 | El Mirador | 29/11/2023 | 4/12/2023 | 420 |
| 29 | El Solarium | 7/12/2023 | 17/12/2023 | 1320 |
| 28 | El Mirador | 4/1/2024 | 15/1/2024 | 840 |
| 27 | El Mirador | 5/1/2024 | 8/1/2024 | 280 |
| 4 | La Estantería | 5/2/2024 | 8/2/2024 | 120 |
| 8 | El Jardín | 11/2/2024 | 14/2/2024 | 160 |
| 17 | El Solarium | 19/2/2024 | 27/2/2024 | 1080 |
| 18 | La Caja | 19/2/2024 | 29/2/2024 | 770 |
| 4 | La Caja | 28/2/2024 | 7/3/2024 | 630 |
| 31 | La Caja | 23/3/2024 | 4/4/2024 | 910 |
| 30 | La Cueva | 3/5/2024 | 10/5/2024 | 800 |
| 22 | El Solarium | 7/5/2024 | 8/5/2024 | 240 |
| 11 | El Mirador | 18/6/2024 | 29/6/2024 | 840 |
| 20 | El Jardín | 25/6/2024 | 2/7/2024 | 320 |
| 32 | El Rascador | 23/7/2024 | 1/8/2024 | 800 |
| 16 | El Rascador | 23/7/2024 | 1/8/2024 | 800 |
| 17 | La Almohada | 15/8/2024 | 25/8/2024 | 1100 |
| 16 | El Solarium | 16/8/2024 | 26/8/2024 | 1320 |
| 7 | La Estantería | 19/8/2024 | 27/8/2024 | 270 |
| 7 | El Rascador | 24/8/2024 | 26/8/2024 | 240 |
| 30 | La Caja | 25/8/2024 | 28/8/2024 | 280 |
| 6 | La Caja | 7/9/2024 | 11/9/2024 | 350 |
| 25 | La Estantería | 26/9/2024 | 8/10/2024 | 390 |
| 1 | La Estantería | 1/10/2024 | 4/10/2024 | 120 |
| 2 | La Almohada | 10/10/2024 | 25/10/2024 | 1600 |
| 1 | El Rascador | 10/1/2023 | 17/1/2023 | 640 |
| 1 | El Solarium | 15/1/2023 | 20/1/2023 | 720 |
| 9 | El Árbol | 15/1/2023 | 22/1/2023 | 400 |
| 2 | La Almohada | 5/2/2023 | 10/2/2023 | 600 |
| 2 | El Mirador | 5/2/2023 | 12/2/2023 | 560 |
| 10 | El Jardín | 10/2/2023 | 17/2/2023 | 320 |
| 3 | La Caja | 1/3/2023 | 8/3/2023 | 560 |
| 11 | La Estantería | 5/3/2023 | 12/3/2023 | 240 |
| 3 | La Cueva | 12/3/2023 | 18/3/2023 | 700 |
| 4 | El Rascador | 1/4/2023 | 7/4/2023 | 560 |
| 12 | El Solarium | 15/4/2023 | 22/4/2023 | 960 |
| 4 | El Árbol | 15/4/2023 | 22/4/2023 | 400 |
| 13 | La Almohada | 5/5/2023 | 12/5/2023 | 800 |
| 5 | El Jardín | 5/5/2023 | 12/5/2023 | 320 |
| 5 | El Mirador | 10/5/2023 | 15/5/2023 | 420 |
| 14 | La Cueva | 1/6/2023 | 8/6/2023 | 800 |
| 6 | La Estantería | 1/6/2023 | 8/6/2023 | 240 |
| 6 | La Caja | 20/6/2023 | 25/6/2023 | 420 |
| 7 | El Árbol | 5/7/2023 | 12/7/2023 | 400 |
| 15 | El Rascador | 10/7/2023 | 17/7/2023 | 640 |
| 7 | El Solarium | 10/7/2023 | 17/7/2023 | 960 |
| 8 | El Jardín | 1/8/2023 | 8/8/2023 | 320 |
| 8 | La Almohada | 15/8/2023 | 22/8/2023 | 800 |
| 16 | El Mirador | 20/8/2023 | 27/8/2023 | 560 |
| 1 | La Caja | 5/9/2023 | 12/9/2023 | 560 |
| 9 | La Estantería | 15/9/2023 | 22/9/2023 | 240 |
| 10 | El Solarium | 10/10/2023 | 17/10/2023 | 960 |
| 2 | El Árbol | 15/10/2023 | 22/10/2023 | 400 |
| 3 | El Jardín | 1/11/2023 | 8/11/2023 | 320 |
| 11 | La Almohada | 5/11/2023 | 12/11/2023 | 800 |
| 12 | La Cueva | 1/12/2023 | 8/12/2023 | 800 |
| 4 | La Estantería | 5/12/2023 | 12/12/2023 | 240 |
| 13 | El Rascador | 5/1/2024 | 10/1/2024 | 480 |
| 5 | El Solarium | 15/1/2024 | 22/1/2024 | 960 |
| 14 | El Mirador | 1/2/2024 | 6/2/2024 | 420 |
| 6 | La Almohada | 5/2/2024 | 12/2/2024 | 800 |
| 15 | La Caja | 1/3/2024 | 8/3/2024 | 560 |
| 7 | La Cueva | 1/3/2024 | 8/3/2024 | 800 |
| 16 | El Árbol | 10/4/2024 | 15/4/2024 | 300 |
| 8 | El Rascador | 15/4/2024 | 22/4/2024 | 640 |
| 9 | El Mirador | 5/5/2024 | 12/5/2024 | 560 |
| 1 | El Jardín | 20/5/2024 | 25/5/2024 | 240 |
| 10 | La Caja | 1/6/2024 | 8/6/2024 | 560 |
| 2 | La Estantería | 10/6/2024 | 18/6/2024 | 270 |
| 3 | El Solarium | 1/7/2024 | 8/7/2024 | 960 |
| 11 | El Árbol | 10/7/2024 | 17/7/2024 | 400 |
| 4 | La Almohada | 15/8/2024 | 22/8/2024 | 800 |
| 12 | El Jardín | 15/8/2024 | 22/8/2024 | 320 |
| 13 | La Estantería | 1/9/2024 | 8/9/2024 | 240 |
| 5 | La Cueva | 10/9/2024 | 17/9/2024 | 800 |
| 6 | El Rascador | 5/10/2024 | 12/10/2024 | 640 |
| 14 | El Solarium | 5/10/2024 | 12/10/2024 | 960 |
| 7 | El Mirador | 1/11/2024 | 7/11/2024 | 490 |
| 15 | La Almohada | 1/11/2024 | 7/11/2024 | 700 |
| 8 | La Caja | 1/12/2024 | 7/12/2024 | 490 |
| 16 | La Cueva | 1/12/2024 | 8/12/2024 | 800 |

