

EJERCICIOS PRIMER CICLO (1-30):

1. Consultas Básicas:

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

```
SELECT *  
FROM CLIENTEPLSQL
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	625 Ricky Owens	25136 Kelly Oval, Cantrellbury, Florida 72984	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 Sumner 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
15	669 Lisa Gray	56386 Richard Mountain, East Andrew, Maryland 86073	6352962563
16	670 Joseph Mitchell	7731 Jennifer Tunnel Suite 505, Rodrigueztown, Utah 09703	827.599.4267
17	673 Hannah Bishop	93988 Stephanie Park, Donaldtown, Minnesota 75086	794.622.4305

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

```
SELECT *  
FROM AUTOPLSQL
```

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2023
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
14	16 Genesis	G80	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	583	844	293 27-AUG-23	20-SEP-23	583
2	584	131	208 01-OCT-22	30-OCT-22	584
3	585	394	543 19-APR-23	11-MAY-23	585
4	586	774	345 19-JUL-23	30-JUL-23	586
5	587	81	250 26-MAR-23	22-APR-23	587
6	588	16	371 23-FEB-23	25-FEB-23	588
7	589	999	584 01-MAY-23	21-MAY-23	589
8	590	796	711 26-MAY-23	17-JUN-23	590
9	591	928	812 08-MAY-23	15-MAY-23	591
10	592	223	847 27-NOV-22	14-DEC-22	592
11	593	987	961 21-AUG-23	20-SEP-23	593
12	594	435	45 18-SEP-23	29-SEP-23	594
13	595	81	435 06-MAY-23	31-MAY-23	595
14	596	397	466 16-SEP-23	25-SEP-23	596
15	597	36	473 06-FEB-23	01-MAR-23	597
16	598	726	669 09-NOV-22	23-NOV-22	598
17	599	999	715 07-OCT-22	25-OCT-22	599
18	600	414	809 29-DEC-22	10-JAN-23	600
19	601	470	463 03-OCT-22	15-OCT-22	601

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

```
SELECT *
FROM SUCURSALPLSQL
```

ID_SUCURSAL	NOMBRE	CIUDAD	PAIS
1	1 Sucursal Bogotá	Bogotá	Colombia
2	2 Sucursal Medellín	Medellín	Colombia
3	3 Sucursal Cali	Cali	Colombia
4	4 Sucursal Barranquilla	Barranquilla	Colombia
5	5 Sucursal Cartagena	Cartagena	Colombia
6	6 Sucursal Villavicencio	Villavicencio	Colombia
7	7 Sucursal Yopal	Yopal	Colombia
8	8 Sucursal Pasto	Pasto	Colombia
9	9 Sucursal Bucaramanga	Bucaramanga	Colombia
10	10 Sucursal Manizales	Manizales	Colombia
11	11 Sucursal São Paulo	São Paulo	Brasil
12	12 Sucursal Ciudad de México	Ciudad de México	México
13	13 Sucursal Lima	Lima	Perú

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

```
SELECT *
FROM RESERVAPLSQL
```

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	862	79	239 05-FEB-23
2	863	780	739 16-MAY-23
3	864	737	729 04-JUN-23
4	865	700	982 10-DEC-22
5	866	197	610 29-JUN-23
6	867	594	755 17-JUN-23
7	868	808	185 07-MAR-23
8	869	315	394 19-MAR-23
9	870	314	627 03-OCT-22
10	871	408	687 23-DEC-22
11	872	128	821 22-SEP-23
12	873	858	380 16-DEC-22
13	874	193	613 15-JAN-23
14	875	256	954 11-AUG-23
15	876	437	903 21-AUG-23
16	877	716	325 14-JUN-23
17	878	708	665 16-AUG-23
18	879	573	287 13-JUN-23

2. Filtros y Ordenamiento:

- Mostrar los clientes que se llaman "Juan".

```
SELECT *
FROM CLIENTEPLSQL WHERE NOMBRE LIKE '%Juan%';
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
675	Juan Wilson	13908 Chase Estates, South Morganmouth, Wisconsin 86282	+1-200-759-2343
483	Juan Baker	15104 Eric Mount, Millermouth, Arizona 23327	9963132346

- Mostrar los autos de marca "Toyota".

```
SELECT *
FROM AUTOPLSQL WHERE MARCA LIKE '%Toyota%';
```

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

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

```
SELECT *
FROM ALQUILERPLSQL WHERE fecha_inicio>'05-APR-23'
```

