



# CUADERNO PL/SQL

Estudiante:

Maria Alejandra Enriquez Serrano  
[menriquezs@ucentral.edu.co](mailto:menriquezs@ucentral.edu.co)

Curso de Bases de Datos

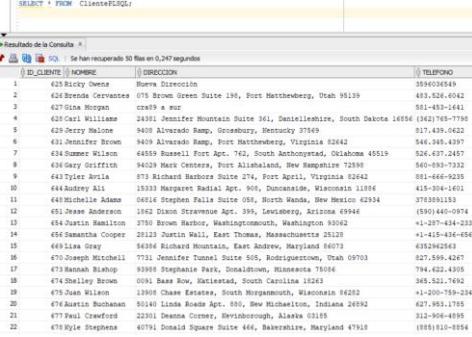
Profesor:

Wilmer Mesias Lopez Lopez

Maestría en Analítica de Datos  
Universidad Central  
Bogotá, Colombia

# EJERCICIOS PRIMER CICLO (1-30):

## 1. Consultas Básicas:

Ejercicio	Resultado	Ánalisis																																																																																												
1. Mostrar todos los clientes en la tabla "Cliente".  SELECT * FROM ClientePLSQL;	 <table border="1"><thead><tr><th>ID_CLIENTE</th><th>NOMBRE</th><th>DIRECCION</th><th>TELEFONO</th></tr></thead><tbody><tr><td>1</td><td>625 Ricky Owens</td><td>Nueva Dirección</td><td>3594034549</td></tr><tr><td>2</td><td>626 Brenda Cervantes</td><td>675 Brown Green Suite 180, Port Matthebweng, Utah 95139</td><td>483.526.4042</td></tr><tr><td>3</td><td>627 Tami</td><td>675 Brown Green Suite 180, Port Matthebweng, Utah 95139</td><td>581-421-7654</td></tr><tr><td>4</td><td>628 Carol Williams</td><td>4381 Jennifer Mountain Suite 361, Danielleshire, South Dakota 16981</td><td>162121765-7798</td></tr><tr><td>5</td><td>629 Jerry Malone</td><td>9405 Alvarado Ramp, Goosbury, Kentucky 37849</td><td>817.429.0422</td></tr><tr><td>6</td><td>631 Jennifer Brown</td><td>9409 Alvarado Ramp, Port Matthebweng, Virginia 32042</td><td>546.345.4397</td></tr><tr><td>7</td><td>634 Summer Wilson</td><td>6459 Russell Port Apt. 762, South Anchorage, Oklahoma 45519</td><td>524.437.2457</td></tr><tr><td>8</td><td>636 Gary Griffith</td><td>84029 Mark Centers, Port Alphaland, New Hampshire 72591</td><td>540-693-7332</td></tr><tr><td>9</td><td>637 Taylor Gillis</td><td>85122 Mark Centers, Port Alphaland, New Hampshire 72591</td><td>851-421-7654</td></tr><tr><td>10</td><td>644 Michael Ali</td><td>13333 Mariana Radial Apt. 969, Dundonville, Wisconsin 11056</td><td>414-404-4621</td></tr><tr><td>11</td><td>645 Michelle Adams</td><td>06418 Stephen Falls Suite 050, North Wanda, New Mexico 62334</td><td>373391153</td></tr><tr><td>12</td><td>651 Jessie Anderson</td><td>1382 Diane Streetview Apt. 390, Levisterg, Arizona 69948</td><td>(590)440-0973</td></tr><tr><td>13</td><td>654 Austin Hamilton</td><td>3780 Brook Harbor, Washingtonsmouth, Washington 93042</td><td>+1-287-414-2337</td></tr><tr><td>14</td><td>656 Samantha Cooper</td><td>28120 Justin Wall, East Thomas, Massachusetts 25120</td><td>+1-415-436-6560</td></tr><tr><td>15</td><td>657 Michaela Anderson</td><td>13333 Mariana Radial Apt. 969, Dundonville, Wisconsin 11056</td><td>414-404-4621</td></tr><tr><td>16</td><td>670 Joseph Mitchell</td><td>7731 Jennifer Tunnel Suite 60, Buderuserson, Utah 69703</td><td>827.599.4247</td></tr><tr><td>17</td><td>673 Hannah Bishop</td><td>93803 Stephanie Park, Donlition, Minnesota 75064</td><td>794.422.4305</td></tr><tr><td>18</td><td>674 Shelley Brown</td><td>0991 Base Row, Kaitiesad, South Carolina 15243</td><td>348.521.7692</td></tr><tr><td>19</td><td>675 Juan Wilson</td><td>13800 Chase Estates, South Mergemont, Wisconsin 86282</td><td>+1-200-759-2343</td></tr><tr><td>20</td><td>676 Austin Bushman</td><td>50140 Linda Roads Apt. 880, New Minneaton, Indiana 24692</td><td>627.953.1785</td></tr><tr><td>21</td><td>677 Paul Crawford</td><td>22301 Deanna Corner, Revioborough, Alaska 33185</td><td>312-904-4995</td></tr><tr><td>22</td><td>678 Kyle Stephens</td><td>6791 Donald Square Suite 446, Bakershire, Maryland 47912</td><td>(888)910-6894</td></tr></tbody></table>	ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO	1	625 Ricky Owens	Nueva Dirección	3594034549	2	626 Brenda Cervantes	675 Brown Green Suite 180, Port Matthebweng, Utah 95139	483.526.4042	3	627 Tami	675 Brown Green Suite 180, Port Matthebweng, Utah 95139	581-421-7654	4	628 Carol Williams	4381 Jennifer Mountain Suite 361, Danielleshire, South Dakota 16981	162121765-7798	5	629 Jerry Malone	9405 Alvarado Ramp, Goosbury, Kentucky 37849	817.429.0422	6	631 Jennifer Brown	9409 Alvarado Ramp, Port Matthebweng, Virginia 32042	546.345.4397	7	634 Summer Wilson	6459 Russell Port Apt. 762, South Anchorage, Oklahoma 45519	524.437.2457	8	636 Gary Griffith	84029 Mark Centers, Port Alphaland, New Hampshire 72591	540-693-7332	9	637 Taylor Gillis	85122 Mark Centers, Port Alphaland, New Hampshire 72591	851-421-7654	10	644 Michael Ali	13333 Mariana Radial Apt. 969, Dundonville, Wisconsin 11056	414-404-4621	11	645 Michelle Adams	06418 Stephen Falls Suite 050, North Wanda, New Mexico 62334	373391153	12	651 Jessie Anderson	1382 Diane Streetview Apt. 390, Levisterg, Arizona 69948	(590)440-0973	13	654 Austin Hamilton	3780 Brook Harbor, Washingtonsmouth, Washington 93042	+1-287-414-2337	14	656 Samantha Cooper	28120 Justin Wall, East Thomas, Massachusetts 25120	+1-415-436-6560	15	657 Michaela Anderson	13333 Mariana Radial Apt. 969, Dundonville, Wisconsin 11056	414-404-4621	16	670 Joseph Mitchell	7731 Jennifer Tunnel Suite 60, Buderuserson, Utah 69703	827.599.4247	17	673 Hannah Bishop	93803 Stephanie Park, Donlition, Minnesota 75064	794.422.4305	18	674 Shelley Brown	0991 Base Row, Kaitiesad, South Carolina 15243	348.521.7692	19	675 Juan Wilson	13800 Chase Estates, South Mergemont, Wisconsin 86282	+1-200-759-2343	20	676 Austin Bushman	50140 Linda Roads Apt. 880, New Minneaton, Indiana 24692	627.953.1785	21	677 Paul Crawford	22301 Deanna Corner, Revioborough, Alaska 33185	312-904-4995	22	678 Kyle Stephens	6791 Donald Square Suite 446, Bakershire, Maryland 47912	(888)910-6894	Se obtiene todos los registros y todas las columnas de la tabla "ClientePLSQL" como resultado
ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO																																																																																											
1	625 Ricky Owens	Nueva Dirección	3594034549																																																																																											
2	626 Brenda Cervantes	675 Brown Green Suite 180, Port Matthebweng, Utah 95139	483.526.4042																																																																																											
3	627 Tami	675 Brown Green Suite 180, Port Matthebweng, Utah 95139	581-421-7654																																																																																											
4	628 Carol Williams	4381 Jennifer Mountain Suite 361, Danielleshire, South Dakota 16981	162121765-7798																																																																																											
5	629 Jerry Malone	9405 Alvarado Ramp, Goosbury, Kentucky 37849	817.429.0422																																																																																											
6	631 Jennifer Brown	9409 Alvarado Ramp, Port Matthebweng, Virginia 32042	546.345.4397																																																																																											
7	634 Summer Wilson	6459 Russell Port Apt. 762, South Anchorage, Oklahoma 45519	524.437.2457																																																																																											
8	636 Gary Griffith	84029 Mark Centers, Port Alphaland, New Hampshire 72591	540-693-7332																																																																																											
9	637 Taylor Gillis	85122 Mark Centers, Port Alphaland, New Hampshire 72591	851-421-7654																																																																																											
10	644 Michael Ali	13333 Mariana Radial Apt. 969, Dundonville, Wisconsin 11056	414-404-4621																																																																																											
11	645 Michelle Adams	06418 Stephen Falls Suite 050, North Wanda, New Mexico 62334	373391153																																																																																											
12	651 Jessie Anderson	1382 Diane Streetview Apt. 390, Levisterg, Arizona 69948	(590)440-0973																																																																																											
13	654 Austin Hamilton	3780 Brook Harbor, Washingtonsmouth, Washington 93042	+1-287-414-2337																																																																																											
14	656 Samantha Cooper	28120 Justin Wall, East Thomas, Massachusetts 25120	+1-415-436-6560																																																																																											
15	657 Michaela Anderson	13333 Mariana Radial Apt. 969, Dundonville, Wisconsin 11056	414-404-4621																																																																																											
16	670 Joseph Mitchell	7731 Jennifer Tunnel Suite 60, Buderuserson, Utah 69703	827.599.4247																																																																																											
17	673 Hannah Bishop	93803 Stephanie Park, Donlition, Minnesota 75064	794.422.4305																																																																																											
18	674 Shelley Brown	0991 Base Row, Kaitiesad, South Carolina 15243	348.521.7692																																																																																											
19	675 Juan Wilson	13800 Chase Estates, South Mergemont, Wisconsin 86282	+1-200-759-2343																																																																																											
20	676 Austin Bushman	50140 Linda Roads Apt. 880, New Minneaton, Indiana 24692	627.953.1785																																																																																											
21	677 Paul Crawford	22301 Deanna Corner, Revioborough, Alaska 33185	312-904-4995																																																																																											
22	678 Kyle Stephens	6791 Donald Square Suite 446, Bakershire, Maryland 47912	(888)910-6894																																																																																											
2. Mostrar todos los autos en la tabla "Auto".  SELECT * FROM AutoPLSQL;	 <table border="1"><thead><tr><th>ID_AUTO</th><th>MARCA</th><th>MODELO</th><th>ANO</th></tr></thead><tbody><tr><td>1</td><td>2 Honda</td><td>Civic</td><td>2025</td></tr><tr><td>2</td><td>3 Nissan</td><td>Versa</td><td>2023</td></tr><tr><td>3</td><td>4 Kia</td><td>Rio</td><td>2023</td></tr><tr><td>4</td><td>6 Chevrolet</td><td>Onix</td><td>2020</td></tr><tr><td>5</td><td>7 Fiat</td><td>Argo</td><td>2023</td></tr><tr><td>6</td><td>9 Renault</td><td>Kwid</td><td>2023</td></tr><tr><td>7</td><td>10 Ford</td><td>Ka</td><td>2023</td></tr><tr><td>8</td><td>11 BMW</td><td>Serie 3</td><td>2020</td></tr><tr><td>9</td><td>16 Genesis</td><td>G80</td><td>2023</td></tr><tr><td>10</td><td>17 Volvo</td><td>S60</td><td>2023</td></tr><tr><td>11</td><td>18 Mazda</td><td>6</td><td>2024</td></tr><tr><td>12</td><td>19 Subaru</td><td>Legacy</td><td>2023</td></tr><tr><td>13</td><td>20 Hyundai</td><td>Sonata</td><td>2023</td></tr><tr><td>14</td><td>21 Kia</td><td>K5</td><td>2023</td></tr><tr><td>15</td><td>22 Honda</td><td>CR-V</td><td>2023</td></tr><tr><td>16</td><td>23 Nissan</td><td>Rogue</td><td>2023</td></tr><tr><td>17</td><td>24 Mazda</td><td>CX-5</td><td>2023</td></tr><tr><td>18</td><td>25 Subaru</td><td>Forester</td><td>2023</td></tr><tr><td>19</td><td>26 Hyundai</td><td>Tucson</td><td>2023</td></tr><tr><td>20</td><td>27 Kia</td><td>Sportage</td><td>2023</td></tr><tr><td>21</td><td>28 Volkswagen</td><td>Tiguan</td><td>2023</td></tr><tr><td>22</td><td>29 Chevrolet</td><td>Equinox</td><td>2023</td></tr></tbody></table>	ID_AUTO	MARCA	MODELO	ANO	1	2 Honda	Civic	2025	2	3 Nissan	Versa	2023	3	4 Kia	Rio	2023	4	6 Chevrolet	Onix	2020	5	7 Fiat	Argo	2023	6	9 Renault	Kwid	2023	7	10 Ford	Ka	2023	8	11 BMW	Serie 3	2020	9	16 Genesis	G80	2023	10	17 Volvo	S60	2023	11	18 Mazda	6	2024	12	19 Subaru	Legacy	2023	13	20 Hyundai	Sonata	2023	14	21 Kia	K5	2023	15	22 Honda	CR-V	2023	16	23 Nissan	Rogue	2023	17	24 Mazda	CX-5	2023	18	25 Subaru	Forester	2023	19	26 Hyundai	Tucson	2023	20	27 Kia	Sportage	2023	21	28 Volkswagen	Tiguan	2023	22	29 Chevrolet	Equinox	2023	Se obtiene todos los registros y todas las columnas de la tabla "AutoPLSQL" como resultado
ID_AUTO	MARCA	MODELO	ANO																																																																																											
1	2 Honda	Civic	2025																																																																																											
2	3 Nissan	Versa	2023																																																																																											
3	4 Kia	Rio	2023																																																																																											
4	6 Chevrolet	Onix	2020																																																																																											
5	7 Fiat	Argo	2023																																																																																											
6	9 Renault	Kwid	2023																																																																																											
7	10 Ford	Ka	2023																																																																																											
8	11 BMW	Serie 3	2020																																																																																											
9	16 Genesis	G80	2023																																																																																											
10	17 Volvo	S60	2023																																																																																											
11	18 Mazda	6	2024																																																																																											
12	19 Subaru	Legacy	2023																																																																																											
13	20 Hyundai	Sonata	2023																																																																																											
14	21 Kia	K5	2023																																																																																											
15	22 Honda	CR-V	2023																																																																																											
16	23 Nissan	Rogue	2023																																																																																											
17	24 Mazda	CX-5	2023																																																																																											
18	25 Subaru	Forester	2023																																																																																											
19	26 Hyundai	Tucson	2023																																																																																											
20	27 Kia	Sportage	2023																																																																																											
21	28 Volkswagen	Tiguan	2023																																																																																											
22	29 Chevrolet	Equinox	2023																																																																																											

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

```
SELECT * FROM AlquilerPLSQL;
```

SELECT * FROM AlquilerPLSQL;					
Resultado de la Consulta					
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
15	598	726	669 09/11/22	23/11/22	598
16	599	999	715 07/10/22	25/10/22	599
17	600	414	809 29/12/22	10/01/23	600
18	601	470	493 02/10/22	16/10/22	601
19	602	830	198 11/08/23	29/08/23	602
20	603	682	526 02/10/22	16/10/22	603
21	604	214	63 20/12/22	24/12/22	604
22	605	171	251 14/06/23	28/06/23	605

Se obtiene todos los registros y todas las columnas de la tabla " AlquilerPLSQL " como resultado

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

```
SELECT * FROM SucursalPLSQL;
```

SELECT * FROM SucursalPLSQL;			
Resultado de la Consulta			
ID_SUCURSAL	NOMBRE	CIUDAD	PAIS
1	584 Smith Ltd	Gardnerstad	Mozambique
2	585 Brown, Bridges and Rice	Jasonfort	Romania
3	586 Thompson-Porter	Lake Kurton	Saint Lucia
4	587 Perez, Farley and Bartlett	Graystad	Liberia
5	588 Schultz LLC	Lake Ryan	Macao
6	589 Morton-Wilson	Haaside	Senegal
7	590 Howell Group	Greensburgh	Uzbekistan
8	591 Clements PLC	West Jaschaven	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	Murch Lisa	Cambodia
13	596 Todd and Sons	New Nathanville	Gambia
14	597 Abbott Ltd	Sparkfurt	Chile
15	598 Stewart-Scott	Lake Robertberg	Sao Tome and Principe
16	599 Simpson, Lee and Weiss	Harrisonstad	Turkmenistan
17	600 Burns, Burnett and Sanders	New Stephanie	Congo
18	601 Daniels and Sons	West Jonathan	Egypt
19	602 Ramirez LLC	Granport	Panama
20	603 Williams-Preston	Andersonville	Kiribati
21	604 Murray-Thomas	Scottfurt	Senegal
22	605 Lozano Inc	Johnsonfort	Netherlands Antilles

Se obtiene todos los registros y todas las columnas de la tabla " SucursalPLSQL " como resultado

