

CONSULTAS BÁSICAS

1. Mostrar todos los clientes en la tabla "Cliente".

Código SQL:

```
MostrarClientes.sql
SELECT *
FROM CLIENTEPLSQL;
```

Resultados:

Resultado de la Consulta x

Se han recuperado 100 filas en 0.359 segundos

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	625 Ricky Owens	94028 Mark Centers, Port Alishaland, New Hampshire 72598	3596036549
2	626 Brenda Cervantes	075 Brown Green Suite 198, Port Matthewberg, Utah 95139	483.526.6042
3	627 Gina Morgan	9409 Alvarado Ramp, Grossbury, Kentucky 37569	581-453-1641
4	628 Carl Williams	24381 Jennifer Mountain Suite 361, Danielleshire, South Dakota 16856	(362)765-7798
5	629 Jerry Malone	981 Fletcher Meadow Apt. 108, East Ashleyberg, Rhode Island 72974	817.439.0622
6	631 Jennifer Brown	8353 Holly Prairie, Port Lawrence, Louisiana 95325	546.345.4397
7	634 Summer Wilson	64559 Russell Fort Apt. 762, South Anthonystad, Oklahoma 45519	526.637.2457
8	636 Gary Griffith	94029 Mark Centers, Port Alishaland, New Hampshire 72598	560-893-7332
9	643 Tyler Avila	873 Richard Harbors Suite 274, Port April, Virginia 82642	881-666-9235
10	644 Audrey Ali	15333 Margaret Radial Apt. 908, Duncanside, Wisconsin 11886	415-304-1601
11	648 Michelle Adams	06816 Stephen Falls Suite 058, North Wanda, New Mexico 62934	3783891153
12	651 Jesse Anderson	1862 Dixon Stravenue Apt. 399, Lewisberg, Arizona 69946	(590)440-0974
13	654 Justin Hamilton	3750 Brown Harbor, Washingtonmouth, Washington 93062	+1-287-434-2337

Línea 2 Columna 19 | Insertar | Windows: C

Explicación:

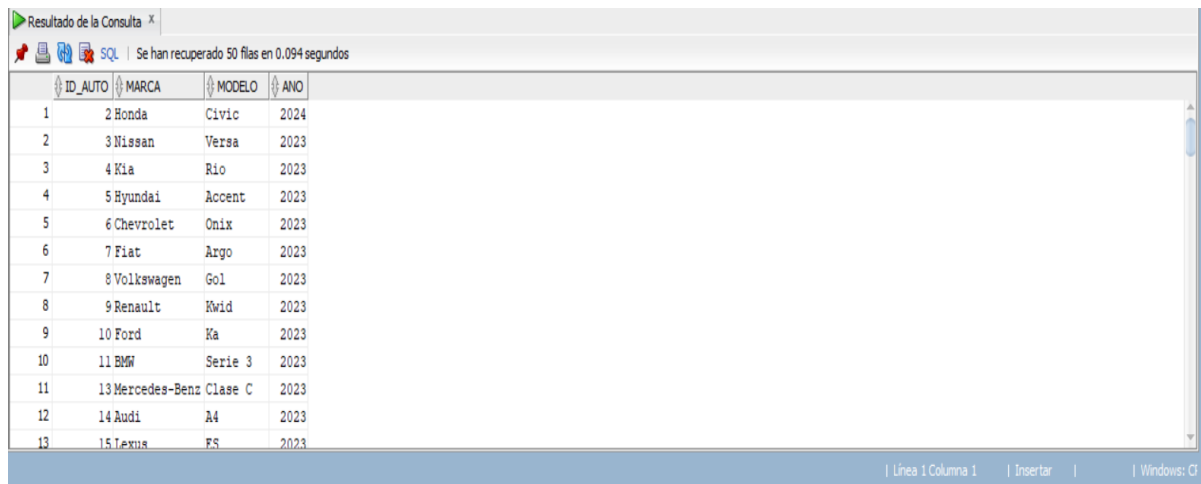
El código utiliza SELECT* para seleccionar todas las columnas de la tabla CLIENTEPLSQL y mostrar todos los clientes, FROM especifica de que tabla va a ser extraída la información.

2. Mostrar todos los autos en la tabla "Auto".

Código SQL:

```
MostrarAutos.sql
SELECT *
FROM AutoPLSQL;
```

Resultados:



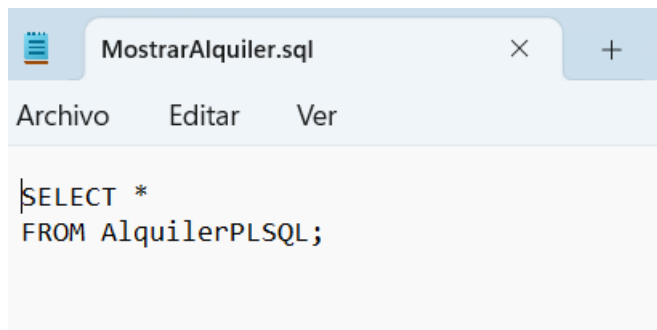
ID_AUTO	MARCA	MODELO	AÑO	
1	2 Honda	Civic	2024	
2	3 Nissan	Versa	2023	
3	4 Kia	Rio	2023	
4	5 Hyundai	Accent	2023	
5	6 Chevrolet	Onix	2023	
6	7 Fiat	Argo	2023	
7	8 Volkswagen	Gol	2023	
8	9 Renault	Kwid	2023	
9	10 Ford	Ka	2023	
10	11 BMW	Serie 3	2023	
11	13 Mercedes-Benz	Clase C	2023	
12	14 Audi	A4	2023	
13	15 Lexus	ES	2023	

Explicación:

El código utiliza `SELECT*` para seleccionar todas las columnas de la tabla `AutoPLSQL` y mostrar todos los autos, `FROM` especifica de que tabla va a ser extraída la información.

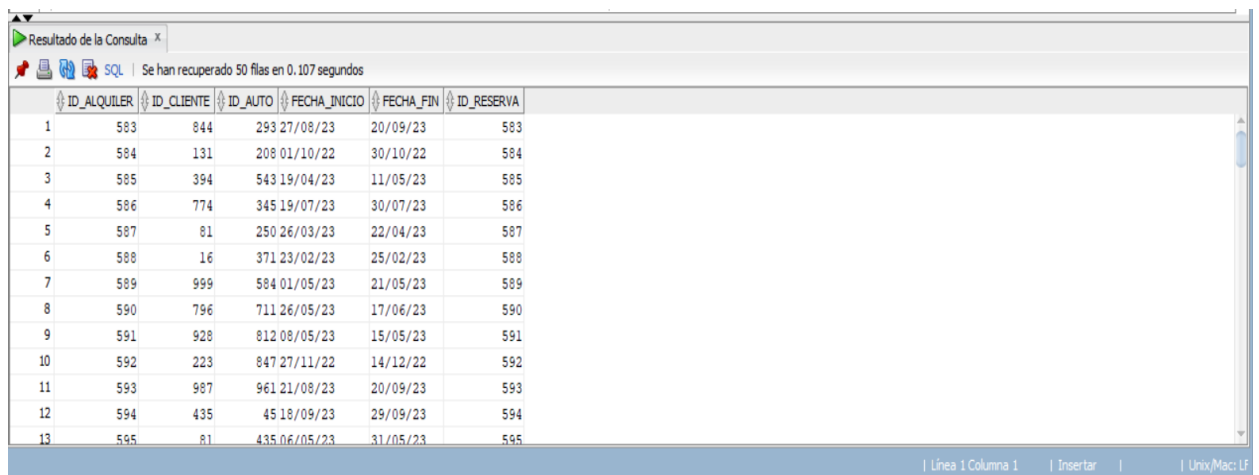
3. Mostrar todos los alquileres en la tabla "Alquiler".