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
583	844	293	27-AUG-23	20-SEP-23	583
585	394	543	19-APR-23	11-MAY-23	585
586	774	345	19-JUL-23	30-JUL-23	586
589	999	584	01-MAY-23	21-MAY-23	589
590	796	711	26-MAY-23	17-JUN-23	590
591	928	812	08-MAY-23	15-MAY-23	591
593	987	961	21-AUG-23	20-SEP-23	593
594	435	45	18-SEP-23	29-SEP-23	594
595	81	435	06-MAY-23	31-MAY-23	595
596	397	466	16-SEP-23	25-SEP-23	596
602	830	198	11-AUG-23	29-AUG-23	602
605	171	251	14-JUN-23	28-JUN-23	605
608	876	328	25-JUL-23	23-AUG-23	608
609	552	167	27-APR-23	02-MAY-23	609
610	155	112	10-SEP-23	06-OCT-23	610
614	138	766	08-MAY-23	29-MAY-23	614
615	668	539	09-JUN-23	17-JUN-23	615
616	478	50	08-APR-23	11-APR-23	616

- Mostrar las sucursales ubicadas en "Madrid".

```
SELECT *
FROM SUCURSALPLSQL WHERE CIUDAD LIKE '%Madrid%';
```

ID_SUCURSAL	NOMBRE	CIUDAD	PAIS
16	Sucursal Madrid	Madrid	España

- Mostrar las reservas realizadas por un cliente específico.

```
SELECT *
FROM RESERVAPLSQL WHERE ID_CLIENTE =844;
```

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
113	844	55	19-JUN-23
583	844	924	27-AUG-23
2714	844	14	15-MAR-23
1813	844	78	03-MAY-23
113	844	55	19-JUN-23
1813	844	78	03-MAY-23
2714	844	14	15-MAR-23

3. Join y Relaciones:

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

```

SELECT T1.*, T2.NOMBRE, T3.MARCA
FROM ALQUILERPLSQL T1
LEFT JOIN CLIENTEPLSQL T2 ON T2.ID_CLIENTE= T1.ID_CLIENTE
LEFT JOIN AUTOPLSQL T3 ON T3.ID_AUTO= T1.ID_AUTO;

```

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA	NOMBRE	MARCA
646	473	584	01-MAY-23	31-MAY-23	646	Nicole Stokes	gas
647	655	894	02-DEC-22	16-DEC-22	647	(null)	do
648	300	18	18-JAN-23	24-JAN-23	648	Jennifer Gamble	Mazda
649	474	719	17-MAY-23	22-MAY-23	649	(null)	item
650	268	272	28-JUN-23	01-JUL-23	650	Michael Ortega	next
651	144	655	15-DEC-22	20-DEC-22	651	Mark Hernandez	current
652	481	351	27-JUL-23	13-AUG-23	652	Martin Vasquez	include
653	279	934	17-JUL-23	03-AUG-23	653	(null)	star
654	91	178	08-JUL-23	12-JUL-23	654	(null)	behind
655	587	571	18-JUN-23	19-JUN-23	655	(null)	behavior
656	489	805	31-OCT-22	23-NOV-22	656	Lisa Wilson	common
657	581	364	10-OCT-22	16-OCT-22	657	Samuel Rowe	still
658	149	463	03-MAY-23	02-JUN-23	658	Scott Harris	without
659	192	488	05-FEB-23	21-FEB-23	659	Karen Black	doctor
660	314	437	08-DEC-22	11-DEC-22	660	Brian Smith	scientist
661	365	950	14-APR-23	27-APR-23	661	Jessica Gilbert	science
662	642	125	31-OCT-22	10-NOV-22	662	(null)	fly
663	943	116	27-MAY-23	08-JUN-23	663	(null)	election

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

```

SELECT T1.FECHA_RESERVA, T2.NOMBRE, T3.CIUDAD, T3.NOMBRE
FROM RESERVAPLSQL T1
LEFT JOIN CLIENTEPLSQL T2 ON T2.ID_CLIENTE= T1.ID_CLIENTE
LEFT JOIN SUCURSALPLSQL T3 ON T3.ID_SUCURSAL= T1.ID_SUCURSAL
WHERE T3.ID_SUCURSAL= 584;

```

FECHA_RESERVA	NOMBRE	CIUDAD	NOMBRE_1
10-FEB-23	(null)	Gardnerstad	Smith Ltd

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

```

SELECT T1.*, T2.NOMBRE
FROM ALQUILERPLSQL T1
LEFT JOIN CLIENTEPLSQL T2 ON T2.ID_CLIENTE= T1.ID_CLIENTE;

```

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA	NOMBRE
583	844	293	27-AUG-23	20-SEP-23	583	Julie Johnson
584	131	208	01-OCT-22	30-OCT-22	584	Alexander Cunningham
585	394	543	19-APR-23	11-MAY-23	585	Christine Snyder
586	774	345	19-JUL-23	30-JUL-23	586	Tara Johnson
587	81	250	26-MAR-23	22-APR-23	587	(null)
588	16	371	23-FEB-23	25-FEB-23	588	Tanya Blake
589	999	584	01-MAY-23	21-MAY-23	589	Jamie Davenport
590	796	711	26-MAY-23	17-JUN-23	590	(null)
591	928	812	08-MAY-23	15-MAY-23	591	Matthew Walter
592	223	847	27-NOV-22	14-DEC-22	592	Jennifer Hansen
593	987	961	21-AUG-23	20-SEP-23	593	Todd Yoder
594	435	45	18-SEP-23	29-SEP-23	594	(null)
595	81	435	06-MAY-23	31-MAY-23	595	(null)
596	397	466	16-SEP-23	25-SEP-23	596	Kevin Hoffman
597	36	473	06-FEB-23	01-MAR-23	597	(null)
598	726	669	09-NOV-22	23-NOV-22	598	Cassandra Davis
599	999	715	07-OCT-22	25-OCT-22	599	Jamie Davenport
600	414	809	29-DEC-22	10-JAN-23	600	Timothy Dixon

