

Taller
Cuaderno PL/SQL
Santiago Monsalve

Ciclo 1

-- 1. Mostrar todos los clientes en la tabla 'Clientepsql'.

SELECT * FROM Clientepsql;

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	9408 Alvarado Ramp, Grossbury, Kentucky 37569	817.439.0622
6	631 Jennifer Brown	9409 Alvarado Ramp, Port Matthewberg, Virginia 82642	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	41-787-434-7337

-- 2. Mostrar todos los autos en la tabla 'Autoplsq'.

SELECT * FROM Autoplsq;

	ID_AUTO	MARCA	MODELO	ANO
1	2	Honda	Civic	2025
2	3	Nissan	Versa	2023
3	4	Kia	Rio	2023
4	5	Hyundai	Accent	2024
5	6	Chevrolet	Onix	2023
6	7	Fiat	Argo	2023
7	9	Renault	Kwid	2023
8	10	Ford	Ka	2023
9	11	BMW	Serie 3	2020
10	15	Lexus	ES	2023
11	16	Genesis	G80	2023
12	17	Volvo	S60	2023
13	18	Mazda	6	2023

-- 3. Mostrar todos los alquileres en la tabla 'Alquilerpsql'.

SELECT * FROM Alquilerplsql;

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131	208	01/10/22	30/10/22	584
2	585	394	543	19/04/23	11/05/23	585
3	586	774	345	19/07/23	30/07/23	586
4	587	81	250	26/03/23	22/04/23	587
5	588	16	371	23/02/23	25/02/23	588
6	589	999	584	01/05/23	21/05/23	589
7	590	796	711	26/05/23	17/06/23	590
8	591	928	812	08/05/23	15/05/23	591
9	592	223	847	27/11/22	14/12/22	592
10	593	987	961	21/08/23	20/09/23	593
11	594	435	45	18/09/23	29/09/23	594
12	595	81	435	06/05/23	31/05/23	595
13	596	307	166	16/09/23	25/09/23	596

-- 4. Mostrar todas las sucursales en la tabla 'Sucursalplsql'.

SELECT * FROM Sucursalplsql;

	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

-- 5. Mostrar todas las reservas en la tabla 'Reservaplsq'.

SELECT * FROM Reservaplsq;

	ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	863	780	739	16/05/23
2	864	737	729	04/06/23
3	865	700	982	10/12/22
4	866	197	610	29/06/23
5	867	594	755	17/06/23
6	868	808	185	07/03/23
7	869	315	394	19/03/23
8	870	314	627	03/10/22
9	871	408	687	23/12/22
10	872	128	821	22/09/23
11	873	858	380	16/12/22
12	875	256	954	11/08/23
12	876	427	002	21/08/22

-- 6. Mostrar los clientes que se llaman 'Juan'.

SELECT * FROM Clientepsql WHERE nombre = 'Juan';

ID_CLI...	NOMBRE	DIREC...	TELEF...
-----------	--------	----------	----------

-- 7. Mostrar los autos de marca 'Toyota'.

SELECT * FROM Autopsql WHERE marca = 'Toyota';

ID_AUTO	MARCA	MODELO	ANO
---------	-------	--------	-----

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

SELECT * FROM Alquilerpsql WHERE fecha_inicio > '21/05/23';

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	586	774	345	19/07/23	30/07/23	586
2	590	796	711	26/05/23	17/06/23	590
3	593	987	961	21/08/23	20/09/23	593
4	594	435	45	18/09/23	29/09/23	594
5	596	397	466	16/09/23	25/09/23	596
6	602	830	198	11/08/23	29/08/23	602
7	605	171	251	14/06/23	28/06/23	605
8	608	876	328	25/07/23	23/08/23	608
9	610	155	112	10/09/23	06/10/23	610
10	615	668	539	09/06/23	17/06/23	615
11	618	748	984	19/07/23	09/08/23	618
12	623	573	16	28/07/23	27/08/23	623
13	625	421	222	05/06/23	06/06/23	625

-- 9. Mostrar las sucursales ubicadas en 'Madrid'.

SELECT * FROM Sucursalplsql WHERE ciudad = 'Madrid';

ID_SUC...	NOMBRE	CIUDAD	PAIS
-----------	--------	--------	------

-- 10. Mostrar las reservas realizadas por un cliente específico.

SELECT * FROM Reservaplsql WHERE id_cliente = 1;

ID_RES...	ID_CLI...	ID_SUC...	FECHA...
-----------	-----------	-----------	----------

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

SELECT Alquilerplsql.*, Clientepsql.nombre, Autopsql.marca
FROM Alquilerplsql
JOIN Clientepsql ON Alquilerplsql.id_cliente = Clientepsql.id_cliente
JOIN Autopsql ON Alquilerplsql.id_auto = Autopsql.id_auto;

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA	NOMBRE	MARCA
1	584	131	208 01/10/22	30/10/22	584	Alexander Cunningham	lot
2	585	394	543 19/04/23	11/05/23	585	Christine Snyder	economic
3	586	774	345 19/07/23	30/07/23	586	Tara Johnson	pressure
4	588	16	371 23/02/23	25/02/23	588	Tanya Blake	account
5	589	999	584 01/05/23	21/05/23	589	Jamie Davenport	gas
6	591	928	812 08/05/23	15/05/23	591	Matthew Walter	cell
7	592	223	847 27/11/22	14/12/22	592	Jennifer Hansen	operation
8	593	987	961 21/08/23	20/09/23	593	Todd Yoder	image
9	596	397	466 16/09/23	25/09/23	596	Kevin Hoffman	later
10	598	726	669 09/11/22	23/11/22	598	Cassandra Davis	strong
11	599	999	715 07/10/22	25/10/22	599	Jamie Davenport	stock
12	600	414	809 29/12/22	10/01/23	600	Timothy Dixon	way
13	602	830	198 11/08/23	29/08/23	602	David Wallace	memorv