Código SQL:



```
SELECT *
FROM AlquilerPLSQL;
```

Resultados:



ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	583	844	29/27/08/23	20/09/23	583
2	584	131	208 01/10/22	30/10/22	584
3	585	394	543 19/04/23	11/05/23	585
4	586	774	345 19/07/23	30/07/23	586
5	587	81	250 26/03/23	22/04/23	587
6	588	16	371 23/02/23	25/02/23	588
7	589	999	584 01/05/23	21/05/23	589
8	590	796	711 26/05/23	17/06/23	590
9	591	928	812 08/05/23	15/05/23	591
10	592	223	847 27/11/22	14/12/22	592
11	593	987	961 21/08/23	20/09/23	593
12	594	435	45 18/09/23	29/09/23	594
13	595	81	435 06/05/23	31/05/23	595

Explicación:

El código utiliza SELECT* para seleccionar todas las columnas de la tabla AlquilerPLSQL y mostrar todos los alquileres, FROM especifica de que tabla va a ser extraída la información.

4. Mostrar todas las sucursales en la tabla "Sucursal".

Código SQL:

```
MostrarAlquiler.sql
```

```
SELECT *
FROM AlquilerPLSQL;
```

Resultados:

Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0.133 segundos

ID_SUCURSAL	NOMBRE	CIUDAD	PAIS
1	584 Smith Ltd	Gardnerstad	Mozambique
2	585 Brown, Bridges and Rice	Jasonfort	Romania
3	586 Thompson-Porter	Lake Kurtton	Saint Lucia
4	587 Perez, Farley and Bartlett	Graystad	Liberia
5	588 Schultz LLC	Lake Ryan	Macao
6	589 Morton-Wilson	Haasside	Senegal
7	590 Howell Group	Greenburgh	Uzbekistan
8	591 Clements PLC	West Jasonhaven	Kenya
9	592 Welch-Webster	North Diana	United States Virgin Islands
10	593 Myers, Smith and Gross	Ashleyville	Cuba
11	594 Dorsey, Jones and Love	Port Emilyburgh	Macao
12	595 Jordan, Johnson and White	North Lisa	Cambodia
13	596 Todd and Sons	New Nathanville	Gambia

Explicación:

El código utiliza SELECT* para seleccionar todas las columnas de la tabla SucursalPLSQL y mostrar todas las sucursales, FROM especifica de que tabla va a ser extraída la información.

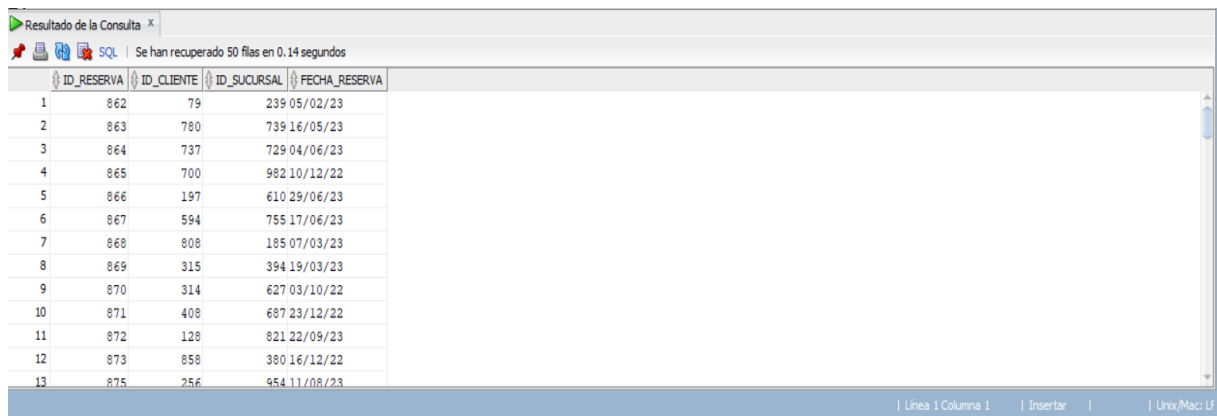
5. Mostrar todas las reservas en la tabla "Reserva".

Código SQL:

```
MostrarReserva.sql
```

```
SELECT *
FROM ReservaPLSQL;
```

Resultados:



ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	862	79	239 05/02/23
2	863	780	739 16/05/23
3	864	737	729 04/06/23
4	865	700	982 10/12/22
5	866	197	610 29/06/23
6	867	594	755 17/06/23
7	868	808	185 07/03/23
8	869	315	394 19/03/23
9	870	314	627 03/10/22
10	871	408	687 23/12/22
11	872	128	821 22/09/23
12	873	858	380 16/12/22
13	875	256	954 11/08/23

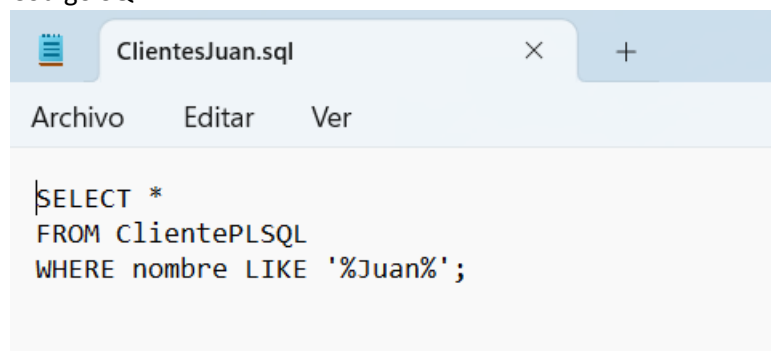
Explicación:

El código utiliza SELECT* para seleccionar todas las columnas de la tabla ReservaPLSQL y mostrar todas las reservas, FROM especifica de que tabla va a ser extraída la información.

FILTROS Y ORDENAMIENTO

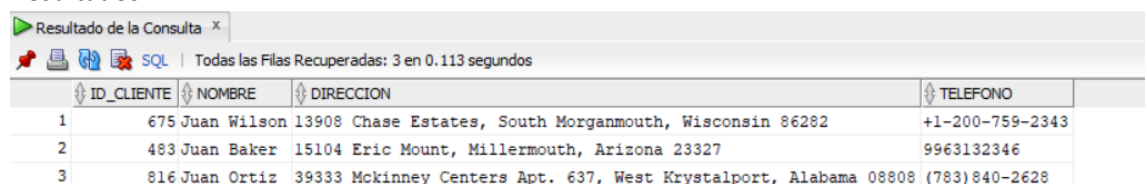
6. Mostrar los clientes que se llaman "Juan".

Código SQL:



```
SELECT *
FROM clientePLSQL
WHERE nombre LIKE '%Juan%';
```

Resultados:



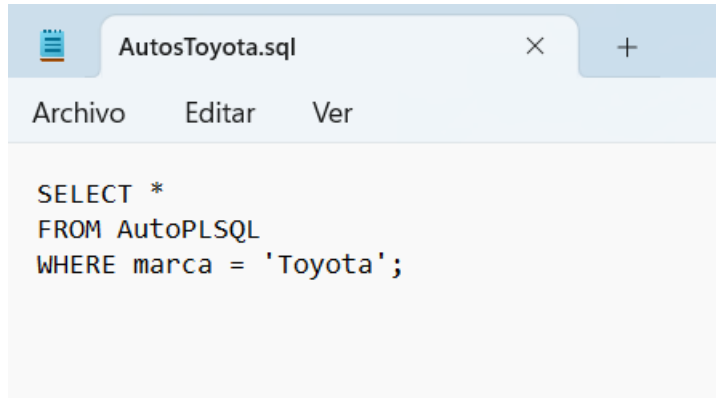
ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	675 Juan Wilson	13908 Chase Estates, South Morganmouth, Wisconsin 86282	+1-200-759-2343
2	483 Juan Baker	15104 Eric Mount, Millermouth, Arizona 23327	9963132346
3	816 Juan Ortiz	39333 Mckinney Centers Apt. 637, West Krystalport, Alabama 08808	(783)840-2628

Explicación:

En esta consulta se utiliza la clausula WHERE que filtra los resultados de la consulta, usando la clausula LIKE para buscar los clientes cuyo nombre contenga la cadena 'Juan'. El operador % representa cualquier número de caracteres, en este caso se usa para buscar cadenas que contengan la cadena 'Juan'.


