

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS



FUNDAMENTOS DE BASES DE DATOS

Proyecto Final - Reporte

Equipo: eSosQLones

Estrada Garcia Luis Gerardo	319013832
Jiménez Hernández Allan	420003478
Mancera Quiroz Javier Alejandro	319274831
Mora Hernández Dulce Julieta	319236448
Peña Nuñez Axel Yael	318279754

1. Introducción

Este documento presenta una solución para el caso de uso para el curso de Fundamentos de Bases de Datos, enfocado en el Zoológico de Huitziltepec. El objetivo principal es desarrollar una base de datos para mejorar la gestión y eficiencia del zoológico, que actualmente utiliza un sistema basado en papel.

2. Problemática

La problemática del Zoológico de Huitziltepec incluye varios aspectos clave:

1. **Digitalización de Registros:** Transición de un sistema de registros en papel a una base de datos digital.
2. **Gestión de Empleados:** Almacenamiento de datos de empleados, incluyendo información personal y profesional.
3. **Manejo de Biomas y Animales:** Registro de información sobre biomas y animales, incluyendo especie, alimentación y cuidados médicos.
4. **Inventario de Insumos:** Control del inventario de insumos con detalles como cantidad, proveedor y fecha de caducidad.
5. **Organización de Eventos:** Gestión de eventos especiales, asistentes y detalles relacionados.
6. **Registro de Clientes/Visitantes:** Almacenamiento de información sobre visitantes, promociones y eventos.
7. **Gestión Financiera:** Sistema para registrar transacciones financieras, incluyendo ventas de servicios y productos, con detalles como costos, descuentos y datos del cliente.

3. Modelo Entidad - Relación

La traducción de la problemática del Zoológico de Huitziltepec a un Modelo Entidad-Relación es un paso crítico en la modernización de su sistema de gestión. Esto permite una visualización estructurada de la información y sus interconexiones, facilitando la implementación de una base de datos relacional. Este enfoque promueve la eficiencia, la integridad de los datos y la escalabilidad del sistema de gestión del zoológico.

Para poder generar el Modelo Entidad-Relacional lo primero que realizamos fue identificar la identificación de nuestras entidades, obteniendo así las siguientes entidades

- Persona. Decidimos definir una entidad 'Persona' dado que los Trabajadores y Clientes compartían los siguientes atributos en nuestro caso de uso, por lo que el uso de la herencia nos permitió generar la reutilización de atributos así como nos permite tener cierta abstracción que ayuda a evitar redundancias. Las características que comparten en común están entidades y que son los atributos de Persona, son

- Id_Persona. La llave de esta entidad
- Nombre Completo (Nombre, Apellido Paterno, Apellido Materno), es un atributo compuesto ya que se compone de los atributos que estan entre parentesis y juntos forman al nombre.
- Correo. Decidimos que fuera multivaluado ya que una persona puede tener más de un correo
- Teléfono. Decidimos que fuera de igual forma multivaluado ya que una persona puede contar con telefono celular, de casa, etc. por lo que nos parecia útil para el diseño colocar este atributo como un atributo multivaluado.

Además, por el caso de uso sabemos que existen tres tipos de trabajadores en nuestro zoológico, a saber: Veterinarios, Cuidadores y Proveedores de los cuales nos interesa guardar información en común así como información en particular de cada uno. Como volvemos a tener el caso en el que se comparten características, decidimos volver a utilizar herencia donde 'Trabajador' es la entidad que contiene los atributos compartidos y cada tipo de trabajador contiene las características particulares de cada uno. Además, decidimos utilizar la disyunción para fines de que al momento de generar el modelo relacional facilitar el manejo de la base de datos así como poder tener restricciones dado que la disyunción nos permite tener una entidad que puede estar vinculada exclusivamente con una las opciones disponibles, en este caso, Veterinario, Cuidador o Proveedor.

A partir de lo anterior tenemos a Trabajador con los siguientes atributos

- Trabajador
 - Dirección(Calle, Colonia, Numero Exterior, Numero Interior, Estado)
 - Fecha Nacimiento
 - Fecha Inicio de Contrato
 - Fecha Fin de Contrato
- Veterinario
 - Salario
 - Especialidad. Decidimos que especialidad fuera un atributo multivaluado ya que un veterinario puede tener más de una especialidad.
- Cuidador. Además de los atributos que toma por parte de 'Trabajador', Cuidador cuenta con los siguientes atributos
 - Hora_Entrada
 - Hora_Salida

- Horario_Lab
 - Dias_Trab
 - Salario
- Proveedor. En el caso de proveedor notemos que además de los atributos que hereda por parte de 'Trabajador', tenemos también los siguientes atributos
- Costo_Prod
 - Nombre_Prod
 - Frecuencia
 - Comision
- Cliente. La entidad cliente no tiene atributos extra a los que hereda dado que el caso de uso solo nos pide los atributos que se heredan.

Además de lo anterior también encontramos herencia en los siguientes casos

- Ticket_Compra
 - Ticket_Serv
 - Ticket_Art
- Producto
 - Servicio
 - Insumo que también tiene herencia
 - Alimento
 - Medicina
 - Articulo
- Notificación
 - Promocion
 - Descuento
 - Noti_Evento

En todos los casos se decidió utilizar herencia con disyunción para evitar inconsistencias.

El caso de uso menciona lo siguiente "*Los biomas representan el habitat donde van a vivir los animales. Para eso es necesario guardar: Tipo de bioma (desierto, pastizales, franja costera, tundra, aviario, bosque templado, bosque tropical), número de jaulas, número de cuidadores, número de veterinarios, número de animales, servicios a visitantes(baños, tiendas, comida).*" Para representar lo anterior tenemos a la entidad 'Bioma' que cuenta con los siguientes atributos

- Tipo, que funciona como nuestra llave

- Num_Jaulas
- Num_Cuid
- Num_Vet
- Num_Anim

Notemos que en todos los casos a excepción de la llave, consideramos a los atributos como atributos derivados dado que son atributos que podemos calcular más adelante.

Para guardar a los animales, creamos la entidad 'Animal' que cuenta con los siguientes atributos

- Id_Animal
- Especie
- indicacionesMed, este atributo se considera multivaluado ya que un animal puede tener diferentes indicaciones médicas, por lo que para almacenarlas, consideramos a este atributo de esa forma
- Num_Jaula
- Nombre
- Alimentación
- Sexo
- Altura
- Peso

Para poder llevar el almacenamiento de los insumos, realizamos la entidad 'Insumo', además como tenemos que existen insumos de tipo Alimento y de tipo Medicina, ocupamos herencia(como se menciono lineas atras), y para cada entidad tendremos los siguientes atributos

- Insumo
 - Cantidad
 - seRefrigera
 - fechaCad
- Alimento
 - tipoAlimento
- Medicina

- labProcedencia

Además, sabemos que el zoologico realiza eventos, para guardar información de ellos tendremos a la entidad 'Evento' cuya entidad tendrá los siguientes atributos

- Id_Evento
- Capacidad
- tipoEvento(en el caso de que tipo de evento es, las restricciones se realizaran cuando se este poblando la base de datos)
- Fecha

Para el caso de las promociones, descuentos y notificaciones tendremos las siguientes entidades

- Notificación de donde heredan las siguientes entidades y que solo tiene Id_Noti como llave
 - Promoción con los siguientes atributos
 - Tipo_Promo
 - Fecha_Inicio
 - Fecha_Fin
 - Nombre_Art
 - Descuento con los siguientes atributos
 - Porcentaje_Desc
 - Nombre_Art
 - Fecha_Fin
 - Fecha_Inicio
 - Noti_Evento que solo tiene Id

Para la generación de tickets una vez más decidimos aplicar herencia, como sigue

- Ticket_Compra con atributos: NumTicket(llave), costoTotal, Cantidad_Prod, Fecha
 - Ticker_Serv, entidad creada para definir los tickets de los servicios que tiene el zoologico.
 - Ticket_Art, entidad creada para definir los tickets de los artículos

Para que las entidades interactuen entre si, nos ayudamos de las Relaciones, que además son definidas por el mismo caso de uso. Las relaciones que tomamos en consideración son las siguientes

1. Para definir "No olvides que un veterinario a lo más puede trabajar en dos biomas" creamos la relación Trabajar entre las entidades Veterinario y Bioma con cardinalidad parcial ya que como se menciona que es a lo más podemos tener el caso en el que no suceda, además consideramos la cardinalidad N:M dado que sabemos que a lo más puede trabajar en dos biomas (esta restricción se coloca cuando se está implementando la base de datos).
2. "mientras que los cuidadores solo pueden trabajar en uno." colocamos la relación Laborar, con participación parcial dado que no forzosamente un cuidador debe de trabajar en un bioma, además, colocamos cardinalidad 1:N entre las entidades Bioma y Cuidador dado que un bioma puede tener muchos cuidadores pero un cuidador solo puede trabajar en un bioma.
3. En el caso de uso se menciona también lo siguiente "Es importante saber que un animal sólo puede estar en una jaula y pertenecer a un bioma." para poder definir la primera parte de las jaulas tendremos que realizarlo cuando se implemente la base de datos cuidando que en efecto se cumpla lo anterior, para la segunda parte de la petición realizamos la relación 'Pertenecer' que tendrá participación total dado que cada animal obligatoriamente debe pertenecer a un bioma y los biomas deben tener animales. La cardinalidad será 1:N entre Bioma y Animal dado que 1 bioma tiene muchos animales pero un animal puede pertenecer a un único bioma.
4. "Además un animal sólo puede ser cuidado por un cuidador" para representar esto colocamos la relación Cuidar cuya cardinalidad es 1:1 con 1 total del lado de Cuidador ya que un animal debe ser cuidado forzosamente por un cuidador pero podemos tener cuidadores que no cuiden animales
5. "y atendido por varios veterinarios solo si son de su mismo bioma." para definir esto tenemos la relación Atender cuya cardinalidad es N:M entre las entidades Veterinario y Animal con participación total del lado de veterinario ya que un animal debe ser atendido forzosamente por varios veterinarios (la restricción de ser del mismo bioma se verifica en la implementación de un disparador) pero podemos tener veterinarios que no atiendan a los animales
6. "Un alimento puede ser distribuido en varios biomas." para dar solución a esta línea, creamos la relación distribuir con cardinalidad N:M con participación parcial
7. "Proveedores: Producto que provee, frecuencia que ofrece el servicio y costo por servicio." para esta línea, interpretamos como que un proveedor solo puede proveer un producto

de los que pueden ser provistos por algún proveedor, pero, restringimos también que un único producto solo puede ser provisto por un solo proveedor, lo que da paso a la relación 1:1, esto ya que creímos que sería mejor tener una relación 1:1 en este caso.