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

```
SELECT Clientepsql.*
FROM Clientepsql
JOIN Reservapsql ON Clientepsql.id_cliente = Reservapsql.id_cliente
WHERE Reservapsql.id_sucursal = 5;
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	627 Gina Morgan	9409 Alvarado Ramp, Grossbury, Kentucky 37569	581-453-1641
2	679 Christopher Smith	824 Anna Trail, Nicolefort, Alaska 86954	2819349284
3	780 Diane Armstrong	378 Edward Hollow, Seanmouth, Hawaii 27595	728-245-2069
4	780 Diane Armstrong	378 Edward Hollow, Seanmouth, Hawaii 27595	728-245-2069
5	785 Gregory Mitchell	945 Melissa Groves Suite 125, Whitneyburgh, Indiana 64561	720-922-7961
6	785 Gregory Mitchell	945 Melissa Groves Suite 125, Whitneyburgh, Indiana 64561	720-922-7961
7	787 Daniel Adams	89184 Young Place Suite 898, New Brandy, North Dakota 98838	920.542.9773
8	853 Daniel Payne	9935 Ward Lakes, Dianehaven, Maine 89593	7552349753
9	853 Daniel Payne	9935 Ward Lakes, Dianehaven, Maine 89593	7552349753
10	859 Michael Boyd	3515 Rivas Stravenue Suite 832, Millsburgh, Kansas 44050	(660)876-7993
11	859 Michael Boyd	3515 Rivas Stravenue Suite 832, Millsburgh, Kansas 44050	(660)876-7993
12	874 Justin Jimenez	598 James Burgs Suite 219, Lake Kyleborough, South Dakota 06445	483-271-4402
13	882 Marv Martinez	901 Diana Spring. East Davidfurt. Florida 43464	5639883860

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

```
SELECT Autopsql.*, Clientepsql.nombre
FROM Autopsql
JOIN Alquilerpsql ON Autopsql.id_auto = Alquilerpsql.id_auto
JOIN Clientepsql ON Alquilerpsql.id_cliente = Clientepsql.id_cliente;
```

	ID_AUTO	MARCA	MODELO	ANO	NOMBRE
1	2	Honda	Civic	2025	Robert Moran
2	3	Nissan	Versa	2023	Steven Velasquez
3	4	Kia	Rio	2023	Gina Morgan
4	6	Chevrolet	Onix	2023	Stefanie Sullivan
5	7	Fiat	Argo	2023	Emily Santana
6	9	Renault	Kwid	2023	Megan West
7	9	Renault	Kwid	2023	Melinda Jackson
8	11	BMW	Serie 3	2020	Kimberly Johnson
9	16	Genesis	G80	2023	Stephanie Clark
10	18	Mazda	6	2023	Jennifer Gamble
11	18	Mazda	6	2023	Nicholas Martin
12	20	Hyundai	Sonata	2023	Michael Koch
13	21	Kia	K5	2023	Leslie Mccov

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

```
SELECT Reservaplsq.*, Clientepsql.nombre, Sucursalpsql.ciudad
FROM Reservaplsq
JOIN Clientepsql ON Reservaplsq.id_cliente = Clientepsql.id_cliente
JOIN Sucursalpsql ON Reservaplsq.id_sucursal = Sucursalpsql.id_sucursal;
```

	ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA	NOMBRE	CIUDAD
1	866	197	610	29/06/23	Christopher Bradshaw	Port Joseph
2	870	314	627	03/10/22	Brian Smith	North Jordanhaven
3	879	573	287	13/06/23	Stephanie Clark	Middletonchester
4	881	647	46	28/05/23	Jenny Brown	Port Alyssa
5	885	493	329	07/02/23	Sarah Duffy	South Melissa
6	890	408	144	11/10/22	Ernest Mora	East Amy
7	893	165	68	08/10/22	Erica Morgan	Fosterland
8	901	588	513	01/06/23	Danielle Williamson	East Barbara
9	904	423	470	26/12/22	Dustin Summers	Port Lawrence
10	916	612	529	22/04/23	Ernest Lawson	North Charlesstad
11	917	361	35	01/05/23	Richard Evans	New Theresaton
12	919	399	504	25/01/23	Benjamin Perez	Port Jadeview
13	921	136	59	27/03/23	Jonathan Webster	New Codview

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

```
SELECT * FROM Clientepsql WHERE id_cliente NOT IN (SELECT id_cliente FROM
Reservaplsq);
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	79Mary Chapman	9270 Lawrence Port, Parrishland, South Dakota 29696	986-323-8823

-- 16. Contar cuántos autos hay de cada marca en la tabla 'Autoplsq'.

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

	MARCA	CANTIDAD
1	Lexus	1
2	Volkswagen	1
3	ball	2
4	each	1
5	apply	2
6	can	2
7	land	2
8	from	1
9	participant	1
10	main	4
11	major	1
12	realize	1
13	vote	3

-- 17. Calcular la duración promedio de los alquileres.

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

	DURACION_PROMEDIO
1	16,24446680080482897384305835010060362173

-- 18. Mostrar el número total de reservas realizadas en cada sucursal.

```
SELECT id_sucursal, COUNT(*) AS num_reservas
FROM Reservaplsq
GROUP BY id_sucursal;
```

	ID_SUCURSAL	NUM_RESERVAS
1	124	2
2	513	1
3	290	1
4	852	2
5	72	70
6	313	6
7	667	1
8	685	1
9	837	1
10	722	2
11	27	95
12	52	82
13	23	85

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

```
SELECT id_cliente, COUNT(id_cliente) AS num_alquileres
FROM Alquilerplsql
GROUP BY id_cliente
ORDER BY num_alquileres DESC
FETCH FIRST 1 ROWS ONLY;
```

	ID_CLIENTE	NUM_ALQUILERES
1	420	5

-- 20. Calcular el promedio de años de los autos en la tabla 'Autoplsql'.

```
SELECT AVG(ano) AS promedio_años
FROM Autoplsql;
```

	PROMEDIO_AÑOS
1	2006,330312185297079556898288016112789527

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

```
SELECT * FROM Clientplsql
WHERE id_cliente IN (SELECT DISTINCT id_cliente FROM Reservaplsql);
```


ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	780 Diane Armstrong	378 Edward Hollow, Seanmouth, Hawaii 27595	728-245-2069
2	737 Tammy Ochoa	749 Hernandez Estate, North Stephanie, Wyoming 78862	833.606.4853
3	197 Christopher Bradshaw	49831 Baker Coves Suite 969, Victorberg, Alaska 24287	(992)281-1696
4	594 Jasmine Burgess	0241 Ward Mountains Apt. 413, Jonathanville, Texas 41605	825.402.7593
5	315 Jessica Murray	17972 Cook Squares Apt. 092, Hollandbury, Pennsylvania 64786	263-700-7643
6	314 Brian Smith	47371 Heath Mountain, Westland, New York 41263	211-494-8869
7	408 Ernest Mora	119 Mary Estates Suite 979, South Katherine, Tennessee 16259	7333224922
8	128 Bradley Walls	36617 Rice Island, Kingland, Mississippi 63054	926.985.0976
9	256 Bryce Shields	9972 Jessica Valley Apt. 090, South Nicholas, Mississippi 44815	542-729-2284
10	437 Lawrence Mann	38415 Palmer Meadow, New Robertchester, South Dakota 39529	499.234.1742
11	573 Stephanie Clark	83559 Flowers Throughway Apt. 375, Michaelport, Iowa 10086	(798)490-6220
12	647 Jenny Brown	610 Stacey Ports, South Riley, Texas 75286	(798)393-2112
13	467 Robert West	324 Jennifer Loop Apt. 779, West Kennethton, Oklahoma 07153	477-777-7081