7. Mostrar los autos de marca "Toyota".

Código SQL:



```
SELECT *  
FROM AutoPLSQL  
WHERE marca = 'Toyota';
```

Resultados:



Resultado de la Consulta

Todas las Filas Recuperadas: 0 en 0.108 segundos

ID_AUTO	MARCA	MODELO	AÑO
---------	-------	--------	-----

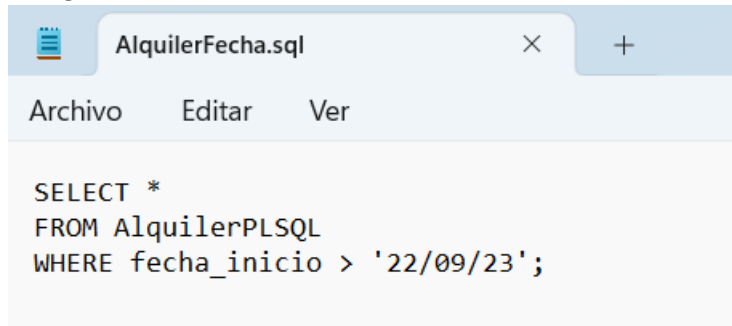
Explicación:

En este caso utilizamos la clausula WHERE marca = 'Toyota' para filtrar los resultados de la consulta, indicando que solo se devolverán los autos de la marca Toyota.

No obtuvimos resultados debido a que la tabla autos no contiene autos de marca Toyota.

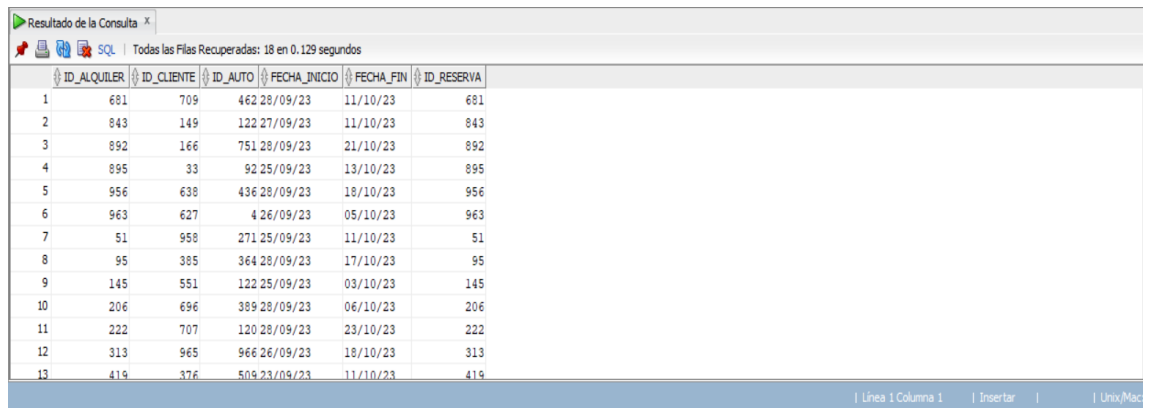
8. Mostrar los alquileres que ocurrieron después de una fecha específica.

Código SQL:



```
SELECT *  
FROM AlquilerPLSQL  
WHERE fecha_inicio > '22/09/23';
```

Resultados:



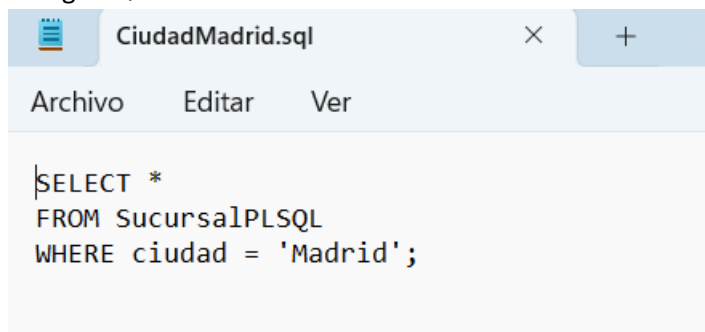
ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	681	709	462 28/09/23	11/10/23	681
2	843	149	122 27/09/23	11/10/23	843
3	892	166	751 28/09/23	21/10/23	892
4	895	33	92 25/09/23	13/10/23	895
5	956	638	436 28/09/23	18/10/23	956
6	963	627	4 26/09/23	05/10/23	963
7	51	958	271 25/09/23	11/10/23	51
8	95	385	364 28/09/23	17/10/23	95
9	145	551	122 25/09/23	03/10/23	145
10	206	696	389 28/09/23	06/10/23	206
11	222	707	120 28/09/23	23/10/23	222
12	313	965	966 26/09/23	18/10/23	313
13	419	376	509 23/09/23	11/10/23	419

Explicación:

Se utiliza la cláusula WHERE para filtrar la fecha_inicio de los alquileres que ocurrieron en una fecha mayor al 22 de septiembre de 2023.

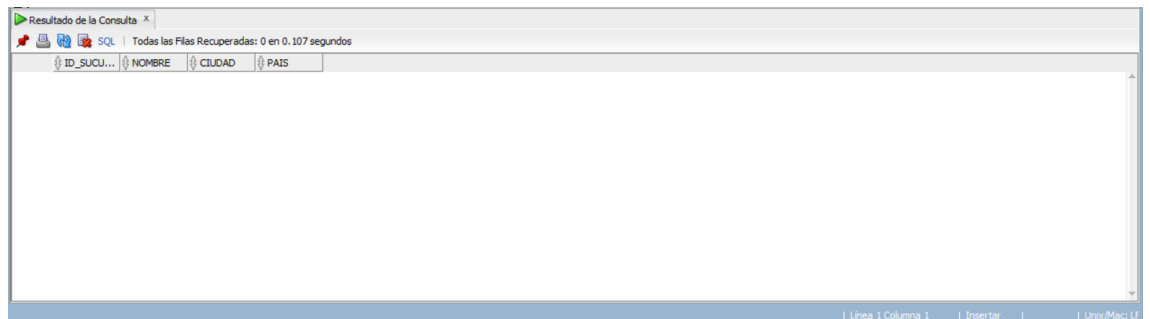
9. Mostrar las sucursales ubicadas en "Madrid".

Código SQL:



```
SELECT *
FROM SucursalPLSQL
WHERE ciudad = 'Madrid';
```

Resultados:



ID_SUCU...	NOMBRE	CIUDAD	PAIS
------------	--------	--------	------

Explicación:

Se utiliza la cláusula WHERE ciudad = 'Madrid' para filtrar las sucursales que se encuentren ubicadas en la ciudad de Madrid de la tabla SucursalPLSQL. Sin embargo la consulta no obtiene resultados ya que la tabla no contiene sucursales ubicadas en Madrid.

10. Mostrar las reservas realizadas por un cliente específico.

Código SQL:

```
ReservaCliente.sql
```

```
SELECT *
FROM ReservaPLSQL
WHERE id_cliente = 700;
```

Resultados:

Resultado de la Consulta x

Todas las Filas Recuperadas: 4 en 0.126 segundos

	ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	865	700	982	10/12/22
2	469	700	44	25/01/23
3	3604	700	100	25/03/23
4	469	700	44	25/01/23

Explicación:

En este caso utiliza la cláusula WHERE id_cliente = 700, para filtrar las reservas que se han realizado solo por el cliente con ese número de ID.

