

Miguel Santiago Feles Gómez
Ciclo 1 y Ciclo 2
Ejercicios Cuaderno PLSQL
Materia: Bases De Datos
Docente: Wilmer López
Universidad Externado de Colombia

Primer Ciclo

- Mostrar todos los clientes en la tabla "Cliente".
SELECT * FROM clienteplsql;

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	+1-287-434-2337
14	656 Samantha Cooper	28123 Justin Wall, East Thomas, Massachusetts 25128	+1-415-436-6560

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

- Mostrar todos los alquileres en la tabla "Alquiler".
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	397	466 16/09/23	25/09/23	596
14	597	36	473 06/02/23	01/03/23	597

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

SELECT * FROM sucursalpsql;

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
14	597 Abbott Ltd	Sparksfurt	Chile

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

SELECT * FROM reservapsql;

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
14	597 Abbott Ltd	Sparksfurt	Chile

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

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

7. Mostrar los autos de marca "Toyota".

SELECT * FROM autoplsq 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 > '05/08/2023';

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	593	987	961 21/08/23	20/09/23	593
2	594	435	45 18/09/23	29/09/23	594
3	596	397	466 16/09/23	25/09/23	596
4	602	830	198 11/08/23	29/08/23	602
5	610	155	112 10/09/23	06/10/23	610
6	632	644	102 11/08/23	03/09/23	632
7	640	462	985 14/09/23	08/10/23	640
8	665	42	311 30/08/23	18/09/23	665
9	671	357	270 06/09/23	22/09/23	671
10	675	929	158 17/09/23	12/10/23	675
11	680	249	590 18/09/23	20/09/23	680
12	681	709	462 28/09/23	11/10/23	681
13	687	989	985 22/08/23	09/09/23	687
14	694	783	585 16/08/23	05/09/23	694

9. Mostrar las sucursales ubicadas en "Madrid".

```
SELECT * FROM sucursalplsql WHERE CIUDAD = 'Madrid';
```

```
SELECT * FROM sucursalplsql WHERE CIUDAD like 'Madrid%';
```

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

10. Mostrar las reservas realizadas por un cliente específico.

```
SELECT * FROM reservaplsq WHERE id_cliente = 17;
```

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	104	17	195 18/11/22
2	466	17	616 11/12/22
3	2350	17	19 03/05/23
4	1407	17	13 23/12/22
5	1762	17	42 10/02/23
6	3338	17	99 19/11/22
7	4385	17	6 23/07/23
8	3422	17	95 05/10/22
9	2350	17	19 03/05/23
10	1407	17	13 23/12/22
11	1762	17	42 10/02/23

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

```
SELECT alq.id_alquiler, alq.id_cliente, c.nombre, alq.id_auto, aut.marca,
```

```
alq.fecha_inicio, alq.fecha_fin, alq.id_reserva
```

```
FROM alquilerplsql alq
```

```
INNER JOIN clienteplsql c ON c.id_cliente = alq.id_cliente
```

```
INNER JOIN autoplsq aut ON aut.id_auto = alq.id_auto;
```

ID_ALQUILER	ID_CLIENTE	NOMBRE	ID_AUTO	MARCA	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131 Alexander Cunningham	208	lot	01/10/22	30/10/22	584
2	585	394 Christine Snyder	543	economic	19/04/23	11/05/23	585
3	586	774 Tara Johnson	345	pressure	19/07/23	30/07/23	586
4	588	16 Tanya Blake	371	account	23/02/23	25/02/23	588
5	589	999 Jamie Davenport	584	gas	01/05/23	21/05/23	589
6	591	928 Matthew Walter	812	cell	08/05/23	15/05/23	591
7	592	223 Jennifer Hansen	847	operation	27/11/22	14/12/22	592
8	593	987 Todd Yoder	961	image	21/08/23	20/09/23	593
9	596	397 Kevin Hoffman	466	later	16/09/23	25/09/23	596
10	598	726 Cassandra Davis	669	strong	09/11/22	23/11/22	598
11	599	999 Jamie Davenport	715	stock	07/10/22	25/10/22	599
12	600	414 Timothy Dixon	809	way	29/12/22	10/01/23	600
13	602	830 David Wallace	198	memory	11/08/23	29/08/23	602
14	604	214 Elizabeth McDowell	63	health	20/12/22	24/12/22	604