## 2.5 Servicios.

|  |  |
| --- | --- |
| **Nombre** | **Precio** |
| CONTROL\_PARASITOS | 35 |
| REVISION\_VETERINARIA | 50 |
| PELUQUERIA | 30 |
| BAÑO | 20 |
| CORTE\_DE\_UNAS | 15 |
| ALIMENTACION\_ESPECIAL | 25 |
| JUEGO\_GUIADO | 10 |

## 2.6 Reservas servicios.

|  |  |  |
| --- | --- | --- |
| **ReservaID** | **Nombre Servicio** | **Cantidad** |
| 3 | PELUQUERIA | 5 |
| 5 | ALIMENTACION\_ESPECIAL | 2 |
| 6 | CONTROL\_PARASITOS | 6 |
| 6 | JUEGO\_GUIADO | 8 |
| 7 | CORTE\_DE\_UNAS | 7 |
| 9 | BAÑO | 3 |
| 9 | PELUQUERIA | 8 |
| 10 | CORTE\_DE\_UNAS | 4 |
| 11 | ALIMENTACION\_ESPECIAL | 2 |
| 11 | PELUQUERIA | 1 |
| 11 | REVISION\_VETERINARIA | 8 |
| 12 | JUEGO\_GUIADO | 7 |
| 15 | PELUQUERIA | 4 |
| 17 | ALIMENTACION\_ESPECIAL | 3 |
| 17 | BAÑO | 3 |
| 17 | CONTROL\_PARASITOS | 1 |
| 17 | CORTE\_DE\_UNAS | 2 |
| 17 | JUEGO\_GUIADO | 5 |
| 17 | PELUQUERIA | 2 |
| 17 | REVISION\_VETERINARIA | 3 |
| 18 | JUEGO\_GUIADO | 6 |
| 19 | PELUQUERIA | 1 |
| 20 | ALIMENTACION\_ESPECIAL | 3 |
| 22 | JUEGO\_GUIADO | 6 |
| 25 | CONTROL\_PARASITOS | 2 |
| 27 | JUEGO\_GUIADO | 5 |
| 29 | ALIMENTACION\_ESPECIAL | 4 |
| 29 | PELUQUERIA | 2 |
| 32 | ALIMENTACION\_ESPECIAL | 2 |
| 32 | BAÑO | 4 |
| 32 | PELUQUERIA | 2 |
| 34 | ALIMENTACION\_ESPECIAL | 4 |
| 35 | ALIMENTACION\_ESPECIAL | 2 |
| 35 | JUEGO\_GUIADO | 3 |
| 36 | PELUQUERIA | 4 |
| 40 | CONTROL\_PARASITOS | 2 |
| 41 | CONTROL\_PARASITOS | 6 |
| 41 | CORTE\_DE\_UNAS | 6 |
| 42 | JUEGO\_GUIADO | 6 |
| 43 | ALIMENTACION\_ESPECIAL | 8 |
| 43 | BAÑO | 7 |
| 43 | CONTROL\_PARASITOS | 3 |
| 44 | CORTE\_DE\_UNAS | 6 |
| 45 | BAÑO | 5 |
| 47 | CORTE\_DE\_UNAS | 2 |
| 48 | CONTROL\_PARASITOS | 1 |
| 49 | PELUQUERIA | 8 |
| 53 | ALIMENTACION\_ESPECIAL | 3 |
| 54 | ALIMENTACION\_ESPECIAL | 1 |
| 54 | JUEGO\_GUIADO | 3 |
| 54 | PELUQUERIA | 3 |
| 55 | PELUQUERIA | 3 |
| 60 | ALIMENTACION\_ESPECIAL | 2 |
| 60 | BAÑO | 1 |
| 61 | CONTROL\_PARASITOS | 1 |
| 61 | JUEGO\_GUIADO | 3 |
| 62 | PELUQUERIA | 1 |
| 62 | CORTE\_DE\_UNAS | 2 |
| 63 | REVISION\_VETERINARIA | 1 |
| 64 | ALIMENTACION\_ESPECIAL | 3 |
| 65 | JUEGO\_GUIADO | 2 |
| 65 | PELUQUERIA | 1 |
| 66 | CONTROL\_PARASITOS | 1 |
| 67 | CORTE\_DE\_UNAS | 2 |
| 68 | REVISION\_VETERINARIA | 1 |
| 69 | ALIMENTACION\_ESPECIAL | 2 |
| 69 | BAÑO | 1 |