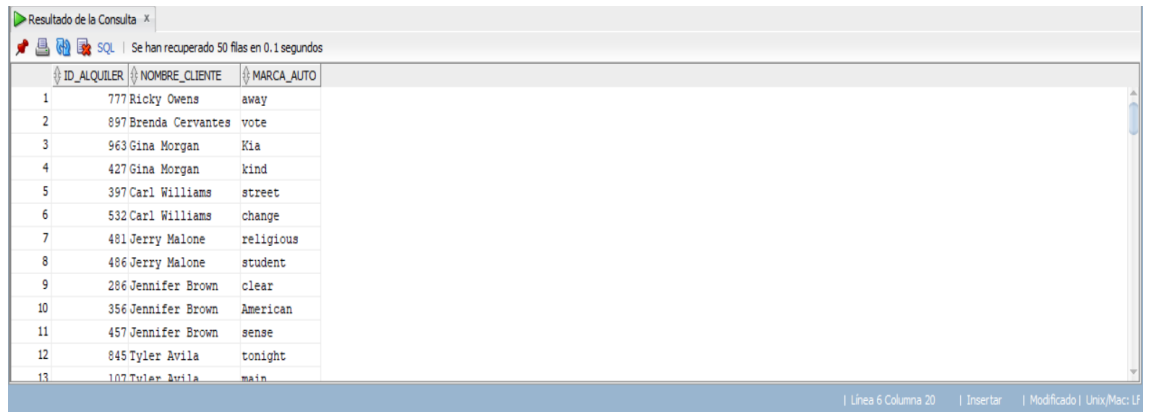
JOIN Y RELACIONES

11. Mostrar los alquileres con los nombres de los clientes y las marcas de los autos.

Código SQL:

```
SELECT
    A.id_alquiler,
    C.nombre AS nombre_cliente,
    Au.marca AS marca_auto
FROM
    AlquilerPLSQL A
JOIN
    ClientePLSQL C ON A.id_cliente = C.id_cliente
JOIN
    AutoPLSQL Au ON A.id_auto = Au.id_auto;
```

Resultados:



Resultado de la Consulta x

Se han recuperado 50 filas en 0.1 segundos

ID_ALQUILER	NOMBRE_CLIENTE	MARCA_AUTO
1	777 Ricky Owens	away
2	897 Brenda Cervantes	vote
3	963 Gina Morgan	Kia
4	427 Gina Morgan	kind
5	397 Carl Williams	street
6	532 Carl Williams	change
7	481 Jerry Malone	religious
8	486 Jerry Malone	student
9	286 Jennifer Brown	clear
10	356 Jennifer Brown	American
11	457 Jennifer Brown	sense
12	845 Tyler Avila	tonight
13	107 Tyler Avila	main

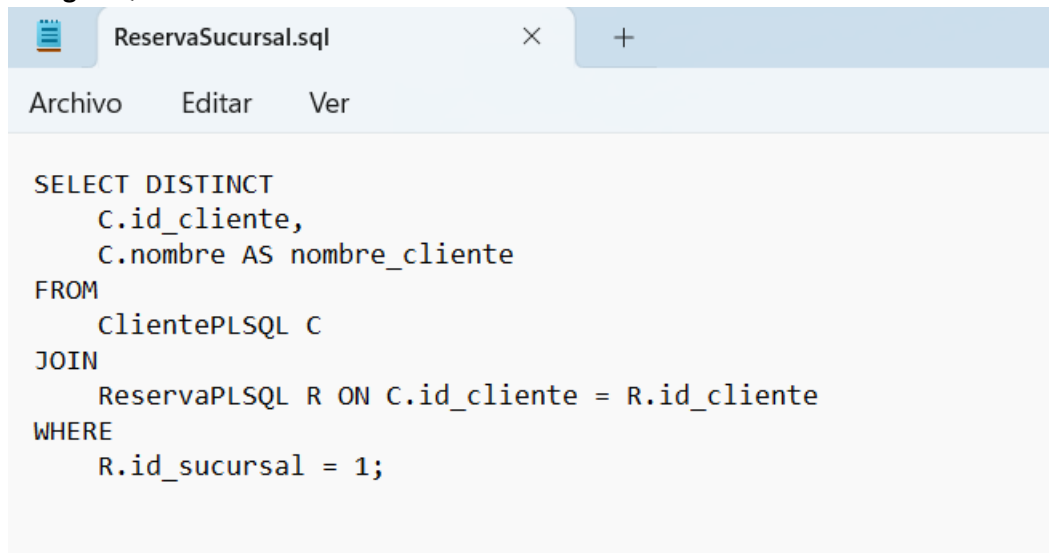
Linea 6 Columna 20 | Insertar | Modificado | Unix/Mac: LF

Explicación:

En este caso inicialmente utiliza la clausula FROM para seleccionar las columnas que se desea mostrar en el resultado de id_cliente, las cuales son nombre_cliente y marca_auto, se especifica la tabla en la cual estarán involucradas las consultas utilizando FROM luego se hace una unión con JOIN de la tabla AlquilerPLSQL con la tabla ClientePLSQL utilizando la columna id_cliente como clave de unión para obtener el nombre del cliente y finalmente se realiza una unión de la tabla AlquilerPLSQL con la tabla AutoPLSQL utilizando la columna id_auto como clave de unión para obtener la marca del auto. El resultado que se obtiene es una lista de alquileres con los nombres de los clientes y las marcas de los autos asociados a cada alquiler.

12. Mostrar los clientes que han realizado reservas en una sucursal específica.

Código SQL:



```
SELECT DISTINCT
    C.id_cliente,
    C.nombre AS nombre_cliente
FROM
    ClientePLSQL C
JOIN
    ReservaPLSQL R ON C.id_cliente = R.id_cliente
WHERE
    R.id_sucursal = 1;
```

Resultados:

Resultado de la Consulta x

Todas las Filas Recuperadas: 23 en 0.105 segundos

ID_CLIENTE	NOMBRE_CLIENTE
699	Kevin Russo
904	Michael Wyatt
556	Mary Vega
802	Rhonda Evans
47	Matthew Dean
150	Logan Johnson
491	Mary Marshall
774	Tara Johnson
625	Ricky Owens
399	Benjamin Perez
440	Sean Rodriguez
516	Tiffany Daniels
809	Raymond Woods

Línea 1 Columna 1 Insertar Unix/Mac

Explicación:

Seleccionamos las columnas que se quieren mostrar, que son id_cliente y nombre_cliente de la tabla ClientePLSQL. Luego realizamos una unión JOIN de la tabla ClientePLSQL con la tabla ReservaPLSQL utilizando la columna id_cliente como llave de unión para relacionar los clientes con sus reservas, después agregamos una cláusula WHERE para filtrar las filas donde el id_sucursal de la reserva sea igual a la sucursal específica que estamos buscando. Utilizamos DISTINCT para asegurarnos de que cada cliente aparezca solo una vez en el resultado, incluso si ha realizado múltiples reservas en la misma sucursal. Esta consulta da como resultado una lista de clientes que han realizado reservas en una sucursal específica.