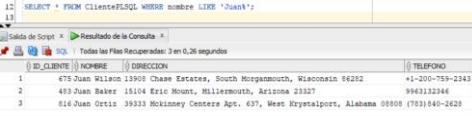
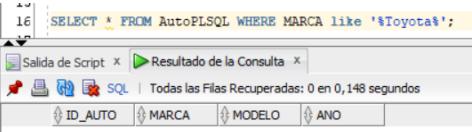
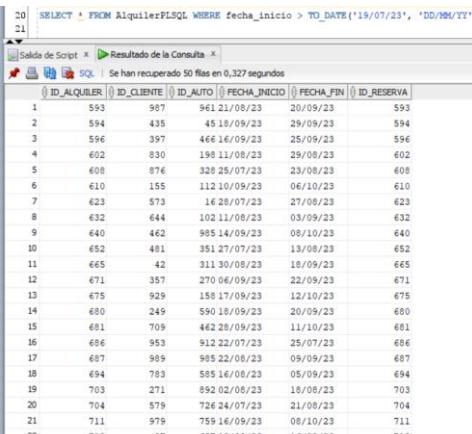
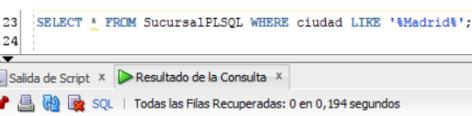
5. Mostrar todas las reservas en la tabla "Reserva".

```
SELECT * FROM ReservaPLSQL;
```

SELECT * FROM ReservaPLSQL;			
Resultado de la Consulta			
ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	863	780	739 16/05/23
2	864	737	729 04/06/23
3	865	700	982 10/12/22
4	866	197	610 29/06/23
5	867	594	755 17/06/23
6	868	808	185 07/03/23
7	869	315	394 19/03/23
8	870	314	627 03/10/22
9	871	408	687 23/12/22
10	872	128	821 22/09/23
11	873	858	380 16/12/22
12	875	256	954 11/08/23
13	876	437	903 21/08/23
14	877	716	325 14/06/23
15	878	708	665 16/08/23
16	879	573	287 13/06/23
17	880	850	568 25/03/23
18	881	647	46 28/05/23
19	882	467	704 12/10/22
20	883	632	890 06/02/23
21	884	997	749 20/12/22
22	885	493	329 07/02/23

Se obtiene todos los registros y todas las columnas de la tabla "ReservaPLSQL" como resultado

## 2. Filtros y Ordenamiento:

Ejercicio	Resultado	Análisis
6. Mostrar los clientes que se llaman "Juan".  SELECT * FROM ClientePLSQL WHERE nombre LIKE 'Juan%';	 12   SELECT * FROM ClientePLSQL WHERE nombre LIKE 'Juan%'; 13   13   Salida de Script   Resultado de la Consulta   13   SQL   Todas las Filas Recuperadas: 3 en 0,26 segundos 13   13   ID_CLIENTE   NOMBRE   DIRECCION   TELFONO   13   1   475 Juan Wilson 13908 Chase Estates, South Moragnouch, Wisconsin 86282 +1-200-759-2343 2   483 Juan Baker 151014 Eric Mount, Millersmouth, Arizona 23327 9961132344 3   816 Juan Ortiz 393133 McKinney Centers Apt. 607, West Krytalport, Alabama 08808 (703)840-2620	Esta consulta devolverá todos los registros de la tabla ClientePLSQL donde el nombre del cliente comience con "Juan".
7. Mostrar los autos de marca "Toyota".  SELECT * FROM AutoPLSQL WHERE MARCA like '%Toyota%';	 16   SELECT * FROM AutoPLSQL WHERE MARCA like '%Toyota%'; 16   16   Salida de Script   Resultado de la Consulta   16   SQL   Todas las Filas Recuperadas: 0 en 0,148 segundos 16   16   ID_AUTO   MARCA   MODELO   ANO   16   16	Esta consulta devolverá todos los registros de la tabla AutoPLSQL cuya marca incluye la palabra "Toyota".
8. Mostrar los alquileres que ocurrieron después de una fecha específica.  SELECT * FROM AlquilerPLSQL WHERE fecha_inicio > TO_DATE('19/07/23', 'DD/MM/YY');	 20   SELECT * FROM AlquilerPLSQL WHERE fecha_inicio > TO_DATE('19/07/23', 'DD/MM/YY'); 21   21   Salida de Script   Resultado de la Consulta   21   SQL   Se han recuperado 50 filas en 0,327 segundos 21   21   ID_ALQUILER   ID_CLIENTE   ID_AUTO   FECHA_INICIO   FECHA_FIN   ID_RESERVA   21   1   593   987   961.21/08/23   20/09/23   593   2   594   435   4518/09/23   29/09/23   594   3   596   397   46616/09/23   25/09/23   596   4   602   830   19611/08/23   29/08/23   602   5   608   876   32025/07/23   23/08/23   608   6   610   155   11210/09/23   06/10/23   610   7   623   573   1628/07/23   27/08/23   623   8   632   644   10211/08/23   03/09/23   632   9   640   462   98514/09/23   08/10/23   640   10   652   401   35127/07/23   13/08/23   652   11   665   42   31130/08/23   18/09/23   665   12   671   357   27006/09/23   22/09/23   671   13   675   929   15817/09/23   12/10/23   675   14   680   249   59018/09/23   20/09/23   680   15   681   709   46228/09/23   11/10/23   681   16   696   953   91222/07/23   25/07/23   696   17   697   989   98522/08/23   09/09/23   687   18   694   703   58516/08/23   05/09/23   694   19   703   271   89202/08/23   18/08/23   703   20   704   579   72624/07/23   21/08/23   704   21   711   979   75916/09/23   08/10/23   711   22   712   97   69703/09/23   16/09/23   712	Esta consulta seleccionará todos los registros de la tabla AlquilerPLSQL donde la fecha de inicio (fecha_inicio) sea posterior al 19 de julio de 2023 (19/07/23).
9. Mostrar las sucursales ubicadas en "Madrid".  SELECT * FROM SucursalPLSQL WHERE ciudad LIKE '%Madrid%';	 23   SELECT * FROM SucursalPLSQL WHERE ciudad LIKE '%Madrid%'; 24   24   Salida de Script   Resultado de la Consulta   24   SQL   Todas las Filas Recuperadas: 0 en 0,194 segundos 24   24   ID_SUCU...   NOMBRE   CIUDAD   PAIS   24   24	Esta consulta devolverá todos los registros de la tabla SucursalPLSQL cuya columna ciudad incluye la palabra "Madrid".

10. Mostrar las reservas realizadas por un cliente específico.

```
SELECT * FROM ReservaPLSQL WHERE id_cliente = 631;
```

The screenshot shows the Oracle SQL Developer interface. On the left, the code editor displays the following SQL query:

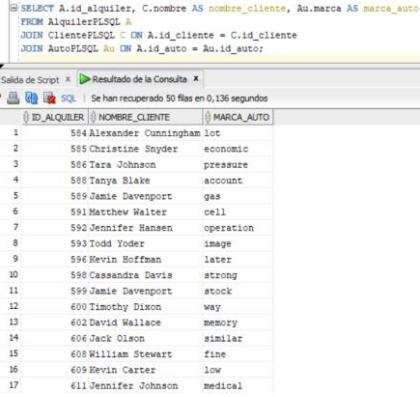
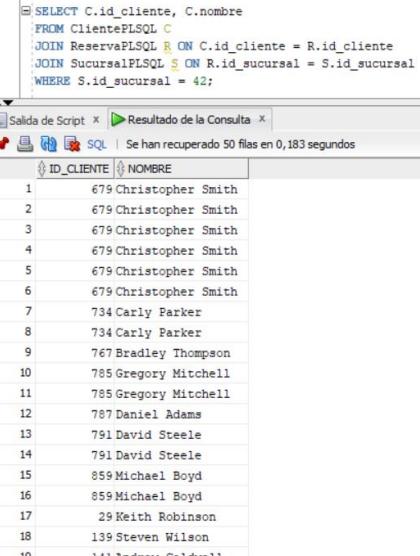
```
25 | SELECT * FROM ReservaPLSQL WHERE id_cliente = 631;
26 |
```

On the right, the results of the query are shown in a grid. The title bar of the results window says "Resultado de la Consulta". Below it, there are tabs for "Salida de Script" and "SQL". A status message at the bottom of the results window reads "Todas las Filas Recuperadas: 7 en 0,247 segundos". The result grid has four columns: ID\_RESERVA, ID\_CLIENTE, ID\_SUCURSAL, and FECHA\_RESERVA. The data is as follows:

ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA
1	286	631	313 21/04/23
2	356	631	343 16/03/23
3	457	631	190 26/07/23
4	2682	631	90 09/04/23
5	1103	631	75 16/05/23
6	1103	631	75 16/05/23
7	2682	631	90 09/04/23

En esta consulta, se seleccionarán todos los registros de la tabla ReservaPLSQL en los que el valor de la columna id\_cliente sea 631.

### 3. Join y Relaciones:

Ejercicio	Resultado	Análisis																				
<p>11. Mostrar los alquileres con los nombres de los clientes y las marcas de los autos.</p> <pre>SELECT A.id_alquiler, C.nombre AS nombre_cliente, Au.marca AS marca_auto FROM AlquilerPLSQL A JOIN ClientePLSQL C ON A.id_cliente = C.id_cliente JOIN AutoPLSQL Au ON A.id_auto = Au.id_auto;</pre>	 <pre>SELECT A.id_alquiler, C.nombre AS nombre_cliente, Au.marca AS marca_auto FROM AlquilerPLSQL A JOIN ClientePLSQL C ON A.id_cliente = C.id_cliente JOIN AutoPLSQL Au ON A.id_auto = Au.id_auto;  Salida de Script   Resultado de la Consulta   Se han recuperado 50 filas en 0,136 segundos ID_ALQUILER   NOMBRE_CLIENTE   MARCA_AUTO 1   584 Alexander Cunningham   lot 2   585 Christine Snyder   economic 3   586 Tara Johnson   pressure 4   588 Tanya Blake   account 5   589 Jamie Davenport   gas 6   591 Matthew Walter   cell 7   592 Jennifer Hansen   operation 8   593 Todd Yoder   image 9   594 Kevin Hoffman   later 10   598 Cassandra Davis   strong 11   599 Jamie Davenport   stock 12   600 Timothy Dixon   way 13   602 David Wallace   memory 14   606 Jack Olson   similar 15   608 William Stewart   fine 16   609 Kevin Carter   low 17   611 Jennifer Johnson   medical</pre>	<p>En esta consulta:</p> <p>Estamos seleccionando el ID del alquiler, el nombre del cliente y la marca del auto.</p> <p>Usamos las cláusulas JOIN para combinar las tablas "AlquilerPLSQL," "ClientePLSQL," y "AutoPLSQL" en función de las relaciones de claves extranjeras. Unimos "AlquilerPLSQL" con "ClientePLSQL" utilizando A.id_cliente = C.id_cliente y luego unimos "AlquilerPLSQL" con "AutoPLSQL" usando A.id_auto = Au.id_auto.</p> <p>Utilizamos el alias AS para darle nombres más legibles a las columnas resultantes, como "nombre_cliente" y "marca_auto."</p>																				
<p>12. Mostrar los clientes que han realizado reservas en una sucursal específica.</p> <pre>SELECT * FROM SucursalPLSQL;</pre> <table border="1"> <thead> <tr> <th>ID_SUCURSAL</th> <th>NOMBRE</th> <th>Ciudad</th> <th>PAIS</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>42 McKay Group</td> <td>Jeffreyton</td> <td>Mongolia</td> </tr> <tr> <td>101</td> <td>45 Nelson Inc</td> <td>Port Scarborough</td> <td>Guadeloupe</td> </tr> <tr> <td>102</td> <td>46 Rodriguez, Scott and Ford</td> <td>Port Alyssa</td> <td>North Macedonia</td> </tr> <tr> <td>103</td> <td>47 Owen Inc</td> <td>Morriston</td> <td>Cyprus</td> </tr> </tbody> </table> <pre>SELECT C.id_cliente, C.nombre FROM ClientePLSQL C JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente JOIN SucursalPLSQL S ON R.id_sucursal = S.id_sucursal WHERE S.id_sucursal = 42;</pre>	ID_SUCURSAL	NOMBRE	Ciudad	PAIS	100	42 McKay Group	Jeffreyton	Mongolia	101	45 Nelson Inc	Port Scarborough	Guadeloupe	102	46 Rodriguez, Scott and Ford	Port Alyssa	North Macedonia	103	47 Owen Inc	Morriston	Cyprus	 <pre>SELECT C.id_cliente, C.nombre FROM ClientePLSQL C JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente JOIN SucursalPLSQL S ON R.id_sucursal = S.id_sucursal WHERE S.id_sucursal = 42;  Salida de Script   Resultado de la Consulta   Se han recuperado 50 filas en 0,183 segundos ID_CLIENTE   NOMBRE 1   679 Christopher Smith 2   679 Christopher Smith 3   679 Christopher Smith 4   679 Christopher Smith 5   679 Christopher Smith 6   679 Christopher Smith 7   734 Carly Parker 8   734 Carly Parker 9   767 Bradley Thompson 10   785 Gregory Mitchell 11   785 Gregory Mitchell 12   787 Daniel Adams 13   791 David Steele 14   791 David Steele 15   859 Michael Boyd 16   859 Michael Boyd 17   29 Keith Robinson 18   139 Steven Wilson 19   141 Andrew Caldwell</pre>	<p>Esta consulta SQL devuelve una lista de clientes que han realizado reservas en la sucursal con el ID 42</p>
ID_SUCURSAL	NOMBRE	Ciudad	PAIS																			
100	42 McKay Group	Jeffreyton	Mongolia																			
101	45 Nelson Inc	Port Scarborough	Guadeloupe																			
102	46 Rodriguez, Scott and Ford	Port Alyssa	North Macedonia																			
103	47 Owen Inc	Morriston	Cyprus																			

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

```
SELECT A.marca, A.modelo, C.nombre
FROM AutoPLSQL A
JOIN AlquilerPLSQL AL ON A.id_auto = AL.id_auto
JOIN ClientePLSQL C ON AL.id_cliente = C.id_cliente;
```

MARCA	MODELO	NOMBRE
1 Honda	Civic	Robert Moran
2 Nissan	Versa	Steven Velasquez
3 Kia	Rio	Gina Morgan
4 Chevrolet	Onix	Stefanie Sullivan
5 Fiat	Argo	Emily Santana
6 Renault	Kwid	Megan West
7 Renault	Kwid	Melinda Jackson
8 BMW	Serie 3	Kimberly Johnson
9 Genesis	G80	Stephanie Clark
10 Mazda	6	Jennifer Gamble
11 Mazda	6	Nicholas Martin
12 Hyundai	Sonata	Michael Koch
13 Kia	K5	Leslie Mccoy
14 name	price	David Schmidt
15 ball	design	Glenda Mendoza
16 thus	and	Sherry Morales
17 thus	and	Mitchell Garcia
18 its	respond	Ashley Farmer
19 traditional	style	Bob Peterson

En esta consulta, seleccionamos la marca y el modelo del automóvil de la tabla AutoPLSQL y el nombre del cliente de la tabla ClientePLSQL. Luego, utilizamos las cláusulas JOIN para combinar las tablas según las relaciones entre AutoPLSQL y AlquilerPLSQL, y entre AlquilerPLSQL y ClientePLSQL.

Esta consulta recuperará una lista de autos alquilados junto con los nombres de los clientes que los alquilaron.

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

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

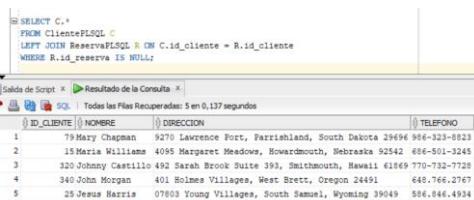
ID_RESERVA	NOMBRE_CLIENTE	CIUDAD_SUCURSAL
1	866 Christopher Bradshaw	Port Joseph
2	870 Brian Smith	North Jordanhaven
3	879 Stephanie Clark	Middletonchester
4	881 Jenny Brown	Port Alyssa
5	885 Sarah Duffy	South Melissa
6	890 Ernest Mora	East Amy
7	893 Erica Morgan	Fosterland
8	901 Danielle Williamson	East Barbara
9	904 Dustin Summers	Port Lawrence
10	916 Ernest Lawson	North Charlesstad
11	917 Richard Evans	New Theresatton
12	919 Benjamin Perez	Port Jadeview
13	921 Jonathan Webster	New Codyview
14	926 Jason Sawyer	Port Lisaborough
15	933 Carol Morales	Powellview
16	938 Kayla Ryan	West Hollyport
17	941 Emily Santana	South Aaronbrough
18	942 Courtney Shannon	Mcpersonbury
19	947 Haley Carroll	West Kathyland

En esta consulta, seleccionamos el ID de la reserva de la tabla ReservaPLSQL, el nombre del cliente de la tabla ClientePLSQL (usando un alias "nombre\_cliente") y la ciudad de la sucursal de la tabla SucursalPLSQL (usando un alias "ciudad\_sucursal"). Luego, utilizamos las cláusulas JOIN para combinar las tablas según las relaciones entre ReservaPLSQL y ClientePLSQL, y entre ReservaPLSQL y SucursalPLSQL.

Esta consulta recuperará una lista de detalles de reservas, incluyendo los nombres de los clientes y las ciudades de las sucursales asociadas a esas reservas.