Para los casos de los tickets y las notificaciones, tendremos las siguientes relaciones

- Asociar entre Articulo y Ticket_Art con cardinalidad N:M y participación total del lado de Ticket_Art dado que los tickets forzosamente estan asociados a un articulo; además la relación cuenta con los atributos costoArt, nombreArt, PromoArt y DescuentoArt
- Registrar entre Servicio y Ticket_Serv con cardinalidad 1:N y participación total del lado de muchos, esto ya que 1 servicio puede tener muchos tickets pero un ticket solo pertenece a un servicio. Además la relación cuenta con los atributos Descuento, costoUnitario y nombreServicio ya que es información que nos interesa guardar al generar el ticket
- La relación Producir entre las entidades Articulo y Promoción con cardinalidad 1:1 y participación total del lado de Promoción, ya que la promoción depende del articulo.
- La relación Hacer entre las entidades Articulo y Descuento con cardinalidad 1:1 y participación total del lado de Descuento, ya que el descuento depende del articulo.
- La relación Generar entre las entidades Evento y Noti_Evento con cardinalidad 1:1 y participación total del lado de Noti_Evento, ya que la notificación depende de si el evento se lleva a cabo.

Además, los clientes pueden asistir a eventos y recibir notificaciones, para esto generamos las siguientes relaciones

- Asistir entre Evento y Cliente con cardinalidad N:M y participación total del lado de Evento ya que los eventos deben de tener clientes pero los clientes pueden ir o no ir. Además de esta relación nos interesa obtener la listaAsistentes y al Cliente
- La relación Recibir entre Notificación y Cliente con cardinalidad N:M y que tiene participación total del lado de notificación ya que una notificación es forzosamente enviada a un cliente pero un cliente no recibe forzosamente una notificación.

3.1. Llaves primarias de las entidades

- Alimento : Id_Alimento.

En su atributo Cantidad, puede existir que haya alimentos con la misma cantidad; en se-Refrigera, como cada instancia puede tomar valores booleanos para este atributo entonces se pueden repetir; en Fecha_Cad, puede haber alimentos que caducan en la misma fecha; en tipoAlimento, puede que dos alimentos sean del mismo tipo. Como no hay algún atributo de Alimento que se pueda ocupar como llave primaria, es decir que no está vacío, que

sea único y se pueda distinguir cada instancia de la entidad, agregamos un atributo que cumpla con las características mencionadas y será nuestra llave primaria para Alimento. Por lo tanto, creamos la llave sustituta Id_Alimento, en la que su tipo de dato asociado es de tipo entero.

- Animal : Id_Animal.

En su atributo Especie, puede haber dos animales de la misma especie; en Indicaciones-Med, puede que dos animales tengan las mismas indicaciones médicas; en NumJaula, dos animales pueden estar en la misma jaula; en Nombre, pueda que dos animales tengan el mismo nombre; en Alimentacion, dos animales pueden tener la misma alimentación; en Sexo, hay animales con el mismo sexo; en Altura y Peso, hay animales con la misma altura y peso. Como no hay algún atributo de Animal que se pueda ocupar como llave primaria, es decir que no esta vacío, que sea único y se pueda distinguir cada instancia de la entidad, agregamos un atributo que cumpla con las características mencionadas y será nuestra llave primaria para Animal. Por lo tanto, creamos la llave sustituta Id_Animal, en la que su tipo de dato asociado es de tipo entero.

- Articulo : Id_Articulo.

En su atributo Cantidad_Arts, como estamos asociandolo a un ticket entonces pueda que haya dos tickets con la misma cantidad de artículos comprados. Como no hay algún atributo de Articulo que se pueda ocupar como llave primaria, es decir que no esta vacío, que sea único y se pueda distinguir cada instancia de la entidad, agregamos un atributo que cumpla con las características mencionadas y será nuestra llave primaria para Articulo. Por lo tanto, creamos la llave sustituta Id_Articulo, en la que su tipo de dato asociado es de tipo entero.

- Bioma : Tipo.

- Cliente : CURP.

- correoClien : CURP, Correo_Clien.

- correoCuid : RFC_Cuid, Correo_Cuid.

- correoProv : RFC_Prov, Correo_Prov.

- correoVet : RFC_Vet, Correo_Vet.

- Cuidador : RFC_Cuid.

- Descuento : Id_Desc.

En su atributo Porcentaje, pueden existir compras que se le aplicaron el mismo porcentaje de descuento; y en sus atributos de Fecha_Inicio y Fecha_Fin, pueda que haya descuentos con diferentes porcentajes pero con las mismas fechas de inicio y de término. Como no hay algún atributo de Descuento que se pueda ocupar como llave primaria, es decir que no esta vacío, que sea único y se pueda distinguir cada instancia de la entidad, agregamos un atributo que cumpla con las características mencionadas y será nuestra llave primaria para Descuento. Por lo tanto, creamos la llave sustituta Id_Desc, en la que su tipo de dato asociado es de tipo entero.

- Especialidad : Especialidad, RFC_Vet.

- Evento : Id_Evento.

En su atributo Capacidad, puede haber eventos con la misma cantidad de asistentes; en su atributo TipoEvento, puede haber eventos del mismo tipo; y en su atributo Fecha, puede haber evento de diferentes tipos pero que se realizan en la misma fecha. Como no hay algún atributo de Evento que se pueda ocupar como llave primaria, es decir que no esté vacío, que sea único y se pueda distinguir cada instancia de la entidad, agregamos un atributo que cumpla con las características mencionadas y será nuestra llave primaria para Evento. Por lo tanto, creamos la llave sustituta Id_Evento, en la que su tipo de dato asociado es de tipo entero.

- indicacionesMed : Id_Animal, IndicacionesMed.

- Medicina : Lote.

- noti_Evento : Id_Noti_Evento.

Como no tiene atributos noti_Evento agregamos un atributo que cumpla de que no esté vacío, que sea único y que se pueda distinguir cada instancia de la entidad. Por lo tanto, creamos la llave sustituta Id_Noti_Evento, en la que su tipo de dato asociado es de tipo entero.

- Promocion : Id_Promo.

En su atributo Tipo, pueden existir promociones del mismo tipo; y en sus atributos de Fecha_Inicio y Fecha_Fin, pueda que haya diferentes promociones pero con las mismas fechas de inicio y de término. Como no hay ningún atributo de Promocion que se pueda ocupar como llave primaria, es decir que no esté vacío, que sea único y se pueda distinguir cada instancia de la entidad, agregamos un atributo que cumpla con las características mencionadas y será nuestra llave primaria para Promocion. Por lo tanto, creamos la llave sustituta Id_Promo, en la que su tipo de dato asociado es de tipo entero.

- Proveedor : RFC_Prov.

- Servicio : Id_Servicio.

Como no tiene atributos Servicio agregamos un atributo que cumpla de que no esté vacío, que sea único y que se pueda distinguir cada instancia de la entidad. Por lo tanto, creamos la llave sustituta Id_Servicio, en la que su tipo de dato asociado es de tipo entero.

- TelClien : CURP, Tel_Clien.

- TelCuid : RFC_Cuid, Tel_Cuid.

- TelProv : RFC_Prov, Tel_Prov.

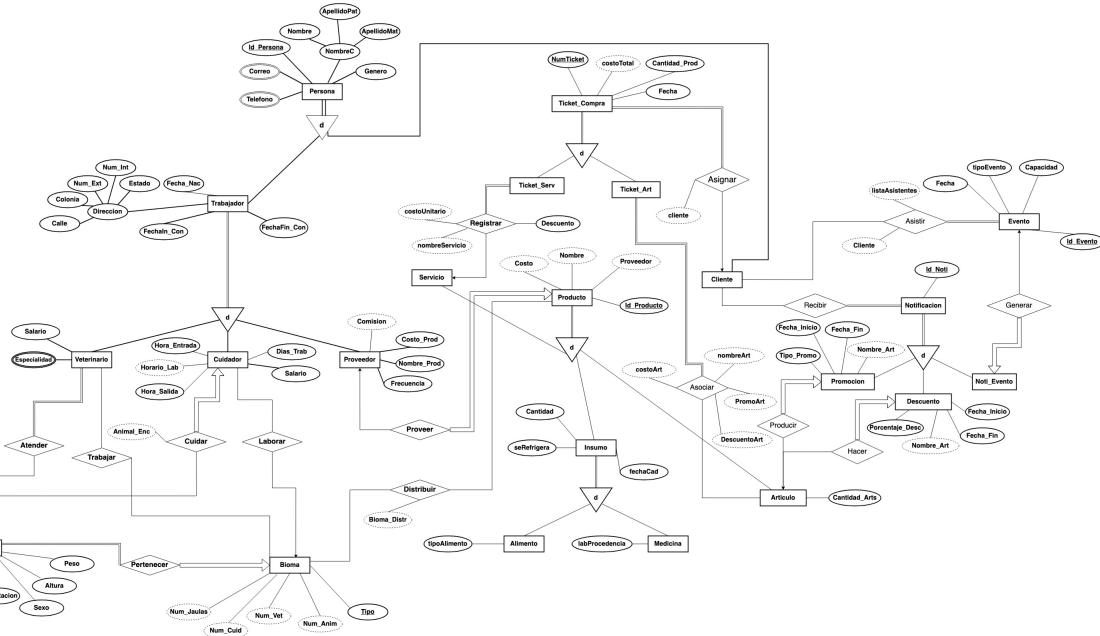
- TelVet : RFC_Vet, Tel_Vet.