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

SELECT * FROM Autoplsql

WHERE id_auto NOT IN (SELECT DISTINCT id_auto FROM Alquilerplsql);

	ID_AUTO	MARCA	MODELO	ANO
1	5	Hyundai	Accent	2024
2	10	Ford	Ka	2023
3	15	Lexus	ES	2023
4	19	Subaru	Legacy	2023
5	22	Honda	CR-V	2023
6	23	Nissan	Rogue	2023
7	24	Mazda	CX-5	2023
8	26	Hyundai	Tucson	2023
9	28	Volkswagen	Tiguan	2023
10	29	Chevrolet	Equinox	2023
11	30	Ford	Escape	2023
12	851	nation	perhaps	2009
13	853	Mrs	need	2013

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

SELECT id_cliente

FROM Alquilerplsql

GROUP BY id_cliente, id_auto

HAVING COUNT(id_auto) > 1;

ID_CLIENTE
1 447

-- 24. Mostrar los clientes que han realizado alquileres en la misma ciudad en la que viven.
SELECT DISTINCT C.* FROM ClientePLSQL C INNER JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente;

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	314 Brian Smith	47371 Heath Mountain, Westland, New York 41263	211-494-8869
2	408 Ernest Mora	119 Mary Estates Suite 979, South Katherine, Tennessee 16259	7333224922
3	165 Erica Morgan	022 Baker Trail Apt. 989, West Nicholasside, Washington 43934	276-859-1890
4	445 John Greene	1495 Sydney Ridges Suite 103, Jamestown, Virginia 77044	+1-833-269-7398
5	186 Jacob Montgomery	70537 April Ranch Suite 067, Travisburgh, Wisconsin 69106	9764875683
6	576 Jonathan Walsh	580 Robert Burgs, Douglasville, Arizona 30817	(311)240-7498
7	365 Jessica Gilbert	7441 Kevin River, Lake Lauren, Virginia 61121	866.383.9179
8	960 Megan West	6190 Matthew Extension, Davisville, Georgia 83012	386.242.4574
9	880 Danielle Sanchez	771 Gregory Road, Russellhaven, Connecticut 75932	4528271923
10	991 Amber Mosley	0211 Kimberly Bridge Apt. 252, South Jamesfurt, Connecticut 23718	917-320-2127
11	227 Melissa Richardson	3516 Alicia Cliffs, Lake Carla, Missouri 34232	879-360-0379
12	62 Deborah Griffith	492 Green Meadow, Baileyshire, Tennessee 89714	3649264959
13	998 Angela Everett	726 Leon Inlet Suite 560, North Autumnstad, Maryland 84649	818.849.6119

-- 25. Encontrar los autos que han sido alquilados en la misma sucursal donde se realizó una reserva.
SELECT A.* FROM AutoPLSQL A LEFT JOIN AlquilerPLSQL AL ON A.id_auto = AL.id_auto
WHERE AL.id_auto IS NULL;

	ID_AUTO	MARCA	MODELO	ANO
1	5	Hyundai	Accent	2024
2	10	Ford	Ka	2023
3	15	Lexus	ES	2023
4	19	Subaru	Legacy	2023
5	22	Honda	CR-V	2023
6	23	Nissan	Rogue	2023
7	24	Mazda	CX-5	2023
8	26	Hyundai	Tucson	2023
9	28	Volkswagen	Tiguan	2023
10	29	Chevrolet	Equinox	2023
11	30	Ford	Escape	2023
12	851	nation	perhaps	2009
13	853	Mrs	need	2013

-- 26. Actualizar la dirección de un cliente específico.

UPDATE Clientes SET direccion = 'Cl45a#116-67' WHERE id_cliente = 5;

-- 26. Actualizar la dirección de un cliente específico.

UPDATE Clientes SET direccion = 'Cl45a#116-67' WHERE id_cliente = 18;

1 fila actualizadas.

-- 27. Eliminar un auto de la tabla 'Autos'.

DELETE FROM Autos WHERE id_auto = 5;

1 fila eliminado

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

UPDATE Alquileres SET fecha_fin = TO_DATE('2023-09-15', 'YYYY-MM-DD') WHERE id_reserva = 6;

1 fila actualizadas.

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

```
DELETE FROM Reservaplsq WHERE id_cliente = 25;
```

11 filas eliminado

-- 30. Actualizar el año de un auto en la tabla 'Autoplsq'.

```
UPDATE Autoplsq SET ano = 2020 WHERE id_auto = 6;
```

1 fila actualizadas.

Ciclo 2:

```
SELECT * FROM ClientePLSQL;
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	625 Ricky Owens	Nueva Dirección	3596036549
2	626 Brenda Cervantes	075 Brown Green Suite 198, Port Matthewberg, Utah 95139	483.526.6042
3	627 Gina Morgan	cra89 a sur	581-453-1641
4	628 Carl Williams	24381 Jennifer Mountain Suite 361, Danielleshire, South Dakota 16856	(362)765-7798
5	629 Jerry Malone	9408 Alvarado Ramp, Grossbury, Kentucky 37569	817.439.0622
6	631 Jennifer Brown	9409 Alvarado Ramp, Port Matthewberg, Virginia 82642	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	Calle 22 sur #315-17	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	41-787-434-7337