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

```
SELECT C.*  
FROM ClientePLSQL C  
LEFT JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente  
WHERE R.id_reserva IS NULL;
```



The screenshot shows the Oracle SQL Developer interface. On the left, there is a script editor window with the following SQL code:

```
SELECT C.*  
FROM ClientePLSQL C  
LEFT JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente  
WHERE R.id_reserva IS NULL;
```

On the right, there is a results window titled "Resultado de la Consulta" (Result of the Query) showing the output of the executed query. The results are as follows:

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	Mary Chapman	9270 Lawrence Port, Parrishland, South Dakota 29496	986-323-6823
2	Maria Williams	4095 Margaret Meadow, Howardmouth, Nebraska 92542	686-501-3245
3	Johnny Castillo	492 Sarah Brook Suite 393, Smithmouth, Hawaii 61869	770-732-7720
4	John Morgan	401 Holmes Villages, West Brett, Oregon 24491	645.766.2767
5	Jesus Harris	07803 Young Villages, South Samuel, Wyoming 39049	586.846.4934

En esta consulta, primero seleccionamos todos los registros de la tabla ClientePLSQL (C). Luego, utilizamos una operación LEFT JOIN para intentar unirlos con la tabla ReservaPLSQL (R) utilizando la columna id\_cliente. La condición WHERE R.id\_reserva IS NULL se asegura de que solo seleccionemos los registros de clientes que no tienen una reserva asociada. Estos son los clientes que no han realizado ninguna reserva.

## 4. Agregación y Agrupamiento:

Ejercicio	Resultado	Análisis																																								
16. Contar cuántos autos hay de cada marca en la tabla "Auto".  SELECT marca, COUNT(*) AS cantidad_de_autos FROM AutoPLSQL GROUP BY marca;	<pre>SELECT marca, COUNT(*) AS cantidad_de_autos FROM AutoPLSQL GROUP BY marca;</pre> <table border="1"> <thead> <tr> <th>MARCA</th> <th>CANTIDAD_DE_AUTOS</th> </tr> </thead> <tbody> <tr><td>1 Volkswagen</td><td>1</td></tr> <tr><td>2 ball</td><td>2</td></tr> <tr><td>3 each</td><td>1</td></tr> <tr><td>4 apply</td><td>2</td></tr> <tr><td>5 can</td><td>2</td></tr> <tr><td>6 land</td><td>2</td></tr> <tr><td>7 from</td><td>1</td></tr> <tr><td>8 participant</td><td>1</td></tr> <tr><td>9 main</td><td>4</td></tr> <tr><td>10 major</td><td>1</td></tr> <tr><td>11 realize</td><td>1</td></tr> <tr><td>12 vote</td><td>3</td></tr> <tr><td>13 draw</td><td>2</td></tr> <tr><td>14 smile</td><td>2</td></tr> <tr><td>15 wind</td><td>1</td></tr> <tr><td>16 ago</td><td>2</td></tr> <tr><td>17 yet</td><td>3</td></tr> <tr><td>18 most</td><td>1</td></tr> <tr><td>19 center</td><td>1</td></tr> </tbody> </table>	MARCA	CANTIDAD_DE_AUTOS	1 Volkswagen	1	2 ball	2	3 each	1	4 apply	2	5 can	2	6 land	2	7 from	1	8 participant	1	9 main	4	10 major	1	11 realize	1	12 vote	3	13 draw	2	14 smile	2	15 wind	1	16 ago	2	17 yet	3	18 most	1	19 center	1	En esta consulta, seleccionamos la columna "marca" de la tabla "AutoPLSQL" y utilizamos la función COUNT(*) para contar la cantidad de autos en cada grupo de marca. Luego, la cláusula GROUP BY agrupa los resultados por marca, lo que te dará el número de autos de cada marca en la tabla. Los resultados mostrarán cada marca y la cantidad de autos de esa marca en la tabla.
MARCA	CANTIDAD_DE_AUTOS																																									
1 Volkswagen	1																																									
2 ball	2																																									
3 each	1																																									
4 apply	2																																									
5 can	2																																									
6 land	2																																									
7 from	1																																									
8 participant	1																																									
9 main	4																																									
10 major	1																																									
11 realize	1																																									
12 vote	3																																									
13 draw	2																																									
14 smile	2																																									
15 wind	1																																									
16 ago	2																																									
17 yet	3																																									
18 most	1																																									
19 center	1																																									
17. Calcular la duración promedio de los alquileres.  SELECT ROUND( AVG(TO_NUMBER(TO_DATE(FECHA_INICIO, 'YYYY-MM-DD') - TO_DATE(FECHA_FIN, 'YYYY-MM-DD')))),0 ) AS duracion_promedio_en_dias FROM ALQUILERPLSQL;	<pre>SELECT ROUND( AVG(TO_NUMBER(TO_DATE(FECHA_INICIO, 'YYYY-MM-DD') - TO_DATE(FECHA_FIN, 'YYYY-MM-DD')))),0 ) AS duracion_promedio_en_dias FROM ALQUILERPLSQL;</pre> <table border="1"> <thead> <tr> <th>DURACION_PROMEDIO_EN_DIAS</th> </tr> </thead> <tbody> <tr><td>6</td></tr> </tbody> </table>	DURACION_PROMEDIO_EN_DIAS	6	Esta consulta calcula la diferencia en días entre las fechas de inicio y fin de cada alquiler en la tabla "AlquilerPLSQL". Luego, la función AVG se utiliza para calcular el promedio de esas diferencias, lo que te dará la duración promedio de los alquileres en días.																																						
DURACION_PROMEDIO_EN_DIAS																																										
6																																										

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

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

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

ID_SUCURSAL	NOMBRE_SUCURSAL	NUMERO_DE_RESERVAS
1	Sucursal 1	59
2	Sucursal 2	74
3	Sucursal 3	72
4	Sucursal 4	92
5	Sucursal 5	60
6	Sucursal 6	75
7	Sucursal 7	75
8	Sucursal 8	85
9	Sucursal 9	79
10	Sucursal 10	82
11	Sucursal 11	86
12	Sucursal 12	77
13	Sucursal 13	63
14	Sucursal 14	73
15	Sucursal 15	60
16	Sucursal 16	72
17	Sucursal 17	69
18	Sucursal 18	58
19	Sucursal 19	67

En esta consulta: Se utiliza una operación LEFT JOIN para combinar la tabla "SucursalPLSQL" con la tabla "ReservaPLSQL" utilizando la columna "id\_sucursal" como clave de unión. Luego, utilizamos la cláusula GROUP BY para agrupar las filas por "id\_sucursal" y "nombre\_sucursal" de la tabla de sucursales. Utilizamos la función COUNT para contar el número de reservas en cada grupo y lo renombramos como "numero\_de\_reservas". Finalmente, utilizamos la cláusula ORDER BY para ordenar los resultados por "id\_sucursal".

Esta consulta muestra el número total de reservas realizadas en cada sucursal junto con el nombre de la sucursal.

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

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

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

ID_CLIENTE	NOMBRE_CLIENTE	CANTIDAD_ALQUILERES
601	Jill Sanford	4

En esta consulta: Se utilizó una operación LEFT JOIN para combinar la tabla "ClientePLSQL" con la tabla "AlquilerPLSQL" utilizando la columna "id\_cliente" como clave de unión. Luego, utilizamos la cláusula GROUP BY para agrupar las filas por "id\_cliente" y "nombre\_cliente" de la tabla de clientes. Utilizamos la función COUNT para contar el número de alquileres realizados por cada cliente y lo renombramos como "cantidad\_alquileres". Ordenamos los resultados en orden descendente según la cantidad de alquileres realizados por cada cliente utilizando la cláusula ORDER BY. Finalmente, utilizamos FETCH FIRST 1 ROWS ONLY para obtener solo la primera fila, que será el cliente que ha realizado la mayor cantidad de alquileres.

El cliente con mayor número de alquileres (4 alquileres) es John Greene

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

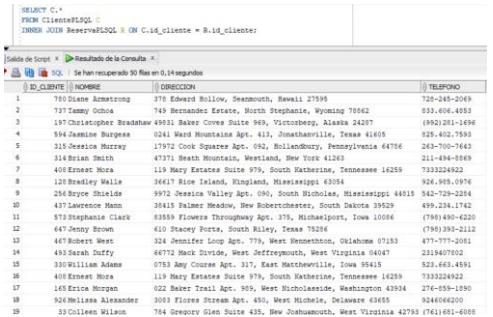
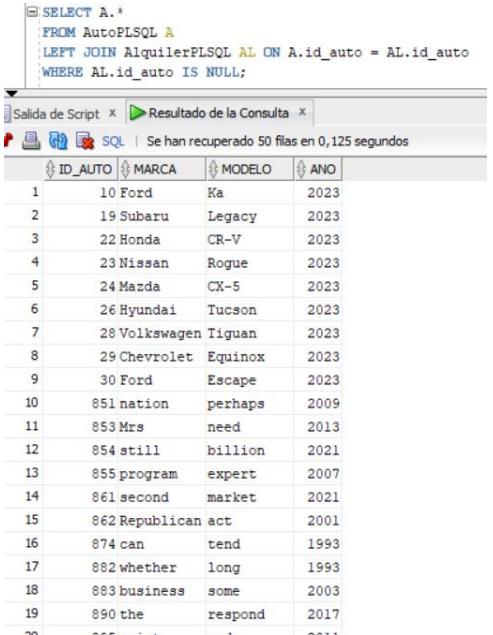
```
SELECT ROUND(AVG(ano)) AS promedio_anos_autos
FROM AutoPLSQL;
```

```
SELECT ROUND(AVG(ano)) AS promedio_anos_autos
FROM AutoPLSQL;
```

PROMEDIO_ANOS_AUTOS
2006

El año promedio de los autos es 2006

## 5. Subconsultas:

Ejercicio	Resultado	Ánalisis
21. Mostrar los clientes que han realizado al menos una reserva.	<pre>SELECT C.* FROM ClientePLSQL C INNER JOIN ReservaPLSQL R ON C.id_cliente = R.id_cliente;</pre> 	<p>En esta consulta, se utiliza una operación INNER JOIN para combinar la tabla "ClientePLSQL" con la tabla "ReservaPLSQL" utilizando la columna "id_cliente" como clave de unión. Seleccionamos todos los campos de la tabla "ClientePLSQL" para los clientes que tienen al menos una reserva asociada.</p>
22. Mostrar los autos que no han sido alquilados aún.	<pre>SELECT A.* FROM AutoPLSQL A LEFT JOIN AlquilerPLSQL AL ON A.id_auto = AL.id_auto WHERE AL.id_auto IS NULL;</pre> 	<p>En esta consulta, se utiliza una operación LEFT JOIN para combinar la tabla "AutoPLSQL" con la tabla "AlquilerPLSQL" utilizando la columna "id_auto" como clave de unión. Utilizamos la cláusula WHERE AL.id_auto IS NULL para seleccionar los autos que no tienen alquileres asociados.</p> <p>Esta consulta mostrará todos los autos que aún no han sido alquilados.</p>

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

```
SELECT c.id_cliente, c.nombre, a.id_auto, COUNT(*) as cantidad_alquileres
FROM ClientePLSQL c
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
GROUP BY c.id_cliente, c.nombre, a.id_auto
HAVING COUNT(*) > 1;
```

ID_CLIENTE	NOMBRE	ID_AUTO	CANTIDAD_ALQUILERES
			1

En esta consulta: Se selecciona la tabla ClientePLSQL y se la une (mediante JOIN) con la tabla AlquilerPLSQL utilizando la columna id\_cliente como el campo de relación. Luego, se agrupa el resultado por id\_cliente, nombre del cliente y id\_auto del auto que han alquilado. La cláusula HAVING se utiliza para filtrar los resultados. La condición COUNT(\*) > 1 verifica si un cliente ha alquilado el mismo auto más de una vez. Si la cuenta de alquileres para un cliente y un auto en particular es mayor que 1, significa que han alquilado el mismo auto más de una vez.

En este caso cada cliente solo ha alquilado una vez el auto

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

```
SELECT DISTINCT c.id_cliente, c.nombre, c.direccion, s.ciudad
FROM ClientePLSQL c
JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
JOIN AutoPLSQL au ON a.id_auto = au.id_auto
JOIN ReservaPLSQL r ON a.id_reserva = r.id_reserva
JOIN SucursalPLSQL s ON r.id_sucursal = s.id_sucursal
WHERE c.direccion = s.ciudad;
```

ID_CLIENTE	NOMBRE	DIRECCION	CIUDAD

Para mostrar los clientes que han realizado alquileres en la misma ciudad en la que viven, se puede puedes usar una consulta SQL que involucre las tablas ClientePLSQL, AlquilerPLSQL y SucursalPLSQL

Pero su resultado servía vacío debido a la estructura de la columna dirección

ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO
1	625 Ricky Owens	Nueva Dirección	3596036549
2	626 Brenda Cervantes	075 Brown Green Suite 198, Port Matthewberg, Utah 95139	483.526.6042
3	627 Gina Morgan	cra89 a sur	581-453-1641
4	628 Carl Williams	24381 Jennifer Mountain Suite 361, Danielleshire, South Dakota 16856 (362)765-7798	817.439.0622
5	629 Jerry Malone	9408 Alvarado Ramp, Grossbury, Kentucky 37569	546.345.4397
6	631 Jennifer Brown	9409 Alvarado Ramp, Port Matthewberg, Virginia 82642	526.637.2457
7	634 Summer Wilson	64559 Russell Fort Apt. 762, South Anthonystad, Oklahoma 45519	560-893-7332
8	636 Gary Griffith	94029 Mark Centers, Port Alishaland, New Hampshire 72598	881-666-9235
9	643 Tyler Avila	Calle 22 sur #315-17	

Como se puede observar en la columna dirección hay mas información aparte de la ciudad

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

```
SELECT R.id_reserva, A.id_auto, A.marca, A.modelo, A.ano,
S.ciudad AS ciudad_sucursal
FROM ReservaPLSQL R
JOIN AlquilerPLSQL ALQ ON R.id_reserva = ALQ.id_reserva
JOIN AutoPLSQL A ON ALQ.id_auto = A.id_auto
JOIN SucursalPLSQL S ON R.id_sucursal = S.id_sucursal
WHERE S.ciudad = S.ciudad;
```

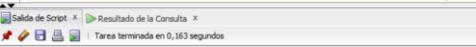
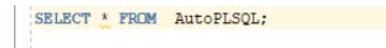
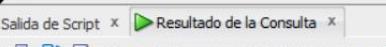
The screenshot shows the SQL tab of the Oracle SQL Developer interface. The query is displayed in the SQL pane, and the results are shown in the Results pane. The results table has columns: ID\_RESERVA, ID\_AUTO, MARCA, MODELO, ANO, and CIUDAD\_SUCURSAL. The data consists of 16 rows of auto rental information.

ID_RESERVA	ID_AUTO	MARCA	MODELO	ANO	CIUDAD_SUCURSAL
1	723	2 Honda	Civic	2025	South Reginaldmouth
2	723	2 Honda	Civic	2025	South Reginaldmouth
3	148	3 Nissan	Versa	2023	Famleshire
4	329	3 Nissan	Versa	2023	Manisales
5	148	3 Nissan	Versa	2023	Famleshire
6	329	3 Nissan	Versa	2023	Manisales
7	293	4 Kia	Rio	2023	Jaconmouth
8	963	4 Kia	Rio	2023	Londres
9	293	4 Kia	Rio	2023	Jaconmouth
10	963	4 Kia	Rio	2023	Londres
11	201	6 Chevrolet	Onix	2020	New Codyview
12	878	6 Chevrolet	Onix	2020	Tonitown
13	201	6 Chevrolet	Onix	2020	New Codyview
14	878	6 Chevrolet	Onix	2020	Tonitown
15	941	7 Fiat	Argo	2023	South Aaronborough
16	28	7 Fiat	Argo	2023	West Ryanbury

En esta consulta: Se selecciona la tabla ReservaPLSQL y se la une (mediante JOIN) con la tabla AlquilerPLSQL utilizando la columna id\_reserva como el campo de relación. Luego, se une la tabla AlquilerPLSQL con la tabla AutoPLSQL usando la columna id\_auto como el campo de relación. También se une la tabla ReservaPLSQL con la tabla SucursalPLSQL utilizando la columna id\_sucursal como el campo de relación. La condición S.ciudad = S.ciudad en la cláusula WHERE garantiza que solo se seleccionen los registros donde la ciudad de la sucursal en la que se realizó la reserva (S.ciudad) sea igual a la ciudad de la sucursal en la que se alquiló el auto (S.ciudad), lo que indica que el auto se alquiló en la misma sucursal de la reserva.

La consulta te devolverá una lista de autos que han sido alquilados en la misma sucursal donde se realizó una reserva, junto con información sobre su identificación, marca, modelo, año y la ciudad de la sucursal correspondiente.