- ticket_Art : Num_Ticket_Art.

- ticket_Serv : Num_Ticket_Serv.

- Veterinario : RFC_Vet.

3.2. Diagrama

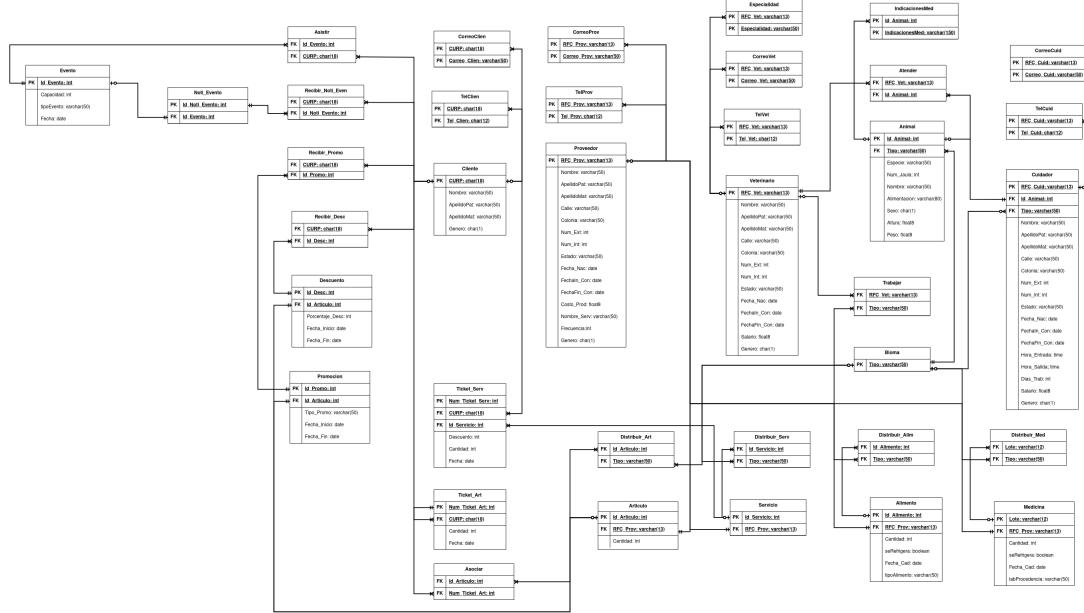


4. Modelo Relacional

La traducción de la operativa y administrativa de un zoológico a un modelo relacional es un paso esencial en la modernización de sus sistemas de información. El modelo relacional aporta múltiples beneficios que son fundamentales para la gestión eficiente de un zoológico:

- Estructura Organizada:** Facilita la representación sistemática y organizada de los datos, lo cual es crucial para manejar la complejidad de la información en un zoológico.
- Integridad de Datos:** Asegura que los datos sean precisos y coherentes, lo que es vital para el registro exacto de información sobre los animales, empleados y eventos.
- Flexibilidad en Consultas:** Permite realizar consultas eficientes y variadas, esencial para obtener información detallada y compleja rápidamente.
- Escalabilidad:** Ofrece la capacidad de expandir la base de datos a medida que crecen las necesidades informativas del zoológico, sin perder rendimiento ni calidad.
- Mantenimiento del Sistema:** Simplifica las actualizaciones y el mantenimiento del sistema, lo que contribuye a una gestión de datos más eficiente y menos propensa a errores.
- Seguridad de los Datos:** Mejora la protección de la información sensible mediante el control de acceso a los datos basado en las relaciones definidas.

4.1. Diagrama



5. Consultas SQL

Las consultas SQL en una base de datos son fundamentales por varias razones:

- Recuperación de Datos:** SQL permite recuperar datos específicos de una base de datos de manera eficiente. Puedes seleccionar una fracción de los datos de una gran base de datos según tus necesidades.
- Manipulación de Datos:** Aparte de la recuperación, SQL permite insertar, actualizar y eliminar datos, lo que ayuda en la gestión y mantenimiento de la base de datos.
- Análisis de Datos:** Las consultas SQL pueden ser utilizadas para analizar datos, como calcular promedios, sumas y otras operaciones estadísticas, lo que es crucial para tomar decisiones basadas en datos.
- Automatización de Tareas:** Las consultas SQL se pueden automatizar y programar para ejecutar tareas regulares, como informes periódicos, lo que ahorra tiempo y reduce los errores humanos.
- Control de Acceso:** SQL permite restringir el acceso a los datos mediante permisos y roles, lo que es esencial para la seguridad y la integridad de la base de datos.

5.1. Consultas aplicadas al caso de uso

Consulta 1: Obtener el listado de proveedores que han tenido contratos por más de 3 años, incluyendo la cantidad total que se ha gastado en sus servicios y el número de contratos que han tenido.

```
/*
 * Consulta 1
 * Obtener el listado de proveedores que han tenido contratos por
 * más de 3 años, incluyendo la cantidad total que se ha gastado en
 * sus servicios y el número de contratos que han tenido.
*/
SELECT
    p.RFC_Prov,
    p.Nombre,
    p.ApellidoPat,
    p.ApellidoMat,
    COUNT(*) AS NumeroDeContratos,
    SUM(p.Costo_Serv) AS GastoTotal
FROM
    proveedor p
WHERE
    (p.FechaFin_Con::DATE - p.FechaIn_Con::DATE) / 365 > 3
GROUP BY
    p.RFC_Prov, p.Nombre, p.ApellidoPat, p.ApellidoMat
ORDER BY
    GastoTotal DESC;
```

proveedor 1 X

Enter a SQL expression to filter results (use Ctrl+Space)

	rfc_prov	nombre	apellidoPat	apellidoMat	numerodecontratos	gastototal
1	a5hAaYACUe6r	Bertha	Sibbons	Spriggin	1	19,252.89
2	LLYyQ7SiF7tT	Franchot	Gladebeck	Ales	1	17,250.1
3	5q0C60eBk9Q1	Maryjo	Jaxon	Stoakley	1	17,221.63
4	wOs37q041N1X	Merl	Raitt	Nicolson	1	17,078.72
5	SOA77ebhSgC8	Elinore	Fearnough	Bonnin	1	17,020.78
6	N6BI1Z054hKe	Ad	Dellit	McConway	1	16,909.44
7	v6O2CE3s1K7	Diana	Willcox	Trassler	1	16,844.14
8	JO22PAByxda0	Maddi	Stansall	Bodsworth	1	16,834.85
9	3dh3pJZ3QVEk	Anthony	Urrey	Faulconbridge	1	16,339.86
10	IkVQybu4Wo6n	Ilyse	Bedbury	Piggens	1	16,104.53
11	6wKq8JTv8ASo	Claribel	Wardrobe	Hush	1	15,897.75
12	8u4X10G23h42	Nevin	Blackader	Bambra	1	15,630.4
13	IP5oOfVVx49	Anselm	Corwin	Gillopp	1	15,620.27
14	290S2mMcOBcz	Jodie	O'Scully	Lambarth	1	15,101.14
15	IOnPrPv1QGKu	Arlvn	Heavyside	Wimmer	1	14,943.41

Renovar ▾ Save ▾ Cancel ▾ Exportar datos ... ▾ 200

Consulta 2: Obtener el promedio de altura y peso para cada especie de animal en el zoológico.

```
/*
 * Consulta 2
 * Obtener el promedio de altura y peso para cada especie de animal en el zoológico
 */
SELECT
    especie,
    AVG(altura) AS promedio_altura,
    AVG(peso) AS promedio_peso
FROM
    animal
GROUP BY
    especie;
```

	especie	promedio_altura	promedio_peso
1	Naulty	56.68	272.5
2	Rennebeck	132.83	210.05
3	Kybert	23.12	111.12
4	Jerwood	96.61	315.62
5	Vanes	28.84	77.38
6	Castro	118.24	14.76
7	Garie	12.39	306.54
8	Cridlon	75.4	324.82
9	Sigart	115.74	392.2
10	Happel	80.38	291.03
11	Beardwood	148.98	6.97
12	Rippingall	148.56	365.74
13	Dilgarno	53.69	196.75
14	Langrick	92.93	223.21
15	Alecock	75.75	389.07
16	Wegenen	137.63	387.11
17	Wallege	106.67	48.29
18	Shord	145.65	48.96
19	Gambie	50.04	108.56
20	Semmence	9.78	166.08
21	Marciek	118.18	261.43
22	Dowse	72.54	116.13
23	Hansana	7.11	213.41

Consulta 3: Obtener una lista de todos los alimentos que requieren refrigeración junto con la información del proveedor y la fecha de caducidad de estos alimentos.

```

/*
 * Consulta 3
 * Obtener una lista de todos los alimentos que requieren refrigeración
 * junto con la información del proveedor y la fecha de caducidad de estos alimentos.
*/
SELECT
    a.tipoalimento,
    a.fecha_cad,
    p.nombre,
    p.apellidopat,
    p.apellidomat
FROM
    alimento a
JOIN
    proveedor p ON a.rfc_prov = p.rfc_prov
WHERE
    a.serefriega = TRUE
ORDER BY
    a.fecha_cad;
```