| 70 | CONTROL\_PARASITOS | 3 |
| 71 | JUEGO\_GUIADO | 1 |
| 72 | PELUQUERIA | 1 |
| 73 | ALIMENTACION\_ESPECIAL | 4 |
| 74 | REVISION\_VETERINARIA | 2 |
| 75 | JUEGO\_GUIADO | 3 |
| 76 | CORTE\_DE\_UNAS | 2 |
| 77 | CONTROL\_PARASITOS | 1 |
| 77 | PELUQUERIA | 2 |
| 78 | ALIMENTACION\_ESPECIAL | 3 |
| 79 | REVISION\_VETERINARIA | 1 |
| 80 | JUEGO\_GUIADO | 2 |
| 81 | BAÑO | 1 |
| 81 | CONTROL\_PARASITOS | 1 |
| 82 | PELUQUERIA | 2 |
| 83 | CORTE\_DE\_UNAS | 1 |
| 84 | JUEGO\_GUIADO | 1 |
| 85 | REVISION\_VETERINARIA | 3 |
| 86 | ALIMENTACION\_ESPECIAL | 2 |
| 86 | BAÑO | 1 |
| 87 | CONTROL\_PARASITOS | 4 |
| 88 | PELUQUERIA | 1 |
| 89 | JUEGO\_GUIADO | 2 |
| 90 | REVISION\_VETERINARIA | 1 |
| 90 | CORTE\_DE\_UNAS | 1 |
| 91 | ALIMENTACION\_ESPECIAL | 3 |

# Consultas.

## 3.1 Consulta a.

Se pide: Mostrar el nombre del gato, el nombre del propietario, la habitación y el monto de la reserva más reciente en la(s) habitación con la capacidad más alta.

Código de consulta:

SELECT TOP 1 g.gatoNombre, p.propietarioNombre, h.habitacionNombre, r.reservaMonto

FROM Gato g, Propietario p, Reserva r, Habitacion h

WHERE g.propietarioDocumento = p.propietarioDocumento and

g.gatoID = r.gatoID and

r.habitacionNombre = h.habitacionNombre and

h.habitacionCapacidad = ( SELECT MAX(h2.habitacionCapacidad)

FROM Habitacion h2)

ORDER BY r.reservaFechaInicio DESC

Resultado:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## 3.2 Consulta b.

Se pide: Mostrar los 3 servicios más solicitados, con su nombre, precio y cantidad total solicitada en el año anterior. Solo listar el servicio si cumple que tiene una cantidad total solicitada mayor o igual que 5.

Código de consulta:

SELECT TOP 3 s.servicioNombre, s.servicioPrecio, SUM(rs.cantidad) AS CantidadSolicitada

FROM Servicio s, Reserva\_Servicio rs, Reserva r

WHERE rs.reservaID = r.reservaID and

rs.servicioNombre = s.servicioNombre and

YEAR(r.reservaFechaInicio) = YEAR(GETDATE())-1

GROUP BY s.servicioNombre, s.servicioPrecio

HAVING SUM(rs.cantidad) >= 5

ORDER BY CantidadSolicitada DESC

Resultado:

Tabla

Descripción generada automáticamente

## 3.3 Consulta c.

Se pide: Listar nombre de gato y nombre de habitación para las reservas que tienen asociados todos los servicios adicionales disponibles.