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

```
SELECT re.id_sucursal, su.nombre, c.id_cliente, c.nombre, direccion, telefono
```

```
FROM clienteplsql c
```

```
INNER JOIN reservaplsq re ON c.id_cliente = re.id_cliente
```

```
INNER JOIN sucursalplsql su ON su.id_sucursal = re.id_sucursal
```

WHERE re.id_sucursal = '10';

ID_SUCURSAL	NOMBRE	ID_CLIENTE	NOMBRE_1	DIRECCION	TELEFONO
1	10 Sucursal 10	626	Brenda Cervantes	075 Brown Green Suite 198, Port Matthewberg, Utah 95139	483.526.6042
2	10 Sucursal 10	626	Brenda Cervantes	075 Brown Green Suite 198, Port Matthewberg, Utah 95139	483.526.6042
3	10 Sucursal 10	670	Joseph Mitchell	7731 Jennifer Tunnel Suite 505, Rodrigueztown, Utah 09703	827.599.4267
4	10 Sucursal 10	670	Joseph Mitchell	7731 Jennifer Tunnel Suite 505, Rodrigueztown, Utah 09703	827.599.4267
5	10 Sucursal 10	686	Blake Mendez	661 Simon Plaza, New Hannahchester, North Dakota 00788	839-465-6600
6	10 Sucursal 10	894	Rachel Russell	250 Colin Ranch, North Tiffanyberg, Connecticut 04535	(799)245-0764
7	10 Sucursal 10	894	Rachel Russell	250 Colin Ranch, North Tiffanyberg, Connecticut 04535	(799)245-0764
8	10 Sucursal 10	904	Michael Wyatt	112 Donald Shoals, Georgeshire, Tennessee 49235	4687607945
9	10 Sucursal 10	904	Michael Wyatt	112 Donald Shoals, Georgeshire, Tennessee 49235	4687607945
10	10 Sucursal 10	915	Bob Peterson	791 Mikayla Oval, Blairville, Minnesota 23895	+1-537-516-4981
11	10 Sucursal 10	915	Bob Peterson	791 Mikayla Oval, Blairville, Minnesota 23895	+1-537-516-4981
12	10 Sucursal 10	964	Haley Carroll	93774 Hunter Greens Suite 645, Biancacher, Utah 82178	(525)711-8847
13	10 Sucursal 10	964	Haley Carroll	93774 Hunter Greens Suite 645, Biancacher, Utah 82178	(525)711-8847
14	10 Sucursal 10	54	John Greene	1001 Mckenzie Brooks Suite 008, Ralphstad, Massachusetts 26257	+1-285-258-3663

13. Mostrar los autos que han sido alquilados junto con los nombres de los clientes.
 SELECT c.NOMBRE AS NOMBRE, a.MARCA AS MARCA, a.MODELO AS MODELO,
 a.AÑO AS AÑO
 FROM alquilerplsql a
 JOIN clienteplsql c ON a.ID_CLIENTE = c.ID_CLIENTE
 JOIN autoplsq a ON a.ID_AUTO = a.ID_AUTO;

NOMBRE	MARCA	MODELO	AÑO
1 Robert Moran	Honda	Civic	2025
2 Steven Velasquez	Nissan	Versa	2023
3 Gina Morgan	Kia	Rio	2023
4 Stefanie Sullivan	Chevrolet	Onix	2023
5 Emily Santana	Fiat	Argo	2023
6 Ernest Mora	Volkswagen	Gol	2023
7 Megan West	Renault	Kwid	2023
8 Melinda Jackson	Renault	Kwid	2023
9 Kimberly Johnson	BMW	Serie 3	2023
10 Stephanie Clark	Genesis	G80	2023
11 Jennifer Gamble	Mazda	6	2023
12 Nicholas Martin	Mazda	6	2023
13 Michael Koch	Hyundai	Sonata	2023
14 Leslie McCoy	Kia	KS	2023