```
SELECT * FROM AutoPLSQL;
```

	ID_AUTO	MARCA	MODELO	ANO
1	2	Honda	Civic	2025
2	3	Nissan	Versa	2023
3	4	Kia	Rio	2023
4	6	Chevrolet	Onix	2020
5	7	Fiat	Argo	2023
6	9	Renault	Kwid	2023
7	10	Ford	Ka	2023
8	11	BMW	Serie 3	2020
9	16	Genesis	G80	2023
10	18	Mazda	6	2024
11	19	Subaru	Legacy	2023
12	20	Hyundai	Sonata	2023
13	21	Kia	K5	2023

SELECT * FROM AlquilerPLSQL;

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131	208	01/10/22	01/11/23	584
2	585	394	543	19/04/23	11/05/23	585
3	586	774	345	19/07/23	30/07/23	586
4	587	81	250	26/03/23	22/04/23	587
5	588	16	371	23/02/23	25/02/23	588
6	589	999	584	01/05/23	21/05/23	589
7	590	796	711	26/05/23	17/10/23	590
8	591	928	812	08/05/23	15/05/23	591
9	592	223	847	27/11/22	14/12/22	592
10	593	987	961	21/08/23	20/09/23	593
11	594	435	45	18/09/23	29/09/23	594
12	595	81	435	06/05/23	31/05/23	595
13	596	397	466	16/09/23	25/09/23	596

SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c
JOIN AlquilerPLSQL al ON c.id_cliente = al.id_cliente
JOIN AutoPLSQL a ON al.id_auto = a.id_auto;

	NOMBRE	MARCA	MODELO
1	Robert Moran	Honda	Civic
2	Steven Velasquez	Nissan	Versa
3	Gina Morgan	Kia	Rio
4	Stefanie Sullivan	Chevrolet	Onix
5	Emily Santana	Fiat	Argo
6	Megan West	Renault	Kwid
7	Melinda Jackson	Renault	Kwid
8	Kimberly Johnson	BMW	Serie 3
9	Stephanie Clark	Genesis	G80
10	Jennifer Gamble	Mazda	6
11	Nicholas Martin	Mazda	6
12	Michael Koch	Hyundai	Sonata
13	Leslie McGraw	Kia	K5

SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto;

	MARCA	MODELO	ANO
1	Honda	Civic	2025
2	Nissan	Versa	2023
3	Nissan	Versa	2023
4	Kia	Rio	2023
5	Kia	Rio	2023
6	Chevrolet	Onix	2020
7	Chevrolet	Onix	2020
8	Fiat	Argo	2023
9	Fiat	Argo	2023
10	Fiat	Argo	2023
11	Renault	Kwid	2023
12	Renault	Kwid	2023
13	BMW	Serie 3	2020

SELECT * FROM AlquilerPLSQL WHERE id_cliente = 1;

ID_ALQ...	ID_CLI...	ID_AUTO	FECHA...	FECHA...	ID_RES...
-----------	-----------	---------	----------	----------	-----------

SELECT * FROM AlquilerPLSQL WHERE id_auto = 1;

ID_ALQ...	ID_CLI...	ID_AUTO	FECHA...	FECHA...	ID_RES...
-----------	-----------	---------	----------	----------	-----------

SELECT * FROM AlquilerPLSQL WHERE id_sucursal = 1;

ORA-00904: "ID_SUCURSAL": identificador no válido
00904. 00000 - "%s: invalid identifier"
***Cause:**
***Action:**
Error en la línea: 13, columna: 35

SELECT * FROM AlquilerPLSQL WHERE fecha_inicio = '27/09/23';

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	843	149	122	27/09/23	11/10/23	843
2	536	372	549	27/09/23	12/10/23	536
3	582	637	502	27/09/23	05/10/23	582

SELECT COUNT(*) FROM AlquilerPLSQL;

	COUNT(*)
1	993

SELECT
c.nombre
FROM

```

ClientePLSQL c
JOIN
  AlquilerPLSQL a ON c.id_cliente = a.id_cliente
JOIN
  ReservaPLSQL r ON a.id_reserva = r.id_reserva
JOIN
  SucursalPLSQL s ON r.id_sucursal = s.id_sucursal
WHERE
  s.nombre = 'Sucursal Central';

```

NOMBRE

```

SELECT a.marca, a.modelo
FROM AutoPLSQL a
JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
WHERE al.id_cliente = 1 AND al.fecha_inicio = TO_DATE('27/09/2023', 'DD/MM/YYYY');

```

MARCA	MODELO
-------	--------

```

SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio > 7;

```

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131	208	01/10/22	01/11/23	584
2	585	394	543	19/04/23	11/05/23	585
3	586	774	345	19/07/23	30/07/23	586
4	587	81	250	26/03/23	22/04/23	587
5	589	999	584	01/05/23	21/05/23	589
6	590	796	711	26/05/23	17/10/23	590
7	592	223	847	27/11/22	14/12/22	592
8	593	987	961	21/08/23	20/09/23	593
9	594	435	45	18/09/23	29/09/23	594
10	595	81	435	06/05/23	31/05/23	595
11	596	397	466	16/09/23	25/09/23	596
12	597	36	473	06/02/23	01/03/23	597
13	598	726	669	09/11/22	23/11/22	598

```

SELECT c.nombre, COUNT(*) AS numero_alquileres

```



```

FROM ClientePLSQL c
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
GROUP BY c.nombre
ORDER BY numero_alquileres DESC
FETCH FIRST 1 ROWS ONLY;

```

	NOMBRE	NUMERO_ALQUILERES
1	John Greene	6

```

SELECT s.nombre, COUNT(*) AS numero_alquileres
FROM SucursalPLSQL s
JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal
GROUP BY s.nombre
ORDER BY numero_alquileres DESC
FETCH FIRST 1 ROWS ONLY;

```

```

ORA-00904: "AL"."ID_SUCURSAL": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error en la línea: 3, columna: 42

```

```

SELECT EXTRACT(MONTH FROM fecha_inicio) AS mes, COUNT(*) AS numero_alquileres
FROM AlquilerPLSQL
GROUP BY EXTRACT(MONTH FROM fecha_inicio)
ORDER BY numero_alquileres DESC
FETCH FIRST 1 ROWS ONLY;

```

	MES	NUMERO_ALQUILERES
1	12	100

```

SELECT TO_CHAR(fecha_inicio, 'D') AS dia_semana, COUNT(*) AS numero_alquileres
FROM AlquilerPLSQL
GROUP BY TO_CHAR(fecha_inicio, 'D')
ORDER BY numero_alquileres DESC
FETCH FIRST 1 ROWS ONLY;

```

	⚡ DIA_SEMANA	⚡ NUMERO_ALQUILERES
1	2	154

SELECT * FROM AlquilerPLSQL ORDER BY precio DESC FETCH FIRST 1 ROWS ONLY;

ORA-00904: "PRECIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
***Cause:**
***Action:**
Error en la línea: 1, columna: 38

SELECT * FROM AlquilerPLSQL ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY;

ORA-00904: "PRECIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
***Cause:**
***Action:**
Error en la línea: 2, columna: 38

SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%';

⚡	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

SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a WHERE precio < 10000;

ORA-00904: "PRECIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
***Cause:**
***Action:**
Error en la línea: 4, columna: 56

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN TO_DATE('2023-09-01', 'YYYY-MM-DD') AND TO_DATE('2023-09-30', 'YYYY-MM-DD');
```