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

```
SELECT T1.*, T2.NOMBRE, T3.CIUDAD
FROM RESERVAPLSQL T1
LEFT JOIN CLIENTEPLSQL T2 ON T2.ID_CLIENTE= T1.ID_CLIENTE
LEFT JOIN SUCURSALPLSQL T3 ON T3.ID_SUCURSAL= T1.ID_SUCURSAL;
```

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA	NOMBRE	CIUDAD
862	79	239	05-FEB-23	Mary Chapman	North Amber
863	780	739	16-MAY-23	Diane Armstrong	(null)
864	737	729	04-JUN-23	Tammy Ochoa	(null)
865	700	982	10-DEC-22	(null)	(null)
866	197	610	29-JUN-23	Christopher Bradshaw	Port Joseph
867	594	755	17-JUN-23	Jasmine Burgess	(null)
868	808	185	07-MAR-23	(null)	East John
869	315	394	19-MAR-23	Jessica Murray	(null)
870	314	627	03-OCT-22	Brian Smith	North Jordanhaven
871	408	687	23-DEC-22	Ernest Mora	(null)
872	128	821	22-SEP-23	Bradley Walls	(null)
873	858	380	16-DEC-22	(null)	(null)
874	193	613	15-JAN-23	(null)	Sanchezview
875	256	954	11-AUG-23	Bryce Shields	(null)
876	437	903	21-AUG-23	Lawrence Mann	(null)
877	716	325	14-JUN-23	(null)	Port Stephaniechester
878	708	665	16-AUG-23	(null)	(null)
879	573	287	13-JUN-23	Stephanie Clark	Middletonchester

- Mostrar los clientes que no han realizado ninguna reserva.

```
SELECT *
FROM CLIENTEPLSQL
WHERE clienteplsql.id_cliente NOT IN (SELECT ID_CLIENTE FROM RESERVAPLSQL);
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
------------	--------	-----------	----------

4. Agregación y Agrupamiento:

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

```
SELECT COUNT(ID_AUTO), MARCA
FROM AUTOPLSQL
GROUP BY MARCA;
```

COUNT(ID_AUTO)	MARCA
2	Volkswagen
1	Audi
1	Lexus
2	ball
1	each
2	apply
2	can
2	land
1	from
1	participant
4	main
1	major
1	realize
3	vote
2	draw
2	smile
1	wind
2	ago

- Calcular la duración promedio de los alquileres.

```
SELECT AVG(FECHA_FIN-FECHA_INICIO) AS RENT_MEAN_DAYS
FROM ALQUILERPLSQL;
```

RENT_MEAN_DAYS
15.95695695695695695695695695695695696

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

```
SELECT T3.NOMBRE, COUNT(T1.ID_RESERVA) AS RESERV_QTY
FROM RESERVAPLSQL T1
LEFT JOIN SUCURSALPLSQL T3 ON T3.ID_SUCURSAL= T1.ID_SUCURSAL
GROUP BY NOMBRE;
```


NOMBRE	RESERV_QTY
Patrick-Rocha	2
Johnson, Martin and Gonzalez	1
Rodriguez, Scott and Ford	81
Phillips-Shepard	2
Reyes, Robinson and Davis	1
Barnes-McLaughlin	1
Garcia-Lee	1
Thomas-Wolfe	4
Lozano Inc	1
Garcia-Wilson	82
Sanford, Davis and Krueger	71
Dixon, Solomon and Tapia	79
Wiley, Trevino and Rowland	83
Sucursal 12	79
Johnson-Mendez	70
Williamson-Harris	2
Morales and Sons	4
Morton-Wilson	1

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

```
SELECT T3.NOMBRE, COUNT(T1.ID RESERVA) AS RESERV_QTY
FROM RESERVAPLSQL T1
LEFT JOIN CLIENTEPLSQL T3 ON T3.ID_CLIENTE= T1.ID_CLIENTE
GROUP BY NOMBRE
ORDER BY RESERV_QTY DESC
FETCH FIRST 1 ROWS ONLY;
```

NOMBRE	RESERV_QTY
(null)	3353