14. Mostrar los detalles de las reservas con los nombres de los clientes y las ciudades de las sucursales.
 SELECT r.*, c.NOMBRE, s.CIUDAD
 FROM reservaplsq r
 JOIN clienteplsql c ON r.ID_CLIENTE = c.ID_CLIENTE
 JOIN sucursalplsql s ON r.ID_SUCURSAL = s.ID_SUCURSAL;

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA	NOMBRE	CIUDAD
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 Codyview
14	926	732	26 23/06/23	Jason Sawyer	Port Lisaborough
15	933	748	569 08/07/23	Carol Morales	Powellview
16	938	499	294 18/01/23	Kayla Ryan	West Hollyport

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

```
SELECT c.* FROM clienteplsql c
LEFT JOIN reservaplsql r ON r.id_cliente = c.id_cliente WHERE r.id_reserva is null;
```

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

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

```
SELECT marca,count(marca) AS total
FROM autoplsql GROUP BY marca;
```

MARCA	TOTAL
1 Volkswagen	2
2 Lexus	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
14 draw	2

17. Calcular la duración promedio de los alquileres.

```
SELECT round(avg(fecha_fin-fecha_inicio),1) AS tiempoPromedio
FROM alquilerplsql;
```

TIEMPO PROMEDIO
1 16

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

```
SELECT nombre, count(id_reserva) as total
FROM sucursalplsql su
INNER JOIN reservaplsql r ON r.id_sucursal = su.id_sucursal GROUP BY nombre;
```

NOMBRE	TOTAL
1 Patrick-Rocha	2
2 Johnson, Martin and Gonzalez	1
3 Rodriguez, Scott and Ford	81
4 Phillips-Shepard	2
5 Reyes, Robinson and Davis	1
6 Barnes-McLaughlin	1
7 Garcia-Lee	1
8 Thomas-Wolfe	4
9 Lozano Inc	1
10 Garcia-Wilson	82
11 Sanford, Davis and Krueger	71
12 Dixon, Solomon and Tapia	79
13 Wiley, Trevino and Rowland	83
14 Sucursal 12	79

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

```

WITH ClienteAlquileres AS (
SELECT
  c.id_cliente,
  c.nombre,
  c.direccion,
  c.telefono,
  COUNT(al.id_alquiler) AS cantidad_alquileres,
  RANK() OVER (ORDER BY COUNT(al.id_alquiler) DESC) AS ranking
FROM clienteplsql c
  INNER JOIN alquilerplsql al ON al.id_cliente = c.id_cliente
GROUP BY
  c.id_cliente, c.nombre, c.direccion, c.telefono)
SELECT
  id_cliente,nombre,direccion,telefono,cantidad_alquileres
FROM ClienteAlquileres
WHERE ranking = 1;

```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO	CANTIDAD_ALQUILERES
1	377 Andrew Chang	31682 Murphy Lakes, Wilsonport, South Dakota 41846	(240)262-6177	4
2	442 Stephanie Young	1390 Corey Plains, Leefort, Pennsylvania 76387	527-673-3176	4
3	420 Pamela Johnson	8694 Wheeler Mills Suite 321, Lake Melissahaven, Arizona 78942	535.761.9239	4
4	693 David Woods	1385 Vasquez Trace, Noahberg, Wisconsin 96993	853.850.0487	4
5	964 Haley Carroll	93774 Hunter Greens Suite 645, Biancacher, Utah 82178	(525)711-8847	4
6	126 Michele Allen	481 Osborne Square, Port Jessica, Colorado 87359	573.570.4517	4
7	601 Jill Sanford	409 Gross Meadow Suite 981, Lauraborough, Louisiana 64286	(419)465-7221	4

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

```

SELECT round(avg(ano)) AS promedio FROM autoplsql;

```

	PROMEDIO
1	2006

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

```

SELECT c.id_cliente,c.nombre, direccion, telefono
FROM clienteplsql c
  LEFT JOIN reservaplsql r ON c.id_cliente = r.id_cliente
WHERE r.id_cliente IS NOT NULL;