13. Mostrar los autos que han sido alquilados junto con los nombres de los clientes.

Código SQL:

```

SELECT
    A.id_alquiler,
    C.nombre AS nombre_cliente,
    Au.marca,
    Au.modelo
FROM
    AlquilerPLSQL A
JOIN
    ClientePLSQL C ON A.id_cliente = C.id_cliente
JOIN
    AutoPLSQL Au ON A.id_auto = Au.id_auto;

```

Resultados:

Salida de Script x Resultado de la Consulta x

Se han recuperado 50 filas en 0.097 segundos

ID_ALQUILER	NOMBRE_CLIENTE	MARCA	MODELO
777	Ricky Owens	away	strategy
897	Brenda Cervantes	vote	interesting
963	Gina Morgan	Kia	Rio
427	Gina Morgan	kind	ball
397	Carl Williams	street	sport
532	Carl Williams	change	value
481	Jerry Malone	religious	generation
486	Jerry Malone	student	yard
286	Jennifer Brown	clear	reality
356	Jennifer Brown	American	Mr
457	Jennifer Brown	sense	major
845	Tyler Avila	tonight	budget
107	Tyler Avila	main	hir

Línea 5 Columna 14 Insertar Modificado Unix/Mac

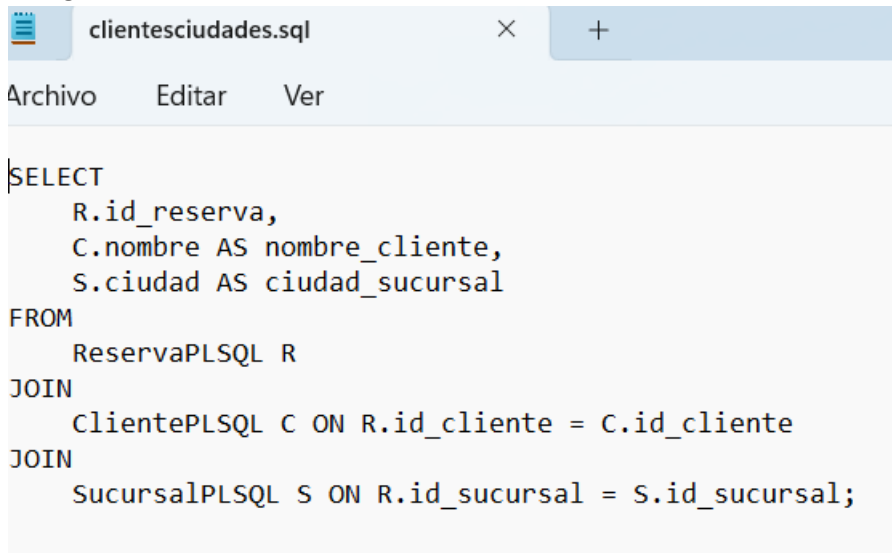
Explicación:

La consulta combina las tablas AlquilerPLSQL, ClientePLSQL y AutoPLSQL utilizando las llaves foráneas id_cliente y id_auto para relacionar los alquileres con los clientes y los autos correspondientes.

El resultado es una lista de alquileres de autos junto con los nombres de los clientes que los han alquilado y la información de marca y modelo del auto alquilado.

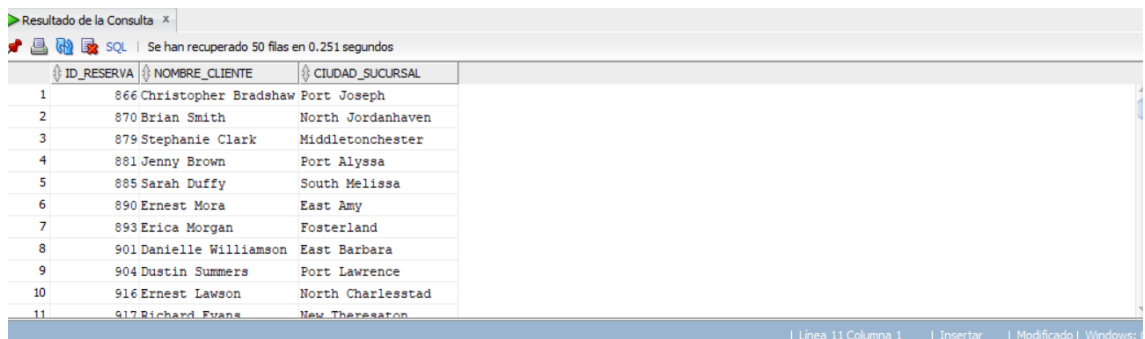
14. Mostrar los detalles de las reservas con los nombres de los clientes y las ciudades de las sucursales.

Código SQL:



```
SELECT
    R.id_reserva,
    C.nombre AS nombre_cliente,
    S.ciudad AS ciudad_sucursal
FROM
    ReservaPLSQL R
JOIN
    ClientePLSQL C ON R.id_cliente = C.id_cliente
JOIN
    SucursalPLSQL S ON R.id_sucursal = S.id_sucursal;
```

Resultados:



ID_RESERVA	NOMBRE_CLIENTE	CIUDAD_SUCURSAL
866	Christopher Bradshaw	Port Joseph
870	Brian Smith	North Jordanhaven
879	Stephanie Clark	Middletonchester
881	Jenny Brown	Port Alyssa
885	Sarah Duffy	South Melissa
890	Ernest Mora	East Amy
893	Erica Morgan	Fosterland
901	Danielle Williamson	East Barbara
904	Dustin Summers	Port Lawrence
916	Ernest Lawson	North Charlesstad
917	Richard Evans	New Theresaton

Explicación:

En esta consulta se combina la información de las tablas ReservaPLSQL, ClientePLSQL y SucursalPLSQL para obtener los detalles de las reservas, los nombres de los clientes que realizaron las reservas y las ciudades de las sucursales asociadas a las reservas.

Iniciamos especificando las columnas que queremos ver en el resultados, luego indicamos las tablas de donde saldrá dicha información y utilizamos abreviaciones para las tablas y finalmente combinamos las tablas utilizando las llaves para relacionar las tablas entre sí.

15. Mostrar los clientes que no han realizado ninguna reserva.

Código SQL:

```
clientes-sin-reserva.sql
Archivo Editar Ver

SELECT C.id_cliente, C.nombre AS nombre_cliente
FROM ClientePLSQL C
LEFT JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente
WHERE R.id_cliente IS NULL;
```

Resultados:

Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0.11 segundos

ID_CLIENTE	NOMBRE_CLIENTE
1	79 Mary Chapman

Explicación:

Seleccionamos las columnas id_cliente y nombre_cliente de la tabla ClientePLSQL, luego utilizamos un LEFT JOIN para combinar la tabla ClientePLSQL con la tabla ReservaPLSQL. Esto nos dará un resultado que incluirá todos los clientes, incluso aquellos que no tienen una reserva correspondiente en la tabla de reservas, después con WHERE, filtramos las filas donde R.id_cliente es nula, para buscar los clientes que no tienen una reserva registrada.

Agregación y Agrupamiento:

16. Contar cuántos autos hay de cada marca en la tabla "Auto".

Código SQL:

```
contarautos.sql
Archivo Editar Ver

SELECT marca, COUNT(*) AS cantidad
FROM AutoPLSQL
GROUP BY marca
```

Resultado:

Resultado de la Consulta x

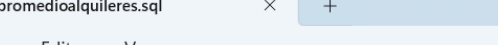
Se han recuperado 50 filas en 0.1 segundos

MARCA	CANTIDAD
1 Volkswagen	2
2 Audi	1
3 Lexus	1
4 ball	2
5 each	1
6 apply	2
7 can	2
8 land	2
9 from	1
10 participant	1

Línea 5 Columna 1 | Insertar | Modificado | Unix/Mac: LF

Explicación:

17. Calcular la duración promedio de los alquileres.



The screenshot shows a SQL editor window with the title 'promedioalquileres.sql'. The editor contains the following SQL query:

```
SELECT AVG(fecha_fin - fecha_inicio) AS duracion_promedio
FROM AlquilerPLSQL;
```

[illegible]

Restamos la fecha de inicio de la fecha de finalización para calcular la diferencia en días entre ambas fechas. Luego, utilizamos la función AVG para calcular el promedio de estas diferencias.

```
reservassucursalsql
SELECT R.id_sucursal, S.nombre AS nombre_sucursal, COUNT(R.id_reserva) AS total_reservas
FROM ReservaPLSQL R
JOIN SucursalPLSQL S ON R.id_sucursal = S.id_sucursal
GROUP BY R.id_sucursal, S.nombre;
```