```
SELECT T3.NOMBRE, COUNT(T1.ID RESERVA) AS RESERV_QTY
FROM RESERVAPLSQL T1
LEFT JOIN CLIENTEPLSQL T3 ON T3.ID_CLIENTE= T1.ID_CLIENTE
WHERE T3.NOMBRE IS NOT NULL
GROUP BY NOMBRE
ORDER BY RESERV_QTY DESC
FETCH FIRST 1 ROWS ONLY;
```

NOMBRE	RESERV_QTY
Haley Carroll	23

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

```
SELECT AVG(ANO)
FROM AUTOPLSQL;
```

AVG(ANO)
2006.342342342342342342342342342342342342

5. Subconsultas:

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

```
SELECT *
FROM CLIENTEPLSQL
WHERE ID_CLIENTE IN (SELECT ID_CLIENTE
                     FROM RESERVAPLSQL
                     GROUP BY ID_CLIENTE
                     HAVING COUNT(ID_RESERVA)>1);
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	780 Diane Armstrong	378 Edward Hollow, Seanmouth, Hawaii 27595	728-245-2069
2	197 Christopher Bradshaw	49831 Baker Coves Suite 969, Victorberg, Alaska 24287	(992)281-1696
3	594 Jasmine Burgess	0241 Ward Mountains Apt. 413, Jonathanville, Texas 41605	825.402.7593
4	408 Ernest Mora	119 Mary Estates Suite 979, South Katherine, Tennessee 16259	7333224922
5	33 Colleen Wilson	784 Gregory Glen Suite 435, New Joshuamouth, West Virginia 42793	(761)681-6088
6	876 William Stewart	00583 Shelly Falls Apt. 415, Jenniferstad, Nevada 58169	573-272-2442
7	472 Angela Garcia	8102 Michael Crest Suite 264, Shelbyburgh, Arizona 54498	(719)202-1549
8	461 Barbara Weaver	3108 Hunt Courts, Gonzalezport, Kansas 93248	958.740.3934
9	93 Nancy Bell	810 David Cape Apt. 023, New Lisamouth, Maryland 05624	359-854-1101
0	131 Alexander Cunningham	3747 Rebekah Grove, New Amandaside, Illinois 08157	8049694589
1	307 Cynthia Moore	8890 Steven Street, Douglasberg, Massachusetts 77056	(570)843-1149
2	290 Michael Morales	2347 Mario Via, Port Matthewfort, Missouri 99809	(221)724-3873
3	344 Michael Haney	32445 Diana Estates, Catherinetown, Hawaii 26112	5574937937

- Mostrar los autos que no han sido alquilados aún.

```
SELECT *
FROM AUTOPLSQL
WHERE ID_AUTO NOT IN (SELECT ID_AUTO
                     FROM ALQUILERPLSQL);
```

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

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

```
SELECT *
FROM CLIENTEPLSQL
WHERE ID_CLIENTE IN (SELECT ID_CLIENTE
                     FROM ALQUILERPLSQL
                     GROUP BY ID_CLIENTE, ID_AUTO
                     HAVING COUNT(*)>1);
```

```

SELECT ID_CLIENTE
FROM ALQUILERPLSQL
GROUP BY ID_CLIENTE, ID_AUTO
HAVING COUNT(*)>1;

SELECT * FROM CLIENTEPLSQL
WHERE ID_CLIENTE IN (447)

```

Script Output x Query Result x

SQL | All Rows Fetched: 0 in 0.1 seconds

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
------------	--------	-----------	----------

El cliente fue eliminado de la base de datos de clientes.

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

```

SELECT T1.NOMBRE
FROM CLIENTEPLSQL T1
LEFT JOIN ALQUILERPLSQL T2 ON T1.id_cliente = T2.id_cliente;
JOIN SUCURSALPLSQL T3 ON T2.ID_SUCURSAL =T3.ID_SUCURSAL
WHERE (TRIM(BOTH ' ' FROM SUBSTR(
    T1.DIRECCION,
    INSTR(T1.DIRECCION, ',', 1, 1) + 1,
    INSTR(T1.DIRECCION, ',', 1, 2) - INSTR(T1.DIRECCION, ',', 1, 1) - 1
))) LIKE '%' || T3.CIUDAD || '%';

```

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

```

SELECT DISTINCT T1.marca, T1.modelo
FROM AutoPLSQL T1
JOIN AlquilerPLSQL T2 ON T1.id_auto = T2.id_auto
WHERE T2.id_sucursal IN ( SELECT id_sucursal FROM ReservaPLSQL);

```

6. Actualizaciones y Eliminaciones:

- Actualizar la dirección de un cliente específico.

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
625	Ricky Owens	94028 Mark Centers, Port Alishaland, New Hampshire 72598	3596036549

```

UPDATE CLIENTEPLSQL
SET DIRECCION= '1600 Amphitheatre Parkway in Mountain View, California'
WHERE ID_CLIENTE=625;

1 row updated.