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	594	435	45	18/09/23	29/09/23	594
2	596	397	466	16/09/23	25/09/23	596
3	610	155	112	10/09/23	06/10/23	610
4	640	462	985	14/09/23	08/10/23	640
5	671	357	270	06/09/23	22/09/23	671
6	675	929	158	17/09/23	12/10/23	675
7	680	249	590	18/09/23	20/09/23	680
8	681	709	462	28/09/23	11/10/23	681
9	711	979	759	16/09/23	08/10/23	711
10	712	97	697	03/09/23	16/09/23	712
11	722	183	38	11/09/23	19/09/23	722
12	756	549	615	14/09/23	19/09/23	756
12	750	27	58	07/09/23	21/09/23	750

```
SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente WHERE c.direccion LIKE '%Bogotá%';
```

```
ORA-00904: "A"."MODELO": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error en la línea: 6, columna: 27
```

```
SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto WHERE al.id_reserva = 1;
```

	MARCA	MODELO	ANO
1	later	too	2003

```
SELECT * FROM AlquilerPLSQL WHERE id_cliente IN (1, 2, 3);
```

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	861	3	171	20/12/22	10/01/23	861
2	292	3	891	19/08/23	13/09/23	292
3	384	2	141	19/12/22	03/01/23	384

```
SELECT * FROM AlquilerPLSQL WHERE id_auto IN (1, 2, 3);
```

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	723	911	2	19/04/23	01/05/23	723
2	148	922	3	13/11/22	25/11/22	148
3	329	668	3	03/07/23	07/07/23	329

```
SELECT * FROM AlquilerPLSQL WHERE id_sucursal IN (1, 2, 3);
```

```
ORA-00904: "ID_SUCURSAL": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error en la línea: 10, columna: 35
```

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN TO_DATE('2023-09-01',
'YYYY-MM-DD') AND TO_DATE('2023-09-30', 'YYYY-MM-DD') AND id_cliente IN (1, 2, 3);
```

ID_ALQ...	ID_CLI...	ID_AUTO	FECHA...	FECHA...	ID_RES...
-----------	-----------	---------	----------	----------	-----------

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN TO_DATE('2023-09-01',
'YYYY-MM-DD') AND TO_DATE('2023-09-30', 'YYYY-MM-DD') AND id_auto IN (1, 2, 3);
```

ID_ALQ...	ID_CLI...	ID_AUTO	FECHA...	FECHA...	ID_RES...
-----------	-----------	---------	----------	----------	-----------

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN TO_DATE('2023-09-01',
'YYYY-MM-DD') AND TO_DATE('2023-09-30', 'YYYY-MM-DD') AND id_sucursal IN (1, 2, 3);
```

ORA-00904: "ID_SUCURSAL": identificador no válido
00904. 00000 - "%s: invalid identifier"
***Cause:**
***Action:**
Error en la línea: 13, columna: 136

```
SELECT c.nombre, COUNT(*) AS numero_alquileres FROM ClientePLSQL c JOIN
AlquilerPLSQL a ON c.id_cliente = a.id_cliente GROUP BY c.nombre ORDER BY
numero_alquileres DESC FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE	NUMERO_ALQUILERES
1	John Greene	6

```
SELECT a.marca, a.modelo, COUNT(*) AS numero_alquileres FROM AutoPLSQL a JOIN
AlquilerPLSQL al ON a.id_auto = al.id_auto GROUP BY a.marca, a.modelo ORDER BY
numero_alquileres DESC FETCH FIRST 1 ROWS ONLY;
```

	MARCA	MODELO	NUMERO_ALQUILERES
1	end	hold	5

```
SELECT EXTRACT(MONTH FROM fecha_inicio) AS mes, COUNT(*) AS numero_alquileres
FROM AlquilerPLSQL GROUP BY EXTRACT(MONTH FROM fecha_inicio) ORDER BY
numero_alquileres DESC FETCH FIRST 1 ROWS ONLY;
```

	MES	NUMERO_ALQUILERES
1	12	100

```
SELECT TO_CHAR(fecha_inicio, 'D') AS dia_semana, COUNT(*) AS numero_alquileres FROM
AlquilerPLSQL GROUP BY TO_CHAR(fecha_inicio, 'D') ORDER BY numero_alquileres DESC
FETCH FIRST 1 ROWS ONLY;
```

	DIA_SEMANA	NUMERO_ALQUILERES
1	2	154

```
SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%' AND fecha_inicio BETWEEN
TO_DATE('2023-09-01', 'YYYY-MM-DD') AND TO_DATE('2023-09-30', 'YYYY-MM-DD');
```

```
ORA-00904: "FECHA_INICIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error en la línea: 18, columna: 59
```

```
SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a WHERE precio < 10000 AND
fecha_inicio BETWEEN TO_DATE('2023-09-01', 'YYYY-MM-DD') AND TO_DATE('2023-09-30',
'YYYY-MM-DD');
```

```
ORA-00904: "FECHA_INICIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error en la línea: 19, columna: 75
```

```
SELECT * FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;
```

```
ORA-00933: comando SQL no terminado correctamente
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error en la línea: 21, columna: 50
```

```
SELECT * FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1;
```

```
ORA-00933: comando SQL no terminado correctamente  
00933. 00000 - "SQL command not properly ended"  
*Cause:  
*Action:  
Error en la línea: 22, columna: 49
```

```
SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%' AND fecha_inicio BETWEEN  
'2023-09-01' AND '2023-09-30'; SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a  
WHERE precio < 10000 AND fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';
```

```
ORA-00904: "FECHA_INICIO": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
*Cause:  
*Action:  
Error en la línea: 2, columna: 76
```

Ciclo 3

```
-- Vista de clientes con alquileres en una sucursal específica  
CREATE OR REPLACE VIEW vista_clientes_alquilados_sucursal AS  
SELECT c.nombre, al.marca, al.modelo  
FROM ClientePLSQL c  
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente  
JOIN SucursalPLSQL s ON a.id_sucursal = s.id_sucursal  
WHERE s.nombre = 'Sucursal Central';
```