## 6. Actualizaciones y Eliminaciones:

Ejercicio	Resultado	Análisis																																																																																								
26. Actualizar la dirección de un cliente específico.	 <pre>UPDATE ClientePLSQL SET direccion = '7229 Browning Crossing, South Michael, Minnesota 28804 - ACTUALIZADA' WHERE id_cliente = 834;</pre>  <p>Salida de Script   Resultado de la Consulta   Tarea terminada en 0,163 segundos</p> <p>1 fila actualizadas.</p> <table border="1"> <thead> <tr> <th>ID_CLIENTE</th> <th>NOMBRE</th> <th>DIRECCION</th> <th>TELEFONO</th> </tr> </thead> <tbody> <tr> <td>80</td> <td>Erica Torres</td> <td>1216 Parker Creek, East Jasontown, Virginia 61331</td> <td>582-303-4865</td> </tr> <tr> <td>81</td> <td>Jasmine Hayes</td> <td>7229 Browning Crossing, South Michael, Minnesota 28804</td> <td>(256)527-9632</td> </tr> <tr> <td>82</td> <td>James Guzman</td> <td>94365 Lewis Mill Apt. 272, Williamport, Massachusetts 16979</td> <td>408-201-5486</td> </tr> <tr> <td>83</td> <td>Melissa Bolton</td> <td>28946 Rachel Highway Suite 436, Johnport, Oklahoma 10988</td> <td>702-281-6916</td> </tr> <tr> <td>84</td> <td>Lauren Cobb</td> <td>72019 Hall Ramp, West Shannenville, California 83899</td> <td>4442647416</td> </tr> <tr> <td>85</td> <td>Julie Johnson</td> <td>815 Gregory Forest, Port Robertchester, North Carolina 69700</td> <td>735-540-0014</td> </tr> </tbody> </table>	ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO	80	Erica Torres	1216 Parker Creek, East Jasontown, Virginia 61331	582-303-4865	81	Jasmine Hayes	7229 Browning Crossing, South Michael, Minnesota 28804	(256)527-9632	82	James Guzman	94365 Lewis Mill Apt. 272, Williamport, Massachusetts 16979	408-201-5486	83	Melissa Bolton	28946 Rachel Highway Suite 436, Johnport, Oklahoma 10988	702-281-6916	84	Lauren Cobb	72019 Hall Ramp, West Shannenville, California 83899	4442647416	85	Julie Johnson	815 Gregory Forest, Port Robertchester, North Carolina 69700	735-540-0014	Esta consulta SQL actualiza la dirección del cliente con el ID 834 con la nueva dirección proporcionada																																																												
ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO																																																																																							
80	Erica Torres	1216 Parker Creek, East Jasontown, Virginia 61331	582-303-4865																																																																																							
81	Jasmine Hayes	7229 Browning Crossing, South Michael, Minnesota 28804	(256)527-9632																																																																																							
82	James Guzman	94365 Lewis Mill Apt. 272, Williamport, Massachusetts 16979	408-201-5486																																																																																							
83	Melissa Bolton	28946 Rachel Highway Suite 436, Johnport, Oklahoma 10988	702-281-6916																																																																																							
84	Lauren Cobb	72019 Hall Ramp, West Shannenville, California 83899	4442647416																																																																																							
85	Julie Johnson	815 Gregory Forest, Port Robertchester, North Carolina 69700	735-540-0014																																																																																							
27. Eliminar un auto de la tabla "Auto".	 <pre>DELETE FROM AutoPLSQL WHERE id_auto = 10;</pre>  <p>Salida de Script   Resultado de la Consulta   Tarea terminada</p> <p>filas eliminado</p>  <p>SELECT * FROM AutoPLSQL;</p>  <p>Salida de Script   Resultado de la Consulta   Tarea terminada en 0,138 s</p> <table border="1"> <thead> <tr> <th>ID_AUTO</th> <th>MARCA</th> <th>MODELO</th> <th>ANO</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2 Honda</td> <td>Civic</td> <td>2025</td> </tr> <tr> <td>2</td> <td>3 Nissan</td> <td>Versa</td> <td>2023</td> </tr> <tr> <td>3</td> <td>4 Kia</td> <td>Rio</td> <td>2023</td> </tr> <tr> <td>4</td> <td>6 Chevrolet</td> <td>Onix</td> <td>2020</td> </tr> <tr> <td>5</td> <td>7 Fiat</td> <td>Argo</td> <td>2023</td> </tr> <tr> <td>6</td> <td>9 Renault</td> <td>Kwid</td> <td>2023</td> </tr> <tr> <td>7</td> <td>10 Ford</td> <td>Ka</td> <td>2023</td> </tr> <tr> <td>8</td> <td>11 BMW</td> <td>Serie 3</td> <td>2020</td> </tr> <tr> <td>9</td> <td>16 Genesis</td> <td>G80</td> <td>2023</td> </tr> <tr> <td>10</td> <td>17 Volvo</td> <td>S60</td> <td>2023</td> </tr> <tr> <td>11</td> <td>18 Mazda</td> <td>6</td> <td>2024</td> </tr> <tr> <td>12</td> <td>19 Subaru</td> <td>Legacy</td> <td>2023</td> </tr> <tr> <td>13</td> <td>20 Hyundai</td> <td>Sonata</td> <td>2023</td> </tr> <tr> <td>14</td> <td>21 Kia</td> <td>K5</td> <td>2023</td> </tr> <tr> <td>15</td> <td>22 Honda</td> <td>CR-V</td> <td>2023</td> </tr> <tr> <td>16</td> <td>23 Nissan</td> <td>Rogue</td> <td>2023</td> </tr> <tr> <td>17</td> <td>24 Mazda</td> <td>CX-5</td> <td>2023</td> </tr> <tr> <td>18</td> <td>25 Subaru</td> <td>Forester</td> <td>2023</td> </tr> <tr> <td>19</td> <td>26 Hyundai</td> <td>Tucson</td> <td>2023</td> </tr> <tr> <td>20</td> <td>27 Kia</td> <td>Sportage</td> <td>2023</td> </tr> <tr> <td>21</td> <td>28 Volkswagen</td> <td>Tiguan</td> <td>2023</td> </tr> </tbody> </table>	ID_AUTO	MARCA	MODELO	ANO	1	2 Honda	Civic	2025	2	3 Nissan	Versa	2023	3	4 Kia	Rio	2023	4	6 Chevrolet	Onix	2020	5	7 Fiat	Argo	2023	6	9 Renault	Kwid	2023	7	10 Ford	Ka	2023	8	11 BMW	Serie 3	2020	9	16 Genesis	G80	2023	10	17 Volvo	S60	2023	11	18 Mazda	6	2024	12	19 Subaru	Legacy	2023	13	20 Hyundai	Sonata	2023	14	21 Kia	K5	2023	15	22 Honda	CR-V	2023	16	23 Nissan	Rogue	2023	17	24 Mazda	CX-5	2023	18	25 Subaru	Forester	2023	19	26 Hyundai	Tucson	2023	20	27 Kia	Sportage	2023	21	28 Volkswagen	Tiguan	2023	Esta consulta SQL eliminará el registro de la tabla "AutoPLSQL" con un ID igual a 10.
ID_AUTO	MARCA	MODELO	ANO																																																																																							
1	2 Honda	Civic	2025																																																																																							
2	3 Nissan	Versa	2023																																																																																							
3	4 Kia	Rio	2023																																																																																							
4	6 Chevrolet	Onix	2020																																																																																							
5	7 Fiat	Argo	2023																																																																																							
6	9 Renault	Kwid	2023																																																																																							
7	10 Ford	Ka	2023																																																																																							
8	11 BMW	Serie 3	2020																																																																																							
9	16 Genesis	G80	2023																																																																																							
10	17 Volvo	S60	2023																																																																																							
11	18 Mazda	6	2024																																																																																							
12	19 Subaru	Legacy	2023																																																																																							
13	20 Hyundai	Sonata	2023																																																																																							
14	21 Kia	K5	2023																																																																																							
15	22 Honda	CR-V	2023																																																																																							
16	23 Nissan	Rogue	2023																																																																																							
17	24 Mazda	CX-5	2023																																																																																							
18	25 Subaru	Forester	2023																																																																																							
19	26 Hyundai	Tucson	2023																																																																																							
20	27 Kia	Sportage	2023																																																																																							
21	28 Volkswagen	Tiguan	2023																																																																																							

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

```
UPDATE AlquilerPLSQL
SET fecha_fin = TO_DATE('2023-11-05', 'YYYY-MM-DD')
WHERE id_reserva = 644;
```

Antes:

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
	55	638	548	723 01/05/23	17/05/23	638
	56	639	886	704 07/11/22	13/11/22	639
	57	640	462	985 14/09/23	08/10/23	640
	58	641	888	701 03/10/22	14/10/22	641
	59	642	297	263 23/01/23	21/02/23	642
	60	643	452	656 08/05/23	28/05/23	643
	61	644	359	184 13/07/23	07/08/23	644
	62	645	228	679 07/11/22	28/11/22	645

```
UPDATE AlquilerPLSQL
SET fecha_fin = TO_DATE('2023-11-05', 'YYYY-MM-DD')
WHERE id_reserva = 644;
```

Despues

	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
	58	641	888	701 03/10/22	14/10/22	641
	59	642	297	263 23/01/23	21/02/23	642
	60	643	452	656 08/05/23	28/05/23	643
	61	644	359	184 13/07/23	05/11/23	644
	62	645	228	679 07/11/22	28/11/22	645
	63	646	473	584 01/05/23	31/05/23	646

Esto marcará la reserva como completada y establecerá la fecha de finalización en el momento actual.

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

SELECT * FROM ReservaPLSQL;																											
Salida de Script x Resultado de la Consulta x																											
SQL   Se han recuperado 150 filas en 1,09 segundos																											
<table border="1"> <thead> <tr> <th>ID_RESERVA</th><th>ID_CLIENTE</th><th>ID_SUCURSAL</th><th>FECHA_RESERVA</th></tr> </thead> <tbody> <tr> <td>1</td><td>863</td><td>780</td><td>739 16/05/23</td></tr> <tr> <td>2</td><td>864</td><td>737</td><td>729 04/06/23</td></tr> <tr> <td>3</td><td>865</td><td>700</td><td>982 10/12/22</td></tr> <tr> <td>4</td><td>866</td><td>197</td><td>610 29/06/23</td></tr> <tr> <td>5</td><td>867</td><td>594</td><td>755 17/06/23</td></tr> </tbody> </table>				ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA	1	863	780	739 16/05/23	2	864	737	729 04/06/23	3	865	700	982 10/12/22	4	866	197	610 29/06/23	5	867	594	755 17/06/23
ID_RESERVA	ID_CLIENTE	ID_SUCURSAL	FECHA_RESERVA																								
1	863	780	739 16/05/23																								
2	864	737	729 04/06/23																								
3	865	700	982 10/12/22																								
4	866	197	610 29/06/23																								
5	867	594	755 17/06/23																								

```
DELETE FROM ReservaPLSQL
WHERE id_cliente = 700;
```

```
DELETE FROM ReservaPLSQL
WHERE id_cliente = 700;

1 fila actualizadas.

4 filas eliminado
```

Esta consulta SQL eliminará todos los registros de la tabla "ReservaPLSQL" donde el ID de cliente sea igual a 700, lo que eliminará todas las reservas realizadas por ese cliente.

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

```
UPDATE AutoPLSQL
SET ano = 2024
WHERE id_auto = 4;
```

Antes:

```
SELECT * FROM AutoPLSQL;
```

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2025
2	3 Nissan	Versa	2023
3	4 Kia	Rio	2023
4	6 Chevrolet	Onix	2020

En esta consulta, se utiliza la cláusula SET para establecer el valor de la columna "ano" en 2024.

```
UPDATE AutoPLSQL  
SET ano = 2024  
WHERE id_auto = 4;
```



1 fila actualizadas.

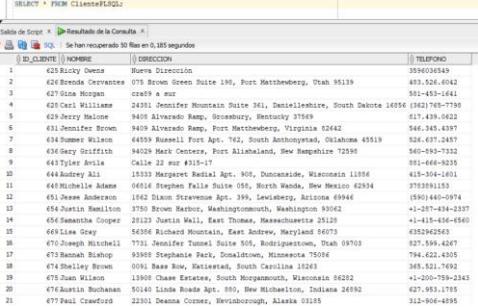
Despues:

```
SELECT * FROM AutoPLSQL;
```

A screenshot of the Oracle SQL Developer interface showing the results of a SELECT query. The results are displayed in a table titled "Salida de Script" with a tab for "Resultado de la Consulta". The message "Se han recuperado 50 filas en 0,115s" is shown. The table has columns: ID\_AUTO, MARCA, MODELO, and ANO. The row where ID\_AUTO is 3 and MARCA is Kia is highlighted with a red border.

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2025
2	3 Nissan	Versa	2023
3	4 Kia	Rio	2024
4	6 Chevrolet	Onix	2020

## EJERCICIOS SEGUNDO CICLO (31-80):

Ejercicio	Resultado	Análisis
31. SELECT * FROM ClientePLSQL;		se utiliza para recuperar todos los registros y columnas de la tabla llamada "ClientePLSQL".
32. SELECT * FROM AutoPLSQL;		se utiliza para recuperar todos los registros y columnas de la tabla llamada "AutoPLSQL".

33. 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
15	598	726	669 09/11/22	23/11/22	598
16	599	999	715 07/10/22	25/10/22	599
17	600	414	809 29/12/22	10/01/23	600
18	601	470	493 02/10/22	16/10/22	601
19	602	830	198 11/08/23	29/08/23	602
20	603	682	526 02/10/22	16/10/22	603

se utiliza para recuperar todos los registros y columnas de la tabla llamada " AlquilerPLSQL " .

34. SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id\_cliente = a.id\_cliente;

```
SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente;
ORA-00936: "A" is not a valid identifier
00936. 0000 - "%s: invalid identifier"
Cause:
        No value
Error at line: 15, column: 27
```

#### Solución

NOMBRE	MARCA	MODELO
1 Robert Mora	Honda	Civic
2 Steven Velasquez	Nissan	Versa
3 Gina Morgan	Kia	Rio
4 Stefanie Sullivan	Chevrolet	Onix
5 Emily Santana	Fiat	Argo
6 Megan West	Renault	Kwid
7 Melinda Jackson	Renault	Kwid
8 Kimberly Johnson	BMW	Serie 3
9 Stephanie Clark	Genesis	G80
10 Jennifer Gamble	Mazda	6
11 Nicholas Martin	Mazda	6
12 Michael Koch	Hyundai	Sonata
13 Leslie McCoy	Kia	K5
14 David Schmidt	name	price
15 Glenda Mendoza	ball	design
16 Sherry Morales	thus	and
17 Mitchell Garcia	thus	and
18 Ashley Farmer	its	respond

El error se debe a que se está intentando unir la tabla ClientePLSQL con la tabla AlquilerPLSQL utilizando la columna id\_cliente como clave de unión, pero no has especificado cómo hacer la unión con la tabla AutoPLSQL.

#### Solución:

En esta consulta, se realiza una unión de tres tablas: ClientePLSQL, AlquilerPLSQL y AutoPLSQL. La primera unión se realiza entre ClientePLSQL y AlquilerPLSQL utilizando la columna id\_cliente como clave de unión, y la segunda unión se realiza entre AlquilerPLSQL y AutoPLSQL utilizando la columna id\_auto como clave de unión. Esto te permitirá obtener el nombre del cliente, la marca y el modelo del auto en la misma consulta.

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

MARCA	MODELO	ANO
1 Honda	Civic	2023
2 Nissan	Versa	2023
3 Nissan	Versa	2023
4 Kia	Rio	2024
5 Kia	Rio	2024
6 Chevrolet	Onix	2020
7 Chevrolet	Onix	2020
8 Fiat	Argo	2023
9 Fiat	Argo	2023
10 Fiat	Argo	2023
11 Renault	Kwid	2023
12 Renault	Kwid	2023
13 BMW	Serie 3	2020
14 Genesis	G80	2023
15 Manda	6	2024
16 Manda	6	2024
17 Manda	6	2024
18 Hyundai	Sonata	2023
19 Kia	K5	2023

Esta consulta da como resultado una lista de autos que están siendo alquilados en la tabla AlquilerPLSQL, con información sobre su marca, modelo y año.

36. `SELECT * FROM AlquilerPLSQL WHERE id_cliente = 1;`

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA

El resultado de esta consulta es una lista de todos los registros de alquiler que corresponden al cliente con id\_cliente igual a 1. En este caso el cliente con ID igual a 1 no tiene ningún registro de haber alquilado algún auto.

37. `SELECT * FROM AlquilerPLSQL WHERE id_auto = 1;`

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA

El resultado de esta consulta será una lista de todos los registros de alquiler que están asociados al auto con id\_auto igual a 1. En este caso el auto con ID igual a 1 no tiene ningún registro de haber sido alquilado.

38. `SELECT * FROM AlquilerPLSQL WHERE id_sucursal = 1;`

ORA-00904: "ID\_SUCURSAL": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:  
Error en la línea: 171, columna: 35

El error está relacionado con el hecho de que la columna id\_sucursal no existe en la tabla 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

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

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

ORA-01861: el literal no coincide con la cadena de formato
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 error en la consulta es debido al formato de la fecha en la base de datos o al tipo de datos de la columna fecha\_inicio.

Solución:

```
SELECT * FROM AlquilerPLSQL WHERE fecha_inicio = TO_DATE('27/09/23', 'DD/MM/YY');
```

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

40. `SELECT COUNT(*) FROM AlquilerPLSQL;`

```
SELECT COUNT(*) FROM AlquilerPLSQL;
```

COUNT(*)	
1	993

La consulta se utiliza para contar el número de registros (filas) en la tabla AlquilerPLSQL.

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';`

```
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": identificador no válido
00904. 00000 - "%s: invalid identifier"
Cause:
Action:
Error en la línea: 183, columna: 25
```

Esta consulta selecciona el nombre de los clientes que tienen alquileres en la sucursal llamada 'Sucursal Central'.

El error está relacionado con el hecho de que la columna id\_sucursal no existe en la tabla AlquilerPLSQL.