```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
625	Ricky Owens	1600 Amphitheatre Parkway in Mountain View, California	3596036549

- Eliminar un auto de la tabla "Auto".

ID_AUTO	MARCA	MODELO	ANO
25	28 Volkswagen	Tiguan	2023
26	29 Chevrolet	Equinox	2023
27	30 Ford	Escape	2023
28	850 name	price	2014
29	851 nation	perhaps	2009
30	852 hall	design	2000

```
DELETE FROM AUTOPLSQL
```

```
WHERE ID_AUTO= 850;
```

```
1 row deleted.
```

```
SELECT * FROM CLIENTEPLSQL WHERE id_cliente=850;
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
------------	--------	-----------	----------

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

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	625	421	222 05-JUN-23	06-JUN-23	625

```
UPDATE ALQUILERPLSQL
```

```
SET FECHA_FIN= '19-NOV-2023'
```

```
WHERE ID_RESERVA=625;
```

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
	625	421	222 05-JUN-23	19-NOV-23	625

- Eliminar todas las reservas realizadas por un cliente específico.

```
DELETE FROM RESERVAPLSQL
```

```
WHERE ID_CLIENTE=421;
```

```
14 rows deleted.
```

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

Script Output x Query Result x

SQL | Fetched 50 rows in 0.096 seconds

ID_AUTO	MARCA	MODELO	ANO
13	16 Genesis	G80	2023
14	17 Volvo	S60	2023
15	18 Mazda	6	2023
16	19 Subaru	Legacy	2023
17	20 Hyundai	Sonata	2023
18	21 Kia	K5	2023

```
UPDATE AUTOPLSQL
```

```
SET ANO= 2024
```

```
WHERE ID_AUTO=18;
```

```
1 row updated.
```

ID_AUTO	MARCA	MODELO	ANO
18	Mazda	6	2024

EJERCICIOS SEGUNDO CICLO (31-80):

- SELECT * FROM ClientePLSQL;

```
SELECT *
FROM CLIENTEPLSQL
```

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	625 Ricky Owens	25136 Kelly Oval, Cantrellbury, Florida 72984	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
14	656 Samantha Cooper	28123 Justin Wall, East Thomas, Massachusetts 25128	+1-415-436-6560
15	669 Lisa Gray	56386 Richard Mountain, East Andrew, Maryland 86073	6352962563
16	670 Joseph Mitchell	7731 Jennifer Tunnel Suite 505, Rodrigueztown, Utah 09703	827.599.4267
17	673 Hannah Bishop	93988 Stephanie Park, Donaldtown, Minnesota 75086	794.622.4305

Muestra todos los elementos de la tabla clientes

- SELECT * FROM AutoPLSQL;

```
SELECT *
FROM AUTOPLSQL
```

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2023
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
14	16 Genesis	G80	2023

Muestra todos los elementos de la tabla autos

- `SELECT * FROM AlquilerPLSQL;`

```
SELECT *
FROM ALQUILERPLSQL
```

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	583	844	293	27-AUG-23	20-SEP-23	583
2	584	131	208	01-OCT-22	30-OCT-22	584
3	585	394	543	19-APR-23	11-MAY-23	585
4	586	774	345	19-JUL-23	30-JUL-23	586
5	587	81	250	26-MAR-23	22-APR-23	587
6	588	16	371	23-FEB-23	25-FEB-23	588
7	589	999	584	01-MAY-23	21-MAY-23	589
8	590	796	711	26-MAY-23	17-JUN-23	590
9	591	928	812	08-MAY-23	15-MAY-23	591
10	592	223	847	27-NOV-22	14-DEC-22	592
11	593	987	961	21-AUG-23	20-SEP-23	593
12	594	435	45	18-SEP-23	29-SEP-23	594
13	595	81	435	06-MAY-23	31-MAY-23	595
14	596	397	466	16-SEP-23	25-SEP-23	596
15	597	36	473	06-FEB-23	01-MAR-23	597
16	598	726	669	09-NOV-22	23-NOV-22	598
17	599	999	715	07-OCT-22	25-OCT-22	599
18	600	414	809	29-DEC-22	10-JAN-23	600
19	601	470	403	03-OCT-22	16-OCT-22	601

Muestra todos los elementos de la tabla alquiler

- `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: 10 Column: 27
```

La columna modelo no existe en la tabla alquiler

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

	MARCA	MODELO	ANO
1	bit	land	1995
2	lot	level	2023
3	economic	approach	2017
4	pressure	become	1990
5	which	picture	2018
6	account	director	2014
7	gas	approach	1993
8	issue	moment	1999
9	cell	test	2018
10	operation	information	2000
11	image	without	2000
12	art	project	2007
13	ball	off	2004
14	lot	level	2023

Los autos que han sido alquilados

- `SELECT * FROM AlquilerPLSQL WHERE id_cliente = 1;`

ID_ALQUI...	ID_CLIENTE	ID_AUTO	FECHA_I...	FECHA_FIN	ID_RESER...
-------------	------------	---------	------------	-----------	-------------