Código de consulta:

SELECT g.gatoNombre, r.habitacionNombre

FROM Gato g, Reserva r, Reserva\_Servicio rs

WHERE g.gatoID = r.gatoID and

rs.reservaID = r.reservaID

GROUP BY g.gatoNombre, r.habitacionNombre

HAVING COUNT(DISTINCT(rs.servicioNombre)) = (SELECT COUNT(servicioNombre) FROM Servicio)

Resultado:

Tabla

Descripción generada automáticamente

## 3.4 Consulta c.

Se pide: Listar monto total de reserva por año y por gato (nombre) para los gatos que tienen más de 10 años de edad, son de raza "Persa" y que en el año tuvieron montos total de reserva superior a 500 dólares.

Código de consulta:

SELECT g.gatoNombre, SUM(r.reservaMonto) AS [Monto total]

FROM Gato g

JOIN Reserva r ON r.gatoID = g.gatoID

WHERE g.gatoEdad > 10 AND

g.gatoRaza = 'Persa' AND

YEAR(r.reservaFechaInicio) = YEAR(GETDATE())

GROUP BY g.gatoNombre

HAVING SUM(r.reservaMonto) > 500

Resultado:

Tabla

Descripción generada automáticamente con confianza media

## 3.5 Consulta e.

Se pide: Mostrar el ranking de reservas más caras, tomando como monto total de una reserva el monto propio de la reserva más los servicios adicionales contratados en la reserva.

Código de consulta:

SELECT r.gatoID, r.habitacionNombre, r.reservaFechaInicio, r.reservaFechaFin, r.reservaID, (

SELECT (SUM(rs2.cantidad\*s2.servicioPrecio))

FROM Reserva\_Servicio rs2

JOIN Servicio s2 ON s2.servicioNombre = rs2.servicioNombre

WHERE rs2.reservaID = r.reservaID) +

r.reservaMonto AS TotalPorReservaYServicio

FROM Reserva r

JOIN Reserva\_Servicio rs ON rs.reservaID = r.reservaID

JOIN Servicio s ON rs.servicioNombre = s.servicioNombre

GROUP BY r.gatoID, r.habitacionNombre, r.reservaFechaInicio, r.reservaFechaFin, r.reservaID, r.reservaMonto

ORDER BY TotalPorReservaYServicio DESC

Resultado:

Tabla

Descripción generada automáticamente

## 3.6 Consulta f.

Se pide: Calcular el promedio de duración en días de las reservas realizadas durante el año en curso. Deben ser consideradas solo aquellas reservas en las que se contrató el servicio "CONTROL\_PARASITOS" pero no se contrató el servicio "REVISION\_VETERINARIA".

Código de consulta:

SELECT AVG(DATEDIFF(DAY,r.reservaFechaInicio, r.reservaFechaFin)) AS PromedioDiasReserva

FROM Reserva r

JOIN Reserva\_Servicio rs ON rs.reservaID = r.reservaID

WHERE YEAR(r.reservaFechaInicio) = YEAR(GETDATE()) AND

rs.servicioNombre = 'CONTROL\_PARASITOS' AND

r.reservaID NOT IN (

SELECT reservaID

FROM Reserva\_Servicio

WHERE servicioNombre = 'REVISION\_VETERINARIA' AND

YEAR(r.reservaFechaInicio) = YEAR(GETDATE())

)

Resultado:

Tabla

Descripción generada automáticamente con confianza media

## 3.7 Consulta g.

Se pide: Para cada habitación, listar su nombre, la cantidad de días que ha estado ocupada y la cantidad de días transcurridos desde la fecha de inicio de la primera reserva en el hotel. Además, incluir una columna adicional que indique la categoría de rentabilidad, asignando el valor "REDITUABLE" si la habitación estuvo ocupada más del 60% de los días, "MAGRO" si estuvo ocupada entre el 40% y el 60%, y "NOESNEGOCIO" si estuvo ocupada menos del 40%.

Código de consulta:

SELECT h.habitacionNombre,

SUM(DATEDIFF(DAY,r.reservaFechaInicio, r.reservaFechaFin)) AS DiasOcupada,

(DATEDIFF(DAY,MIN(r.reservaFechaInicio),GETDATE())) AS DiasDesdeInicio ,

CASE

WHEN (SUM(DATEDIFF(DAY,r.reservaFechaInicio, r.reservaFechaFin)) / DATEDIFF(DAY,MIN(r.reservaFechaInicio),GETDATE()) ) >= 0.6 THEN 'REDITUABLE'

WHEN (SUM(DATEDIFF(DAY,r.reservaFechaInicio, r.reservaFechaFin)) / DATEDIFF(DAY,MIN(r.reservaFechaInicio),GETDATE()) ) >= 0.4 THEN 'MAGRO'

ELSE 'NOESNEGOCIO'

END

FROM Habitacion h

JOIN Reserva r ON r.habitacionNombre = h.habitacionNombre

GROUP BY h.habitacionNombre

Resultado:

Tabla

Descripción generada automáticamente con confianza media

# T-QSL

## 4.1 Ejercicio a.

Se pide: Escribir un procedimiento almacenado para reservar una habitación. Se debe actualizar el estado de DISPONIBLE a LLENA si se alcanzó la capacidad de la habitación con la reserva en cuestión No permitir realizar la reserva si el estado de la habitación es LLENA o LIMPIANDO. Se debe retornar el número de reserva asignado (cero sino se logró reservar).