The screenshot shows a database interface with several tabs at the top: proveedor 1, Resultados 2, animal 3, alimento(+) 4, and alimento(+) 5. The active tab is 'Resultados 2'. Below the tabs, there is a search bar with placeholder text 'Enter a SQL expression to filter results (use Ctrl+Space)'. The main area displays a table with 19 rows of data. The columns are: tipoalimento, fecha_cad, nombre, apellidopat, and apellidomat. The data includes various food items like Insectos, Plancton, Granos, Forraje, etc., along with their expiration dates and supplier names. At the bottom of the interface, there are buttons for Renovar, Save, Cancel, and Exportar datos, along with a page number indicator (48).

	tipoalimento	fecha_cad	nombre	apellidopat	apellidomat
1	Insectos	2023-12-23	Jarid	Yashunin	Doumenc
2	Plancton	2024-01-03	Lanette	Agerskow	Rubinowitsch
3	Granos	2024-01-11	Tallie	Cotmore	Danjoie
4	Forraje	2024-02-10	Jodie	O'Scully	Lambarth
5	Granos	2024-02-17	Isa	Startin	Van Der Vlies
6	Frutas	2024-02-24	Nil	Lawford	Sennock
7	Hierbas	2024-03-03	Chas	Partener	Trebilcock
8	Frutas	2024-03-07	Pepita	Canete	Tincombe
9	Pescados	2024-03-13	Jo-anne	Kampshell	Ousley
10	Semillas	2024-03-17	Alisun	Dykins	Tanzer
11	Pescados	2024-03-20	Odessa	Fradson	Fricke
12	Algas	2024-04-08	Clementia	Fodden	Piburn
13	Verduras	2024-04-10	Marice	Glison	Ioan
14	Verduras	2024-04-20	Thomasine	Clappson	Bafford
15	Pescados	2024-05-13	Merridie	Mandres	Jovis
16	Forraje	2024-06-05	Anselm	Corwin	Gillopp
17	Algas	2024-08-20	Louie	Dumphries	Dagnall
18	Frutas	2024-09-17	Astrix	Singers	Applegarth
19	Forraje	2024-09-22	Elinore	Fearnough	Bonnin

Consulta 4: Identificar los proveedores cuyos contratos están próximos a vencer en los próximos 180 días mostrando detalles del proveedor y la fecha de finalización del contrato.

```

    /**
     * Consulta 4
     * Identificar los proveedores cuyos contratos están próximos a vencer en los próximos 30 días
     * mostrando detalles del proveedor y la fecha de finalización del contrato
     */
    SELECT
        p.rfc_prov,
        p.nombre,
        p.apellidopat,
        p.apellidomat,
        p.costo_serv,
        p.fechaфин_кон
    FROM
        proveedor p
    WHERE
        p.fechaфин_кон BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '180 days';

```

	rfc_prov	nombre	apellido pat	apellido mat	costo_serv	fechaфин_кон
1	4zHYs5HJ0UsV	Celestina	Mallindine	Smerdon	8,047.53	2023-12-11
2	XY2pl303lexZ	Anabelle	Bennoe	David	12,827.41	2023-12-05
3	Qur2Q7YK4FhL	Marice	Glison	loan	5,762	2023-12-03
4	eYcv2Z4CSVtG	Susette	Bernocchi	Kean	6,243.39	2024-05-25
5	Yg8k92gzCQeH	Dierdre	Easun	Martinovsky	8,340.72	2024-05-07
6	1h918Tx8y2y5	Jeannette	Thom	Twyford	10,620.8	2024-04-29
7	2rin3S59Yr5Y	Ross	Castagne	Garces	9,201.3	2024-02-17
8	VXgPF7iQzgns	Nealon	Hundal	Dast	13,943.84	2023-12-04
9	YpedH3iCEbWj	Collette	Scown	Skews	12,342.01	2024-04-11
10	LLYyQ7SiF7tT	Franchot	Gladebeck	Ales	17,250.1	2024-02-03
11	v602CE3s11K7	Diana	Willcox	Trassler	16,844.14	2024-05-21
12	Gb92T3Jw9TYY	Darell	MacRedmond	Wordley	10,126.14	2023-12-16
13	8AfhV0IA0JDR	Shannah	Gabriel	WatSam	13,449.66	2024-01-25
14	IkVQybu4Wo6n	Ilyse	Bedbury	Piggens	16,104.53	2024-02-03
15	3XpGp0dY0XwX	Buckie	Aisthorpe	Stourton	11,484.47	2024-04-18
16	08lq30z40OPo	Louie	Dumphries	Dagnall	15,874.83	2023-12-17
17	1fz9x82vgeOF	Odessa	Fradson	Fricke	13,214.86	2024-05-22
18	SOA77ebhSgC8	Elinore	Fearnighough	Bonnin	17,020.78	2024-01-10
19	f4OvHhG4fqwY	Davey	Buckthorpe	Condon	692.85	2024-02-26

Consulta 5: Identificar los tipos de alimentos y medicamentos más distribuidos en cada tipo de bioma lo que puede ayudar a optimizar la logística y el inventario.

```

/*
 * Consulta 5
 * Identificar los tipos de alimentos y medicamentos más distribuidos en cada tipo de bioma
 * lo que puede ayudar a optimizar la logística y el inventario
 */

SELECT
    b.tipo AS tipo_bioma,
    da.tipo AS tipo_alimento,
    COUNT(da.id_alimento) AS cantidad_alimentos_distribuidos,
    dm.tipo AS tipo_medicamento,
    COUNT(dm.lote) AS cantidad_medicamentos_distribuidos
FROM
    bioma b
LEFT JOIN
    distribuir_alim da ON b.tipo = da.tipo
LEFT JOIN
    distribuir_med dm ON b.tipo = dm.tipo
GROUP BY
    b.tipo, da.tipo, dm.tipo
ORDER BY
    cantidad_alimentos_distribuidos DESC, cantidad_medicamentos_distribuidos DESC;

```

bioma(+) 8 ×

Enter a SQL expression to filter results (use Ctrl+Space)

	b.tipo_bioma	da.tipo	cantidad_alimentos_distribuidos	dm.tipo	cantidad_medicamentos_distribuidos
1	Estepa	Estepa	70	Estepa	70
2	Selva	Selva	60	Selva	60
3	Pastizales	Pastizales	54	Pastizales	54
4	Desierto	Desierto	48	Desierto	48
5	Franja costera	Franja costera	45	Franja costera	45
6	Avíario	Avíario	36	Avíario	36
7	Manglar	Manglar	35	Manglar	35
8	Bosque Templado	Bosque Templado	30	Bosque Templado	30
9	Pradera	Pradera	30	Pradera	30
10	Arrecife de Coral	Arrecife de Coral	30	Arrecife de Coral	30
11	Pantano	Pantano	30	Pantano	30
12	Chaparral	Chaparral	28	Chaparral	28
13	Bosque de Coníferas	Bosque de Coníferas	24	Bosque de Coníferas	24
14	Sabana	Sabana	18	Sabana	18
15	Bosque Tropical	Bosque Tropical	10	Bosque Tropical	10
16	Tundra	Tundra	9	Tundra	9
17	Taiga	Taiga	2	Taiga	2

Renovar ▾ Save ▾ Cancel ▾ Exportar datos ... ▾ 200 17 ... 17 row(s) fetched - 15ms, on 2023-11-30 at 18:12:02

Consulta 6: Calcular la cantidad promedio de animales asignados a cada cuidador y su salario promedio, para evaluar la relación entre la carga de trabajo y la compensación.

```

/*
 * Consulta 6
 * Calcular la cantidad promedio de animales asignados a cada cuidador y su salario promedio, para evaluar
 * la relación entre la carga de trabajo y la compensación
 */

SELECT
    cu.nombre,
    cu.apellidopat,
    cu.apellidomat,
    COUNT(a.id_animal) AS cantidad_animales_cuidados,
    AVG(cu.salario) AS salario_promedio
FROM
    cuidador cu
JOIN
    animal a ON cu.id_animal = a.id_animal
GROUP BY
    cu.rfc_cuid, cu.nombre, cu.apellidopat, cu.apellidomat;

```

The screenshot shows a database interface with the following details:

- Biología (+ 8)**: The current database or schema name.
- Resultados 2**: The number of results found.
- cuidador 3**: The table being queried.
- SQL Query:** `SELECT cu.nombre, cu.apellidopat, cu.apellido`
- Table Headers:** nombre, apellidopat, apellidomat, cantidad_animales_cuidados, salario_promedio
- Data Rows (17 total):**

	nombre	apellidopat	apellidomat	cantidad_animales_cuidados	salario_promedio
1	Trixie	Loody	Ambrosoli	1	13,059
2	Verna	Chaize	Gildersleeve	1	24,275
3	Blanca	Matushevich	Maciunas	1	23,597
4	Ruddy	Massimi	Pennock	1	12,625
5	Stanislaus	Oades	Lunt	1	18,890
6	Boniface	Huby	Worsell	1	26,091
7	Ashil	Plesing	Lensch	1	12,862
8	Leanna	Dargie	Dallewater	1	25,514
9	Renee	Carling	Wedmore.	1	22,907
10	Kipp	Manley	Girard	1	13,443
11	Edmon	Sherston	Blackleech	1	21,016
12	Allyn	Lurriman	Kingcote	1	12,221
13	Leroi	Crosier	Martill	1	20,268
14	Chere	Eggleton	Gomby	1	17,349
15	Marilee	Beddow	Butting	1	13,748
16	Xever	Hillett	Buntin	1	20,635
17	Joane	Jiggen	Avey	1	11,683
- Bottom Buttons:** Renovar, Save, Cancel, and various navigation and export options.