No hay alquileres para el cliente cuyo ID es 1

- `SELECT * FROM AlquilerPLSQL WHERE id_auto = 1;`

ID_ALQUI...	ID_CLIENTE	ID_AUTO	FECHA_I...	FECHA_FIN	ID_RESER...
-------------	------------	---------	------------	-----------	-------------

No hay alquileres para el auto cuyo ID es 1

- `SELECT * FROM AlquilerPLSQL WHERE id_sucursal = 1;`

```
ORA-00904: "ID_SUCURSAL": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error at Line: 14 Column: 35
```

No existe la columna ID sucursal en la tabla alquiler

- `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.
```

El formato de fecha usado para filtrar no es compatible con el formato de fecha de los datos

- `SELECT COUNT(*) FROM AlquilerPLSQL;`

	COUNT(*)
1	999

El total de alquileres realizados

- `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: 17 Column: 113
```

La columna sucursal ID no existe en la tabla alquiler

```
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 s.id_sucursal = s.id_sucursal
WHERE s.nombre = 'Sucursal Central';
```

- `SELECT a.marca, a.modelo FROM AutoPLSQL a`

	MARCA	MODELO
1	Honda	Civic
2	Nissan	Versa
3	Kia	Rio
4	Hyundai	Accent
5	Chevrolet	Onix
6	Fiat	Argo
7	Volkswagen	Gol
8	Renault	Kwid
9	Ford	Ka
10	BMW	Serie 3
11	Mercedes-Benz	Clase C
12	Lexus	ES
13	Genesis	G80
14	Volvo	S60
15	Mazda	6

Selecciona las columnas Marca y modelo de la tabla Auto

- `JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto WHERE al.id_cliente = 1 AND al.fecha_inicio = '2023-09-27';`

```
Error starting at line : 26 in command -
```

```
JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto WHERE al.id_cliente = 1 AND al.fecha_inicio = '2023-09-27'
```

```
Error report -
```

```
Unknown Command
```

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

```
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'
```


No hay autos con fecha de inicio de alquiler del 27 de septiembre de 2023.

- `SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio > 7;`

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
584	131	208	01-OCT-22	30-OCT-22	584
585	394	543	19-APR-23	11-MAY-23	585
586	774	345	19-JUL-23	30-JUL-23	586
587	81	250	26-MAR-23	22-APR-23	587
589	999	584	01-MAY-23	21-MAY-23	589
590	796	711	26-MAY-23	17-JUN-23	590
592	223	847	27-NOV-22	14-DEC-22	592
593	987	961	21-AUG-23	20-SEP-23	593
594	435	45	18-SEP-23	29-SEP-23	594
595	81	435	06-MAY-23	31-MAY-23	595
596	397	466	16-SEP-23	25-SEP-23	596

Seleccionar todos los alquileres con una duración mayor a 7 días

- `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;`

Cliente que tiene la mayor cantidad de alquileres, no obstante, limit 1 no puede ser usado.

```
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 LIMIT 1;`

Identifica el auto más popular en términos de alquileres, mostrando así la marca y modelo del auto más frecuentemente alquilado y el número de veces que fue alquilado. Limit 1 no puede ser usado.

```
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 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;`

Sucursal con el mayor número de alquileres. Limit 1 no puede ser usado.

```
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;
```

- `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;`

Mes con el mayor número de alquileres. Limit 1 no puede ser usado.

```
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 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;
```

Día de la semana en el que se realizan más alquileres. Limit 1 no puede ser usado.

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

Alquiler más costoso. Limit 1 no puede ser usado.

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

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

Alquiler más económico. Limit 1 no puede ser usado.

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

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

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

Seleccionar todos los datos de clientes cuyo nombre sea o contenga Juan

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

Precio no se encuentra entre las columnas de la tabla auto.

- `SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';`

```
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.
```

SE

- `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%';`

Marca y modelo pertenecen a la tabla auto, no a la tabla alquiler.

- `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
later	too	2003

Año, marca y modelo del auto alquilado para la reserva cuyo ID es igual a 1

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

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
861	3	171	20-DEC-22	10-JAN-23	861
292	3	891	19-AUG-23	13-SEP-23	292
384	2	141	19-DEC-22	03-JAN-23	384

Alquileres realizados por los clientes cuyos ID son 1 2 o 3

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

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
723	911	2	19-APR-23	01-MAY-23	723
148	922	3	13-NOV-22	25-NOV-22	148
329	668	3	03-JUL-23	07-JUL-23	329

Alquileres realizados de los autos cuyos ID son 1 2 o 3

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

El formato de fecha no coincide.