Por otra parte, en este caso 'Sucursal Central' no existe en la tabla "SucursalPLSQL" por lo tanto no tiene un ID\_SUCURSAL

42. `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';`

```
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
-------	--------

El resultado de esta consulta será una lista de los autos (marca y modelo) que fueron alquilados por el cliente con id\_cliente igual a 1 en la fecha de inicio '2023-09-27'.

El resultado es vacío debido a que ID\_CLIENTE = 1 no existe en la tabla "ClientePLSQL"

		<pre>SELECT * FROM ClientePLSQL WHERE ID_CLIENTE=1;</pre> <p>Salida de Script   Resultado de la Consulta   SQL   Todas las Filas Recuperadas: 0 en 0,416 segundos</p> <table border="1"> <thead> <tr> <th>ID_CLIENTE</th> <th>NOMBRE</th> <th>DIRECCION</th> <th>TELEFONO</th> </tr> </thead> </table>	ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO																																																																																																																				
ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO																																																																																																																							
43. SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio > 7;	<pre>SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio &gt; 7;</pre> <p>Salida de Script   Resultado de la Consulta   SQL   Se han recuperado 50 filas en 0,251 segundos</p> <table border="1"> <thead> <tr> <th>ID_ALQUILER</th> <th>ID_CLIENTE</th> <th>ID_AUTO</th> <th>FECHA_INICIO</th> <th>FECHA_FIN</th> <th>ID_RESERVA</th> </tr> </thead> <tbody> <tr><td>1</td><td>594</td><td>131</td><td>208/01/10/22</td><td>01/11/23</td><td>594</td></tr> <tr><td>2</td><td>585</td><td>394</td><td>543/19/04/23</td><td>11/05/23</td><td>585</td></tr> <tr><td>3</td><td>586</td><td>774</td><td>345/19/07/23</td><td>30/07/23</td><td>586</td></tr> <tr><td>4</td><td>587</td><td>81</td><td>250/26/03/23</td><td>22/04/23</td><td>587</td></tr> <tr><td>5</td><td>589</td><td>999</td><td>584/01/05/23</td><td>21/05/23</td><td>589</td></tr> <tr><td>6</td><td>590</td><td>796</td><td>711/26/05/23</td><td>17/06/23</td><td>590</td></tr> <tr><td>7</td><td>592</td><td>223</td><td>847/27/11/22</td><td>14/12/22</td><td>592</td></tr> <tr><td>8</td><td>593</td><td>987</td><td>961/21/08/23</td><td>20/09/23</td><td>593</td></tr> <tr><td>9</td><td>594</td><td>435</td><td>45/18/09/23</td><td>29/09/23</td><td>594</td></tr> <tr><td>10</td><td>595</td><td>81</td><td>435/06/05/23</td><td>31/05/23</td><td>595</td></tr> <tr><td>11</td><td>596</td><td>397</td><td>466/16/09/23</td><td>25/09/23</td><td>596</td></tr> <tr><td>12</td><td>597</td><td>36</td><td>473/06/02/23</td><td>01/03/23</td><td>597</td></tr> <tr><td>13</td><td>598</td><td>726</td><td>669/09/11/22</td><td>23/11/22</td><td>598</td></tr> <tr><td>14</td><td>599</td><td>999</td><td>715/07/10/22</td><td>25/10/22</td><td>599</td></tr> <tr><td>15</td><td>600</td><td>414</td><td>809/29/12/22</td><td>10/01/23</td><td>600</td></tr> <tr><td>16</td><td>601</td><td>470</td><td>493/02/10/22</td><td>16/10/22</td><td>601</td></tr> <tr><td>17</td><td>602</td><td>830</td><td>198/11/08/23</td><td>29/08/23</td><td>602</td></tr> <tr><td>18</td><td>603</td><td>682</td><td>526/02/10/22</td><td>16/10/22</td><td>603</td></tr> <tr><td>19</td><td>605</td><td>171</td><td>251/14/06/23</td><td>28/06/23</td><td>605</td></tr> </tbody> </table>	ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA	1	594	131	208/01/10/22	01/11/23	594	2	585	394	543/19/04/23	11/05/23	585	3	586	774	345/19/07/23	30/07/23	586	4	587	81	250/26/03/23	22/04/23	587	5	589	999	584/01/05/23	21/05/23	589	6	590	796	711/26/05/23	17/06/23	590	7	592	223	847/27/11/22	14/12/22	592	8	593	987	961/21/08/23	20/09/23	593	9	594	435	45/18/09/23	29/09/23	594	10	595	81	435/06/05/23	31/05/23	595	11	596	397	466/16/09/23	25/09/23	596	12	597	36	473/06/02/23	01/03/23	597	13	598	726	669/09/11/22	23/11/22	598	14	599	999	715/07/10/22	25/10/22	599	15	600	414	809/29/12/22	10/01/23	600	16	601	470	493/02/10/22	16/10/22	601	17	602	830	198/11/08/23	29/08/23	602	18	603	682	526/02/10/22	16/10/22	603	19	605	171	251/14/06/23	28/06/23	605	<p>La consulta realiza una operación de selección en la tabla AlquilerPLSQL para encontrar registros donde la diferencia entre las fechas de finalización (fecha_fin) y de inicio (fecha_inicio) sea mayor que 7 días.</p>
ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA																																																																																																																					
1	594	131	208/01/10/22	01/11/23	594																																																																																																																					
2	585	394	543/19/04/23	11/05/23	585																																																																																																																					
3	586	774	345/19/07/23	30/07/23	586																																																																																																																					
4	587	81	250/26/03/23	22/04/23	587																																																																																																																					
5	589	999	584/01/05/23	21/05/23	589																																																																																																																					
6	590	796	711/26/05/23	17/06/23	590																																																																																																																					
7	592	223	847/27/11/22	14/12/22	592																																																																																																																					
8	593	987	961/21/08/23	20/09/23	593																																																																																																																					
9	594	435	45/18/09/23	29/09/23	594																																																																																																																					
10	595	81	435/06/05/23	31/05/23	595																																																																																																																					
11	596	397	466/16/09/23	25/09/23	596																																																																																																																					
12	597	36	473/06/02/23	01/03/23	597																																																																																																																					
13	598	726	669/09/11/22	23/11/22	598																																																																																																																					
14	599	999	715/07/10/22	25/10/22	599																																																																																																																					
15	600	414	809/29/12/22	10/01/23	600																																																																																																																					
16	601	470	493/02/10/22	16/10/22	601																																																																																																																					
17	602	830	198/11/08/23	29/08/23	602																																																																																																																					
18	603	682	526/02/10/22	16/10/22	603																																																																																																																					
19	605	171	251/14/06/23	28/06/23	605																																																																																																																					
44. 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;	<pre>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;</pre> <p>Salida de Script   Resultado de la Consulta   SQL   En Ejecución:SELECT c.nombre, COUNT(*) AS numero_alquileres F...</p> <p>ORA-00933: comando SQL no terminado correctamente 00933. 00000 - "SQL command not properly ended" *Cause: *Action: Error en la línea: 200, columna: 51</p>	<p>La consulta cuenta el número de alquileres de cada cliente y luego mostrar el cliente con el mayor número de alquileres.</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p>Solución:</p> <pre>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;</pre> <p>Salida de Script   Resultado de la Consulta   SQL   Todas las Filas Recuperadas: 1 en 0,295 segundos</p> <table border="1"> <thead> <tr> <th>NOMBRE</th> <th>NUMERO_ALQUILERES</th> </tr> </thead> <tbody> <tr><td>John Greene</td><td>6</td></tr> </tbody> </table>	NOMBRE	NUMERO_ALQUILERES	John Greene	6																																																																																																																				
NOMBRE	NUMERO_ALQUILERES																																																																																																																									
John Greene	6																																																																																																																									

45. `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;`

```
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;
```

Esta consulta SQL devuelve la marca y modelo del automóvil más alquilado en la base de datos, junto con la cantidad de alquileres que tiene.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Solución:

```
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

Esta consulta SQL devuelve el nombre de la sucursal con el mayor número de alquileres registrados en la base de datos, junto con la cantidad de alquileres que tiene.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Además, El error está relacionado con el hecho de que la columna id\_sucursal no existe en la tabla AlquilerPLSQL.

46. `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;`

```
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;
```

Esta consulta SQL devuelve el nombre de la sucursal con el mayor número de alquileres registrados en la base de datos, junto con la cantidad de alquileres que tiene.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Además, El error está relacionado con el hecho de que la columna id\_sucursal no existe en la tabla AlquilerPLSQL.

47. `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;`

```
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;
```

Este código SQL devuelve el mes con el mayor número de alquileres registrados en la tabla "AlquilerPLSQL" y muestra la cantidad de alquileres realizados en ese mes.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Solución:

		<pre>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;</pre> <table border="1"> <thead> <tr> <th>MES</th> <th>NUMERO_ALQUILERES</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>100</td> </tr> </tbody> </table>	MES	NUMERO_ALQUILERES	12	100
MES	NUMERO_ALQUILERES					
12	100					
48. 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;	<pre>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;</pre>	<p>Este código SQL devuelve el día de la semana con el mayor número de alquileres registrados en la tabla "AlquilerPLSQL" y muestra la cantidad de alquileres realizados en ese día.</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p><b>Solución:</b></p> <pre>260   SELECT TO_NUMBER(TO_CHAR(fecha_inicio, 'D')) AS dia_semana, COUNT(*) AS numero_alquileres 261   FROM AlquilerPLSQL 262   GROUP BY TO_NUMBER(TO_CHAR(fecha_inicio, 'D')) 263   ORDER BY numero_alquileres DESC 264   fetch first 1 rows only;</pre> <table border="1"> <thead> <tr> <th>DIA_SEMANA</th> <th>NUMERO_ALQUILERES</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>154</td> </tr> </tbody> </table>	DIA_SEMANA	NUMERO_ALQUILERES	1	154
DIA_SEMANA	NUMERO_ALQUILERES					
1	154					
49. SELECT * FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;	<pre>266   SELECT * FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;</pre>	<p>Esta consulta SQL devuelve la fila de la tabla "AlquilerPLSQL" que tiene el precio más alto registrado.</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p>Además, en la tabla "AlquilerPLSQL" no existe la columna "precio"</p>				

		<pre>271   SELECT * FROM AlquilerPLSQL; 272  </pre> <p>Salida de Script x Resultado de la Consulta x   SQL   Se han recuperado 50 filas en 0,138 segundos</p> <table border="1"> <thead> <tr> <th>ID_ALQUILER</th> <th>ID_CLIENTE</th> <th>ID_AUTO</th> <th>FECHA_INICIO</th> <th>FECHA_FIN</th> <th>ID_RESERVA</th> </tr> </thead> <tbody> <tr><td>1</td><td>584</td><td>131</td><td>208 01/10/22</td><td>01/11/23</td><td>584</td></tr> <tr><td>2</td><td>585</td><td>394</td><td>543 19/04/23</td><td>11/05/23</td><td>585</td></tr> <tr><td>3</td><td>586</td><td>774</td><td>345 19/07/23</td><td>30/07/23</td><td>586</td></tr> <tr><td>4</td><td>587</td><td>81</td><td>250 26/03/23</td><td>22/04/23</td><td>587</td></tr> </tbody> </table>	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
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																											
50. SELECT * FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1;	<pre>274   SELECT * FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1; 275  </pre> <p>Salida de Script x Resultado de la Consulta x   SQL   En Ejecución:SELECT * FROM AlquilerPLSQL ORDER BY precio AS ORA-00933: comando SQL no terminado correctamente 00933. 00000 - "SQL command not properly ended" *Cause: *Action: Error en la línea: 273, columna: 49</p>	<p>Esta consulta SQL devuelve la fila de la tabla "AlquilerPLSQL" que tiene el precio más bajo registrado.</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p>Además, en la tabla "AlquilerPLSQL" no existe la columna "precio"</p>																														
51. SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%';	<pre>277   SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%'; 278  </pre> <p>Salida de Script x Resultado de la Consulta x   SQL   Todos los Filtros Recuperados: 3 en 0,22 segundos</p> <table border="1"> <thead> <tr> <th>ID_CLIENTE</th> <th>NOMBRE</th> <th>DIRECCION</th> <th>TELEFONO</th> </tr> </thead> <tbody> <tr><td>1</td><td>475 Juan Wilson</td><td>1348 Chase Estates, South Morganmouth, Wisconsin 84282</td><td>+1-200-759-2343</td></tr> <tr><td>2</td><td>453 Juan Baker</td><td>15104 Eric Mount, Millersmouth, Arizona 23327</td><td>9943112344</td></tr> <tr><td>3</td><td>816 Juan Ortiz</td><td>39353 Molinley Cestera Apt. 457, West Krytalport, Alabama 08508 (783)840-2428</td><td></td></tr> </tbody> </table>	ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO	1	475 Juan Wilson	1348 Chase Estates, South Morganmouth, Wisconsin 84282	+1-200-759-2343	2	453 Juan Baker	15104 Eric Mount, Millersmouth, Arizona 23327	9943112344	3	816 Juan Ortiz	39353 Molinley Cestera Apt. 457, West Krytalport, Alabama 08508 (783)840-2428		<p>Esta consulta recupera todas las filas de la tabla "ClientePLSQL" donde el valor en la columna "nombre" contiene la subcadena "Juan". El operador LIKE con el patrón '%Juan%' busca cualquier instancia de "Juan" en cualquier parte del nombre del cliente, independientemente de lo que haya antes o después de "Juan".</p>														
ID_CLIENTE	NOMBRE	DIRECCION	TELEFONO																													
1	475 Juan Wilson	1348 Chase Estates, South Morganmouth, Wisconsin 84282	+1-200-759-2343																													
2	453 Juan Baker	15104 Eric Mount, Millersmouth, Arizona 23327	9943112344																													
3	816 Juan Ortiz	39353 Molinley Cestera Apt. 457, West Krytalport, Alabama 08508 (783)840-2428																														
52. SELECT a.marca, a.modelo, a.anو FROM AutoPLSQL a WHERE precio < 10000;	<pre>280   SELECT a.marca, a.modelo, a.anو FROM AutoPLSQL a WHERE precio &lt; 10000; 281   282  </pre> <p>Salida de Script x Resultado de la Consulta x   SQL   En Ejecución:SELECT a.marca, a.modelo, a.anو FROM AutoPLSQL a WHERE precio &lt; ORA-00904: "PRECIO": identificador no válido 00904. 00000 - "%s: invalid identifier" *Cause: *Action: Error en la línea: 280, columna: 56</p>	<p>Esta consulta recupera las columnas "marca", "modelo" y "anو" de la tabla "AutoPLSQL" donde el valor en la columna "precio" es menor a 10,000. Esto significa que se seleccionarán todos los automóviles cuyo precio sea inferior a 10,000.</p> <p>Pero sale error debido a que la columna "precio" no existe en la tabla "AutoPLSQL"</p>																														

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

53. SELECT \* FROM AlquilerPLSQL WHERE fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30';

```

282 | SELECT * FROM AlquilerPLSQL
283 | WHERE fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';
284 |
285 |

```

ORA-01861: el literal no coincide con la cadena de formato  
01861. 00000 - "Literal does not match format string"  
"Cause: Literale 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.

Esta consulta SQL devuelve todos los registros de alquiler que comenzaron en el mes de septiembre de 2023.

Pero sale error debido al formato de la columna "fecha\_inicio" está en formato TO\_DATE('DD/MM/YY'), se debe ajustar la consulta para que pueda comparar las fechas correctamente.

#### Solución:

```

285 | SELECT * FROM AlquilerPLSQL
286 | WHERE TO_DATE(fecha_inicio, 'DD/MM/YY')
287 | BETWEEN TO_DATE('01/09/23', 'DD/MM/YY') AND TO_DATE('30/09/23', 'DD/MM/YY');
288 |

```

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	594	435	45 18/09/23	29/09/23	594
2	596	397	466 16/09/23	25/09/23	596
3	610	155	112 10/09/23	06/10/23	610
4	640	462	985 14/09/23	08/10/23	640
5	671	357	270 06/09/23	22/09/23	671
6	675	929	158 17/09/23	12/10/23	675
7	680	249	590 18/09/23	20/09/23	680
8	681	709	462 28/09/23	11/10/23	681
9	711	979	759 16/09/23	08/10/23	711
10	712	97	697 03/09/23	16/09/23	712
11	722	183	38 11/09/23	19/09/23	722
12	756	549	615 14/09/23	19/09/23	756
13	759	37	58 07/09/23	21/09/23	759
14	763	549	76 09/09/23	11/09/23	763
15	767	786	92 20/09/23	03/10/23	767
16	809	877	693 18/09/23	12/10/23	809
17	831	995	108 03/09/23	09/09/23	831
18	843	149	122 27/09/23	11/10/23	843
19	844	902	795 14/09/23	07/10/23	844
20	872	128	774 22/09/23	10/10/23	872

54. 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%';

```

291 | SELECT c.nombre, a.marca, a.modelo
292 |   FROM ClientePLSQL c
293 | JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
294 | WHERE c.direccion LIKE '%Bogotá%';
295 |
296 |

```

Esta consulta SQL devuelve los nombres de los clientes y los detalles de los autos alquilados por aquellos clientes cuya dirección contiene la subcadena "Bogotá". El resultado será un conjunto de filas que cumplen con esta condición, incluyendo el nombre del cliente y los detalles del auto alquilado.

Pero sale error debido a que la tabla "AlquilerPLSQL" no existe la columna "modelo"

```

271 | SELECT * FROM AlquilerPLSQL;
272 |

```

Solución:

La consulta debería ser modificada para incluir la tabla AutoPLSQL para obtener información sobre la marca y el modelo de los autos alquilados por clientes cuya dirección contenga 'Bogotá'.

```

296 | SELECT c.nombre, a.marca, a.modelo
297 |   FROM ClientePLSQL c
298 | JOIN AlquilerPLSQL al ON c.id_cliente = al.id_cliente
299 | JOIN AutoPLSQL a ON al.id_auto = a.id_auto
300 | WHERE c.direccion LIKE '%Bogotá%';
301 |
302 |

```

55. 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;

```

302 | SELECT a.marca, a.modelo, a.ano
303 |   FROM AutoPLSQL a
304 | JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
305 | WHERE al.id_reserva = 1;
306 |
307 |

```

La consulta dará la información de la marca, modelo y año de los autos que están alquilados en la reserva con id\_reserva igual a 1 en la base de datos.