```

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 Kennethnton, Oklahoma 07153	477-777-2081
14	493 Sarah Duffy	66772 Mack Divide, West Jeffreyemouth, West Virginia 04047	2319407802

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

```
SELECT * FROM autoplsq
WHERE id_auto NOT IN (SELECT id_auto FROM alquilerpsql);
```

ID_AUTO	MARCA	MODELO	ANO
1	10 Ford	Ka	2023
2	19 Subaru	Legacy	2023
3	22 Honda	CR-V	2023
4	23 Nissan	Rogue	2023
5	24 Mazda	CX-5	2023
6	26 Hyundai	Tucson	2023
7	28 Volkswagen	Tiguan	2023
8	29 Chevrolet	Equinox	2023
9	30 Ford	Escape	2023
10	851 nation	perhaps	2009
11	853 Mrs	need	2013
12	854 still	billion	2021

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

```
SELECT c.NOMBRE FROM clientepsq c
INNER JOIN alquilerpsql a ON c.ID_CLIENTE = a.ID_CLIENTE
GROUP BY c.NOMBRE
HAVING COUNT(a.ID_AUTO) > 1;
```

NOMBRE
1 Christine Snyder
2 Jennifer Gamble
3 Mitchell Bennett
4 Jodi Proctor
5 Amanda Foster
6 Joseph Mitchell
7 Melinda Jackson
8 Tyler Avila
9 James Brown
10 Michele Allen
11 Richard Evans
12 April Lewis

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

```
SELECT DISTINCT * FROM clientepsq c
INNER JOIN alquilerpsql al ON c.id_cliente = al.id_cliente
INNER JOIN reservapsql r ON r.id_reserva = al.id_reserva
INNER JOIN sucursalpsql su ON su.id_sucursal = r.id_sucursal
WHERE ciudad = 'Bogotá';
```

ID_CLIENTE_1	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA	ID_RESERVA_1	ID_CLIENTE_2	ID_SUCURSAL	FECHA_RESERVA	ID_SUCURSAL_1	NOMBRE_1	CIUDAD	PAIS
467	251 20/08/23	26/08/23		395	395	802		121/12/22		1 Sucursal 1	Bogotá	Colombia
297	263 23/01/23	21/02/23		642	642	888		126/03/23		1 Sucursal 1	Bogotá	Colombia
631	150 16/03/23	03/04/23		356	356	182		103/09/23		1 Sucursal 1	Bogotá	Colombia
892	87 16/11/22	05/12/22		209	209	809		111/06/23		1 Sucursal 1	Bogotá	Colombia
111	137 26/07/23	02/08/23		131	131	384		106/04/23		1 Sucursal 1	Bogotá	Colombia

11 rows deleted.

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

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

```
SELECT DISTINCT au.*, su.* FROM autoplsq au
INNER JOIN alquilerpls al ON (au.id_auto = al.id_auto)
INNER JOIN reservapls r ON (r.id_reserva = al.id_reserva AND al.id_cliente =
r.id_cliente)
INNER JOIN sucursalpls su ON (su.id_sucursal = r.id_sucursal)
WHERE su.id_sucursal = 210;
```

ID_AUTO	MARCA	MODELO	ANO	ID_SUCURSAL	NOMBRE	CIUDAD	PAIS
1	42	high	adult	2012	210 Wilson PLC	Lake Angelaton	Kyrgyz Republic

26. Actualizar la dirección de un cliente específico.

```
UPDATE CLIENTEPLSQL SET DIRECCION = 'Carrera 5 #90 12 Sincelejo' WHERE
ID_CLIENTE = '744';
```

1 row updated.

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	744 Jeffrey Butler	Carrera 5 #90 12 Sincelejo	(407)961-3681

27. Eliminar un auto de la tabla "Auto". Primero, se visualiza el registro de un auto en específico

```
DELETE FROM autoplsq WHERE id_auto = 7;
```

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

1 row deleted.