Código de procedimiento:

CREATE OR ALTER PROCEDURE SP\_EJ4A

@NombreHabitacion CHAR(30),

@GatoID INT,

@ReservaFechaFin DATE

AS

BEGIN

DECLARE @Estado VARCHAR(20) = 'Sin asignar'

DECLARE @GatoEncontrado BIT = 0

DECLARE @VerifFecha INT = DATEDIFF(DAY, GETDATE(), @ReservaFechaFin)

DECLARE @MontoReserva DECIMAL(7,2) = 0

SELECT @Estado = h.habitacionEstado, @MontoReserva = (H.habitacionPrecio\*(@VerifFecha+1))

FROM Habitacion h

WHERE h.habitacionNombre = @NombreHabitacion

SELECT @GatoEncontrado = 1

FROM Gato g

WHERE g.gatoID = @GatoID

IF(@Estado = 'LLENA' OR @Estado = 'LIMPIANDO' OR @Estado = 'Sin asignar' OR @GatoEncontrado = 0 OR @VerifFecha < 0)

RETURN 0

DECLARE @ReservaID INT = 0

BEGIN TRY

BEGIN TRANSACTION

INSERT INTO Reserva (gatoID, habitacionNombre, reservaFechaInicio, reservaFechaFin, reservaMonto)

VALUES (@GatoID, @NombreHabitacion, GETDATE(), @ReservaFechaFin, @MontoReserva)

SELECT @ReservaID = MAX(r.reservaID)

FROM Reserva r

UPDATE Habitacion

SET habitacionEstado = 'LLENA'

WHERE habitacionNombre = @NombreHabitacion AND

habitacionCapacidad <= (

SELECT COUNT(r.gatoID)

FROM Reserva r

WHERE r.habitacionNombre = @NombreHabitacion AND GETDATE() BETWEEN r.reservaFechaInicio AND r.reservaFechaFin)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

END CATCH

RETURN @ReservaID

END

Pruebas:

Estado actual de las habitaciones

Tabla

Descripción generada automáticamente

Se realiza la inserción de una reserva en la habitación “El Solarium” que se encuentra disponible y tiene capacidad para 1 gato.

DECLARE @Insert INT

EXEC @Insert = SP\_EJ4A @NombreHabitacion = 'El Solarium', @GatoID = 31, @ReservaFechaFin = '2024-11-30'

PRINT @Insert

Resultado del store procedure.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Estado de las habitaciones luego de ejecutarlo:

Tabla

Descripción generada automáticamente

Se intenta insertar una nueva reserva en la misma habitación

DECLARE @Insert2 INT

EXEC @Insert2 = SP\_EJ4A @NombreHabitacion = 'El Solarium', @GatoID = 30, @ReservaFechaFin = '2024-11-28'

PRINT @Insert2

Resultado:

Texto

Descripción generada automáticamente con confianza baja

## 4.2 Ejercicio b.

Se pide: Mediante una función que reciba un nombre de servicio, devolver un booleano indicando si este año el servicio fue contratado más veces que el año pasado.

Código de función:

CREATE OR ALTER FUNCTION F\_EJ4B (@NombreServicio CHAR(30))

RETURNS BIT

BEGIN

DECLARE @Res Bit = 0

SELECT @Res = 1

FROM Reserva r

LEFT JOIN Reserva\_Servicio rs on rs.reservaID = r.reservaID

WHERE rs.servicioNombre = @NombreServicio AND

YEAR(GETDATE()) = YEAR(r.reservaFechaInicio)

HAVING ISNULL(SUM(rs.cantidad), 0) > (

SELECT ISNULL(SUM(rs.cantidad), 0)

FROM Reserva r

JOIN Reserva\_Servicio rs ON rs.reservaID = r.reservaID

WHERE rs.servicioNombre = @NombreServicio AND

YEAR(GETDATE())-1 = YEAR(r.reservaFechaInicio))

RETURN @Res

END

Prueba de función.

DECLARE @PruebaFuncion BIT

EXEC @PruebaFuncion = dbo.F\_EJ4B 'BAÑO'

PRINT @PruebaFuncion

Resultado:

Texto

Descripción generada automáticamente con confianza baja

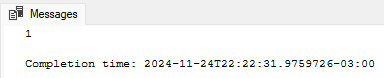
Prueba caso 2.