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

```
309 | SELECT * FROM AlquilerPLSQL
310 | WHERE id_cliente IN (1, 2, 3);
311 |
312 | Salida de Script x Resultado de la Consulta x
313 | SQL | Todas las Filas Recuperadas: 3 en 0,243 segundos
314 | ID_ALQUILER | ID_CLIENTE | ID_AUTO | FECHA_INICIO | FECHA_FIN | ID_RESERVA
315 | 1 | 861 | 3 | 171 20/12/22 | 10/01/23 | 861
316 | 2 | 292 | 3 | 891 19/08/23 | 13/09/23 | 292
317 | 3 | 384 | 2 | 141 19/12/22 | 03/01/23 | 384
```

Esta consulta te devolverá todos los registros de alquileres que están asociados a clientes cuyos id\_cliente sea igual a 1, 2 o 3.

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

```
312 | SELECT * FROM AlquilerPLSQL
313 | WHERE id_auto IN (1, 2, 3);
314 | Salida de Script x Resultado de la Consulta x
315 | SQL | Todas las Filas Recuperadas: 3 en 0,217 segundos
316 | ID_ALQUILER | ID_CLIENTE | ID_AUTO | FECHA_INICIO | FECHA_FIN | ID_RESERVA
317 | 1 | 723 | 911 | 219/04/23 | 01/05/23 | 723
318 | 2 | 148 | 922 | 3 13/11/22 | 25/11/22 | 148
319 | 3 | 329 | 668 | 3 03/07/23 | 07/07/23 | 329
```

Esta consulta te devolverá todos los registros de alquileres que involucran autos cuyos id\_auto sea igual a 1, 2 o 3.

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

```
315 | SELECT * FROM AlquilerPLSQL
316 | WHERE id_sucursal IN (1, 2, 3);
317 |
318 | Salida de Script x Resultado de la Consulta x
319 | SQL | En Ejecución:SELECT * FROM AI
320 | ORA-00904: "ID_SUCURSAL": identificador no válido
321 | 00904. 00000 - "%s: invalid identifier"
322 | *Cause:
323 | *Action:
324 | Error en la línea: 316, columna: 7
```

Esta consulta te devolverá todos los registros de alquileres que están asociados a sucursales cuyos id\_sucursal sea igual a 1, 2 o 3.

Pero sale error debido a que la tabla “AlquilerPLSQL” no existe la columna “ID\_SUCURSAL”

```
271 | SELECT * FROM AlquilerPLSQL;
272 |
273 | Salida de Script x Resultado de la Consulta x
274 | SQL | Se han recuperado 50 filas en 0,138 segundos
275 | ID_ALQUILER | ID_CLIENTE | ID_AUTO | FECHA_INICIO | FECHA_FIN | ID_RESERVA
276 | 1 | 584 | 131 | 208 01/10/22 | 01/11/23 | 584
277 | 2 | 585 | 394 | 543 19/04/23 | 11/05/23 | 585
278 | 3 | 586 | 774 | 345 19/07/23 | 30/07/23 | 586
279 | 4 | 587 | 81 | 250 26/03/23 | 22/04/23 | 587
```

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

```
319 | SELECT * FROM AlquilerPLSQL
320 | WHERE fecha_inicio
321 | BETWEEN '2023-09-01' AND '2023-09-30' AND id_cliente IN (1, 2, 3);
322 |
323 | Salida de Script x Resultado de la Consulta x
324 | SQL | En Ejecución:SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN '202
325 | ORA-01861: el literal no coincide con la cadena de formato
326 | 01861. 00000 - "Literal does not match format string"
327 | *Cause: Literals in the input must be the same length as literals in
328 | the format string (with the exception of leading whitespace). If the
329 | "FX" modifier has been toggled on, the literal must match exactly,
330 | with no extra whitespace.
331 | *Action: Correct the format string to match the literal.
```

La consulta devuelve todos los registros de alquiler que cumplen con ambas condiciones: que tengan una fecha de inicio en el rango de septiembre de 2023 y que estén asociados a los clientes con id\_cliente igual a 1, 2 o 3.

Pero sale error debido al formato de la columna "fecha\_inicio" está en formato TO\_DATE('DD/MM/YY'), se debe ajustar la consulta para que pueda comparar las fechas correctamente.

Solución:

		<pre>323   SELECT * FROM AlquilerPLSQL 324   WHERE TO_DATE(fecha_inicio, 'DD/MM/YY') 325   BETWEEN TO_DATE('01/09/23', 'DD/MM/YY') AND TO_DATE('30/09/23', 'DD/MM/YY') AND id_cliente IN (1, 2, 3); 326   327  </pre>
60. <code>SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30' AND id_auto IN (1, 2, 3);</code>	<pre>328   SELECT * FROM AlquilerPLSQL 329   WHERE fecha_inicio 330   BETWEEN '2023-09-01' AND '2023-09-30' AND id_auto IN (1, 2, 3); 331   332  </pre>	<p>La consulta devuelve todos los registros de alquiler que cumplen con ambas condiciones: que tengan una fecha de inicio en el rango de septiembre de 2023 y que estén asociados a los autos con id_auto igual a 1, 2 o 3.</p> <p>Pero sale error debido al formato de la columna "fecha_inicio" está en formato TO_DATE('DD/MM/YY'), se debe ajustar la consulta para que pueda comparar las fechas correctamente.</p> <p>Solución:</p> <pre>332   SELECT * FROM AlquilerPLSQL 333   WHERE TO_DATE(fecha_inicio, 'DD/MM/YY') 334   BETWEEN TO_DATE('01/09/23', 'DD/MM/YY') AND TO_DATE('30/09/23', 'DD/MM/YY') AND id_auto IN (1, 2, 3); 335   336  </pre>
61. <code>SELECT * FROM AlquilerPLSQL WHERE fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30' AND id_sucursal IN (1, 2, 3);</code>	<pre>337   SELECT * FROM AlquilerPLSQL 338   WHERE fecha_inicio 339   BETWEEN '2023-09-01' AND '2023-09-30' AND id_sucursal IN (1, 2, 3); 340   341  </pre>	<p>La consulta devuelve todos los registros de alquiler que cumplen con ambas condiciones: que tengan una fecha de inicio en el rango de septiembre de 2023 y que estén asociados a las sucursales con id_sucursal igual a 1, 2 o 3.</p> <p>Pero sale error debido al formato de la columna "fecha_inicio" está en formato TO_DATE('DD/MM/YY'), se debe ajustar la consulta para que pueda comparar las fechas correctamente.</p> <p>Solución:</p>

```

341 | SELECT * FROM AlquilerPLSQL
342 | WHERE TO_DATE(fecha_inicio, 'DD/MM/YY')
343 | BETWEEN TO_DATE('01/09/23', 'DD/MM/YY') AND TO_DATE('30/09/23', 'DD/MM/YY') AND id_sucursal IN (1, 2, 3);
344 |
345 |

```

Salida de Script x Resultado de la Consulta x  
 En Ejecución:SELECT \* FROM AlquilerPLSQL WHERE TO\_DATE(fecha\_inicio, DD/MM/YY) BETWEEN TO\_DATE(01/09/23, DD/MM/YY) AND TO\_DATE(30/09/23, DD/MM/YY) AND id\_sucursal IN (1, 2, 3);  
ORA-00904: "ID\_SUCURSAL": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:  
Error en la linea: 343, columna: 81

Pero sale error debido a que la tabla “AlquilerPLSQL” no existe la columna “ID\_SUCURSAL”

```

271 | SELECT * FROM AlquilerPLSQL;
272 |

```

Salida de Script x Resultado de la Consulta x  
 Se han recuperado 50 filas en 0,138 segundos  

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

62. 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;

```

345 | SELECT c.nombre, COUNT(*) AS numero_alquileres
346 | FROM ClientePLSQL c
347 | JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
348 | GROUP BY c.nombre
349 | ORDER BY numero_alquileres DESC LIMIT 1;

```

Salida de Script x Resultado de la Consulta x  
 En Ejecución:SELECT c.nombre, COUNT(\*) AS numero\_alquileres  
ORA-00933: comando SQL no terminado correctamente  
00933. 00000 - "SQL command not properly ended"  
\*Cause:  
\*Action:  
Error en la linea: 349, columna: 33

La consulta devuelve el nombre del cliente que tiene el mayor número de alquileres en la base de datos y muestra ese número.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Solución:

```

351 | SELECT c.nombre, COUNT(*) AS numero_alquileres
352 | FROM ClientePLSQL c
353 | JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
354 | GROUP BY c.nombre
355 | ORDER BY numero_alquileres DESC
356 | fetch first 1 rows only;
357 |

```

Salida de Script x Resultado de la Consulta x  
 Todas las Filas Recuperadas: 1 en 0,226 segundos  

NOMBRE	NUMERO_ALQUILERES
John Greene	6

```
63. 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;
```

```
358 | SELECT a.marca, a.modelo, COUNT(*) AS numero_alquileres
359 | FROM AutoPLSQL a
360 | JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
361 | GROUP BY a.marca, a.modelo
362 | ORDER BY numero_alquileres DESC LIMIT 1;
363 |

```

La consulta devuelve la marca y el modelo del automóvil que ha sido más alquilado en la base de datos y muestra la cantidad de alquileres realizados para ese automóvil.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Solución:

```
364 | SELECT a.marca, a.modelo, COUNT(*) AS numero_alquileres
365 | FROM AutoPLSQL a
366 | JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
367 | GROUP BY a.marca, a.modelo
368 | ORDER BY numero_alquileres DESC
369 | fetch first 1 rows only;
370 |

```

La consulta devuelve el nombre de la sucursal que tiene el mayor número de alquileres en la base de datos y muestra esa cantidad de alquileres.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Solución:

```
371 | SELECT s.nombre, COUNT(*) AS numero_alquileres
372 | FROM SucursalPLSQL s
373 | JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal
374 | GROUP BY s.nombre
375 | ORDER BY numero_alquileres DESC LIMIT 1;
376 |

```

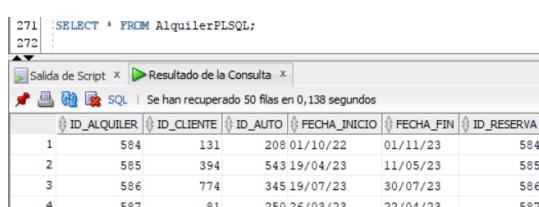
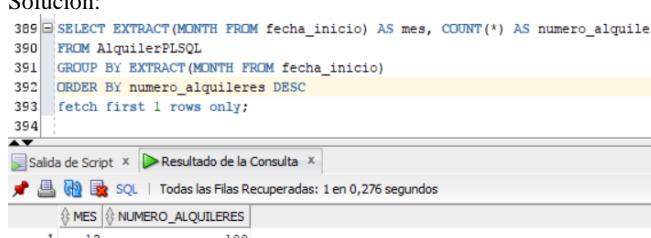
La consulta devuelve el nombre de la sucursal que tiene el mayor número de alquileres en la base de datos y muestra esa cantidad de alquileres.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Solución:

```
377 | SELECT s.nombre, COUNT(*) AS numero_alquileres
378 | FROM SucursalPLSQL s
379 | JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal
380 | GROUP BY s.nombre
381 | ORDER BY numero_alquileres DESC
382 | fetch first 1 rows only;
383 |

```

		<p>Pero sale error debido a que la tabla “AlquilerPLSQL” no existe la columna “ID_SUCURSAL”</p> 
65.	<pre>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;</pre>	<p>La consulta devuelve el mes con la mayor cantidad de alquileres en la base de datos y muestra esa cantidad de alquileres.</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p>Solución:</p> 
66.	<pre>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;</pre>	<p>La consulta devuelve el día de la semana con la mayor cantidad de alquileres en la base de datos y muestra esa cantidad de alquileres.</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p>Solución:</p>

		<pre> 400   SELECT EXTRACT(DAYOFWEEK FROM fecha_inicio) AS dia_semana, COUNT(*) AS numero_alquileres 401   FROM AlquilerPLSQL 402   GROUP BY EXTRACT(DAYOFWEEK FROM fecha_inicio) 403   ORDER BY numero_alquileres DESC 404   fetch first 1 rows only; </pre> <p>Salida de Script x Resultado de la Consulta x  ORA-00907: falta el paréntesis derecho  00907. 00000 - "missing right parenthesis"  *Cause:  *Action:  Error en la línea: 400, columna: 26</p>						
		<p>Pero sale error debido al formato de la columna "fecha_inicio" está en formato TO_DATE('DD/MM/YY'), se debe ajustar la consulta para que pueda comparar las fechas correctamente.</p> <pre> 406   SELECT TO_NUMBER(TO_CHAR(fecha_inicio, 'D')) AS dia_semana, COUNT(*) AS numero_alquileres 407   FROM AlquilerPLSQL 408   GROUP BY TO_NUMBER(TO_CHAR(fecha_inicio, 'D')) 409   ORDER BY numero_alquileres DESC 410   fetch first 1 rows only; 411   </pre> <p>Salida de Script x Resultado de la Consulta x  Todas las Filas Recuperadas: 1 en 0,137 segundos</p> <table border="1"> <thead> <tr> <th>DIA_SEMANA</th> <th>NUMERO_ALQUILERES</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> </tr> <tr> <td></td> <td>154</td> </tr> </tbody> </table>	DIA_SEMANA	NUMERO_ALQUILERES	1	2		154
DIA_SEMANA	NUMERO_ALQUILERES							
1	2							
	154							
67. SELECT * FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;		<p>la consulta devuelve todos los detalles del alquiler que tiene el precio más alto en la base de datos.</p> <pre> 412   SELECT * FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1; 413   </pre> <p>Salida de Script x Resultado de la Consulta x  ORA-00933: comando SQL no terminado correctamente  00933. 00000 - "SQL command not properly ended"  *Cause:  *Action:  Error en la línea: 412, columna: 50</p> <p>Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.</p> <p>Solución:</p>						

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

```
274 | SELECT * FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1;
275 |
```

Salida de Script x Resultado de la Consulta x  
SQL | En Ejecución:SELECT \* FROM AlquilerPLSQL ORDER BY precio AS  
ORA-00933: comando SQL no terminado correctamente  
00933. 00000 - "SQL command not properly ended"  
\*Cause:  
\*Action:  
Error en la línea: 273, columna: 49

```
416 | SELECT * FROM AlquilerPLSQL ORDER BY precio DESC
417 | fetch first 1 rows only;
418 |
419 |
```

Salida de Script x Resultado de la Consulta x  
SQL | En Ejecución:SELECT \* FROM AlquilerPLSQL ORDER BY  
ORA-00904: "PRECIO": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:  
Error en la línea: 416, columna: 38

Pero sale error debido a que la tabla “AlquilerPLSQL” no existe la columna “precio”

```
271 | SELECT * FROM AlquilerPLSQL;
272 |
```

Salida de Script x Resultado de la Consulta x  
SQL | Se han recuperado 50 filas en 0,138 segundos

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

Esta consulta SQL devuelve la fila de la tabla "AlquilerPLSQL" que tiene el precio más bajo registrado.

Pero sale error porque la cláusula LIMIT no es una cláusula SQL estándar en todos los sistemas de gestión de bases de datos. La cláusula LIMIT se utiliza en algunos sistemas, como MySQL, para limitar el número de filas devueltas en un resultado.

Además, en la tabla “AlquilerPLSQL” no existe la columna “precio”

```
271 | SELECT * FROM AlquilerPLSQL;
272 |
```

Salida de Script x Resultado de la Consulta x  
SQL | Se han recuperado 50 filas en 0,138 segundos

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

69. SELECT \* FROM ClientePLSQL WHERE nombre LIKE '%Juan%' AND fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30';

The screenshot shows the Oracle SQL Developer interface. A script editor window contains the following code:

```
419 | SELECT * FROM ClientePLSQL
420 | WHERE nombre LIKE '%Juan%' AND fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';
421 |
422 |  
Salida de Script | Resultado de la Consulta |  
SQL | En Ejecución:SELECT * FROM ClientePLSQL WHERE nombre LIKE '%Juan%' AND fecha_inicio BETWEEN  
ORA-00904: 'FECHA_INICIO' identificador no válido  
00904. 00000 - "%: invalid identifier"  
*Cause:  
*Action:  
Error en la línea: 420, columna: 32
```

An error message is displayed: ORA-00904: 'FECHA\_INICIO' identificador no válido (ORA-00904: 'FECHA\_INICIO' identifier not valid).

La consulta devuelve todos los registros de clientes cuyos nombres contienen "Juan" y que han tenido alquileres con fechas de inicio dentro del rango especificado.

Pero sale un error debido a que se está tratando de buscar datos de dos tablas diferentes, ClientePLSQL y AlquilerPLSQL, pero no se ha establecido ninguna relación entre ellas en la consulta. Además, la columna fecha\_inicio pertenece a la tabla AlquilerPLSQL, pero estás intentando filtrarla en la tabla ClientePLSQL, lo que también genera un error.

Para lograr el resultado, se debe realizar una consulta que involucre una operación JOIN entre las tablas ClientePLSQL y AlquilerPLSQL y aplicar los filtros apropiados.