Alquileres realizados por los clientes cuyos ID son 1 2 o 3 y realizaron alquileres entre el 1 de septiembre y el 30 de septiembre

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN '01-SEP-2023' AND '30-SEP-2023' AND id_cliente IN (1, 2, 3);
```

ID_ALQUI...	ID_CLIENTE	ID_AUTO	FECHA_I...	FECHA_FIN	ID_RESER...
-------------	------------	---------	------------	-----------	-------------

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

El formato de fecha no coincide.

Alquileres realizados para los autos cuyos ID son 1 2 o 3 y realizaron alquileres entre el 1 de septiembre y el 30 de septiembre

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN '01-SEP-2023' AND '30-SEP-2023' AND id_auto IN (1, 2, 3);
```

ID_ALQUI...	ID_CLIENTE	ID_AUTO	FECHA_I...	FECHA_FIN	ID_RESER...
-------------	------------	---------	------------	-----------	-------------

- `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;`

Limit 1 no puede ser usado.

Seleccionar el cliente con mayor número de alquileres.

```
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
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 LIMIT 1;`

Limit 1 no puede ser usado.

```
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
end	hold	5

Marca y modelo del auto más alquilado.

- `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;`
- Limit 1 no puede ser usado.

```
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;
```

Sucursal con mayor número de alquileres.

- 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;

Limit 1 no puede ser usado.

```
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
12	100

Mes con mayor número de alquileres

- 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;

Limit 1 no puede ser usado.

DayOfweek no puede ser usado.

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

DIA_SEMANA	NUMERO_ALQUILERES
MONDAY	154

Día de la semana con mayor número de alquileres.

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

Limit 1 no puede ser usado, la tabla alquiler no tiene una columna precio

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

Limit 1 no puede ser usado, la tabla alquiler no tiene una columna precio

- SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%' AND fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';

Falta el join entre las tablas de cliente y alquiler.

- SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a WHERE precio < 10000 AND fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';

Falta el join entre las tablas de auto y alquiler, la tabla alquiler no tiene una columna precio

EJERCICIOS TERCER CICLO (81-90):

- ```
CREATE VIEW vista_clientes_alquilados_sucursal AS SELECT
c.nombre, a.marca, a.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';
```

Crea una vista con el nombre del cliente la marca y modelo alquilado en la sucursal central.

- ```
CREATE 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 = '2023-09-27';
```

```
CREATE 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 = '2023-09-27';
```

View VISTA_AUTOS_ALQUILADOS_CLIENTE_FECHA created.

Crea una vista de la marca y modelo del auto rentado por el cliente cuyo ID es 1 y la fecha de inicio del alquiler es el 27 de septiembre de 2023.

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

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

View VISTA\_ALQUILERES\_MAS\_7DIAS created.

Crea una vista de todos los alquileres cuya duración fue superior a 7 días.

- ```
CREATE 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;
```

```
CREATE 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;
```

View VISTA_CLIENTES_MAS_ALQUILERES created.

Crea una vista de los clientes y el número de alquileres ordenando de mayor a menor.

- ```
CREATE 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;
```

```
CREATE 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;
```

View VISTA\_AUTOS\_MAS\_ALQUILERES created.

Crea una vista de los autos por modelo y marca y la cantidad de veces que han sido alquilados y los ordena de mayor a menor.

- ```
CREATE 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;
```

Crea una vista de las sucursales y la cantidad de alquileres por sucursal y ordena de mayor a menor.

- ```
CREATE 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;
```

```
CREATE 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;
```

View VISTA\_MESES\_MAS\_ALQUILERES created.

Crea una vista de la cantidad de alquileres realizados por mes y los ordena de mayor a menor



- `CREATE VIEW vista_dias_semana_mas_alquileres AS 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;`

DayOfweek no puede ser usado.

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

View **VISTA\_DIAS\_SEMANA\_MAS\_ALQUILERES** created.

Crea una vista de la cantidad de alquileres por día de la semana y los ordena de mayor a menor.

- `CREATE VIEW vista_alquileres_mas_caros AS SELECT * FROM  
AlquilerPLSQL ORDER BY precio DESC;`

Crea una vista de los precios de los alquileres los ordena de mayor a menor.

- `CREATE VIEW vista_alquileres_mas_baratos AS SELECT * FROM  
AlquilerPLSQL ORDER BY precio ASC;`

Crea una vista de los precios de los alquileres los ordena de menor a mayor.

## **EJERCICIOS TERCER CICLO (91-100):**

```
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;
```

Este trigger funciona durante el proceso de inserción, actualiza el contador de autos disponible en la tabla "AutoPLSQL". Sin embargo, debido a que la columna "número\_disponibles" no existe en la estructura de la tabla "AutoPLSQL", el desencadenante presenta un error relacionado con la actualización de la columna.

```
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;
```

Este trigger funciona durante el proceso de eliminación, actualiza el contador de autos disponible en la tabla "AutoPLSQL". Sin embargo, debido a que la columna "número\_disponibles" no existe en la estructura de la tabla "AutoPLSQL", el desencadenante presenta un error relacionado con la actualización de la columna.