Resultado de la Consulta x

SQL | Se han recuperado 50 filas en 0.11 segundos

ID_SUCURSAL	NOMBRE_SUCURSAL	TOTAL_RESERVAS
1	627 Rodriguez-Ramos	1
2	569 Reyes, Robinson and Davis	1
3	72 Friedman-Tran	70
4	476 Garcia-Lee	1
5	38 Garcia-Wilson	82
6	16 Sucursal 16	72
7	22 Harris Inc	89
8	55 West, Miller and Johnson	72
9	13 Sucursal 13	83
10	69 Dixon, Solomon and Tapia	79

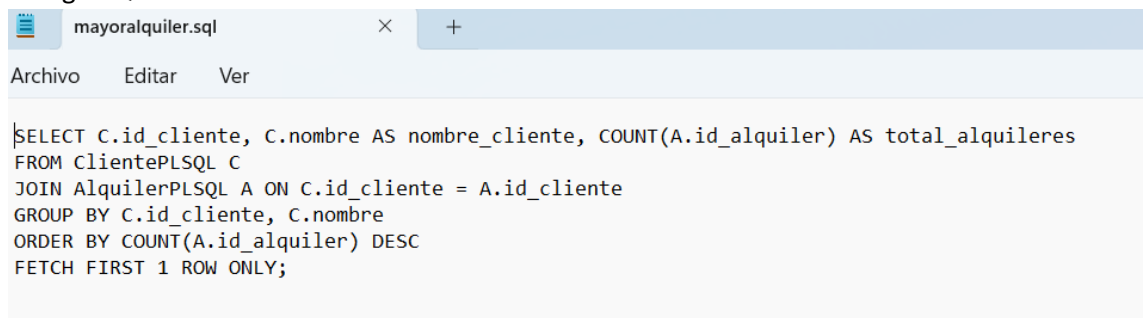
| Línea 5 Columna 1
| Insertar
| Modificar

Explicación:

Seleccionamos las columnas necesarias como R.id_sucursal, S.nombre, luego utilizamos la función COUNT para contar el número de reservas en cada sucursal. Utilizamos la cláusula FROM para especificar las tablas "ReservaPLSQL" y "SucursalPLSQL" de las cuales se utilizarán los datos, después usamos JOIN para combinar las tablas y finalmente GROUP BY para agrupar los resultados por id_sucursal y nombre.

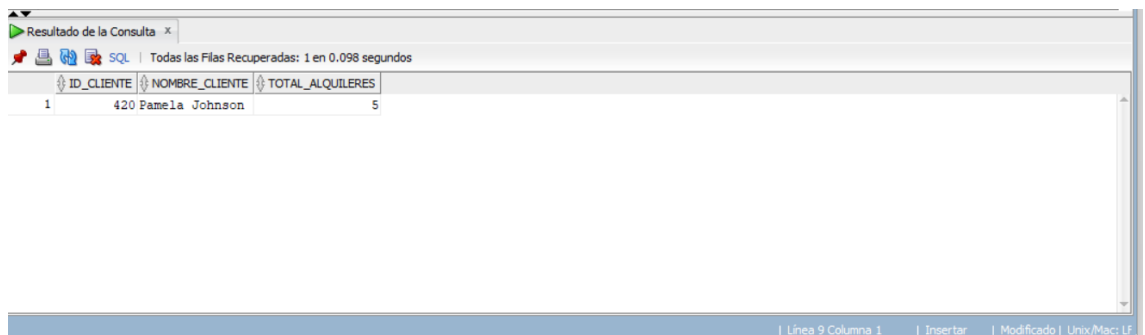
19. Encontrar el cliente que ha realizado la mayor cantidad de alquileres.

Código SQL:



```
SELECT C.id_cliente, C.nombre AS nombre_cliente, COUNT(A.id_alquiler) AS total_alquileres
FROM ClientePLSQL C
JOIN AlquilerPLSQL A ON C.id_cliente = A.id_cliente
GROUP BY C.id_cliente, C.nombre
ORDER BY COUNT(A.id_alquiler) DESC
FETCH FIRST 1 ROW ONLY;
```

Resultado:



ID_CLIENTE	NOMBRE_CLIENTE	TOTAL_ALQUILERES
1	420 Pamela Johnson	5

Explicación:

Seleccionamos las columnas necesarias para encontrar el cliente con mayor cantidad de alquileres, utilizamos COUNT para realizar el conteo de la cantidad de alquileres por cliente, luego seleccionamos las tablas de la cual saldrá la información, después usamos JOIN para combinar las tablas alquiler y cliente. Utilizamos la cláusula GROUP BY para agrupar los resultados por id_cliente y nombre para obtener el número total de alquileres para cada cliente, siguiente a esto usamos ORDER BY para ordenar los resultados en orden descendente y finalmente utilizamos FETCH FIRST 1 ROW ONLY para limitar el resultado a la primera fila, que contendrá al cliente con el mayor número de alquileres.

20. Calcular el promedio de años de los autos en la tabla "Auto".

Código SQL:

```
promedioanos.sql
Archivo Editar Ver

SELECT AVG(ano) AS promedio_anos
FROM AutoPLSQL;
```

Resultado:

Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0.109 segundos

	PROMEDIO_ANOS
1	2006,337675350701402805611222444889779559

Línea 1 Columna 1 | Insertar | Unix/Mac:

Explicación:

La función AVG(ano) toma todos los valores de la columna "ano", suma estos valores y luego divide la suma total por el número de registros para obtener el promedio. El resultado será el promedio de los años de los autos en la tabla "AutoPLSQL".

Subconsultas:

21. Mostrar los clientes que han realizado al menos una reserva.

Código SQL:

```
1reservaalmenos.sql
Archivo Editar Ver

SELECT id_cliente, nombre AS nombre_cliente
FROM ClientePLSQL
WHERE EXISTS (
    SELECT 1
    FROM ReservaPLSQL
    WHERE ReservaPLSQL.id_cliente = ClientePLSQL.id_cliente
);
```

Resultado:

Resultado de la Consulta x

Se han recuperado 50 filas en 0.098 segundos

	ID_CLIENTE	NOMBRE_CLIENTE
1	780	Diane Armstrong
2	737	Tammy Ochoa
3	197	Christopher Bradshaw
4	594	Jasmine Burgess
5	315	Jessica Murray
6	314	Brian Smith
7	408	Ernest Mora
8	128	Bradley Walls
9	256	Bryce Shields
10	437	Lawrence Mann

Línea 1 Columna 1 | Insertar | Unix/Mac:

Explicación:

Inicialmente seleccionamos las columnas id_cliente y nombre de la tabla ClientePLSQL, luego utilizamos la cláusula FROM para especificar la tabla "ClientePLSQL" de la que

obtendremos los datos. Utilizamos WHERE para filtrar las filas en las que al menos una reserva existe para el cliente actual. La cláusula EXISTS la usamos para verificar si hay al menos una fila en la tabla "ReservaPLSQL" que tiene un id_cliente igual al id_cliente en la tabla "ClientePLSQL". El resultado es una lista de clientes que han realizado al menos una reserva. Cada fila representa un cliente que cumple con la condición EXISTS.

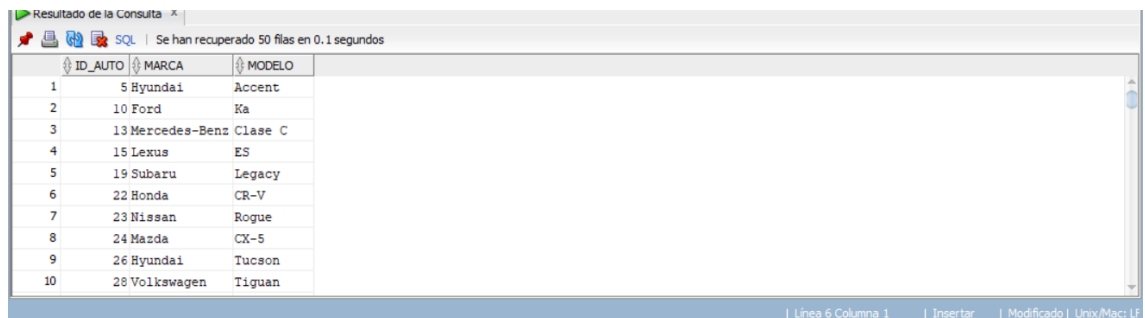
22. Mostrar los autos que no han sido alquilados aún.