Solución:

The screenshot shows the Oracle SQL Developer interface. A script editor window contains the following corrected code:

```
423 | SELECT C.*  
424 | FROM ClientePLSQL C  
425 | JOIN AlquilerPLSQL A ON C.id_cliente = A.id_cliente  
426 | WHERE C.nombre LIKE '%Juan%'  
427 | AND A.fecha_inicio BETWEEN TO_DATE('2023-09-01', 'YYYY-MM-DD') AND TO_DATE('2023-09-30', 'YYYY-MM-DD');  
428 |  
Salida de Script | Resultado de la Consulta |  
SQL | Todas las Filas Recuperadas: 0 en 0,128 segundos  
ID_CLIENTE NOMBRE DIRECCION TELEFONO
```

The results pane shows a header row with columns: ID\_CLIENTE, NOMBRE, DIRECCION, and TELEFONO.

En esta consulta se está haciendo referencia a una columna llamada precio en la cláusula WHERE, pero en la definición de la tabla AutoPLSQL no existe una columna llamada precio. Además, la columna fecha\_inicio pertenece a la tabla AlquilerPLSQL, no a la tabla AutoPLSQL.

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

The screenshot shows the Oracle SQL Developer interface. A script editor window contains the following code:

```
430 | SELECT a.marca, a.modelo, a.ano
431 | FROM AutoPLSQL a
432 | WHERE precio < 10000 AND fecha_inicio BETWEEN '2023-09-01' AND '2023-09-30';
433 |
434 |  
Salida de Script | Resultado de la Consulta |  
SQL | En Ejecución:SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a WHERE precio < 10000
ORA-00904: 'FECHA_INICIO' identificador no válido
00904. 00000 - "%: invalid identifier"
*Cause:
*Action:
Error en la línea: 432, columna: 26
```

An error message is displayed: ORA-00904: 'FECHA\_INICIO' identificador no válido (ORA-00904: 'FECHA\_INICIO' identifier not valid).

## EJERCICIOS TERCER CICLO (81-90):

Ejercicio	Resultado	Análisis																														
71. 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';	<pre> 436   CREATE VIEW vista_clientes_alquilados_sucursal_100388 AS 437     SELECT c.nombre, a.marca, a.modelo 438       FROM ClientePLSQL c 439     JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente 440     JOIN SucursalPLSQL s ON a.id_sucursal = s.id_sucursal 441     WHERE s.nombre = 'Sucursal Central'; 442   </pre> <p>Salida de Script x Resultado de la Consulta x     Tarea terminada en 0,889 segundos</p> <p>Error que empieza en la linea: 436 del comando :  CREATE VIEW vista_clientes_alquilados_sucursal_100388 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'  Informe de error -  ORA-00904: "%."."ID_SUCURSAL": identificador no válido  00904. 00000 - "%s: invalid identifier"  *Cause:  *Action:</p>	<p>La vista resultante, vista_clientes_alquilados_sucursal, contendrá los nombres de los clientes, las marcas de los autos y los modelos de los autos alquilados en la "Sucursal Central".</p> <p>Sale error debido a que en la tabla "AlquilerPLSQL" no existe la columna "ID_SUCURSAL"</p> <pre> 271   SELECT * FROM AlquilerPLSQL; 272   </pre> <p>Salida de Script x Resultado de la Consulta x     Se han recuperado 50 filas en 0,138 segundos</p> <table border="1"> <thead> <tr> <th>ID_ALQUILER</th> <th>ID_CLIENTE</th> <th>ID_AUTO</th> <th>FECHA_INICIO</th> <th>FECHA_FIN</th> <th>ID_RESERVA</th> </tr> </thead> <tbody> <tr><td>1</td><td>584</td><td>131</td><td>208 01/10/22</td><td>01/11/23</td><td>584</td></tr> <tr><td>2</td><td>585</td><td>394</td><td>543 19/04/23</td><td>11/05/23</td><td>585</td></tr> <tr><td>3</td><td>586</td><td>774</td><td>345 19/07/23</td><td>30/07/23</td><td>586</td></tr> <tr><td>4</td><td>587</td><td>81</td><td>250 26/03/23</td><td>22/04/23</td><td>587</td></tr> </tbody> </table>	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
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																											
72. 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';	<pre> 445   CREATE VIEW vista_autos_alquilados_cliente_fecha_100388 AS 446     SELECT a.marca, a.modelo 447       FROM AutoPLSQL a 448     JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto 449     WHERE al.id_cliente = 1 AND al.fecha_inicio = '2023-09-27'; 450   </pre> <p>Salida de Script x Resultado de la Consulta x     Tarea terminada en 0,875 segundos</p> <p>View VISTA_AUTOS_ALQUILADOS_CLIENTE_FECHA_100388 creado.</p>	<p>La vista resultante, vista_autos_alquilados_cliente_fecha_100388, contendrá las marcas y modelos de los autos que han sido alquilados por el cliente con el ID 1 en la fecha '2023-09-27'.</p> <pre> 450   451   SELECT * FROM vista_autos_alquilados_cliente_fecha_100388; 452   </pre> <p>Salida de Script x Resultado de la Consulta x     Todas las Filas Recuperadas: 0 en 0,165 segundos</p> <table border="1"> <thead> <tr> <th>MARCA</th> <th>MODELO</th> </tr> </thead> </table> <p>En este caso el cliente con ID igual a 1 no ha alquilado un auto en la fecha '2023-09-27'.</p>	MARCA	MODELO																												
MARCA	MODELO																															

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

The screenshot shows the Oracle SQL Developer interface. On the left, a code editor window displays the SQL code for creating a view. On the right, a results window titled 'Resultado de la Consulta' shows the message 'View VISTA\_ALQUILERES\_MAS\_7DIAS creado.' (View VISTA\_ALQUILERES\_MAS\_7DIAS created).

```
457 | CREATE VIEW vista_alquileres_mas_7dias_100388 AS  
458 | SELECT * FROM AlquilerPLSQL  
459 | WHERE fecha_fin - fecha_inicio > 7;  
460 |  
461 |
```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0,199 segundos

View VISTA\_ALQUILERES\_MAS\_7DIAS creado.

La vista resultante, vista\_alquileres\_mas\_7dias, contendrá todos los registros de alquiler de la tabla AlquilerPLSQL donde la duración del alquiler es mayor que 7 días.

The screenshot shows the Oracle SQL Developer interface. A results window titled 'Resultado de la Consulta' displays a table with 50 rows of data. The table has columns: ID\_ALQUILER, ID\_CLIENTE, ID\_AUTO, FECHA\_INICIO, FECHA\_FIN, and ID\_RESERVA. The data represents various car rental records.

ID_ALQUILER	ID_CLIENTE	ID_AUTO	FECHA_INICIO	FECHA_FIN	ID_RESERVA
1	584	131	208 01/10/22	01/11/23	584
2	585	394	543 19/04/23	11/05/23	585
3	586	774	345 19/07/23	30/07/23	586
4	587	81	250 26/03/23	22/04/23	587
5	589	999	584 01/05/23	21/05/23	589
6	590	796	711 26/05/23	17/06/23	590
7	592	223	847 27/11/22	14/12/22	592
8	593	987	961 21/08/23	20/09/23	593
9	594	435	45 18/09/23	29/09/23	594
10	595	81	435 06/05/23	31/05/23	595
11	596	397	466 16/09/23	25/09/23	596
12	597	36	473 06/02/23	01/03/23	597
13	598	726	669 09/11/22	23/11/22	598
14	599	999	715 07/10/22	25/10/22	599
15	600	414	809 29/12/22	10/01/23	600
16	601	470	493 02/10/22	16/10/22	601
17	602	830	198 11/08/23	29/08/23	602
18	603	682	526 02/10/22	16/10/22	603
19	604	171	251 14/02/23	28/02/23	604

```

74. 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;

```

```

463 | CREATE VIEW vista_clientes_mas_alquileres_100388 AS
464 |   SELECT c.nombre, COUNT(*) AS numero_alquileres
465 |     FROM ClientePLSQL c
466 |     JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente
467 |       GROUP BY c.nombre
468 |       ORDER BY numero_alquileres DESC;

```

Salida de Script | Resultado de la Consulta | Tarea terminada en 0,216 segundos

View VISTA\_CLIENTES\_MAS\_ALQUILERES\_100388 creado.

La vista resultante contendrá el nombre de cada cliente y la cantidad de alquileres que ha realizado, ordenados de manera descendente por la cantidad de alquileres.

```
472 | SELECT * FROM vista_clientes_mas_alquileres_100388;
```

Salida de Script | Resultado de la Consulta | SQL | Se han recuperado 50 filas en 0,169 segundos

NOMBRE	NUMERO_ALQUILERES
1 John Greene	6
2 Michele Allen	4
3 Haley Carroll	4
4 Stephanie Young	4
5 Jill Sanford	4
6 Pamela Johnson	4
7 David Woods	4
8 Andrew Chang	4
9 Melinda Jackson	3
10 Kyle Stephens	3
11 Zachary George	3
12 Steven Velasquez	3
13 Ashley Farmer	3
14 Danielle Williamson	3
15 Lauren Jones	3
16 Travis Myers	3
17 Karen Black	3
18 Courtney Shannon	3
19 Stephanie Gonzalez	3

```

75. 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;

```

```

475 CREATE VIEW vista_autos_mas_alquileres_100388 AS
476   SELECT a.marca, a.modelo, COUNT(*) AS numero_alquileres
477   FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto
478   GROUP BY a.marca, a.modelo
479   ORDER BY numero_alquileres DESC;
480

```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0,155 segundos

View VISTA\_AUTOS\_MAS\_ALQUILERES\_100388 creado.

La vista resultante, vista\_autos\_mas\_alquileres, contendrá la marca y el modelo de cada automóvil, así como la cantidad de alquileres realizados para ese automóvil, ordenados de manera descendente por la cantidad de alquileres.

```
483 | SELECT * FROM vista_autos_mas_alquileres_100388;
```

Salida de Script x Resultado de la Consulta x  
SQL | Se han recuperado 50 filas en 0,138 segundos

MARCA	MODELO	NUMERO_ALQUILERES
1 end	hold	5
2 lot	level	5
3 significant	pretty	5
4 tough	year	4
5 analysis	reveal	4
6 just	morning	4
7 economic	approach	4
8 worry	behavior	4
9 similar	few	4
10 heavy	improve	4
11 same	fund	4
12 special	back	4
13 scientist	media	4
14 culture	far	4
15 Mazda	6	3
16 happy	hour	3
17 decade	include	3
18 tonight	budget	3
19 later	too	3
20 mind	all	3

76. 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;

```

486 | CREATE VIEW vista_sucursales_mas_alquileres_100388 AS
487 |   SELECT s.nombre, COUNT(*) AS numero_alquileres
488 |     FROM SucursalPLSQL s
489 |   JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal
490 |   GROUP BY s.nombre ORDER BY numero_alquileres DESC;
491 |

```

View VISTA\_AUTOS\_MAS\_ALQUILERES\_100388 creado.

Error que empieza en la linea: 486 del comando :  
CREATE VIEW vista\_sucursales\_mas\_alquileres\_100388 AS  
SELECT s.nombre, COUNT(\*) AS numero\_alquileres  
FROM SucursalPLSQL s  
JOIN AlquilerPLSQL al ON s.id\_sucursal = al.id\_sucursal  
GROUP BY s.nombre ORDER BY numero\_alquileres DESC  
Informe de error -  
ORA-00904: "AL"."ID\_SUCURSAL": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:

La vista resultante contendrá el nombre de cada sucursal y la cantidad de alquileres que se han realizado en cada una de ellas, ordenados de manera descendente por la cantidad de alquileres.

Sale error debido a que en la tabla “AlquilerPLSQL” no existe la columna “ID\_SUCURSAL”

```

271 | SELECT * FROM AlquilerPLSQL;
272 |

```

Salida de Script x Resultado de la Consulta x | Se han recuperado 50 filas en 0,138 segundos

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

77. 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;

```

493 | CREATE VIEW vista_meses_mas_alquileres_100388 AS
494 |   SELECT EXTRACT(MONTH FROM fecha_inicio) AS mes, COUNT(*) AS numero_alquileres
495 |     FROM AlquilerPLSQL
496 |   GROUP BY EXTRACT(MONTH FROM fecha_inicio)
497 |   ORDER BY numero_alquileres DESC;
498 |

```

View VISTA\_MESES\_MAS\_ALQUILERES\_100388 creado.

La vista resultante contendrá el número de mes y la cantidad de alquileres que se han realizado en cada mes, ordenados de manera descendente por la cantidad de alquileres.

```

499 | SELECT * FROM vista_meses_mas_alquileres_100388;
500 |

```

Salida de Script x Resultado de la Consulta x | Todas las Filas Recuperadas: 12 en 0,19 segundos

MES	NUMERO_ALQUILERES
1	12
2	5
3	10
4	11
5	9
6	1
7	4
8	3
9	8
10	6
11	7
12	2

```

78. 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;

```

The screenshot shows the SQL script being run in the 'Salida de Script' tab. The error message is as follows:

```

Error que aparece en la linea: 504 del comando :
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;
Informe de error -
ORA-00907: falta el parentesis derecho
00907. 00000 - "missing right parenthesis"
*Cause:
*Action:

```

La vista resultante contendrá el número de día de la semana y la cantidad de alquileres que se han realizado en cada día de la semana, ordenados de manera descendente por la cantidad de alquileres.

Pero sale error debido al formato de la columna "fecha\_inicio" está en formato TO\_DATE('DD/MM/YY'), se debe ajustar la consulta para que pueda comparar las fechas correctamente.

Solución:

```

504 CREATE VIEW vista_dias_semana_mas_alquileres_100388 AS
505 SELECT TO_NUMBER(TO_CHAR(fecha_inicio, 'D')) AS dia_semana, COUNT(*) AS numero_alquileres
506 FROM AlquilerPLSQL
507 GROUP BY TO_NUMBER(TO_CHAR(fecha_inicio, 'D'))
508 ORDER BY numero_alquileres DESC;
509
510 DROP VIEW vista_dias_semana_mas_alquileres;

```

The screenshot shows the 'Resultado de la Consulta' tab displaying the message: 'View VISTA\_DIAS\_SEMANA\_MAS\_ALQUILERES\_100388 creado.'

Resultado

```

510 | SELECT *FROM vista_dias_semana_mas_alquileres_100388;
511 |

```

The screenshot shows the 'Resultado de la Consulta' tab with the following data:

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

79. CREATE VIEW vista\_alquileres\_mas\_caros AS  
SELECT \* FROM AlquilerPLSQL ORDER BY precio  
DESC;

```
514 | CREATE VIEW vista_alquileres_mas_caros_100388 AS
515 | SELECT * FROM AlquilerPLSQL
516 | ORDER BY precio DESC;
517 |

```

Salida de Script | Resultado de la Consulta | Tarea terminada en 0,429 segundos

Error que empieza en la linea: 514 del comando :  
CREATE VIEW vista\_alquileres\_mas\_caros\_100388 AS  
SELECT \* FROM AlquilerPLSQL  
ORDER BY precio DESC  
Informe de error -  
ORA-00904: "PRECIO": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:

El código crea una vista llamada vista\_alquileres\_mas\_caros. Sin embargo, hay un problema en esta vista. En la consulta SELECT dentro de la vista, se intenta ordenar los resultados por una columna llamada precio, pero no se realiza ningún cálculo o selección de esa columna en la consulta. La columna precio no está presente en la vista AlquilerPLSQL

```
271 | SELECT * FROM AlquilerPLSQL;
272 |

```

Salida de Script | Resultado de la Consulta | Se han recuperado 50 filas en 0,138 segundos

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

80. CREATE VIEW vista\_alquileres\_mas\_baratos AS  
SELECT \* FROM AlquilerPLSQL ORDER BY precio  
ASC;

```
519 | CREATE VIEW vista_alquileres_mas_baratos AS
520 | SELECT * FROM AlquilerPLSQL
521 | ORDER BY precio ASC;
522 |

```

Salida de Script | Resultado de la Consulta | Tarea terminada en 0,275 segundos

Error que empieza en la linea: 519 del comando :  
CREATE VIEW vista\_alquileres\_mas\_baratos AS  
SELECT \* FROM AlquilerPLSQL  
ORDER BY precio ASC  
Informe de error -  
ORA-00904: "PRECIO": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:

El código crea una vista, la vista selecciona todos los campos y registros de la tabla AlquilerPLSQL y los ordena en orden ascendente (de menor a mayor) por la columna precio.