Error que empieza en la línea: 5 del comando :

```
CREATE OR REPLACE VIEW vista_clientes_alquilados_sucursal AS
SELECT c.nombre, al.marca, al.modelo
FROM ClientePLSQL c
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
JOIN SucursalPLSQL s ON a.id_sucursal = s.id_sucursal
WHERE s.nombre = 'Sucursal Central'
Informe de error -
ORA-00904: "A"."ID_SUCURSAL": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
```

-- Vista de autos alquilados por un cliente en una fecha específica

```
CREATE OR REPLACE VIEW vista_autos_alquilados_cliente_fecha AS
SELECT a.marca, a.modelo
FROM AutoPLSQL a
JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
WHERE al.id_cliente = 1 AND al.fecha_inicio = TO_DATE('2023-09-27', 'YYYY-MM-DD');
```

↕	MARCA	↕	MODELO
---	-------	---	--------

-- Vista de alquileres que duran más de 7 días

```
CREATE OR REPLACE VIEW vista_alquileres_mas_7dias AS
SELECT *
FROM AlquilerPLSQL
WHERE fecha_fin - fecha_inicio > 7;
```


	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131	208	01/10/22	01/11/23	584
2	585	394	543	19/04/23	11/05/23	585
3	586	774	345	19/07/23	30/07/23	586
4	587	81	250	26/03/23	22/04/23	587
5	589	999	584	01/05/23	21/05/23	589
6	590	796	711	26/05/23	17/10/23	590
7	592	223	847	27/11/22	14/12/22	592
8	593	987	961	21/08/23	20/09/23	593
9	594	435	45	18/09/23	29/09/23	594
10	595	81	435	06/05/23	31/05/23	595
11	596	397	466	16/09/23	25/09/23	596
12	597	36	473	06/02/23	01/03/23	597
13	598	776	669	09/11/22	23/11/22	598

-- Vista de clientes con más alquileres

```
CREATE OR REPLACE VIEW vista_clientes_mas_alquileres AS
SELECT c.nombre, COUNT(*) AS numero_alquileres
FROM ClientePLSQL c
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
GROUP BY c.nombre
ORDER BY numero_alquileres DESC;
```

	NOMBRE	NUMERO_ALQUILERES
1	John Greene	6
2	Michele Allen	4
3	Haley Carroll	4
4	Stephanie Young	4
5	Jill Sanford	4
6	Pamela Johnson	4
7	David Woods	4
8	Andrew Chang	4
9	Melinda Jackson	3
10	Kyle Stephens	3
11	Zachary George	3
12	Steven Velasquez	3
13	Ashley Farmer	2

-- Vista de clientes con más alquileres

```
CREATE OR REPLACE VIEW vista_clientes_mas_alquileres AS
```

```

SELECT c.nombre, COUNT(*) AS numero_alquileres
FROM ClientePLSQL c
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
GROUP BY c.nombre
ORDER BY numero_alquileres DESC;

```

	NOMBRE	NUMERO_ALQUILERES
1	John Greene	6
2	Michele Allen	4
3	Haley Carroll	4
4	Stephanie Young	4
5	Jill Sanford	4
6	Pamela Johnson	4
7	David Woods	4
8	Andrew Chang	4
9	Melinda Jackson	3
10	Kyle Stephens	3
11	Zachary George	3
12	Steven Velasquez	3
13	Ashley Farmer	2

```

-- Vista de autos con más alquileres
CREATE OR REPLACE VIEW vista_autos_mas_alquileres AS
SELECT a.marca, a.modelo, COUNT(*) AS numero_alquileres
FROM AutoPLSQL a
JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
GROUP BY a.marca, a.modelo
ORDER BY numero_alquileres DESC;

```

	⚡ MARCA	⚡ MODELO	⚡ NUMERO_ALQUILERES
1	end	hold	5
2	lot	level	5
3	significant	pretty	5
4	tough	year	4
5	analysis	reveal	4
6	just	morning	4
7	economic	approach	4
8	worry	behavior	4
9	similar	few	4
10	heavy	improve	4
11	same	fund	4
12	special	back	4
13	scientist	media	4

-- Vista de sucursales con más alquileres

```
CREATE OR REPLACE VIEW vista_sucursales_mas_alquileres AS
```

```
SELECT s.nombre, COUNT(*) AS numero_alquileres
```

```
FROM SucursalPLSQL s
```

```
JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal
```

```
GROUP BY s.nombre
```

```
ORDER BY numero_alquileres DESC;
```

Error que empieza en la línea: 45 del comando :

```
CREATE OR REPLACE VIEW vista_sucursales_mas_alquileres AS
```

```
SELECT s.nombre, COUNT(*) AS numero_alquileres
```

```
FROM SucursalPLSQL s
```

```
JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal
```

```
GROUP BY s.nombre
```

```
ORDER BY numero_alquileres DESC
```

```
Informe de error -
```

```
ORA-00904: "AL"."ID_SUCURSAL": identificador no válido
```

```
00904. 00000 - "%s: invalid identifier"
```

```
*Cause:
```

```
*Action:
```

-- Vista de meses con más alquileres

```
CREATE OR REPLACE VIEW vista_meses_mas_alquileres AS
SELECT EXTRACT(MONTH FROM fecha_inicio) AS mes, COUNT(*) AS numero_alquileres
FROM AlquilerPLSQL
GROUP BY EXTRACT(MONTH FROM fecha_inicio)
ORDER BY numero_alquileres DESC;
```

	↕ MES	↕ NUMERO_ALQUILERES
1	12	100
2	5	98
3	10	89
4	11	88
5	9	86
6	1	84
7	4	84
8	3	81
9	8	77
10	6	76
11	7	69
12	2	61

-- Vista de días de la semana con más alquileres

```
CREATE OR REPLACE VIEW vista_dias_semana_mas_alquileres AS
SELECT TO_CHAR(fecha_inicio, 'D') AS dia_semana, COUNT(*) AS numero_alquileres
FROM AlquilerPLSQL
GROUP BY TO_CHAR(fecha_inicio, 'D')
ORDER BY numero_alquileres DESC;
```

	↕ DIA_SEMANA	↕ NUMERO_ALQUILERES
1	2	154
2	4	146
3	5	144
4	1	143
5	7	140
6	6	135
7	3	131

```
-- Vista de alquileres más caros
CREATE OR REPLACE VIEW vista_alquileres_mas_caros AS
SELECT *
FROM AlquilerPLSQL
ORDER BY precio DESC;
```