DECLARE @PruebaFuncion2 BIT

EXEC @PruebaFuncion2 = dbo.F\_EJ4B 'ALIMENTACION\_ESPECIAL'

PRINT @PruebaFuncion2

Resultado:



# Disparadores.

## 5.1 Ejercicio a.

Se pide: Cada vez que se crea una nueva reserva se debe crear un registro de auditoria con todos los datos ingresados en una tabla ReservaLog (definir su estructura libremente). Y adicionalmente cada vez que se modifica el campo monto de una reserva: debe registrar monto previo y nuevo monto en la tabla ReservaLog. En todos los casos se debe grabar fecha-hora de registro, usuario(login), nombre de equipo desde el que se realizó la modificación.

Se crea la tabla ReservaLog.

CREATE TABLE ReservaLog(

LogId INT IDENTITY(1,1) PRIMARY KEY,

Fecha DATETIME NOT NULL DEFAULT GETDATE(),

Usuario NVARCHAR(100) DEFAULT SUSER\_NAME(),

Equipo NVARCHAR (100) DEFAULT HOST\_NAME(),

Accion NVARCHAR (50) NOT NULL,

ReservaID INT NOT NULL,

GatoID INT NOT NULL,

HabitacionNombre CHAR(30) NOT NULL,

ReservaFechaInicio DATE NOT NULL,

ReservaFechaFin DATE NOT NULL,

ReservaMonto DECIMAL(7,2) NOT NULL,

NuevaReservaMonto DECIMAL(7,2) DEFAULT NULL

)

Código del trigger.

CREATE OR ALTER TRIGGER TR\_EJ5A ON Reserva

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON

IF EXISTS(SELECT 1 FROM deleted) BEGIN

IF UPDATE(ReservaMonto)

INSERT INTO ReservaLog (Accion, ReservaID, GatoID, HabitacionNombre, ReservaFechaInicio, ReservaFechaFin, ReservaMonto, NuevaReservaMonto)

SELECT 'UPDATE DEL MONTO', d.\*, i.reservaMonto

FROM inserted i, deleted d

END

ELSE BEGIN

INSERT INTO ReservaLog (Accion, ReservaID, GatoID, HabitacionNombre, ReservaFechaInicio, ReservaFechaFin, ReservaMonto)

SELECT 'INSERT', i.\*

FROM inserted i

END

END

Registros en la tabla reserva

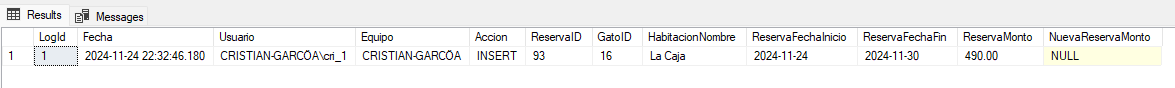


Se realiza un insert.

INSERT INTO Reserva

VALUES (16, 'La Caja', GETDATE(), '2024-11-30', 490)

Registros en la tabla luego de haber realizado un insert.



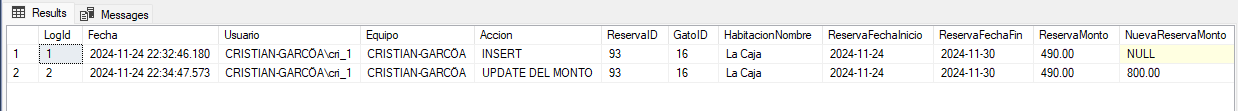
Se realiza el UPDATE del monto de la reserva 93.

UPDATE Reserva

SET reservaMonto = 800

WHERE reservaID = 93

Registros en la tabla de auditoria luego de hacer el update.



## 5.2 Ejercicio b.

Se pide: Antes de insertar una nueva reserva, se debe controlar posibles solapamientos de reservas (un gato no podría estar alojado simultáneamente 2 veces en el hotel). Se debe dar de alta las reservas válidas y simplemente ignorar las reservas solapadas.

Código del trigger

CREATE OR ALTER TRIGGER TR\_EJ5B ON Reserva

INSTEAD OF INSERT

AS

BEGIN

SET NOCOUNT ON

INSERT INTO Reserva (gatoID, habitacionNombre, reservaFechaInicio, reservaFechaFin, reservaMonto)

SELECT i.gatoID, i.habitacionNombre, i.reservaFechaInicio, i.reservaFechaFin, i.reservaMonto

FROM inserted i

WHERE i.gatoID NOT IN (

SELECT r.gatoID

FROM Reserva r

WHERE i.reservaFechaInicio <= r.reservaFechaFin AND

i.reservaFechaFin >= r.reservaFechaInicio

)

END

Muestra de las últimas reservas:

Tabla

Descripción generada automáticamente

Se realizan 4 inserts, contemplando gatos que solapen reservas.