Consulta 7: Para cada veterinario especializado en un tipo específico de animal, obtener el promedio de peso de los animales que ha atendido. Esto puede ayudar a entender qué tan equipados están los veterinarios para manejar diferentes tamaños y tipos de animales.

```

/*
 * Consulta 7
 * Para cada veterinario especializado en un tipo específico de animal, obtener el promedio de peso de los animales que ha atendido
 * Esto puede ayudar a entender qué tan equipados están los veterinarios para manejar diferentes tamaños y tipos de animales
 */

SELECT
    v.nombre,
    v.apellidopat,
    v.apellidomat,
    t.tipo,
    AVG(a.peso) AS peso_promedio
FROM
    veterinario v
JOIN
    atender at ON v.rfc_vet = at.rfc_vet
JOIN
    animal a ON at.id_animal = a.id_animal
JOIN
    trabajar t ON v.rfc_vet = t.rfc_vet
WHERE
    a.tipo = t.tipo
GROUP BY
    v.rfc_vet, t.tipo
ORDER BY
    peso_promedio DESC;

```

Resultados:

	ABC nombre	ABC apellidopat	ABC apellidomat	ABC tipo	123 peso_promedio
1	Drusilla	Philips	Stoving	Tundra	387.11
2	Jerrilyn	Biset	Proback	Selva	270.29
3	Townie	Fourcade	Brand-Hardy	Bosque Templado	259.14
4	Welch	Ferenczy	Jocey	Bosque Templado	171.18
5	Deena	Brunel	Houndsen	Sabana	161.52
6	Arlan	Folliott	Browett	Tundra	48.96
7	Kori	Lartice	Brody	Taiga	38.51

Consulta 8: Encontrar la proporción de animales atendidos por cada veterinario en comparación con la duración de su contrato para evaluar la eficiencia del veterinario en relación con su experiencia en el zoológico.

```

/*
 * Consulta 8
 * Encontrar la proporción de animales atendidos por cada veterinario en comparación con la duración de su contrato
 * para evaluar la eficiencia del veterinario en relación con su experiencia en el zoológico
 */
SELECT
    v.nombre,
    v.apellidopat,
    v.apellidomat,
    AGE(CURRENT_DATE, v.fechain_con) AS duracion_contrato,
    (COUNT(at.id_animal) / EXTRACT(YEAR FROM AGE(CURRENT_DATE, v.fechain_con))) AS promedio_anual_atendidos
FROM
    veterinario v
JOIN
    atender at ON v.rfc_vet = at.rfc_vet
GROUP BY
    v.rfc_vet
HAVING
    AGE(CURRENT_DATE, v.fechain_con) > INTERVAL '1 year'
ORDER BY
    promedio_anual_atendidos DESC;

```

SELECT v.nombre, v.apellidopat, v.apellidomat, Enter a SQL expression to filter results (use Ctrl+Space)

	RBC nombre	RBC apellido paterno	RBC apellido materno	duracion_contrato	promedio_anual_atendidos
1	Tad	Grunder	Dikelin	1 year 4 mons 27 days	1
2	Pen	Waison	Clampe	1 year 27 days	1
3	Tamarah	Tooker	Queree	1 year 3 mons 26 days	1
4	Kristel	Alexandre	Comiskey	1 year 6 mons	1
5	Abel	Arthur	Jailler	1 year 5 mons 20 days	1
6	Adan	Herrero	Garret	1 year 15 days	1
7	Gwenny	Petchell	Sully	1 year 9 mons 25 days	1
8	Tammy	Sudron	Fosten	2 years 11 mons 7 days	0.5
9	Judie	Chung	Rabbet	2 years 6 mons 25 days	0.5
10	Ike	Jahncke	Hanlin	2 years 6 mons 12 days	0.5
11	Valle	Michurin	Seacombe	2 years 11 mons 6 days	0.5
12	Brunhilde	Nimmo	Gookey	3 years 10 mons 2 days	0.3333333333
13	Deena	Brunel	Houndson	3 years 27 days	0.3333333333
14	Arlan	Folliott	Browett	3 years 11 mons 18 days	0.3333333333
15	Leeann	Andreix	Vere	3 years 1 mon 27 days	0.3333333333
16	Sari	Keogh	Frede	4 years 10 mons 19 days	0.25
17	Richmound	Joddens	Jiranek	4 years 6 mons 18 days	0.25
18	Alaine	Bagley	Acock	4 years 11 mons 18 days	0.25

Renovar ▾ Save ▾ Cancel ▾ Exportar datos ... 200 99 99 row(s) fetch

CST | es_MX | Editable | Inserción inteligente | 4 : 93 : 226 | Sel: 0 | 0

Consulta 9: Encontrar clientes que han recibido promociones pero que no han asistido a eventos lo que puede indicar oportunidades perdidas para aumentar la participación en eventos futuros.

```
/*
 * Consulta 9
 * Encontrar clientes que han recibido promociones pero que no han asistido a eventos
 * lo que puede indicar oportunidades perdidas para aumentar la participación en eventos futuros
 */
SELECT
    cl.nombre,
    cl.apellidopat,
    cl.apellidomat,
    COUNT(rp.id_promo) AS promociones_recibidas,
    COUNT(DISTINCT asist.id_evento) AS eventos_asistidos
FROM
    cliente cl
LEFT JOIN
    recibir_promo rp ON cl.curp = rp.curp
LEFT JOIN
    asistir asist ON cl.curp = asist.curp
GROUP BY
    cl.curp
HAVING
    COUNT(rp.id_promo) > 0 AND COUNT(asist.id_evento) = 0;
```

veterinario 6 | veterinario 7 | veterinario 3 | cliente 4 | cliente 5 | cliente 6 X

SELECT cl.nombre, cl.apellidopat, cl.apellidom: Enter a SQL expression to filter results (use Ctrl+Space)

	ABC nombre	ABC apellidopat	ABC apellidom	123 promociones_recibidas	123 eventos_asistidos
1	Bard	Beverley	Cable	2	0
2	Stillmann	Ritchley	Stovin	2	0
3	Lian	Valenti	MacGuigan	2	0
4	Ange	Spender	Smeuin	1	0
5	Lionello	Brimblecomb	Goadby	4	0
6	Chelsae	Ebsworth	Hopewell	2	0
7	Marijn	Attenburrow	Giovannazzi	4	0
8	Nancee	Maier	O'Hannay	2	0
9	Edi	Monelli	Engelbrecht	2	0
10	Shelley	Gudge	Treacy	7	0

Consulta 10: Listar los clientes según el número de notificaciones de eventos que han recibido lo que puede ser un indicador de su interés en las actividades del zoológico y una oportunidad para dirigir promociones futuras.

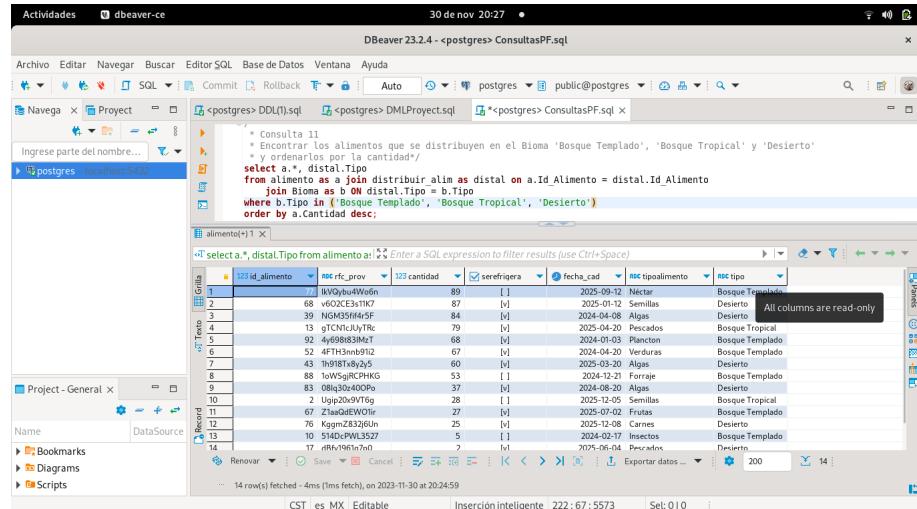
```
/*
 * Consulta 10
 * Listar los clientes según el número de notificaciones de eventos que han recibido
 * lo que puede ser un indicador de su interés en las actividades del zoológico
 * y una oportunidad para dirigir promociones futuras
*/
SELECT
    cl.nombre,
    cl.apellidopat,
    cl.apellidomat,
    COUNT(ne.id_noti_evento) AS cantidad_notificaciones_recibidas
FROM
    cliente cl
LEFT JOIN
    recibir_noti_even ne ON cl.curp = ne.curp
GROUP BY
    cl.curp
ORDER BY
    cantidad_notificaciones_recibidas DESC
LIMIT 15;
```

cliente 7 X

Enter a SQL expression to filter results (use Ctrl+Space)

	ABC nombre	ABC apellidopat	ABC apellidomat	123 cantidad_notificaciones_recibidas
1	Lelah	Kaesmans	HassekI	6
2	Rollins	Muat	Sole	5
3	Esra	Riteley	Rubrow	5
4	Haroun	Malloy	Prestedge	4
5	Natka	Cluer	Lidgard	4
6	Magnum	Buckles	Korneichik	4
7	Vanya	Robe	Vogeler	4
8	Cherrita	Cooper	Dudley	4
9	Ian	Ceci	Goldwater	4
10	Jesselyn	Ballefant	Graudoux	4
11	Boycey	Milburne	Doidge	4
12	Laurie	Feare	Alekseev	4
13	Julian	Newbold	Mannion	4
14	Vera	Quinnell	Rowan	3
15	Brian	Caherny	Gutowska	3