Pero no se realiza ningún cálculo o selección de esa columna en la consulta. La columna precio no está presente en la vista AlquilerPLSQL

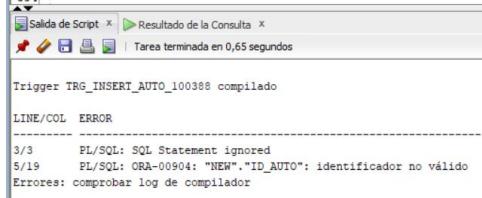
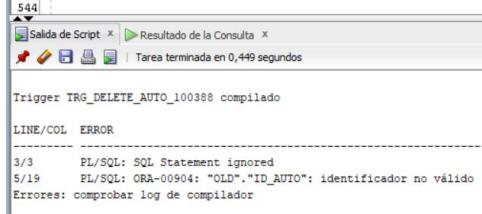
```
271 | SELECT * FROM AlquilerPLSQL;
272 |

```

Salida de Script | Resultado de la Consulta | Se han recuperado 50 filas en 0,138 segundos

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

## EJERCICIOS TERCER CICLO (91-100):

Ejercicio	Resultado	Análisis
<p>81. 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;</p>	<pre> 525: CREATE TRIGGER trg_insert_auto_100388 526: BEFORE INSERT ON AutoPLSQL 527: FOR EACH ROW 528: BEGIN 529:     -- Actualizar el número de autos disponibles 530:     UPDATE AutoPLSQL 531:         SET numero_disponibles = numero_disponibles + 1 532:         WHERE id_auto = NEW.id_auto; 533: END; 534: </pre> 	<p>Cuando se realiza una inserción en la tabla AutoPLSQL, este trigger se activa antes de la inserción y aumenta el número de autos disponibles en la fila correspondiente en 1.</p> <p>El error que se está experimentando se debe a que se está intentando actualizar la columna numero_disponibles en la tabla AutoPLSQL, pero en la definición de la tabla AutoPLSQL, no hay una columna llamada numero_disponibles. Por lo tanto, Oracle muestra un error porque la columna que se está intentando actualizar no existe.</p>
<p>82. 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;</p>	<pre> 535: CREATE TRIGGER trg_delete_auto_100388 536: BEFORE DELETE ON AutoPLSQL 537: FOR EACH ROW 538: BEGIN 539:     -- Actualizar el número de autos disponibles 540:     UPDATE AutoPLSQL 541:         SET numero_disponibles = numero_disponibles - 1 542:         WHERE id_auto = OLD.id_auto; 543: END; 544: </pre> 	<p>Cuando se elimina un registro de la tabla AutoPLSQL, este trigger disminuirá el valor de numero_disponibles en 1 en la fila correspondiente. Esto se utiliza para mantener un registro del número de autos disponibles y actualizarlo automáticamente cuando se elimina un auto de la tabla.</p> <p>El error que se está experimentando se debe a que se está intentando actualizar la columna “numero_disponibles” en la tabla AutoPLSQL, pero en la definición de la tabla AutoPLSQL, no hay una columna llamada “numero_disponibles”. Por lo tanto, Oracle muestra un error porque la columna que se está intentando actualizar no existe.</p>

```
83. 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;
```

```
547 ┌ CREATE TRIGGER trg_update_auto_100388
548   ┌ BEFORE UPDATE ON AutoPLSQL
549   ┌ FOR EACH ROW
550 ┌ BEGIN
551   -- Actualizar el número de autos disponibles
552   ┌ IF NEW.numero_disponibles != OLD.numero_disponibles THEN
553     UPDATE AutoPLSQL
554     SET numero_disponibles = NEW.numero_disponibles
555     WHERE id_auto = NEW.id_auto;
556   END IF;
557 END;
558
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,409 segundos

Trigger TRG\_UPDATE\_AUTO\_100388 compilado

LINE/COL ERROR

---

3/3 PL/SQL: Statement ignored

3/9 PLS-00201: el identificador 'NEW.NUMERO\_DISPONIBLES' se debe declarar

Errores: comprobar log de compilador

Este trigger asegura que el valor de numero\_disponibles en la tabla AutoPLSQL se mantenga actualizado cuando se realicen actualizaciones en la tabla, pero solo si el valor de numero\_disponibles ha cambiado en el proceso de actualización. Esto ayuda a mantener coherencia en los datos.

El error que se está experimentando se debe a que se está intentando actualizar la columna “numero\_disponibles” en la tabla AutoPLSQL, pero en la definición de la tabla AutoPLSQL, no hay una columna llamada “numero\_disponibles”. Por lo tanto, Oracle muestra un error porque la columna que se está intentando actualizar no existe.

```
84. 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;
```

```
560 [+] CREATE TRIGGER trg_insert_cliente_100388
561   BEFORE INSERT ON ClientePLSQL
562   FOR EACH ROW
563   BEGIN
564     -- Actualizar el número de clientes
565     UPDATE ClientePLSQL
566     SET numero_clientes = numero_clientes + 1;
567   END;
568
569
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,995 segundos

Trigger TRG\_INSERT\_CLIENTE\_100388 compilado

LINE/COL ERROR

---

/3 PL/SQL: SQL Statement ignored  
/27 PL/SQL: ORA-00904: "NUMERO\_CLIENTES": identificador no válido  
Cinturones: comprobar log del compilador

Este trigger asegura que el valor de numero\_clientes en la tabla ClientePLSQL se incremente en 1 cada vez que se inserta un nuevo cliente en la tabla.

El error que se está experimentando se debe a que se está intentando actualizar la columna “numero\_clientes” en la tabla ClientePLSQL, pero en la definición de la tabla ClientePLSQL,, no hay una columna llamada “numero\_clientes”. Por lo tanto, Oracle muestra un error porque la columna que se está intentando actualizar no existe.

```

85. 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;

```

The screenshot shows the Oracle SQL Developer interface. On the left, the code for the trigger `trg_delete_cliente_100388` is displayed. The trigger is defined to run before a delete operation on the `ClientePLSQL` table, updating the `numero_clientes` column by subtracting 1. A compilation error is shown in the bottom pane: `PL/SQL: SQL Statement ignored` and `PL/SQL: ORA-00904: "NUMERO_CLIENTES": identificador no válido`. The bottom pane also shows the results of a `SELECT * FROM ClientePLSQL;` query.

Este trigger asegura que el valor de `numero_clientes` en la tabla `ClientePLSQL` se disminuye en 1 cada vez que se elimina un cliente de la tabla.

El error que se está experimentando se debe a que se está intentando actualizar la columna “`numero_clientes`” en la tabla `ClientePLSQL`, pero en la definición de la tabla `ClientePLSQL`, no hay una columna llamada “`numero_clientes`”. Por lo tanto, Oracle muestra un error porque la columna que se está intentando actualizar no existe.

The screenshot shows the Oracle SQL Developer interface. On the left, the code for the trigger `TRG_DELETE_CLIENTE_100388` is displayed. The trigger is defined to run before a delete operation on the `ClientePLSQL` table, updating the `numero_clientes` column by subtracting 1. A compilation error is shown in the bottom pane: `PL/SQL: SQL Statement ignored` and `PL/SQL: ORA-00904: "NUMERO_CLIENTES": identificador no válido`. The bottom pane also shows the results of a `SELECT * FROM ClientePLSQL;` query.

```

86. 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;

```

The screenshot shows the Oracle SQL Developer interface. On the left, the code for the trigger `trg_update_cliente_100388` is displayed. The trigger is defined to run before an update operation on the `ClientePLSQL` table, updating the `numero_alquileres` column. An if condition checks if the new value is not equal to the old value. A compilation error is shown in the bottom pane: `PL/SQL: Statement ignored` and `PLS-00201: el identificador 'NEW.NUMERO_ALQUILERES' se debe declarar`. The bottom pane also shows the results of a `SELECT * FROM ClientePLSQL;` query.

El error que se está experimentando se debe a que se está intentando actualizar la columna “`numero_alquileres`” en la tabla `ClientePLSQL`, pero en la definición de la tabla `ClientePLSQL`, no hay una columna llamada “`numero_alquileres`”. Por lo tanto, Oracle muestra un error porque la columna que se está intentando actualizar no existe.

The screenshot shows the Oracle SQL Developer interface. On the left, the code for the trigger `TRG_UPDATE_CLIENTE_100388` is displayed. The trigger is defined to run before an update operation on the `ClientePLSQL` table, updating the `numero_alquileres` column. An if condition checks if the new value is not equal to the old value. A compilation error is shown in the bottom pane: `PL/SQL: Statement ignored` and `PLS-00201: el identificador 'NEW.NUMERO_ALQUILERES' se debe declarar`. The bottom pane also shows the results of a `SELECT * FROM ClientePLSQL;` query.

```

87. 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

```

The screenshot shows the Oracle SQL Developer interface. On the left, the code for the procedure `proc_calcular_precio_alquiler_100388` is displayed. The procedure takes four parameters: `id_alquiler`, `id_auto`, `fecha_inicio`, and `fecha_fin`. It calculates the price of the rental. A compilation error is shown in the bottom pane: `PLS-00102: De ha encontrado el símbolo "ID" cuando se esperaba uno de los siguientes: <an identifier> <a double-quoted delimited-identifier>`. The bottom pane also shows the results of a `SELECT * FROM ClientePLSQL;` query.

Este procedimiento se utiliza para calcular el precio de un alquiler de automóvil en función de la fecha de inicio, la fecha de finalización y el precio base del automóvil, pero el uso de `NEW.precio` parece estar fuera de contexto y requiere un contexto específico para su correcta ejecución.

Pero genera error porque la sintaxis es de SQL server. Además la tabla “`AlquilerPLSQL`” y “`Auto PLSQL`” no tienen la columna “`precio`”

```

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;

```

```

88. 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;

```

```

proc:CREATE PROCEDURE proc_listar_alquileres_cliente_10038
401:(
402:    IN id_cliente INT
403:)
404:AS
405:BEGIN
406:    -- Listar los alquileres del cliente
407:    SELECT *
408:    FROM AlquilerPLSQL
409:    WHERE id_cliente = id_cliente
410:END;

```

Salida de Script x | Resultado de la Consulta x | Tarea terminada en 0,07 segundos

```

Procedure PROC_LISTAR_ALQUILERES_CLIENTE_10038 compilado

LINE/COL ERROR
-----|-----
100|  PL/SQL: Se ha encontrado el símbolo "ID" cuando se esperaba uno de los siguientes: ( un identificador ) ( un double-quoted delimited-identifier )

```

Salida de Script x | Resultado de la Consulta x | Tarea terminada en 0,07 segundos

Este procedimiento se utiliza para recuperar todos los registros de alquiler pertenecientes a un cliente específico identificado por su id\_cliente. Al llamar a este procedimiento y proporcionar el id\_cliente como argumento, obtendrás una lista de alquileres asociados a ese cliente en la tabla AlquilerPLSQL.

Pero genera error porque la sintaxis es de SQL server.

Solución:

```

748: CREATE OR REPLACE PROCEDURE proc_listar_alquileres_cliente_10038
749: (
750:    p_id_cliente NUMBER
751: )
752: AS
753: BEGIN
754:    -- Utilizar un cursor para recuperar los alquileres del cliente específico
755:    FOR alquiler IN (SELECT * FROM AlquilerPLSQL WHERE id_cliente = p_id_cliente) LOOP
756:        -- Realizar acciones adicionales si es necesario
757:        -- En este caso, simplemente imprimiremos los valores de alquiler
758:        DBMS_OUTPUT.PUT_LINE('ID Alquiler: ' || alquiler.id_alquiler ||
759:                             ', ID Cliente: ' || alquiler.id_cliente || ', ID Auto: ' ||
760:                             alquiler.id_auto || ', Fecha Inicio: ' || alquiler.fecha_inicio ||
761:                             ', Fecha Fin: ' || alquiler.fecha_fin);
762:
763:    END LOOP;
764: END proc_listar_alquileres_cliente_10038;
765:
766:

```

Salida de Script x | Resultado de la Consulta x | Tarea terminada en 0,675 segundos

Procedure PROC\_LISTAR\_ALQUILERES\_CLIENTE\_10038 compilado

```

89. 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;

```

```

799: CREATE PROCEDURE proc_listar_autos_sucursal
799: (
799:     IN id_sucursal INT
799: )
799: AS
799: BEGIN
799:     -- Listar los autos de la sucursal
799:     SELECT *
799:     FROM AutoPLSQL
799:     WHERE id_sucursal = id_sucursal;
799: END;

```

Cuando se llama a este procedimiento proc\_listar\_autos\_sucursal con un valor específico de id\_sucursal, devolverá todas las filas de la tabla AutoPLSQL que pertenezcan a esa sucursal en particular.

Pero genera error porque la sintaxis es de SQL server, ademas este procedimiento genera error debido a que en la tabla "AutoPLSQL" no existe la columna "ID\_SUCURSAL"

```

799 | SELECT * FROM AutoPLSQL;

```

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2025
2	3 Nissan	Versa	2023
3	4 Kia	Rio	2023
4	6 Chevrolet	Onix	2020

```

90. 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;

```

```

899: CREATE OR REPLACE PROCEDURE proc_agregar_auto
899: (
899:     IN marca VARCHAR(255),
899:     IN modelo VARCHAR(255),
899:     IN ano INT,
899:     IN numero_disponibles INT
899: )
899: AS
899: BEGIN
899:     -- Insertar un nuevo auto
899:     INSERT INTO AutoPLSQL (marca, modelo, ano,
899:     numero_disponibles)
899:     VALUES (marca, modelo, ano, numero_disponibles);
899: END;

```

Este procedimiento genera error debido a que en la tabla "AutoPLSQL" no existe la columna "numero\_disponibles"

```

799 | SELECT * FROM AutoPLSQL;

```

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2025
2	3 Nissan	Versa	2023
3	4 Kia	Rio	2023
4	6 Chevrolet	Onix	2020

Pero genera error porque la sintaxis es de SQL server. Además, para solucionar este problema, se debe modificar el procedimiento para que

coincida con la estructura de la tabla AutoPLSQL

```
834 CREATE OR REPLACE PROCEDURE proc_agregar_auto  
835  (  
836    ID_auto_p INTEGER,  
837    marca_p VARCHAR2,  
838    modelo_p VARCHAR2,  
839    ano_p INTEGER  
840  )  
841  AS  
842 BEGIN  
843   -- Insertar un nuevo auto  
844   INSERT INTO AutoPLSQL (id_auto,marca,modelo,ano)  
845   VALUES (ID_auto_p,marca_p,modelo_p,ano_p);  
846   COMMIT;  
847 END;  
848  
849
```

Salida de Script | Resultado de la Consulta | Tarea terminada en 0,605 segundos

Procedure PROC\_AGREGAR\_AUTO compilado

Visualizar registro de la tabla “AutoPLSQL”

```
849 SELECT *FROM AutoPLSQL ;
```

Salida de Script | Resultado de la Consulta | Todas las Filas Recuperadas: 988 en 0,78

ID_AUTO	MARCA	MODELO	ANO
975	835win	bill	2003
976	836put	design	2023
977	837easy	experi...	2005
978	838street	sport	2002
979	839environm...	put	2001
980	840religious	because	1994
981	841act	televi...	2012
982	842identify	seek	1994
983	843education	enough	1991
984	844generation	work	2013
985	845kind	himself	2003
986	846back	support	1997
987	847operation	inform...	2000
988	848hot	try	1996

Agregar por medio del procedimiento “proc\_agregar\_auto”

```

851 BEGIN
852     proc_agregar_auto(1035488084,'MARCA NUEVA', 'MODELO NUEVO', 2023);
853 END;
854

```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0,275 segundos

Procedimiento PL/SQL terminado correctamente.

Visualizar nuevamente la tabla “AutoPLSQL”

```

849 SELECT *FROM AutoPLSQL ;
850
851

```

Salida de Script x Resultado de la Consulta x  
SQL | Todas las Filas Recuperadas: 989 en 5,292 segundos

ID_AUTO	MARCA	MODELO	ANO	
1	1035488084	MARCA NUEVA	MODELO NUEVO	2023
2	2	Honda	Civic	2025
3	3	Nissan	Versa	2023
4	4	Kia	Rio	2023

Efectivamente se agrego un nuevo registro

91. CREATE PROCEDURE proc\_eliminar\_auto  
(  
 IN id\_auto INT  
)  
AS  
BEGIN  
 -- Eliminar un auto  
 DELETE FROM AutoPLSQL  
 WHERE id\_auto = id\_auto;  
END;

```

871 CREATE OR REPLACE PROCEDURE proc_eliminar_auto
872 (
873     IN id_auto INT
874 )
875 AS
876 BEGIN
877     -- Eliminar un auto
878     DELETE FROM AutoPLSQL
879     WHERE id_auto = id_auto;
880 END;

```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0,720 segundos

Procedure PROC\_ELIMINAR\_AUTO compilado

LÍNEA/COL ERROR  
1/3 :SQL00101: Se ha encontrado el símbolo "ID" cuando se esperaba uno de los siguientes: *(an identifier) a double-quoted delimited-identifier*

Crea un procedimiento almacenado llamado proc\_eliminar\_auto que tiene un parámetro de entrada id\_auto de tipo INT. Este procedimiento está diseñado para eliminar un registro de la tabla AutoPLSQL con el id\_auto especificado como argumento.

Pero genera error porque la sintaxis es de SQL server.

Solución

```
883 CREATE OR REPLACE PROCEDURE proc_eliminar_auto
884 (
885     id_auto_p INTEGER
886 )
887 AS
888 BEGIN
889     -- Eliminar un auto
890     DELETE FROM AutoPLSQL
891     WHERE id_auto = id_auto_p;
892     COMMIT;
893 END;
894
```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0,606 segundos

Procedure PROC\_ELIMINAR\_AUTO compilado

Eliminar el ID (1035488084) por medio del procedimiento "proc\_eliminar\_auto"

```
901 BEGIN
902     proc_eliminar_auto(1035488084);
903 END;
904
```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0,3 segundos

Procedimiento PL/SQL terminado correctamente.

Visualizar la tabla "AutoPLSQL"

904 |

905 | SELECT \*FROM AutoPLSQL ;

Salida de Script | Resultado de la Consulta | Se han recuperado 50 filas en 0,434 segundos

The screenshot shows a database client interface with a query editor and a results window. The query editor contains two lines of SQL: '904 |' and '905 | SELECT \*FROM AutoPLSQL ;'. The results window is titled 'Resultado de la Consulta' and shows a table with 50 rows. The table has four columns: ID\_AUTO, MARCA, MODELO, and ANO. The data includes various car models from different manufacturers like Honda, Nissan, Kia, Chevrolet, Fiat, Renault, and Ford, all from the year 2023.

ID_AUTO	MARCA	MODELO	ANO
1	2 Honda	Civic	2025
2	3 Nissan	Versa	2023
3	4 Kia	Rio	2023
4	6 Chevrolet	Onix	2020
5	7 Fiat	Argo	2023
6	9 Renault	Kwid	2023
7	10 Ford	Ka	2023

Se puede observar que se elimino correctamente