ID_AUTO	MARCA	MODELO	ANO
2	Honda	Civic	2025
3	Nissan	Versa	2023
4	Kia	Rio	2023
6	Chevrolet	Onix	2020
9	Renault	Kwid	2023

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

```
UPDATE ALQUILERPLSQL
SET FECHA_FIN = '05/11/2023'
WHERE ID_RESERVA = 521;
```

1 row updated.

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	521	266	462 05/09/23	05/11/23	521

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

```
SELECT * FROM reservaplsq;
DELETE FROM reservaplsq WHERE id_cliente = 863;
```

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
863	780	739	16/05/23
864	737	729	04/06/23
865	700	982	10/12/22
866	197	610	29/06/23

30. Actualizar el año de un auto en la tabla "Auto".
 SELECT * FROM AUTOPLSQL WHERE ID_AUTO = 521;

1 row updated.

	ID_AUTO	MARCA	MODELO	ANO
1	521	sister	air	2020

Segundo Ciclo

1. 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, Daniellshire, 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	+1-287-434-2337
14	656 Samantha Cooper	28123 Justin Wall, East Thomas, Massachusetts 25128	+1-415-436-6560

2. SELECT * FROM AutoPLSQL;

ID_AUTO	MARCA	MODELO	ANO
1	3 Nissan	Versa	2023
2	4 Kia	Rio	2023
3	6 Chevrolet	Onix	2020
4	7 Fiat	Argo	2023
5	9 Renault	Kwid	2023
6	10 Ford	Ka	2023
7	11 BMW	Serie 3	2020
8	16 Genesis	G80	2023
9	17 Volvo	S60	2023
10	18 Mazda	6	2024
11	19 Subaru	Legacy	2023
12	20 Hyundai	Sonata	2023
13	21 Kia	K5	2023
14	22 Honda	CR-V	2023

3. 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	17/06/23	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	397	466 16/09/23	25/09/23	596
14	597	36	473 06/02/23	01/03/23	597

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

```
ORA-00904: "A"."MODELO": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error at Line: 3 Column: 27
```

Para corregirlo se implementó
 SELECT cli.nombre AS NombreCliente,
 aut.marca AS MarcaAuto,
 aut.modelo AS ModeloAuto
 FROM ClientePLSQL cli
 INNER JOIN AlquilerPLSQL alq ON cli.id_cliente = alq.id_cliente
 INNER JOIN AutoPLSQL aut ON alq.id_auto = aut.id_auto;

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

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

	MARCA	MODELO	ANO
1	Nissan	Versa	2023
2	Nissan	Versa	2023
3	Kia	Rio	2023
4	Kia	Rio	2023
5	Chevrolet	Onix	2020
6	Chevrolet	Onix	2020
7	Fiat	Argo	2023
8	Fiat	Argo	2023
9	Fiat	Argo	2023
10	Renault	Kwid	2023
11	Renault	Kwid	2023
12	BMW	Serie 3	2020
13	Genesis	G80	2023
14	Volvo	S60	2023

6. SELECT * FROM AlquilerPLSQL WHERE id_cliente = 1;

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

7. SELECT * FROM AlquilerPLSQL WHERE id_auto = 1;

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

8. SELECT * FROM AlquilerPLSQL WHERE id_sucursal = 1;

ORA-00904: "ID_SUCURSAL": invalid identifier
00904. 00000 - "%s: invalid identifier"

*Cause:

*Action:

Error at Line: 1 Column: 35

SELECT DISTINCT alquilerplsql.* FROM alquilerplsql JOIN reservaplsq ON
alquilerplsql.id_reserva = reservaplsq.id_reserva WHERE id_sucursal = 1;

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	218	327	68 10/05/23	12/05/23	218
2	395	467	251 20/08/23	26/08/23	395
3	131	111	137 26/07/23	02/08/23	131
4	642	297	263 23/01/23	21/02/23	642
5	686	953	912 22/07/23	25/07/23	686
6	429	451	319 14/05/23	16/05/23	429
7	209	892	87 16/11/22	05/12/22	209
8	356	631	150 16/03/23	03/04/23	356

9. SELECT * FROM AlquilerPLSQL WHERE fecha_inicio = '2023-09-27';

ORA-01861: literal does not match format string
01861. 00000 - "literal does not match format string"

*Cause: Literals in the input must be the same length as literals in the format string (with the exception of leading whitespace). If the "FX" modifier has been toggled on, the literal must match exactly, with no extra whitespace.

