

Cuaderno PL/SQL

Primera parte (Ejercicios 1-30):

1. `SELECT ID_CLIENTE FROM ClientePLSQL;` (Esto selecciona la columna ID_CLIENTE)

Resultado:

	ID_CLIENTE
1	449
2	450
3	453
4	454
5	455
6	457
7	463
8	467
9	478
10	481
11	483
12	487
13	491
14	497
15	498
16	499
17	500

2. `SELECT MARCA FROM AutoPLSQL;` (Esto selecciona la marca de los autos)

Resultado:

	MARCA
1	Toyota
2	Honda
3	Nissan
4	Kia
5	Hyundai
6	Chevrolet
7	Fiat
8	Volkswagen
9	Renault
10	Ford
11	BMW
12	Mercedes-Benz
13	Audi
14	Lexus
15	Genesis
16	Volvo
17	Mazda

3. `SELECT id_alquiler FROM AlquilerPLSQL;` (Esto selecciona todos los alquileres)

Resultado:

	ID_ALQUILER
1	584
2	585
3	586
4	587
5	588
6	589
7	590
8	591
9	592
10	593
11	594
12	595
13	596
14	597
15	598
16	599
17	600

4. SELECT NOMBRE FROM SucursalPLSQL; (Esto selecciona todas las sucursales)

Resultados:

	NOMBRE
1	Sucursal Bogotá
2	Sucursal Medellín
3	Sucursal Cali
4	Sucursal Barranquilla
5	Sucursal Cartagena
6	Sucursal Villavicencio
7	Sucursal Yopal
8	Sucursal Pasto
9	Sucursal Bucaramanga
10	Sucursal Manizales
11	Sucursal São Paulo
12	Sucursal Ciudad de México
13	Sucursal Lima
14	Sucursal Santiago de Chile
15	Sucursal Buenos Aires
16	Sucursal Madrid
17	Sucursal París

5. SELECT id_reserva FROM ReservaPLSQL; (esto selecciona todas las reservas)

Resultado:

	ID_RESERVA
1	863
2	864
3	865
4	866
5	867
6	868
7	869
8	870
9	871
10	872
11	873
12	875
13	876
14	877
15	878
16	879
17	880

6. SELECT NOMBRE FROM ClientePLSQL WHERE NOMBRE = 'Juan'; (Esto toma la columna nombre de ClientePLSQL donde el nombre sea igual a Juan)

Resultados: (No hay ningún Juan)

NOMBRE

7. SELECT MARCA FROM AutoPLSQL WHERE MARCA = 'Toyota'; (Toma la columna Marca de AutoPLSQL donde el nombre sea igual a Toyota)

Resultados:

MARCA
1 Toyota

8. SELECT * FROM AlquilerPLSQL WHERE FECHA_INICIO > '22-10-01';
(toma valores de la tabla donde la fecha sea mayor a la indicada)

Resultado:

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	585	394	543	23-04-19	23-05-11	585
2	586	774	345	23-07-19	23-07-30	586
3	587	81	250	23-03-26	23-04-22	587
4	588	16	371	23-02-23	23-02-25	588
5	589	999	584	23-05-01	23-05-21	589
6	590	796	711	23-05-26	23-06-17	590
7	591	928	812	23-05-08	23-05-15	591
8	592	223	847	22-11-27	22-12-14	592
9	593	987	961	23-08-21	23-09-20	593
10	594	435	45	23-09-18	23-09-29	594
11	595	81	435	23-05-06	23-05-31	595
12	596	397	466	23-09-16	23-09-25	596
13	597	36	473	23-02-06	23-03-01	597
14	598	726	669	22-11-09	22-11-23	598
15	599	999	715	22-10-07	22-10-25	599
16	600	414	809	22-12-29	23-01-10	600
17	601	470	463	22-10-02	22-10-16	601

9. SELECT CIUDAD FROM SucursalPLSQL WHERE CIUDAD = 'Madrid';
(Toma la columna Ciudad de SucursalPLSQL donde la ciudad sea Madrid)

Resultados:

CIUDAD
1 Madrid

10. select * from reservaplsq where id_cliente = '780'; (reservas realizadas por el cliente con ID 780)

Resultado:

	ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	863	780	739	23-05-16
2	2607	780	33	22-10-01
3	2643	780	58	23-04-27
4	806	780	5	23-01-09
5	1713	780	69	23-02-25
6	3053	780	76	23-03-05
7	3220	780	14	23-01-30
8	3341	780	66	23-09-04
9	2607	780	33	22-10-01
10	2643	780	58	23-04-27
11	806	780	5	23-01-09
12	1713	780	69	23-02-25

11. select clienteplsq.nombre, reservaplsq.id_reserva from clienteplsq inner join reservaplsq on clienteplsq.id_cliente = reservaplsq.id_cliente; (realiza una unión para mostrar los nombres de los clientes y sus respectivas reservas)

Resultado:

	NOMBRE	ID_RESERVA
1	Diane Armstrong	863
2	Tammy Ochoa	864
3	Christopher Bradshaw	866
4	Jasmine Burgess	867
5	Jessica Murray	869
6	Brian Smith	870
7	Ernest Mora	871
8	Bradley Walls	872
9	Bryce Shields	875
10	Lawrence Mann	876
11	Stephanie Clark	879
12	Jenny Brown	881
13	Robert West	882
14	Sarah Duffy	885
15	William Adams	887
16	Ernest Mora	890
17	Erica Morgan	892

12. select sucursalpsql.nombre, reservapsql.id_cliente from sucursalpsql inner join reservapsql on sucursalpsql.id_sucursal = reservapsql.id_sucursal; (Nombre de la sucursal y su respectivo cliente)

Resultado:

	NOMBRE	ID_CLIENTE
1	Anderson-Cuevas	197
2	Tucker, Taylor and Jones	808
3	Rodriguez-Ramos	314
4	Patrick-Rocha	716
5	Johnson, Martin and Gonzalez	573
6	Mills Ltd	850
7	Rodriguez, Scott and Ford	647
8	West-Weber	493
9	Wheeler-Leonard	793
10	Dixon and Sons	997
11	Lester Group	391
12	Dixon and Sons	408
13	Nelson-Kelly	165
14	Reid and Sons	588
15	Nielsen Ltd	4
16	Malone-Harris	423
17	Watts Group	118

13. select clientepsql.nombre, alquilerpsql.id_alquiler from clientepsql inner join alquilerpsql on clientepsql.id_cliente = alquilerpsql.id_cliente; (El nombre de la persona y su alquiler)

Resultado:

	NOMBRE	ID_ALQUILER
1	Alexander Cunningham	584
2	Christine Snyder	585
3	Tara Johnson	586
4	Tanya Blake	588
5	Jamie Davenport	589
6	Matthew Walter	591
7	Jennifer Hansen	592
8	Todd Yoder	593
9	Kevin Hoffman	596
10	Cassandra Davis	598
11	Jamie Davenport	599
12	Timothy Dixon	600
13	David Wallace	602
14	Elizabeth McDowell	604
15	Jack Olson	606
16	William Stewart	608
17	Kevin Carter	609

14. select clienteplsql.nombre, sucursalplsql.ciudad from clienteplsql inner join sucursalplsql on clienteplsql.id_cliente = sucursalplsql.id_sucursal; (Nombre y ciudad de las reservas)

Resultado:

	NOMBRE	CIUDAD
1	Daniel Leonard	Cali
2	Christian Martin	Pereira
3	Jacqueline Bryant	Manizales
4	Gina Walker	Bucaramanga
5	Anthony Wood	Pasto
6	Alyssa Lawrence	Santa Marta
7	Edward Rogers	Nueva Delhi
8	Jennifer Johnson	Shanghái
9	Maria Williams	Nueva York
10	Tanya Blake	París
11	Danielle Johnson	Buenos Aires
12	Jennifer Johnson	Donaldshire
13	Mark Walter	Suarezstad
14	Amy Park	Edwardstad
15	Jesus Harris	Beardfort
16	Micheal Williams	Port Lisaborough
17	David Schmidt	South Reginaldmouth