Error que empieza en la línea: 9 del comando :

```
CREATE OR REPLACE VIEW vista_alquileres_mas_caros AS
SELECT *
FROM AlquilerPLSQL
ORDER BY precio DESC
Informe de error -
ORA-00904: "PRECIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
```

```
-- Vista de alquileres más baratos
CREATE OR REPLACE VIEW vista_alquileres_mas_baratos AS
SELECT *
FROM AlquilerPLSQL
ORDER BY precio ASC;
```

Error que empieza en la línea: 11 del comando :

```
CREATE OR REPLACE VIEW vista_alquileres_mas_baratos AS
SELECT *
FROM AlquilerPLSQL
ORDER BY precio ASC
Informe de error -
ORA-00904: "PRECIO": identificador no válido
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
```

Ciclo 4

```
CREATE TRIGGER trg_insert_auto
BEFORE INSERT ON AutoPLSQL
```

```

FOR EACH ROW
BEGIN
  -- Actualizar el número de autos disponibles
  UPDATE AutoPLSQL
    SET numero_disponibles = numero_disponibles + 1
  WHERE id_auto = NEW.id_auto;
END;

```

Error que empieza en la línea: 1 del comando :

```

CREATE TRIGGER trg_insert_auto
BEFORE INSERT ON AutoPLSQL
FOR EACH ROW
BEGIN
  -- Actualizar el número de autos disponibles
  UPDATE AutoPLSQL
    SET numero_disponibles = numero_disponibles + 1
  WHERE id_auto = NEW.id_auto;
END;
Informe de error -
ORA-04081: el disparador 'TRG_INSERT_AUTO' ya existe
ORA-04081. 00000 - "trigger '%s' already exists"
*Cause:      The TRIGGER name or type already exists.
*Action:     Use a different trigger name or drop the trigger which is of the
              same name.

```

```

CREATE TRIGGER trg_delete_auto
BEFORE DELETE ON AutoPLSQL
FOR EACH ROW
BEGIN
  -- Actualizar el número de autos disponibles
  UPDATE AutoPLSQL
    SET numero_disponibles = numero_disponibles - 1
  WHERE id_auto = OLD.id_auto;
END;

```

Error que empieza en la línea: 11 del comando :

```
CREATE TRIGGER trg_delete_auto
BEFORE DELETE ON AutoPLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de autos disponibles
    UPDATE AutoPLSQL
        SET numero_disponibles = numero_disponibles - 1
        WHERE id_auto = OLD.id_auto;
END;
Informe de error -
ORA-04081: el disparador 'TRG_DELETE_AUTO' ya existe
04081. 00000 - "trigger '%s' already exists"
*Cause:      The TRIGGER name or type already exists.
*Action:     Use a different trigger name or drop the trigger which is of the
              same name.
```

```
CREATE TRIGGER trg_update_auto
BEFORE UPDATE ON AutoPLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de autos disponibles
    IF NEW.numero_disponibles != OLD.numero_disponibles THEN
        UPDATE AutoPLSQL
            SET numero_disponibles = NEW.numero_disponibles
            WHERE id_auto = NEW.id_auto;
    END IF;
END;
```

Error que empieza en la línea: 21 del comando :

```
CREATE TRIGGER trg_update_auto
BEFORE UPDATE ON AutoPLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de autos disponibles
    IF NEW.numero_disponibles != OLD.numero_disponibles THEN
        UPDATE AutoPLSQL
            SET numero_disponibles = NEW.numero_disponibles
            WHERE id_auto = NEW.id_auto;
    END IF;
END;
Informe de error -
ORA-04081: el disparador 'TRG_UPDATE_AUTO' ya existe
04081. 00000 - "trigger '%s' already exists"
*Cause:      The TRIGGER name or type already exists.
*Action:     Use a different trigger name or drop the trigger which is of the
```

```
CREATE TRIGGER trg_insert_cliente
```

```

BEFORE INSERT ON ClientePLSQL
FOR EACH ROW
BEGIN
  -- Actualizar el número de clientes
  UPDATE ClientePLSQL
    SET numero_clientes = numero_clientes + 1;
END;

```

same name.

Error que empieza en la línea: 33 del comando :

```

CREATE TRIGGER trg_insert_cliente
BEFORE INSERT ON ClientePLSQL
FOR EACH ROW
BEGIN
  -- Actualizar el número de clientes
  UPDATE ClientePLSQL
    SET numero_clientes = numero_clientes + 1;
END;

```

Informe de error -

ORA-04081: el disparador 'TRG_INSERT_CLIENTE' ya existe

ORA-04081. 00000 - "trigger '%s' already exists"

*Cause: The TRIGGER name or type already exists.

*Action: Use a different trigger name or drop the trigger which is of the same name.

```

CREATE TRIGGER trg_delete_cliente
BEFORE DELETE ON ClientePLSQL
FOR EACH ROW
BEGIN
  -- Actualizar el número de clientes
  UPDATE ClientePLSQL
    SET numero_clientes = numero_clientes - 1;
END;

```


Error que empieza en la línea: 42 del comando :

```
CREATE TRIGGER trg_delete_cliente
BEFORE DELETE ON ClientePLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de clientes
    UPDATE ClientePLSQL
        SET numero_clientes = numero_clientes - 1;
END;
Informe de error -
ORA-04081: el disparador 'TRG_DELETE_CLIENTE' ya existe
04081. 00000 - "trigger '%s' already exists"
*Cause:      The TRIGGER name or type already exists.
*Action:     Use a different trigger name or drop the trigger which is of the
              same name.
```