Consulta 11: Encontrar los alimentos que se distribuyen en el Bioma 'Bosque Templado', 'Bosque Tropical' y 'Desierto' y ordenarlos por la cantidad



```

30 de nov 20:27 •
DBeaver 23.2.4 - <postgres> ConsultasPF.sql

Archivo Editar Navegar Buscar Editor SQL Base de Datos Ventana Ayuda
SQL Rollback Auto postgres public@postgres ConsultasPF.sql
Navega Proyecto
Ingrese parte del nombre...
postgres localhost:5432
* Consulta 11
* Encontrar los alimentos que se distribuyen en el Bioma 'Bosque Templado', 'Bosque Tropical' y 'Desierto'
* y ordenarlos por la cantidad*
select a.*, distal.Tipo
from alimento a join distribuir alim as distal on a.Id_Alimento = distal.Id_Alimento
join Bioma b ON distal.Tipo = b.Tipo
where b.Tipo IN ('Bosque Templado', 'Bosque Tropical', 'Desierto')
order by a.Cantidad desc;

alimento(+1)
select a.*,
       distal.Tipo from alimento a
Enter a SQL expression to filter results (use Ctrl+Space)
1 id_alimento | kVOjAu4Wv6n | cantidad | servirpara | fecha_cad | tipoAlimento | tipo
2 68 v6OCxE3t1k7 89 | l | 2025-09-12 | Nector | Desierto
3 39 NGM35f64d5F 87 | v | 2025-01-12 | Semillas | Desierto
4 13 g1CNrJUjTR 84 | v | 2024-04-08 | Algas | Desierto
5 92 4y69883MzT 79 | v | 2025-04-20 | Pescados | Bosque Tropical
6 52 4FTH3n69t9L 68 | v | 2024-01-03 | Plancton | Bosque Templado
7 43 4qC9XcP9Z 60 | v | 2024-04-20 | Verduras | Bosque Templado
8 80 t0V5gRCPHKG 53 | l | 2025-01-12 | Desierto | Desierto
9 83 08i3Qb40OOrs 37 | v | 2024-12-21 | Fornale | Bosque Templado
10 2 Usqj20x9Vfg 28 | l | 2025-12-05 | Semillas | Bosque Tropical
11 67 Z1aaQeWEWOI 27 | v | 2025-07-02 | Frutas | Bosque Templado
12 76 Kggmz83jGJUn 25 | v | 2025-12-08 | Carnes | Desierto
13 10 514DcPW13527 5 | l | 2024-02-17 | Insectos | Bosque Templado
14 17 rRk-rHnLn7nD ? | v | 2025-06-04 | Pescados | Desierto
Record Renovar Save Cancel Exportar datos ... 200
14 row(s) fetched - 4ms (1ms fetch), on 2023-11-30 at 20:24:59
CST es_MX Editable Inserción inteligente 222:67:5573 Sel: 0 | 0

```

Consulta 12: Devolver el RFC de veterinarios que atienden a animales cuya altura es mayor a 50

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Actividades, dbeaver-ce, Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Connections:** postgres - localhost:5432 (selected).
- SQL Editor:** Contains the following SQL code:

```
/*
 * Consulta 12
 * Devolver el RFC de veterinarios que atienden a animales
 * cuya altura es mayor a 50
 */
select distinct v.RFC_Vet
from Veterinario as v join atender as aten on v.RFC_Vet = aten.RFC_Vet
join animal as a on aten.Id_Animal = a.Id_Animal
where a.altura > 50;
```
- Results Grid:** Shows a table titled "veterinario1" with one column "rfc_vet". The data is as follows:

rfc_vet
1 y72V6DTznhM+W
2 lbSSk5L577hw
3 4Qgl3Mdfrczc
4 2S670J010PUf
5 Gt6s1MmGawv0
6 C0B9yXqfjwOy
7 MaD5+1Ae8Nx7
8 aKD9W2RqipB8
9 A37g7PPYqUIN
10 Kc7XT3aF89WVT
11
- Bottom Status Bar:** CST | es_MX Editable, Inserción inteligente 225:1:5602, Set: 0 | 0

Consulta 13: Devolver el laboratorio de procedencia de todas las medicinas que se distribuyan en algun tipo de Bosque

The screenshot shows the DBeaver interface with the following details:

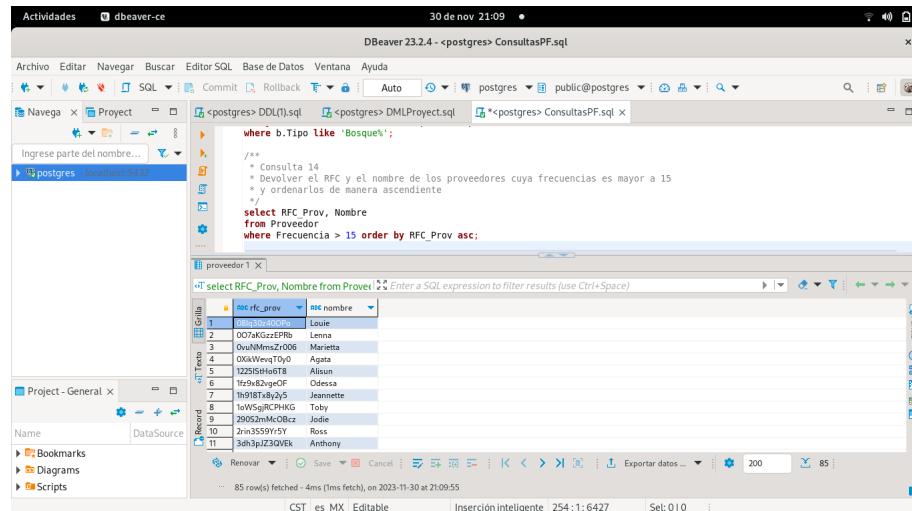
- Toolbar:** Actividades, dbeaver-ce, 30 de nov 21:08 •
- Menu Bar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Project Tab:** Navega, Proyecto, postgres - localhost:5432.
- SQL Editor:** <postgres> DDL(l).sql, <postgres> DMLProyect.sql, *<postgres> ConsultasPF.sql.
- Query Content:**

```
/** Consulta 13
 * Devolver el laboratorio de procedencia de todas las medicinas
 * que se distribuyan en algun tipo de Bosque
 */
select distinct m.labProcedencia
from medicina as m join distribuir_med as distmed on m.Lote = distmed.Lote
join bioma as b on distmed.Tipo = b.Tipo
where b.Tipo like 'Bosques';
```
- Result Grid:** medicina1 X | Enter a SQL expression to filter results (use Ctrl+Space)

	labProcedencia
1	Bergstrom and Sons
2	Bogisch, Cole and Rutherford
3	Brekke, West and Saweyn
4	Cartwright Inc
5	Crona-Hygen
6	Diamond, Brown and Lynch
7	Kedney and Sons
8	Lehner and Sons
9	Mayer-Robel
10	Mills, Price and McDermott
11	

- Bottom Status Bar:** 16 row(s) fetched - 3ms, on 2023-11-30 at 21:08:37, CST | es_MX Editable, Inserción inteligente 235:1:5883, Set: 0 | 0

Consulta 14: Devolver el RFC y el nombre de los proveedores cuya frecuencias es mayor a 15 y ordenarlos de manera ascendente



The screenshot shows the DBBeaver interface with a SQL query window open. The query is:

```

/*
 * Consulta 14
 * Devolver el RFC y el nombre de los proveedores cuya frecuencias es mayor a 15
 * y ordenarlos de manera ascendente
 */
select RFC_Prov, Nombre
from Proveedor
where Frecuencia > 15 order by RFC_Prov asc;

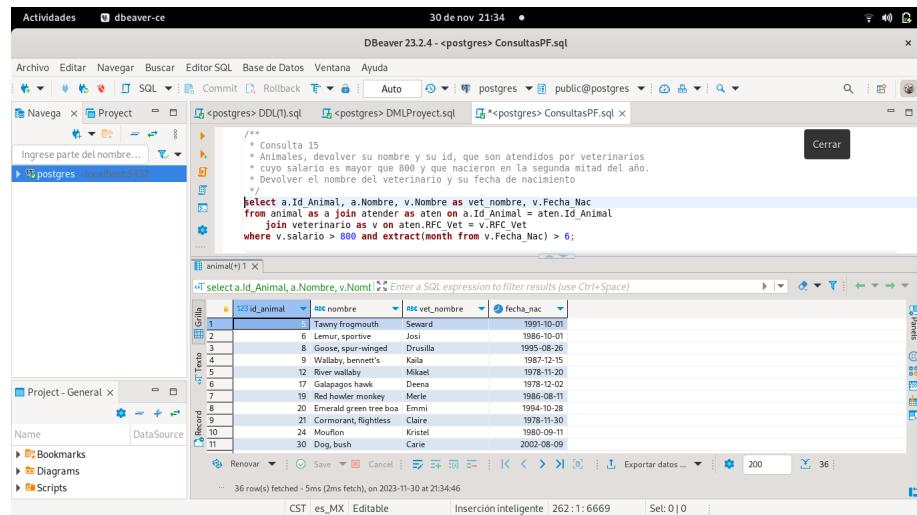
```

The results grid shows 11 rows of data:

	RFC_Prov	Nombre
1	0074KgzzEPfb	Lenna
2	OuuNhmmsZrD06	Marietta
3	OXAWeqvqToY0	Agata
4	122515HoGtB	Alixun
5	1fz9x82vgOF	Odessa
6	1wV5qHcPHIG	Jeanette
7	2905XjmMcOBzC2	Sab
8	2rn3S5O9y5V	Jodie
9	3eb3pJZ3QVfk	Ross
10		Anthony
11		

Below the grid, the message "85 row(s) fetched - 4ms (1ms fetch), on 2023-11-30 at 21:09:55" is displayed.