```
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;
```

Este trigger funciona durante el proceso de actualización, actualiza el contador de autos disponible en la tabla "AutoPLSQL" cada vez que se ingresa un nuevo auto. Sin embargo, debido a que la columna "número\_disponibles" no existe en la estructura de la tabla "AutoPLSQL", el desencadenante presenta un error relacionado con la actualización de la columna.

```
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;
```

Este trigger funciona durante el proceso de inserción, actualiza el contador de clientes en la tabla " ClientePLSQL ". Sin embargo, debido a que la columna " numero\_clientes " no existe en la estructura de la tabla " ClientePLSQL ", el desencadenante presenta un error relacionado con la actualización de la columna.

```
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;
```

Este trigger funciona durante el proceso de eliminación, actualiza el contador de clientes en la tabla " ClientePLSQL ". Sin embargo, debido a que la columna " numero\_clientes " no existe en la estructura de la tabla " ClientePLSQL ", el desencadenante presenta un error relacionado con la actualización de la columna.

```
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;
```

Este trigger funciona durante el proceso de actualización, actualiza el contador de clientes en la tabla " ClientePLSQL ". Sin embargo, debido a que la columna " numero\_clientes " no existe en la estructura de la tabla " ClientePLSQL ", el desencadenante presenta un error relacionado con la actualización de la columna.



```

CREATE PROCEDURE proc_calcular_precio_alquiler
(
 IN id_alquiler INT,
 IN id_auto INT,
 IN fecha_inicio DATE,
 IN fecha_fin DATE
)
AS
BEGIN
 -- Calcular el precio del alquiler
 DECLARE
 precio_base NUMERIC(10, 2);
 dias_alquiler INT;
 BEGIN
 precio_base := (SELECT precio FROM AutoPLSQL WHERE id_auto =
id_auto);
 dias_alquiler := (fecha_fin - fecha_inicio) + 1;
 SET NEW.precio = precio_base * dias_alquiler;
 END;
END;

```

La sintaxis no es correcta, se están mezclando Oracle PL/SQL y MySQL y adicionalmente no existe una columna precio en la tabla auto que permita calcular el costo total del alquiler.

```

CREATE PROCEDURE proc_listar_alquileres_cliente
(
 IN id_cliente INT
)
AS
BEGIN
 -- Listar los alquileres del cliente
 SELECT *
 FROM AlquilerPLSQL
 WHERE id_cliente = id_cliente;
END;

```

La sintaxis no es correcta, se están mezclando Oracle PL/SQL y otro DBMS, se puede corregir de la siguiente forma

```

CREATE OR REPLACE PROCEDURE proc_listar_alquileres_cliente(
 p_id_cliente IN INT
)
IS
BEGIN
 -- Listar los alquileres del cliente
 FOR alquiler_rec IN (SELECT * FROM AlquilerPLSQL WHERE id_cliente = p_id_cliente)
 LOOP
 -- Imprimir
 DBMS_OUTPUT.PUT_LINE('ID Alquiler: ' || alquiler_rec.id_alquiler ||
 ', ID Auto: ' || alquiler_rec.id_auto ||
 ', Fecha Inicio: ' || alquiler_rec.fecha_inicio ||
 ', Fecha Fin: ' || alquiler_rec.fecha_fin);
 END LOOP;
END proc_listar_alquileres_cliente;
/

```

Procedure PROC\_LISTAR\_ALQUILERES\_CLIENTE compiled

```

CREATE PROCEDURE proc_listar_autos_sucursal
(
 IN id_sucursal INT
)
AS
BEGIN
 -- Listar los autos de la sucursal
 SELECT *
 FROM AutoPLSQL
 WHERE id_sucursal = id_sucursal;
END;

```

Error de sintaxis nuevamente, la columna id\_sucursal no existe en la tabla de autos.

```

CREATE PROCEDURE proc_agregar_auto
(
 IN marca VARCHAR(255),
 IN modelo VARCHAR(255),
 IN ano INT,
 IN numero_disponibles INT
)
AS
BEGIN
 -- Insertar un nuevo auto
 INSERT INTO AutoPLSQL (marca, modelo, ano, numero_disponibles)
 VALUES (marca, modelo, ano, numero_disponibles);
END;

```

Error de sintaxis nuevamente, numero\_disponibles no existe en la tabla auto

```

CREATE PROCEDURE proc_eliminar_auto
(
 IN id_auto INT
)
AS
BEGIN
 -- Eliminar un auto
 DELETE FROM AutoPLSQL
 WHERE id_auto = id_auto;
END;

```

Error de sintaxis nuevamente, se corrige así:

```

CREATE PROCEDURE proc_eliminar_auto
(
 id_auto IN INT
)
AS
BEGIN
 -- Eliminar un auto
 DELETE FROM AutoPLSQL
 WHERE id_auto = id_auto;
END;

```

Procedure PROC\_ELIMINAR\_AUTO compiled