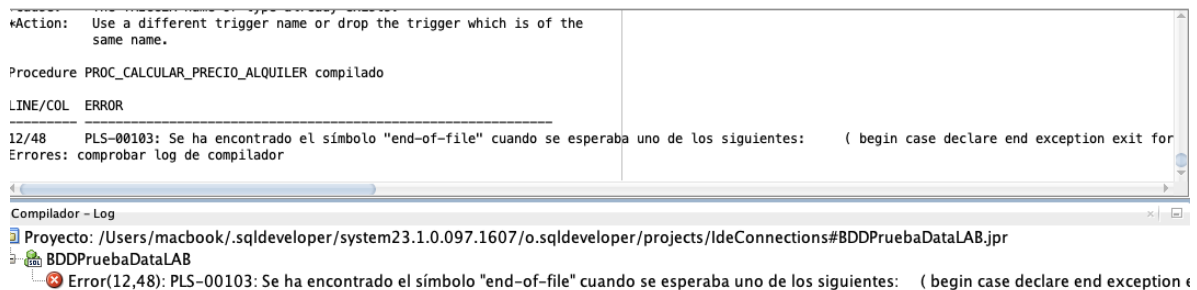
```
CREATE TRIGGER trg_update_cliente
BEFORE UPDATE ON ClientePLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de clientes
    IF NEW.numero_alquileres != OLD.numero_alquileres THEN
        UPDATE ClientePLSQL
            SET numero_alquileres = NEW.numero_alquileres
            WHERE id_cliente = NEW.id_cliente;
    END IF;
END;
```

```
CREATE TRIGGER trg_update_cliente
BEFORE UPDATE ON ClientePLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de clientes
    IF NEW.numero_alquileres != OLD.numero_alquileres THEN
        UPDATE ClientePLSQL
            SET numero_alquileres = NEW.numero_alquileres
            WHERE id_cliente = NEW.id_cliente;
    END IF;
END;
Informe de error -
ORA-04081: el disparador 'TRG_UPDATE_CLIENTE' ya existe
04081. 00000 - "trigger '%s' already exists"
*Cause:      The TRIGGER name or type already exists.
*Action:     Use a different trigger name or drop the trigger which is of the
              same name.
```

```

CREATE OR REPLACE PROCEDURE proc_calcular_precio_alquiler(
  id_alquiler IN INT,
  id_auto IN INT,
  fecha_inicio IN DATE,
  fecha_fin IN DATE
)
IS
  precio_base NUMERIC(10, 2);
  dias_alquiler INT;
BEGIN
  SELECT precio INTO precio_base FROM AutoPLSQL WHERE id_auto = id_auto;
  dias_alquiler := fecha_fin - fecha_inicio + 1;
END;

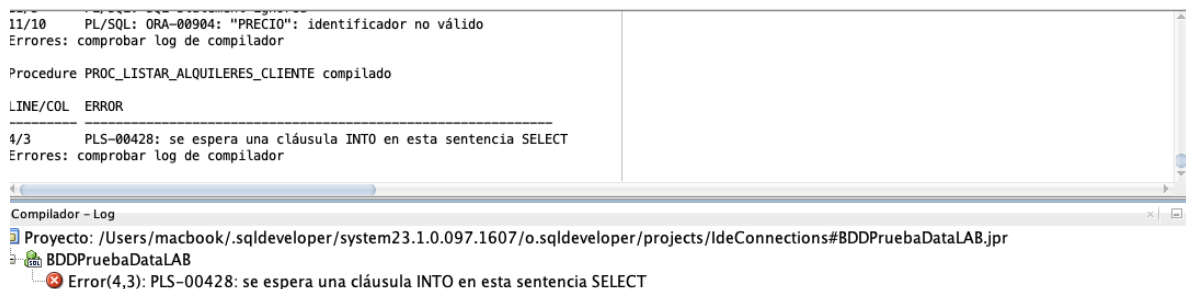
```



```

CREATE OR REPLACE PROCEDURE proc_listar_alquileres_cliente(id_cliente IN INT)
IS
BEGIN
  SELECT * FROM AlquilerPLSQL WHERE id_cliente = id_cliente;
END;

```



```

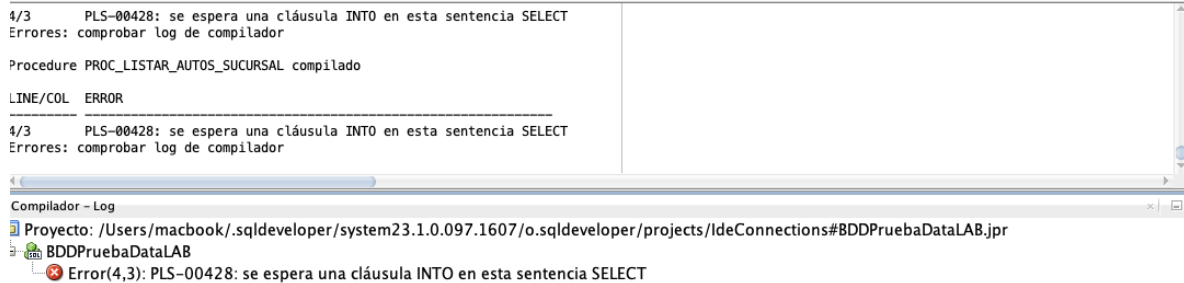
CREATE OR REPLACE PROCEDURE proc_listar_autos_sucursal(id_sucursal IN INT)

```

```

IS
BEGIN
  SELECT * FROM AutoPLSQL WHERE id_sucursal = id_sucursal;
END;

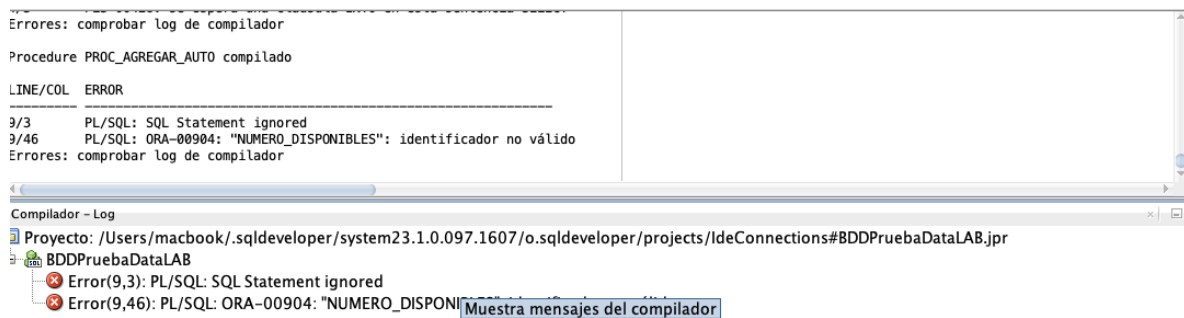
```



```

CREATE OR REPLACE PROCEDURE proc_agregar_auto(
  marca IN VARCHAR2,
  modelo IN VARCHAR2,
  ano IN INT,
  numero_disponibles IN INT
)
IS
BEGIN
  INSERT INTO AutoPLSQL (marca, modelo, ano, numero_disponibles)
  VALUES (marca, modelo, ano, numero_disponibles);
END;

```



```

CREATE OR REPLACE PROCEDURE proc_eliminar_auto(id_auto IN INT)
IS
BEGIN
  DELETE FROM AutoPLSQL WHERE id_auto = id_auto;

```

END;

Procedure PROC_ELIMINAR_AUTO compilado