Consulta 15: Animales, devolver su nombre y su id, que son atendidos por veterinarios cuyo salario es mayor que 800 y que nacieron en la segunda mitad del año. Devolver el nombre del veterinario y su fecha de nacimiento



The screenshot shows a DBeaver interface with a SQL editor and a results grid.

```

/* Consulta 15
 * Animales, devolver su nombre y su id, que son atendidos por veterinarios
 * cuyo salario es mayor que 800 y que nacieron en la segunda mitad del año.
 * Devolver el nombre del veterinario y su fecha de nacimiento
 */
select a.id_Animal, a.Nombre, v.Nombre as vet_nombre, v.Fecha_Nac
from animal as a join atencion as aten on a.Id_Animal = aten.Id_Animal
join veterinario as v on aten.RFC_Vet = v.RFC_Vet
where v.salario > 800 and extract(month from v.Fecha_Nac) > 6;

```

The results grid displays the following data:

	id_animal	Nombre	vet_nombre	fecha_nac
1	Taxmy frogmouth	Seward		1993-10-01
2	Lemur, sportive	Josi		1986-10-01
3	Goose,spur-winged	Drusilla		1995-08-26
4	Wallaby, bennett's	Kala		1987-12-15
5	River wallaby	Mikael		1978-11-20
6	Galapagos hawk	Doris		1978-02-02
7	Red spider monkey	Merle		1994-08-11
8				
9	Emerald green tree boa	Emmi		1994-10-28
10	Cormorant, flightless	Claire		1978-11-30
11	Mouflon	Kristel		1980-09-11
12	Dog, bush	Carie		2002-08-09

36 row(s) fetched - Sims (2ms fetch), on 2023-11-30 at 21:34:46

6. Procedimientos almacenados

- **Procedimiento almacenado 1:** Permite agregar un nuevo animal:

```

dBeaver 23.1.5 - <postgres> TriggerYSP.sql
Archivo Editar Navegar Search Editor SQL Base de Datos Ventana Ayuda
Auto SQL Commit Rollback
<postgres> DDLsql <postgres> DMLsql <postgres> Trigg... <postgres> Query... <postgres> DDL_... <postgres> DML_P <postgres> Trig... <postgres> Script...
Navegador de Bases de Datos
Ingresé parte del nombre de un objeto aquí
DBBeaver Sample Database (SQLite)
postres - localhost:5432
649 SELECT * FROM cuidador WHERE Id_Animal = 51;
650 SELECT * FROM indicacionesMed WHERE Id_Animal = 51;
651 SELECT * FROM atender WHERE Id_Animal = 51;
652 /*
653
654--/SP QUE PERMITE AGREGAR UN NUEVO ANIMAL/
655 CREATE OR REPLACE PROCEDURE afadir_animal (
656     p_Id_Animal INT,
657     p_Tipo VARCHAR(50),
658     p_Especie VARCHAR(50),
659     p_Num_Jaula INT,
660     p_Nombre VARCHAR(50),
661     p_Alimentacion VARCHAR(50),
662     p_Sexo CHAR,
663     p_Altura FLOAT8,
664     p_Peso FLOAT8)
665 LANGUAGE plpgsql
666 AS $$*
667 BEGIN
668     INSERT INTO animal (Id_Animal, Tipo, Especie, Num_Jaula, Nombre, Alimentacion, Sexo, Altura, Peso)
669     VALUES (p_Id_Animal, p_Tipo, p_Especie, p_Num_Jaula, p_Nombre, p_Alimentacion, p_Sexo, p_Altura, p_Peso);
670 END;
671 $$;
672
673
674-- Probar procedimiento:
675 CALL afadir_animal(200, 'Desierto', 'Elefante', 10, 'Dumbo', 'Hierbas', 'M', 3.5, 6000);
676 SELECT * FROM animal WHERE Id_Animal = 200;

```

Estadísticas 1 animal 2 x

id_animal	tipo	especie	num_jaula	nombre	alimentacion	sexo	altura	peso
200	Desierto	Elefante	10	Dumbo	Hierbas	M	3.5	6.000

Record Texto

Refresh Save Cancel Exportar datos ... CSV es Writable Smart Insert 676:1 [43] Set:43 | 1 | 1 row(s) fetched - 7ms, on 2023-12-01 at 23:05:22

- **Procedimiento almacenado 2:** Elimina un animal y los registro en las tablas que contienen llave foránea correspondiente a la tupla, además solo se puede eliminar animales si su id es mayor a 50:

```

625--~/SP2, Elimina un animal y los registro en las tablas que contienen llave foranea correspondiente a la tupla, ademas solo se puede eliminar
626 CREATE OR REPLACE PROCEDURE delete_animal (p_Id_Animal INT)
627 LANGUAGE plpgsql
628 AS $$
629 BEGIN
630     IF p_Id_Animal <= 50 THEN
631         RAISE EXCEPTION 'El ID del animal debe ser mayor a 50. ID proporcionado: %', p_Id_Animal;
632     END IF;
633
634     -- Primero eliminar referencias en tablas con claves foraneas que apuntan a 'animal'
635     DELETE FROM atender WHERE Id_Animal = p_Id_Animal;
636     DELETE FROM indicacionesMed WHERE Id_Animal = p_Id_Animal;
637     DELETE FROM cuidador WHERE Id_Animal = p_Id_Animal;
638
639     --eliminar el animal de la tabla 'animal'
640     DELETE FROM animal WHERE Id_Animal = p_Id_Animal;
641 END;
642 $$;
643
644-- Probar procedimiento:
645 CALL delete_animal(51);
646
647/*
648 SELECT * FROM animal WHERE Id_Animal = 51;
649 SELECT * FROM cuidador WHERE Id_Animal = 51;
650 SELECT * FROM indicacionesMed WHERE Id_Animal = 51;
651 SELECT * FROM atender WHERE Id_Animal = 51;
652 */

```

The screenshot shows the DBeaver interface with the following details:

- SQL Editor:** Contains the stored procedure code.
- Results Tab:** Shows the output of the executed procedure. It lists four queries:
 - SELECT * FROM animal WHERE Id_Animal = 51;
 - SELECT * FROM cuidador WHERE Id_Animal = 51;
 - SELECT * FROM indicacionesMed WHERE Id_Animal = 51;
 - SELECT * FROM atender WHERE Id_Animal = 51;
 All four queries return empty result sets.
- Table View:** A grid view of the 'animal' table with columns: id_animal, tipo, especie, num_jaula, nombre, alimentacion, sexo, altura, peso. The table is currently empty.

- **Procedimiento almacenado 3:** Muestra el RFC de los veterinarios, que tienen un salario mayor a \$15,000,00, que trabajan en un bioma y que tienen cierta especialidad:

The screenshot shows the DBeaver 23.1.5 interface with the following details:

- Toolbar:** Archivo, Editar, Navegar, Search, Editor SQL, Base de Datos, Ventana, Ayuda.
- Left Panel:** Navegador de Bases de Datos (Database Navigator) showing the database structure.
- Central Area:** Editor SQL window containing the following PostgreSQL code:

```

601 $$
602 declare
603   RFC_V VARCHAR(13);
604 begin
605
606   raise notice 'Lista de RFC de los vete';
607
608   for RFC_V in select * from ((select r.*;
609
610   loop
611     raise notice '- %', RFC_V;
612   end loop;
613
614   if RFC_V not in (select * from ((select r.*;
615     raise notice '-> No hay veterinario';
616
617   end;
618
619   $$;
620
621-- Ejecutar procedimiento:
622 call lista_veterinarios('Tundra', 'Especialidad en Animales Exóticos');
623 --call lista_veterinarios('Chaparral', 'Espec');
624
625--/SP2, Elimina un animal y los registro
626 CREATE OR REPLACE PROCEDURE delete_animal
627 LANGUAGE plpgsql
628 AS $$
```
- Right Panel:** Salida (Output) window showing the result of the query:

```

Lista de RFC de los veterinarios que tienen un salario mayor a $15,000.00, trabajan en el biom
- 0Gb3Sp6iR07X
```
- Bottom Status Bar:** Includes buttons for CST, es, Writable, Smart Insert, and statistics (622:1 [71], Sel: 71 | 1).

- **Procedimiento almacenado 4:** Muestra una lista de artículos de un bioma, que tengan mayor del 50% de descuento:

```

--select * from asistir;
572
573 -- Procedimiento almacenado que muestra una lista de articulos de un tipo de bioma
574 create or replace procedure lista_articulos(tipo_B
575 as
576 $$
577 declare
578   id_Art INT;
579 begin
580
581   raise notice 'Lista de articulos que tienen mayor a 50 de descuento en el tipo de bioma Selva';
582   for id_Art in select id_articulo from ((select
583     loop
584     raise notice '- %', id_Art;
585   end loop;
586 end;
587 end;
588 $$;
589 language plpgsql;
590
591 -- Ejecutar procedimiento:
592 call lista_articulos('Selva');
593 call lista_articulos('Bosque Templado');
594 call lista_articulos('Bosque Tropical');
595 --call lista_articulos('Desierto');
596 --call lista_articulos('Fastígliales');
597
598 -- Procedimiento almacenado que muestre el RFC de los articulos
599

```

Salida

```

Enter a part of a message to search for here
Lista de articulos que tienen mayor a 50 de descuento en el tipo de bioma Selva :
- 24
Lista de articulos que tienen mayor a 50 de descuento en el tipo de bioma Bosque Temp :
- 39
Lista de articulos que tienen mayor a 50 de descuento en el tipo de bioma Bosque Trop :
- 45