Código SQL:

```
AUTOSINALQUILAR.sql
Archivo Editar Ver

SELECT A.id_auto, A.marca, A.modelo
FROM AutoPLSQL A
LEFT JOIN AlquilerPLSQL AL ON A.id_auto = AL.id_auto
WHERE AL.id_auto IS NULL;
```

Resultado:



Resultado de la Consulta x

Se han recuperado 50 filas en 0.1 segundos

ID_AUTO	MARCA	MODELO
1	5 Hyundai	Accent
2	10 Ford	Ka
3	13 Mercedes-Benz	Clase C
4	15 Lexus	ES
5	19 Subaru	Legacy
6	22 Honda	CR-V
7	23 Nissan	Rogue
8	24 Mazda	CX-5
9	26 Hyundai	Tucson
10	28 Volkswagen	Tiguan

Linea 6 Columna 1 Insertar Modificado Unix/Mac: LF

Explicación:

Seleccionamos las columnas id_auto, marca, modelo y ano de la tabla "AutoPLSQL".

Utilizamos una cláusula LEFT JOIN para combinar la tabla de autos ("AutoPLSQL") con la tabla de alquileres ("AlquilerPLSQL") utilizando la columna id_auto como llave. Esto nos permite relacionar cada auto con sus registros de alquiler correspondientes incluso los que no tienen registros.

Utilizamos la cláusula WHERE para filtrar las filas donde AL.id_auto (la columna de la tabla de alquileres) es nula. Para buscar los autos que no tienen un registro de alquiler correspondiente en la tabla de alquileres.

23. Encontrar los clientes que han alquilado el mismo auto más de una vez.

Código SQL:

```
23.sql
Archivo  Editar  Ver

SELECT
    c.nombre,
    a.marca,
    COUNT(DISTINCT aq.id_alquiler) AS cantidad_alquileres
FROM ALQUILERPLSQL aq
JOIN AUTOPLSQL a ON a.id_auto = aq.id_auto
JOIN CLIENTEPLSQL c ON aq.id_cliente = c.id_cliente
GROUP BY c.nombre, a.marca
HAVING COUNT(DISTINCT aq.id_alquiler) > 1;
```

Resultado:

Resultado de la Consulta x

Todas las Filas Recuperadas: 0 en 0.122 segundos

NOMBRE	MARCA	CANTIDA...
--------	-------	------------

Línea 10 Columna 1 | Insertar | Modificado | Unix/Mac: L3

Explicación:

Comenzamos especificando las columnas que queremos obtener en el resultado luego utilizamos COUNT para contar el número de registros en la tabla ALQUILERPLSQL con un id_alquiler distinto. Esto nos dará la cantidad de alquileres que ha realizado cada cliente para cada marca de auto. Luego relacionamos los registros entre las tablas ALQUILERPLSQL y AUTOPLSQL y luego entre ALQUILERPLSQL y CLIENTEPLSQL. Después agrupamos los resultados por el nombre del cliente y la marca del auto. Finalmente, con la cláusula HAVING . Filtra los grupos de resultados y solo selecciona aquellos donde la cantidad de alquileres (COUNT(DISTINCT aq.id_alquiler)) es mayor que 1. El resultado es vacío indicándonos que no hay clientes que alquilaran un auto más de una vez

24. Mostrar los clientes que han realizado alquileres en la misma ciudad en la que viven.

Código SQL:

```
24.sql
Archivo  Editar  Ver

SELECT c.*, s.* from reservaplsq r
JOIN alquilerplsqa a on r.id_reserva = a.id_reserva
JOIN clienteplsqa c on a.id_cliente = c.id_cliente
JOIN sucursalplsqa s on r.id_sucursal = s.id_sucursal
WHERE c.direccion LIKE '%' || s.ciudad || '%';
```

Resultado:

Resultado de la Consulta x

Todas las Filas Recuperadas: 0 en 0.126 segundos

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO	ID_SUCU...	NOMBRE_1	CIUDAD	PAIS
------------	--------	-----------	----------	------------	----------	--------	------

Linea 1 Columna 1 | Insertar | Modificado | Unix/Mac: L

Explicación:

Comenzamos seleccionando las columnas necesarias indicando la tabla reservaPLSQL. Realizamos una unión con JOIN usando la tabla alquilerplsql utilizando la columna id_reserva para relacionar los registros entre las tablas reservaplsq y alquilerplsql. Esto se hace para vincular las reservas con los alquileres. Continuamos con otra unión, esta vez con la tabla clientepsq y realizamos una tercera unión con la tabla sucursalplsql. Finalmente se aplica una condición con la cláusula WHERE. La consulta filtra los resultados para que solo se seleccionen aquellos donde la columna direccion de la tabla clientepsq contenga la ciudad de la sucursal, LIKE se utiliza para buscar la ciudad en la dirección del cliente. El símbolo % se utiliza como comodín, lo que significa que coincidirá con cualquier cadena que contenga la ciudad en la dirección del cliente.

25. Encontrar los autos que han sido alquilados en la misma sucursal donde se realizó una reserva.

Código SQL:

```
25.sql
SELECT a.* from autoplsq a
JOIN alquilerplsql al ON a.id_auto = al.id_auto
JOIN reservaplsq r ON al.id_reserva = r.id_reserva
JOIN sucursalplsql s ON r.id_sucursal = s.id_sucursal
```

Resultado:

Resultado de la Consulta x

Se han recuperado 50 filas en 0.207 segundos

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2025
2	2 Honda	Civic	2025
3	3 Nissan	Versa	2023
4	3 Nissan	Versa	2023
5	3 Nissan	Versa	2023
6	3 Nissan	Versa	2023
7	4 Kia	Rio	2023
8	4 Kia	Rio	2023
9	4 Kia	Rio	2023
10	4 Kia	Rio	2023

Linea 5 Columna 1 | Insertar | Modificado | Unix/Ma

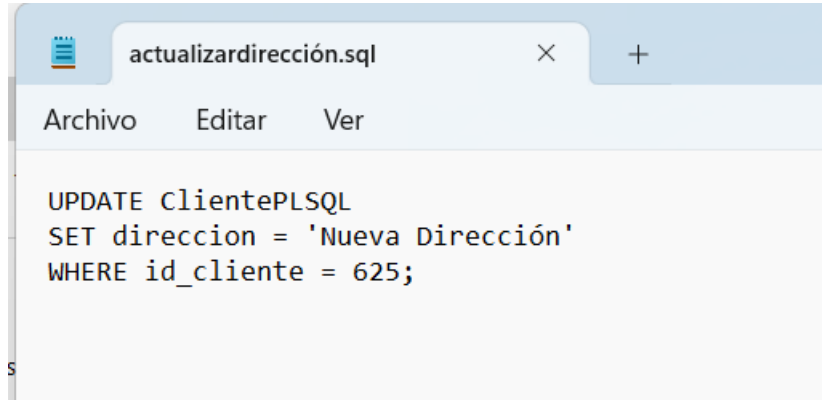
Explicación:

Esta consulta está relacionando las tablas de autos, alquileres, reservas y sucursales uniéndolas mediante llaves utilizando la clausula JOIN para obtener información sobre los autos que han sido alquilados en reservas que se llevaron a cabo en una sucursal específica. La consulta seleccionará todos los datos de la tabla de autos para esos casos particulares.

Actualizaciones y Eliminaciones:

26. Actualizar la dirección de un cliente específico.

Código SQL:



```
UPDATE ClientePLSQL
SET direccion = 'Nueva Dirección'
WHERE id_cliente = 625;
```

Resultado:



Salida de Script x

Tarea terminada en 0.114 segundos

1 fila actualizadas.