15. select clienteplsql.nombre, reservaplsq.id_reserva from clienteplsql left join reservaplsq on clienteplsql.id_cliente = reservaplsq.id_cliente where reservaplsq.id_reserva is null; (Los clientes que no han hecho ninguna reserva)

Resultado:

NOMBRE	ID_RES...
1 Mary Chapman	(null)

16. SELECT MARCA, COUNT(*) FROM AutoPLSQL GROUP BY MARCA;

Resultados:

MARCA	COUNT(*)
1 Volkswagen	2
2 Audi	1
3 Lexus	1
4 Honda	2
5 Nissan	2
6 Mercedes-Benz	1
7 Volvo	1
8 Chevrolet	2
9 Fiat	1
10 Kia	3
11 Ford	2
12 Renault	1
13 Hyundai	3
14 BMW	1
15 Genesis	1
16 Toyota	1
17 Mazda	2

17. SELECT AVG(fecha_fin - fecha_inicio) AS promedio_dias FROM alquilerplsql; (Duración promedio en días de los alquileres)

Resultado:

PROMEDIO_DIAS
1 15.94889779559118236472945891783567134269

18. SELECT sucursalplsql.nombre, COUNT(reservaplsq.id_reserva) AS total_reservations FROM sucursalplsql LEFT JOIN reservaplsq ON sucursalplsql.id_sucursal = reservaplsq.id_sucursal GROUP BY sucursalplsql.nombre ORDER BY sucursalplsql.nombre; (Todas las reservaciones que hay por cada sucursal)

Resultado:

NOMBRE	TOTAL_RESERVATIONS
1 Abbott Ltd	0
2 Adkins-Dunn	0
3 Aguilar Group	2
4 Allen Group	81
5 Allen Inc	0
6 Allen, Montgomery and Hebert	0
7 Alvarado, Torres and Little	90
8 Anderson and Sons	1
9 Anderson-Cuevas	2
10 Armstrong Inc	0
11 Austin-Clark	1
12 Avila, Gutierrez and Bowers	0
13 Avila, Price and Strickland	0
14 Ayers, Kemp and Smith	2
15 Baker-Powell	0
16 Baker-Stein	83
17 Baker-Williams	1

19. SELECT nombre, total_alquiler FROM (SELECT clienteplsql.nombre, COUNT(alquilerplsql.id_cliente) AS total_alquiler FROM clienteplsql INNER JOIN alquilerplsql ON clienteplsql.id_cliente = alquilerplsql.id_cliente GROUP BY clienteplsql.nombre ORDER BY total_alquiler DESC) WHERE ROWNUM <= 10; (La cantidad de alquileres por persona, en este caso fue limitado a mostrar los 10 primeros)

Resultado:

	NOMBRE	TOTAL_ALQUILER
1	John Greene	6
2	Pamela Johnson	5
3	Michele Allen	4
4	Jill Sanford	4
5	David Woods	4
6	Andrew Chang	4
7	Stephanie Young	4
8	Haley Carroll	4
9	Melinda Jackson	3
10	Carol Morales	3

20. SELECT AVG(ANO) FROM AutoPLSQL; (lo que hace es seleccionar el promedio de la tabla AutPLSQL)

Resultado:

	AVG(ANO)
1	2023

21. SELECT sucursalplsql.nombre, COUNT(reservaplsq.id_reserva) AS total_reservations FROM sucursalplsql LEFT JOIN reservaplsq ON sucursalplsql.id_sucursal = reservaplsq.id_sucursal GROUP BY sucursalplsql.nombre HAVING COUNT(reservaplsq.id_reserva) > 0 ORDER BY sucursalplsql.nombre; (esto primero saca la cantidad total de reservas que ha hecho cada cliente y luego mira quienes tengan más de 0)

Resultado:

NOMBRE	TOTAL_RESERVATIONS
28 Brown, Bridges and Rice	1
29 Brown, Coffey and Johnson	2
30 Bryant LLC	1
31 Bryant PLC	2
32 Bush, Olson and Krueger	85
33 Butler-Moreno	2
34 Cannon LLC	2
35 Carey PLC	1
36 Carter, Woods and Williams	3
37 Case PLC	2
38 Castillo-Dennis	1
39 Chan, Clark and Munoz	2
40 Chandler Inc	1
41 Chavez, Walker and Martinez	1
42 Chen Inc	2
43 Christian and Sons	1
44 Clark, Morris and Jones	2

22. SELECT AutoPLSQL.marca, AutoPLSQL.modelo FROM AutoPLSQL LEFT JOIN AlquilerPLSQL ON AutoPLSQL.id_auto = AlquilerPLSQL.id_auto WHERE AlquilerPLSQL.id_auto IS NULL; (este código muestra los autos que no han sido alquilados usando ambas tablas de alquiler y auto)

Resultado:

MARCA	MODELO
1 Hyundai	Accent
2 Ford	Ka
3 Mercedes-Benz	Clase C
4 Lexus	ES
5 Subaru	Legacy
6 Honda	CR-V
7 Nissan	Rogue
8 Mazda	CX-5
9 Hyundai	Tucson
10 Volkswagen	Tiguan
11 Chevrolet	Equinox
12 Ford	Escape
13 nation	perhaps
14 Mrs	need
15 still	billion
16 program	expert
17 second	market

23. SELECT ClientePLSQL.nombre AS nombre_cliente, AutoPLSQL.marca AS marca_auto, AutoPLSQL.modelo AS modelo_auto FROM ClientePLSQL INNER JOIN AlquilerPLSQL ON ClientePLSQL.id_cliente = AlquilerPLSQL.id_cliente INNER JOIN AutoPLSQL ON AlquilerPLSQL.id_auto = AutoPLSQL.id_auto GROUP BY ClientePLSQL.nombre, AutoPLSQL.marca, AutoPLSQL.modelo HAVING COUNT(AlquilerPLSQL.id_alquiler) > 1; (Selecciona las columnas como los

nombres y las marcas de los autos para hacer un join que me junte ambas informaciones, para luego darle la condición que buscamos, aparentemente no existe ningún registro que cumpla con esta condición)

Resultado:

NOMBRE_...	MARCA_A...	MODELO_...
------------	------------	------------

24. SELECT ClientePLSQL.nombre AS nombre_cliente, ReservaPLSQL.fecha_reserva, SucursalPLSQL.ciudad AS ciudad_residencia FROM ClientePLSQL INNER JOIN ReservaPLSQL ON ClientePLSQL.id_cliente = ReservaPLSQL.id_cliente INNER JOIN SucursalPLSQL ON ReservaPLSQL.id_sucursal = SucursalPLSQL.id_sucursal WHERE ClientePLSQL.direccion = SucursalPLSQL.ciudad; (La relación que se busca es que el lugar donde vive el cliente se encuentre en la ciudad que aparece en la tabla de sucursal)

Resultado:

NOMBRE_...	FECHA_R...	CIUDAD_...
------------	------------	------------

25. SELECT AutoPLSQL.marca AS marca_auto, AutoPLSQL.modelo AS modelo_auto, SucursalPLSQL.ciudad AS ciudad_sucursal FROM AutoPLSQL INNER JOIN AlquilerPLSQL ON AutoPLSQL.id_auto = AlquilerPLSQL.id_auto INNER JOIN ReservaPLSQL ON AlquilerPLSQL.id_reserva = ReservaPLSQL.id_reserva INNER JOIN SucursalPLSQL ON ReservaPLSQL.id_sucursal = SucursalPLSQL.id_sucursal WHERE SucursalPLSQL.ciudad = SucursalPLSQL.pais; (Junta las reservas con la tabla de sucursal, de la cual se relaciona primero con los autos alquilados para encontrar si efectivamente hubo una reserva en la misma sucursal donde se alquiló un carro)

Resultado:

MARCA_A...	MODELO_...	CIUDAD_...
------------	------------	------------

26. Update clientepsql set direccion = '94024 Mark Centers, Port Alishaland, New Hampshire 73476' where nombre = 'Ricky Owens'; (actualiza la dirección del cliente llamado Ricky Owens)

Resultado:

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	625 Ricky Owens	94024 Mark Centers, Port Alishaland, New Hampshire 73476	3596036549

27. DELETE FROM AutoPLSQL WHERE MARCA = 'Toyota'; (Busca la tabla AutoPLSQL donde la marca sea igual a Toyota y la elimina)

Resultados:

1 row deleted.