INSERT INTO Reserva

VALUES (16, 'El Rascador', '2024-11-23', '2024-11-28', 480),

(4, 'El Árbol', '2024-11-15', '2024-11-30', 800),

(31,'El Mirador', '2024-11-25', '2024-11-28', 280),

(7, 'La Cueva', '2024-11-18', '2024-11-19', 200)

Registros en la tabla reserva luego de realizar los insert.

Tabla

Descripción generada automáticamente

Se verifica que solo se registraron las reservas que no se solapaban.

# Vista

Se pide: Crear una vista que liste el monto total a facturar por propietario por las reservas y servicios del mes pasado. Se debe listar el nombre del propietario, el monto total de sus reservas, el monto total de servicios adicionales que contrató y la suma de ambos montos (monto a facturar).

Se crean dos funciones para obtener los montos de las reservas y de los servicios con el fin de utilizarlas en la vista.

CREATE OR ALTER FUNCTION F\_MontoReservasMesAnterior (

@DocPropietario CHAR(30) )

RETURNS INT

BEGIN

DECLARE @res DECIMAL(10,2)

SELECT @res = ISNULL(SUM(r.reservaMonto),0)

FROM Reserva r

JOIN Gato g ON g.gatoID = r.gatoID

WHERE g.propietarioDocumento = @DocPropietario AND

MONTH(GETDATE())-1 = MONTH(r.reservaFechaInicio) AND

YEAR(GETDATE()) = YEAR(r.reservaFechaInicio)

RETURN @res

END

CREATE OR ALTER FUNCTION F\_MontoServiciosMesAnterior (

@DocPropietario CHAR(30) )

RETURNS INT

BEGIN

DECLARE @res DECIMAL(10,2)

SELECT @res = ISNULL(SUM(rs.cantidad\*s.servicioPrecio),0)

FROM Reserva\_Servicio rs

JOIN Servicio s ON s.servicioNombre = rs.servicioNombre

JOIN Reserva r ON r.reservaID = rs.reservaID

JOIN Gato g ON g.gatoID = r.gatoID

WHERE g.propietarioDocumento = @DocPropietario AND

MONTH(GETDATE())-1 = MONTH(r.reservaFechaInicio) AND

YEAR(GETDATE()) = YEAR(r.reservaFechaInicio)

RETURN @res

END

Código de la vista

CREATE OR ALTER VIEW View\_Ej6

AS

SELECT p.propietarioNombre,

dbo.F\_MontoReservasMesAnterior (p.propietarioDocumento) AS MontoReservas,

dbo.F\_MontoServiciosMesAnterior (p.propietarioDocumento) AS MontoServicios,

dbo.F\_MontoReservasMesAnterior (p.propietarioDocumento) + dbo.F\_MontoServiciosMesAnterior (p.propietarioDocumento)

AS MontoAFacturar

FROM Propietario p

Ejecutar la vista:

SELECT \*

FROM View\_Ej6

Resultado:

Tabla

Descripción generada automáticamente con confianza media

# MongoDB

## 7.1 Ejercicio a.

Se pide:

Cree la(s) colecciones de documentos JSON para almacenar una reserva y que luego se pueda consultar de forma óptima todos los datos de una reserva (habitación, servicios adicionales, gato, dueño, etc.). Oriente su diseño a performance de lectura como objetivo principal por encima de los criterios de integridad de datos que nos tiene acostumbrados el modelo relacional.

Creación de colecciones y datos para cumplir con las consultas solicitadas.

show databases

use CatHotel

db.createCollection("reservas", {})

db.createCollection("propietarios", {})

db.createCollection("gatos", {})

db.createCollection("habitaciones", {})

db.propietarios.insertMany([

{ propietarioDocumento: '12345678', propietarioNombre: 'Alejandro Torres', propietarioTelefono: '+598 91 234 567', propietarioEmail: 'alejandro.torres@gmail.com' },

{ propietarioDocumento: '51953448', propietarioNombre: 'Beatriz Mendoza', propietarioTelefono: '+598 92 345 678', propietarioEmail: 'beatriz.mendoza@hotmail.com' },

{ propietarioDocumento: '55525512', propietarioNombre: 'Inés Romero', propietarioTelefono: '+598 99 012 345', propietarioEmail: 'ines.romero@outlook.com' },

{ propietarioDocumento: '60150754', propietarioNombre: 'Karina Paredes', propietarioTelefono: '+598 2623 4567' }

])

db.gatos.insertMany([

{ gatoNombre: 'Darius', gatoRaza: 'Persa', gatoEdad: 11, gatoPeso: 8.2, propietarioDocumento: '55525512' },

{ gatoNombre: 'Luna', gatoRaza: 'Ragdoll', gatoEdad: 3, gatoPeso: 5.7, propietarioDocumento: '12345678' },

{ gatoNombre: 'Chloe', gatoRaza: 'Siberiano', gatoEdad: 5, gatoPeso: 6.9, propietarioDocumento: '51953448' },

{ gatoNombre: 'Cleo', gatoRaza: 'Siames', gatoEdad: 3, gatoPeso: 4.3, propietarioDocumento: '12345678' },