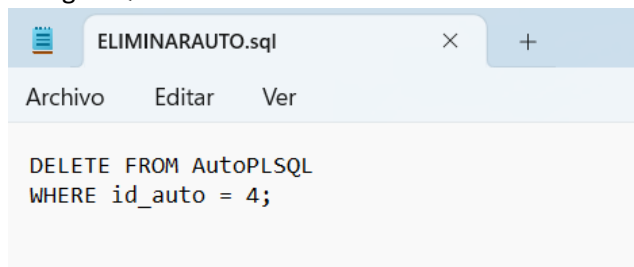
Linea 5 Columna 1 | Insertar | Modificado | Windows: C

Explicación:

Con UPDATE se especifica la tabla que se desea actualizar, luego con SET escogemos la columna a la cual se establecerá un nuevo valor y finalmente indicamos la condición para señalar al cliente al cual se le realizara dicha actualización, utilizando como llave el id_cliente.

27. Eliminar un auto de la tabla "Auto".

Código SQL:



```
DELETE FROM AutoPLSQL
WHERE id_auto = 4;
```

Resultado:



Salida de Script x

Tarea terminada en 0.124 segundos

1 fila eliminado

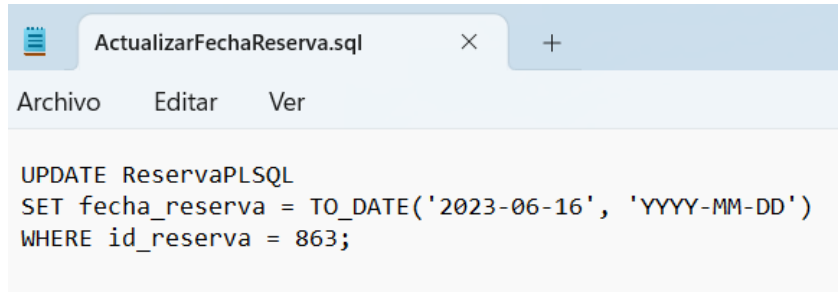
Linea 2 Columna 20 | Insertar | Modificado | Unix/Mac: C

Explicación:

Se especifica la tabla de la que se quiere eliminar un registro, en este caso, "AutoPLSQL", y luego establecemos la condición que indica cual registro se debe eliminar.

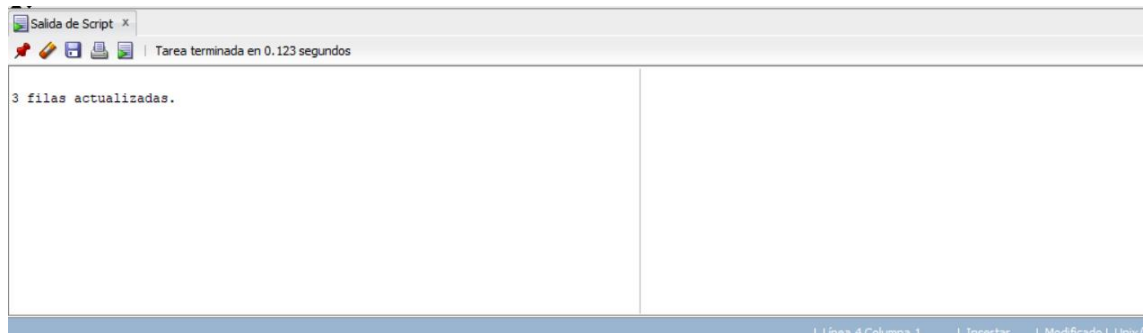
28. Marcar una reserva como completada actualizando la fecha de fin.

Código SQL:



```
UPDATE ReservaPLSQL
SET fecha_reserva = TO_DATE('2023-06-16', 'YYYY-MM-DD')
WHERE id_reserva = 863;
```

Resultado:



Salida de Script x

Tarea terminada en 0.123 segundos

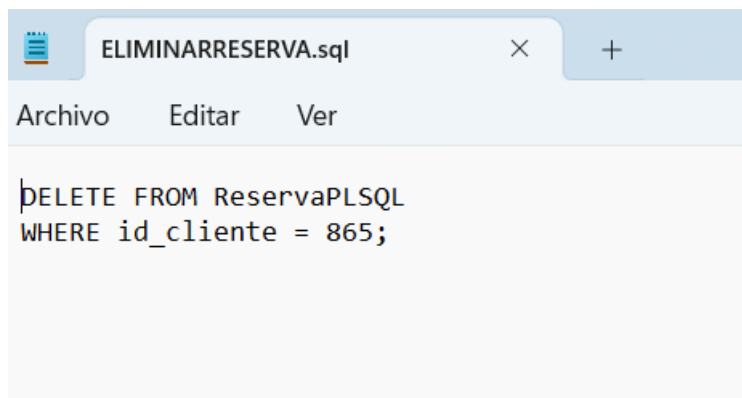
3 filas actualizadas.

Explicación:

Seleccionamos la tabla que se desea actualizar, y luego se selecciona la columna en la que se debe actualizar la fecha, sin embargo en la instrucción dice que se debe actualizar la columna fecha de fin, pero la tabla no cuenta con esa columna, así que se trabajó con la columna fecha_reserva para cumplir con la instrucción.

29. Eliminar todas las reservas realizadas por un cliente específico.

Código SQL:



```
DELETE FROM ReservaPLSQL
WHERE id_cliente = 865;
```

Resultado:

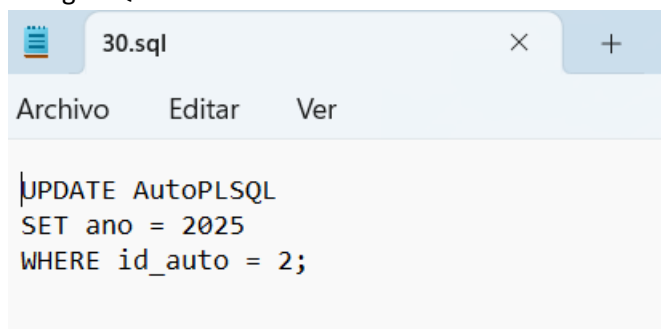


Explicación:

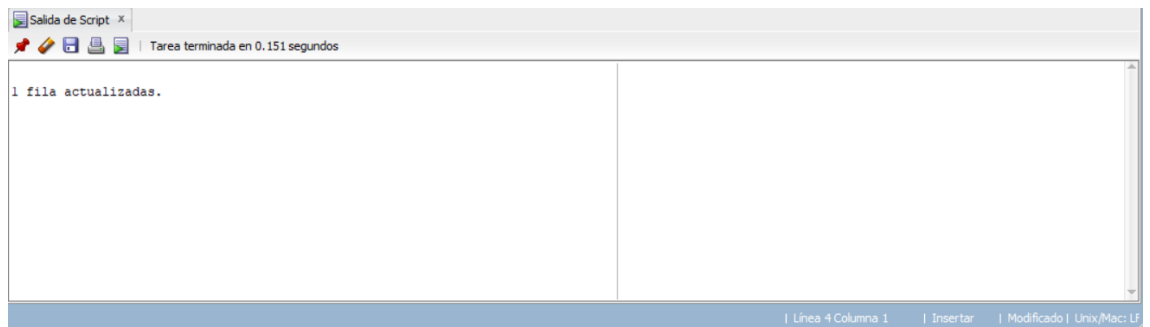
Especificamos la tabla de la cual queremos borrar la información, y luego establecemos la condición indicando a cual cliente se le deberá borrar la información.

30. Actualizar el año de un auto en la tabla "Auto".

Código SQL:



Resultado:



Explicación:

Señalamos la tabla de la cual queremos actualizar la información, después actualizamos la columna de la cual vamos a modificar el valor año e indicamos la condición para señalar a que auto vamos a aplicarle la actualización.