28. update reservaplsq set fecha_reserva = '23-10-23' where fecha_reserva = '23-05-16'; (actualiza cada registro que tenga la fecha como 23-05-16, en este caso

fueron 21 en total)

Resultado:

	ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	863	780	739	23-10-23
2	864	737	729	23-06-04
3	865	700	982	22-12-10
4	866	197	610	23-06-29
5	867	594	755	23-06-17
6	868	808	185	23-03-07
7	869	315	394	23-03-19
8	870	314	627	22-10-03
9	871	408	687	22-12-23
10	872	128	821	23-09-22
11	873	858	380	22-12-16
12	875	256	954	23-08-11
13	876	437	903	23-08-21
14	877	716	325	23-06-14
15	878	708	665	23-08-16
16	879	573	287	23-06-13
17	880	850	568	23-03-25

29. delete from reservaplsql where id_cliente = '780'; (esto remueve todas las reservas hechas por el cliente con id 780)

Resultado:

	ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	864	737	729	23-06-04
2	865	700	982	22-12-10
3	866	197	610	23-06-29
4	867	594	755	23-06-17
5	868	808	185	23-03-07
6	869	315	394	23-03-19
7	870	314	627	22-10-03
8	871	408	687	22-12-23
9	872	128	821	23-09-22
10	873	858	380	22-12-16
11	875	256	954	23-08-11
12	876	437	903	23-08-21
13	877	716	325	23-06-14
14	878	708	665	23-08-16
15	879	573	287	23-06-13
16	880	850	568	23-03-25
17	881	647	46	23-05-28

30. update autoplsql set marca = 'Nissan' where id_auto = '2'; (va a buscar el id_auto 2 y va a modificar su marca a Nissan)

Resultado:

	ID_AUTO	MARCA	MODELO	ANO
1	2	Nissan	Civic	2025

Segunda parte (Ejercicios 30-80):

31. Selecciona todos los elementos de la tabla (Columnas y filas) ClientePLSQL

Resultados:

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	83888 Stephanie Park, Donaldtown, Minnesota 75086	784.622.4305

32. Selecciona todos los elementos de la tabla (Columnas y filas) AutoPLSQL

Resultados:

	ID_AUTO	MARCA	MODELO	ANO
1	1	Toyota	Corolla	2023
2	2	Honda	Civic	2023
3	3	Nissan	Versa	2023
4	4	Kia	Rio	2023
5	5	Hyundai	Accent	2023
6	6	Chevrolet	Onix	2023
7	7	Fiat	Argo	2023
8	8	Volkswagen	Gol	2023
9	9	Renault	Kwid	2023
10	10	Ford	Ka	2023
11	11	BMW	Serie 3	2023
12	13	Mercedes-Benz	Clase C	2023
13	14	Audi	A4	2023
14	15	Lexus	ES	2023
15	16	Genesis	G80	2023
16	17	Volvo	S60	2023
17	18	Mazda	6	2023

33. Selecciona todos los elementos de la tabla (Columnas y filas) AlquilerPLSQL

Resultados:

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131	208 22-10-01	22-10-30	584
2	585	394	543 23-04-19	23-05-11	585
3	586	774	345 23-07-19	23-07-30	586
4	587	81	250 23-03-26	23-04-22	587
5	588	16	371 23-02-23	23-02-25	588
6	589	999	584 23-05-01	23-05-21	589
7	590	796	711 23-05-26	23-06-17	590
8	591	928	812 23-05-08	23-05-15	591
9	592	223	847 22-11-27	22-12-14	592
0	593	987	961 23-08-21	23-09-20	593
1	594	435	45 23-09-18	23-09-29	594
2	595	81	435 23-05-06	23-05-31	595
3	596	397	466 23-09-16	23-09-25	596
4	597	36	473 23-02-06	23-03-01	597
5	598	726	669 22-11-09	22-11-23	598
6	599	999	715 22-10-07	22-10-25	599
7

34. SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente; (El problema es que alquilerplsqli no tiene ninguna columna con nombre modelo, por ende da error)
Resultado:

```
ORA-00904: "A"."MODELO": invalid identifier
00904. 00000 - "%s": invalid identifier"
*Cause:
*Action:
Error at Line: 179 Column: 27
```

35. SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto; (da la marca, el modelo y el año en que se alquiló un auto)
Resultado:

	MARCA	MODELO	ANO
1	Nissan	Civic	2025
2	Nissan	Versa	2023
3	Nissan	Versa	2023
4	Kia	Rio	2023
5	Kia	Rio	2023
6	Chevrolet	Onix	2023
7	Chevrolet	Onix	2023
8	Fiat	Argo	2023
9	Fiat	Argo	2023
10	Fiat	Argo	2023
11	Volkswagen	Gol	2023
12	Renault	Kwid	2023
13	Renault	Kwid	2023
14	BMW	Serie 3	2023
15	Genesis	G80	2023
16	Volvo	S60	2023
17	Mazda	5	2023

36. SELECT * FROM AlquilerPLSQL WHERE id_cliente = 1; (selecciona todas las variables de la tabla dónde el cliente sea el 1, pero este no existe)
Resultado:

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

37. SELECT * FROM AlquilerPLSQL WHERE id_auto = 1; (selecciona las mismas variables, pero igual, no existe el id 1)

Resultado:

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

38. SELECT * FROM AlquilerPLSQL WHERE id_sucursal = 1; (En esta situación sale error, esto se debe a que se está buscando una columna que no existe en alquilerpls, en este caso id_sucursal)

Resultado:

```
ORA-00904: "ID_SUCURSAL": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error at Line: 160 Column: 35
```

39. SELECT * FROM AlquilerPLSQL WHERE fecha_inicio = '2023-09-27'; (Nada raro, solo filtra por la fecha de inicio proveida)

Resultado:

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

40. SELECT COUNT(*) FROM AlquilerPLSQL; (Cuenta todos los registros en la tabla)

Resultado:

COUNT(*)
1 998

41. 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'; (Da error porque al momento de realizar el segundo Join se están considerando columnas que no existen en la tabla. Si removiéramos el segundo Join en adelante no sale error)

Resultado:

```
ORA-00904: "A"."ID_SUCURSAL": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Error at Line: 207 Column: 113
```

42. SELECT a.marca, a.modelo FROM AutoPLSQL a; (selecciona la marca y modelo de los autos)

Resultado:

	MARCA	MODELO
1	Nissan	Civic
2	Nissan	Versa
3	Kia	Rio
4	Hyundai	Accent
5	Chevrolet	Onix
6	Fiat	Argo
7	Volkswagen	Gol

43. JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto WHERE al.id_cliente = 1 AND al.fecha_inicio = '2023-09-27'; (el código está incompleto, desde el JOIN, porque no dice ni siquiera qué va a juntar. Sale error)

Resultado:

```
Error starting at line : 213 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
```

44. SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio > 7; (Saca el promedio de las fechas donde este sea mayor a 7)

Resultado:

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

45. 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;

Resultado:

```
ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error at Line: 225 Column: 181
```

46. 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;
47. 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: 225 Column: 181
```

48. 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: 225 Column: 181
```

49. 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; (En este caso me aparece un error distinto pero dado por el mismo problema (sintaxis))

Resultado:

```
ORA-00907: missing right parenthesis
00907. 00000 - "missing right parenthesis"
*Cause:
*Action:
Error at Line: 237 Column: 26
```

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

Resultado:

```
ORA-00933: SQL command not properly ended
00933. 00000 - "SQL command not properly ended"
*Cause:
*Action:
Error at Line: 225 Column: 181
```

El Limit 1 es lo que causa error de sintaxis, y lo mismo para estas últimas 5 (honestamente no sé si es por cosa del programa, pero el comando correcto a usar sería 'Where rownum = 1;')