*Action: Correct the format string to match the literal.

SELECT * FROM AlquilerPLSQL WHERE fecha_inicio = '27/09/2023';

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

10. SELECT COUNT(*) FROM AlquilerPLSQL;

COUNT(*)
1 993

11. SELECT c.nombre 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';

ORA-00904: "A"."ID_SUCURSAL": invalid identifier
00904. 00000 - "%s: invalid identifier"

*Cause:

*Action:

Error at Line: 1 Column: 113

Se implementó para mejorar

SELECT DISTINCT c.*

FROM ClientePLSQL c

JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente

JOIN ReservaPLSQL r ON a.id_reserva = r.id_reserva AND a.id_cliente = r.id_cliente

JOIN SucursalPLSQL s ON r.id_sucursal = s.id_sucursal

WHERE s.nombre = 'Sucursal Central';

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

12. SELECT a.marca, a.modelo FROM AutoPLSQL a

MARCA	MODELO
Nissan	Versa
Kia	Rio
Chevrolet	Onix
Fiat	Argo
Renault	Kwid
Ford	Ka
BMW	Serie 3
Genesis	G80
Volvo	S60

13. 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 = '2023-09-27';

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

14. SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio > 7;

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESE...
584	131	208	01/10/22	01/11/23	584
585	394	543	19/04/23	11/05/23	585
586	774	345	19/07/23	30/07/23	586
587	81	250	26/03/23	22/04/23	587
589	999	584	01/05/23	21/05/23	589
590	796	711	26/05/23	17/06/23	590
592	223	847	27/11/22	14/12/22	592
593	987	961	21/08/23	20/09/23	593

15. 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 LIMIT 1;

ORA-00933: SQL command not properly ended
 00933. 00000 - "SQL command not properly ended"
 *Cause:
 *Action:
 Error at Line: 1 Column: 170

Para solucionar se implementó
 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 COUNT(*) DESC
 FETCH FIRST 1 ROWS ONLY;

	NOMBRE	NUMERO_ALQUILERES
1	John Greene	6

16. 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 LIMIT 1;

ORA-00933: SQL command not properly ended
 00933. 00000 - "SQL command not properly ended"
 *Cause:
 *Action:
 Error at Line: 1 Column: 181

SELECT marca, modelo, numero_alquileres
 FROM (
 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 COUNT(*) DESC
)
 WHERE ROWNUM <= 1;

	MARCA	MODELO	NUMERO_ALQUILERES
1	end	hold	5

17. 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 LIMIT 1;

ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error at Line: 1 Column: 175

18. 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 LIMIT 1;

ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error at Line: 2 Column: 75

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

	MES	NUMERO_ALQUILERES
1	12	100

19. SELECT EXTRACT(DAYOFWEEK FROM fecha_inicio) AS dia_semana, COUNT(*) AS numero_alquileres FROM AlquilerPLSQL GROUP BY EXTRACT(DAYOFWEEK FROM fecha_inicio) ORDER BY numero_alquileres DESC LIMIT 1;

ORA-00907: missing right parenthesis
00907. 00000 - "missing right parenthesis"
*Cause:
*Action:
Error at Line: 1 Column: 26

SELECT dia_semana AS DIA, numero_alquileres AS Alquileres
FROM (
SELECT TRIM(TO_CHAR(fecha_inicio, 'Day')) AS dia_semana, COUNT(*) AS numero_alquileres
FROM AlquilerPLSQL
GROUP BY TRIM(TO_CHAR(fecha_inicio, 'Day'))
ORDER BY COUNT(*) DESC
)
WHERE ROWNUM <= 1

	DIA	ALQUILERES
1	Monday	154

20. SELECT * FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;

ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error at Line: 1 Column: 50

21. SELECT * FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1;

ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error at Line: 1 Column: 49