```

Estadísticas

Name	Value
Updated Rows	0
Query	call lista_articulos('Bosque Templado'); call lista_articulos('Bosque Tropical');
Start time	Fri Dec 01 22:46:32 CST 2023
Finish time	Fri Dec 01 22:46:32 CST 2023

7. Diccionario

- Alimento
 - Id_Alimento: Identificador del alimento
 - RFC_Prov: RFC del proveedor
 - Cantidad: Cantidad que hay del alimento
 - seRefrigera: Si se refrigerara el alimento
 - Fecha_Cad: Fecha de caducidad del alimento
 - tipoAlimento: Tipo de alimento
- Animal
 - Id_Animal: Identificador único para el animal
 - Tipo: Tipo de bioma del animal
 - Especie: Especie del animal
 - Num_Jaula: Número de la jaula del animal
 - Nombre: Nombre del animal
 - Alimentacion: Descripción de la alimentación del animal
 - Sexo: Género del animal: H para hombre, F para mujer
 - Altura: Altura del animal en metros
 - Peso: Peso del animal en kilogramos
- Articulo
 - Id_Articulo: Identificador del articulo
 - RFC_Prov: RFC del proveedor
 - Cantidad: Cantidad que hay del articulo
- asistir
 - Id_Evento: Identificador del evento
 - CURP: CURP del cliente
- asociar
 - Id_Articulo: Identificador del articulo
 - Num_Ticket_Art: Numero de ticket
- atender
 - RFC_Vet: RFC del veterinario
 - Id_Animal: Identificador del animal

- Bioma
 - Tipo: Tipo de bioma
- Cliente
 - CURP: CURP de el cliente
 - Nombre: Nombre de el cliente
 - ApellidoPat: Apellido paterno de el cliente
 - ApellidoMat: Apellido materno de el cliente
 - Genero: Género de la persona: H para hombre, F para mujer
- correoClien
 - CURP: CURP de el cliente
 - Correo_Clien: Correo de el cliente
- correoCuid
 - RFC_Cuid: RFC de el cuidador
 - Correo_Cuid: Correo de el cuidador
- correoProv
 - RFC_Prov: RFC de el proveedor
 - Correo_Prov: Correo de el proveedor
- correoVet
 - RFC_Vet: RFC de el veterinario
 - Correo_Vet: Correo de el veterinario
- Cuidador
 - RFC_Cuid: RFC de el cuidador
 - Id_Animal: Id del animal que cuida el cuidador
 - Tipo: Tipo de bioma donde trabaja el cuidador
 - Nombre: Nombre de el cuidador
 - ApellidoPat: Apellido paterno de el cuidador
 - ApellidoMat: Apellido materno de el cuidador
 - Calle: Calle donde reside el cuidador
 - Colonia: Colonia o barrio donde reside el cuidador
 - Num_Ext: Número exterior de la vivienda

- Num_Int: Número interior de la vivienda
 - Estado: Estado o provincia donde reside el cuidador
 - Fecha_Nac: Fecha de nacimiento de el cuidador
 - FechaIn_Con: Fecha de inicio de contrato (si aplica)
 - FechaFin_Con: Fecha de finalización de contrato (si aplica)
 - Hora_Entrada: Hora de entrada del cuidador
 - Hora_Salida: Hora de salida del cuidador
 - Dias_Trab: Dias que trabaja el cuidador
 - Salario: Salario del cuidador
- Descuento
 - Id_Desc: Identificador del descuento
 - Id_Articulo: Identificador del articulo
 - Porcentaje_Desc: Procentaje del descuento
 - Fecha_Inicio: Fecha de inicio del descuento
 - Fecha_Fin: Fecha de fin del descuento
 - distribuir_Alim
 - Id_Alimento: Identificador del alimento
 - Tipo: Tipo de Bioma
 - distribuir_Art
 - Id_Articulo: Identificador del articulo
 - Tipo: Tipo de Bioma
 - distribuir_Med
 - Lote: Identificador de la medicina
 - Tipo: Tipo de Bioma
 - distribuir_Serv
 - Id_Servicio: Identificador del servicio
 - Tipo: Tipo de Bioma
 - Especialidad
 - Especialidad: Especialidad de el veterinario
 - RFC_Vet: RFC de el veterinario
 - Evento

- Id_Evento: Id del evento
 - Capacidad: La capacidad que tendra el evento
 - tipoEvento: El tipo de evento
 - Fecha: La fecha del evento
- indicacionesMed
 - Id_Animal: Id del animal
 - IndicacionesMed: Indicaciones medicas del animal
- Medicina
 - Lote: Identificador de la medicina
 - RFC_Prov: RFC del proveedor
 - Cantidad: Cantidad que hay de la medicina
 - seRefrigera: Si se refrigerara la medicina
 - Fecha_Cad: Fecha de caducidad de la medicina
 - labProcedencia: Tipo de medicina
- noti_Evento
 - Id_Noti_Evento: Identificador de la notificacion del evento
 - Id_Evento: Identificador del evento sobre el que se notifica
- Promocion
 - Id_Promo: Identificador de la promocion
 - Id_Articulo: Identificador del articulo
 - Tipo_Promo: Tipo de la promocion
 - Fecha_Inicio: Fecha de inicio de la promocion
 - Fecha_Fin: Fecha de fin de la promocion
- Proveedor
 - RFC_Prov: RFC de el proveedor
 - Nombre: Nombre de el proveedor
 - ApellidoPat: Apellido paterno de el proveedor
 - ApellidoMat: Apellido materno de el proveedor
 - Calle: Calle donde reside el proveedor
 - Colonia: Colonia o barrio donde reside el proveedor
 - Num_Ext: Número exterior de la vivienda
 - Num_Int: Número interior de la vivienda

- Estado: Estado o provincia donde reside el proveedor
 - Fecha_Nac: Fecha de nacimiento de el proveedor
 - FechaIn_Con: Fecha de inicio de contrato (si aplica)
 - FechaFin_Con: Fecha de finalización de contrato (si aplica)
 - Costo_Serv: Costo del servicio del proveedor
 - Producto: Producto que ofrece el proveedor
 - Frecuencia: Frecuencia con la que el proveedor realiza
- recibir_Desc
 - CURP: CURP del cliente
 - Id_Desc: Identificador del descuento
 - recibir_Noti_Even
 - CURP: CURP del cliente
 - Id_Noti_Evento: Identificador de la notificación de un evento
 - recibir_Promo
 - CURP: CURP del cliente
 - Id_Promo: Identificador de la promoción
 - Servicio
 - Id_Servicio: Identificador del servicio
 - RFC_Prov: RFC del proveedor
 - TelClien
 - CURP: CURP de el cliente
 - Tel_Clien: Telefono de el cliente
 - TelCuid
 - RFC_Cuid: RFC de el cuidador
 - Tel_Cuid: Telefono de el cuidador
 - TelProv
 - RFC_Prov: RFC de el proveedor
 - Tel_Prov: Telefono de el proveedor
 - TelVet
 - RFC_Vet: RFC de el veterinario

- Tel_Vet: Telefono de el veterinario
- ticket_Art
 - Num_Ticket_Art: Número único para el ticket
 - CURP: CURP del cliente
 - Cantidad: Cantidad del ticket
 - Fecha: Fecha de emisión del ticket
- ticket_Serv
 - Num_Ticket_Serv: Número único para el ticket
 - CURP: CURP del cliente
 - Id_Servicio: Identificador del servicio
 - Descuento: Descuento del ticket
 - Cantidad: Cantidad del ticket
 - Fecha: Fecha de emisión del ticket
- trabajar
 - RFC_Vet: RFC del veterinario
 - Tipo: Tipo de bioma
- Veterinario
 - RFC_Vet: RFC de el veterinario
 - Nombre: Nombre de el veterinario
 - ApellidoPat: Apellido paterno de el veterinario
 - ApellidoMat: Apellido materno de el veterinario
 - Calle: Calle donde reside el veterinario
 - Colonia: Colonia o barrio donde reside el veterinario
 - Num_Ext: Número exterior de la vivienda
 - Num_Int: Número interior de la vivienda
 - Estado: Estado o provincia donde reside el veterinario
 - Fecha_Nac: Fecha de nacimiento de el veterinario
 - FechaIn_Con: Fecha de inicio de contrato (si aplica)
 - FechaFin_Con: Fecha de finalización de contrato (si aplica)
 - Salario: Salario del veterinario

8. Conclusión

Este proyecto demostró cómo la digitalización y la implementación de una base de datos integral pueden revolucionar la gestión del Zoológico de Huitziltepec. A través de la digitalización de registros, la gestión eficiente de recursos y la optimización de procesos internos, el zoológico está ahora mejor equipado para enfrentar desafíos modernos. Los resultados sugieren mejoras significativas en la administración de personal, el cuidado animal, y la experiencia del visitante. A futuro, recomendamos continuar la expansión y actualización de la base de datos para adaptarse a las necesidades cambiantes y aprovechar las nuevas tecnologías, asegurando así una gestión sostenible y eficiente del zoológico.

A su vez, mediante la realización del proyecto aprendimos a enfrentar los distintos tipos de problemas que implica diseñar una base de datos, junto con los problemas que surgieron durante el proceso de implementación, haciendo que al finalizar el proyecto, después de buscar soluciones a los múltiples errores o desacuerdos que surgieron, hayamos aprendido mucho acerca del proceso de construir y probar el funcionamiento de una base de datos a partir de un caso de uso. Además de que aprendimos herramientas fundamentales para el proceso de probar o hacer que todo funcione correctamente como consultas, disparadores o procedimientos almacenados, así que no solo aprendimos diseñar la estructura de la base de datos, sino que también aprendimos a manipular y probar que todo lo que hayamos hecho este bien, poblando la base de datos y/o haciendo consultas, procedimientos, etc. Aprendimos a como solucionar las problemáticas que pueden surgir en el proceso.