{ gatoNombre: 'Felix', gatoRaza: 'Bengalí', gatoEdad: 3, gatoPeso: 6.5, propietarioDocumento: '60150754' }

])

db.habitaciones.insertMany([

{ habitacionNombre: 'El Rascador', habitacionCapacidad: 3, habitacionPrecio: 80, habitacionEstado: 'DISPONIBLE' },

{ habitacionNombre: 'Suite1', habitacionCapacidad: 1, habitacionPrecio: 120, habitacionEstado: 'DISPONIBLE' },

{ habitacionNombre: 'La Estantería', habitacionCapacidad: 7, habitacionPrecio: 30, habitacionEstado: 'LLENA' },

{ habitacionNombre: 'La Caja', habitacionCapacidad: 4, habitacionPrecio: 70, habitacionEstado: 'DISPONIBLE' }

])

db.reservas.insertMany([

{

gato: { nombre: 'Luna', propietarioDocumento: '12345678', propietarioNombre: 'Alejandro Torres' },

habitacionNombre: 'La Caja',

estadia: { fechaInicio: '2024-11-23', fechaFin: '2024-12-04' },

monto: 840,

serviciosContratados: [{ nombre: 'PELUQUERIA', cantidad: 1, precio: 25 }, { nombre: 'BAÑO', cantidad: 1, precio: 20 }],

montoTotal: 885,

resenia: 'Muy buen trabajao con el servicio de peluquería, mi gata quedó hermosa'

},

{

gato: { nombre: 'Luna', propietarioDocumento: '12345678', propietarioNombre: 'Alejandro Torres' },

habitacionNombre: 'La Caja',

estadia: { fechaInicio: '2024-07-23', fechaFin: '2024-07-25' },

monto: 210,

serviciosContratados: [{ nombre: 'BAÑO', cantidad: 1, precio: 20 }],

montoTotal: 230,

resenia: 'Mi gata estuvo en buenas manos'

},

{

gato: { nombre: 'Felix', propietarioDocumento: '60150754', propietarioNombre: 'Karina Paredes' },

habitacionNombre: 'El Rascador',

estadia: { fechaInicio: '2024-03-23', fechaFin: '2024-04-04' },

monto: 1040,

serviciosContratados: [{ nombre: 'PELUQUERIA', cantidad: 5, precio: 125 }, { nombre: 'CONTROL\_PARASITOS', cantidad: 1, precio: 50 }],

montoTotal: 1215

},

{

gato: { nombre: 'Cleo', propietarioDocumento: '12345678', propietarioNombre: 'Alejandro Torres' },

habitacionNombre: 'La Estanteria',

estadia: { fechaInicio: '2024-07-01', fechaFin: '2024-07-10' },

serviciosContratados: [{ nombre: 'ALIMENTACION\_ESPECIAL', cantidad: 7, precio: 175 }],

montoTotal: 475,

},

{

gato: { nombre: 'Darius', propietarioDocumento: '55525512', propietarioNombre: 'Ines Romero' },

habitacionNombre: 'Suite1',

estadia: { fechaInicio: '2024-11-15', fechaFin: '2024-11-30' },

serviciosContratados: [{ nombre: 'ALIMENTACION\_ESPECIAL', cantidad: 7, precio: 175 }],

montoTotal: 2095,

propietarioSatisfecho: true

}

])

## 7.2 Ejercicio b.

Se pide: Listar reservas del propietario con documento "12345678"

Código:

db.reservas.find({

"gato.propietarioDocumento": "12345678"

})

Resultado:

Captura de pantalla de computadora

Descripción generada automáticamente

## 7.3 Ejercicio c.

Se pide: Listar las reservas que incluyen el servicio "PELUQUERIA "

Código:

db.reservas.find({

"serviciosContratados.nombre": "PELUQUERIA"

})

Resultado:

Captura de pantalla de computadora

Descripción generada automáticamente

## 7.4 Ejercicio d.

Se pide: Actualizar el estado de la habitación "Suite1" asegurándose que está en estado "DISPONIBLE" y pasándolo a estado "LLENA"

Se muestra el estado de la habitación inicial en disponible.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Código:

db.habitaciones.update(

{ habitacionNombre: "Suite1", habitacionEstado: "DISPONIBLE" },

{ $set: { "habitacionEstado": "LLENA" } }

)

Resultado:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

## 7.5 Ejercicio e.

Se pide: Listar nombre de propietario y cantidad de reservas con fecha de inicio en julio 2024, para los propietarios que tengan más de una reserva en ese mes.

Código:

db.reservas.aggregate(

{

$match: {

"estadia.fechaInicio": {

$gte: "2024-07-01",

$lt: "2024-08-01"

}

}

},

{ $group: { \_id: "$gato.propietarioNombre", count: { $sum: 1 } } },

{

$match: {

count: { $gt: 1 }

}

}

)

Resultado:

