

Universidad Nacional Autónoma de México

Facultad de ingeniería

Ingeniería en Computación

Laboratorio de Bases de Datos

Práctica 10: Consultas básicas en SQL y funciones de agregación

29 de junio de 2021

6 de junio de 2021

Alumno: Correa González Alfredo



## 2. Desarrollo.

1. Los 10 registros más altos del banco Santander.

Actualizo la tabla primero porque Santander estaba escrito mal

```
SQL> UPDATE RECIBOS SET OBSERVACION = 'santander'
  2  WHERE OBSERVACION = 'santader';
445 rows updated.
```

```
SQL> SELECT *
  2   FROM(
  3   SELECT * FROM RECIBOS
  4   ORDER BY MONTO DESC
  5   )
  6   WHERE OBSERVACION IN ('santander') AND ROWNUM <= 10;
```

NUM_RECIBO	MATRICULA	ID_PAGO	MONTO	MES	OBSERVACION
2568	10378	2	7960	OCTUBRE	santander
1258	10318	3	7950	JULIO	santander
1973	10533	1	7940	MARZO	santander
901	10461	1	7930	ENERO	santander
1506	10316	2	7900	JUNIO	santander
972	10532	1	7890	ENERO	santander
2805	10365	2	7830	SEPTIEMBRE	santander
2671	10481	1	7810	OCTUBRE	santander
2657	10467	1	7780	OCTUBRE	santander
1592	10402	3	7750	JUNIO	santander

10 rows selected.

2.Los 10 registros más alto del banco Banamex

```
SQL> SELECT *
  2   FROM(
  3   SELECT * FROM RECIBOS
  4   ORDER BY MONTO DESC
  5   )
  6   WHERE OBSERVACION IN ('banamex') AND ROWNUM <= 10;
```

NUM_RECIBO	MATRICULA	ID_PAGO	MONTO	MES	OBSERVACION
2477	10537	1	7970	NOVIEMBRE	banamex
1983	10543	1	7970	MARZO	banamex
2	10312	1	7950	ABRIL	banamex
1485	10545	1	7930	JULIO	banamex
221	10531	1	7920	ABRIL	banamex
2439	10499	1	7910	NOVIEMBRE	banamex
1442	10502	1	7880	JULIO	banamex
682	10492	1	7860	DICIEMBRE	banamex
281	10341	1	7850	AGOSTO	banamex
2702	10512	1	7850	OCTUBRE	banamex

10 rows selected.

3.Los 10 registros más alto entre el banco Santander y Banamex

```
SQL> SELECT *
  2   FROM(
  3   SELECT * FROM RECIBOS
  4   ORDER BY MONTO DESC
  5   )
  6   WHERE (OBSERVACION IN ('santander') OR OBSERVACION IN ('banamex')) AND ROWNUM <= 10;
```

NUM_RECIBO	MATRICULA	ID_PAGO	MONTO	MES	OBSERVACION
2477	10537	1	7970	NOVIEMBRE	banamex
1983	10543	1	7970	MARZO	banamex
2568	10378	2	7960	OCTUBRE	santander
1258	10318	3	7950	JULIO	santander
2	10312	1	7950	ABRIL	banamex
1973	10533	1	7940	MARZO	santander
901	10461	1	7930	ENERO	santander
1485	10545	1	7930	JULIO	banamex
221	10531	1	7920	ABRIL	banamex
2439	10499	1	7910	NOVIEMBRE	banamex

10 rows selected.

4.Mostrar el promedio de montos por mes que sean arriba de 4 500 000

Con promedio de 4 500 000 ningún mes cumple esa condición.

```
SQL> SELECT MES, avg(MONTO) from RECIBOS
  2   GROUP by MES
  3   HAVING avg(MONTO)>4500000;
```

no rows selected

Pero al bajarle el valor del promedio pedido si va a ver meses que cumplan esa condición.

```
SQL> SELECT MES, avg(MONTO) from RECIBOS
  2   GROUP by MES
  3   HAVING avg(MONTO)>4500;
```

MES	AVG(MONTO)
DICIEMBRE	4751.72
JULIO	4764.04
MAYO	4623.88
ABRIL	4615.56
ENERO	4568.68
JUNIO	4585.08
SEPTIEMBRE	4601.24

7 rows selected.

```
SQL> SELECT MES, avg(MONTO) from RECIBOS
  2   GROUP by MES
  3   HAVING avg(MONTO)>500;
```

MES	AVG(MONTO)
DICIEMBRE	4751.72
JULIO	4764.04
MAYO	4623.88
OCTUBRE	4395.84
ABRIL	4615.56
ENERO	4568.68
JUNIO	4585.08
AGOSTO	4395.2
FEBRERO	4469.56
NOVIEMBRE	4261.48
SEPTIEMBRE	4601.24

MES	AVG(MONTO)
MARZO	4486.16

12 rows selected.

```
SQL> SELECT MES, avg(MONTO) from RECIBOS
  2   GROUP by MES
  3   HAVING avg(MONTO)>0;
```

MES	AVG(MONTO)
DICIEMBRE	4751.72
JULIO	4764.04
MAYO	4623.88
OCTUBRE	4395.84
ABRIL	4615.56
ENERO	4568.68
JUNIO	4585.08
AGOSTO	4395.2
FEBRERO	4469.56
NOVIEMBRE	4261.48
SEPTIEMBRE	4601.24

MES	AVG(MONTO)
MARZO	4486.16

12 rows selected.

5. Mostrar los apellidos únicamente 1 vez

```
SQL> SELECT APELLIDO, COUNT(APELLIDO)
  2  FROM ALUMNO
  3  GROUP BY APELLIDO
  4  HAVING COUNT(APELLIDO) = 1;
```

APELLIDO	COUNT(APELLIDO)
Albarran	1
Avonce	1
Camera	1
Cano	1
Chavarria	1
Contreras	1
Escobar	1
Hernan	1
Kimura	1
Lugo	1
Macarro	1

  

APELLIDO	COUNT(APELLIDO)
Maximo	1
Mendoza	1
Mercado	1
Muela	1
Oliva	1
Olivera	1
Torres	1
Aguilera	1
Baca	1
Baylon	1
Clemente	1

6. Mostrar cuántos alumnos tenemos por apellido, pero solo en aquellos apellidos con más de 5 alumnos.

```
SQL> SELECT APELLIDO, COUNT(*)
  2  FROM ALUMNO
  3  GROUP BY APELLIDO
  4  HAVING COUNT(*) > 5;
```

APELLIDO	COUNT(*)
Lopez	6
Garcia	22
López	17
Calvillo	19

Comprobando que hay 6 alumno con apellido Lopez

```
SQL> SELECT * FROM ALUMNO
2  WHERE APELLIDO IN ('Lopez');

MATRICULA NOMBRE      APELLIDO      ID_CAMPUS  ID_CARRERA
-----
10545 Jose      Lopez          4          14
10546 José      Lopez          1          14
10547 Josefina   Lopez          3          12
10548 Juan       Lopez          4          13
10550 Joahana    Lopez          1          11
10565 Karla      Lopez          4          12

6 rows selected.
```

7. Mostrar los 10 recibos más bajos por forma de pago 1 realizados en algún mes que inicie con J o termine con E .

```
SQL> SELECT *
2  FROM(
3  SELECT * FROM RECIBOS
4  ORDER BY MONTO ASC
5  )
6  WHERE (MES LIKE 'J%' OR MES LIKE '%E') AND ROWNUM <= 10;

NUM_RECIBO MATRICULA      ID_PAGO      MONTO MES      OBSERVACION
-----
1541      10351          2          1000 JUNIO      coppel
685      10495          1          1000 DICIEMBRE  bbva
2367      10427          1          1010 NOVIEMBRE  santander
2416      10476          1          1010 NOVIEMBRE  santander
656      10466          1          1010 DICIEMBRE  banamex
2498      10558          1          1020 NOVIEMBRE  elektra
2696      10506          1          1030 OCTUBRE   banorte
2509      10319          1          1040 OCTUBRE   santander
1451      10511          1          1040 JULIO      bbva
1648      10458          1          1040 JUNIO      hsbc

10 rows selected.
```

### **3.- CONCLUSIONES.**

En esta práctica utilicé las consultas de agregación count y avg, en algunos ejercicios como en los primeros 2 tenía en mente utilizar MAX(monto) para que me mostrará los primeros 10 valores más grandes de monto pero vi que esa no era la manera correcta de hacerlo. También tuve problemas ya que como tengo la versión 11 g del Oracle, no existe la instrucción fetch para mostrarme los primeros o últimos 10 valores así que tuve que ver la forma de hacerlo con ROWNUM, pero tenía que saber dónde ponerlo correctamente ya que por ejemplo en el ejercicio 3 ponía ROWNUM antes del ORDER BY y los resultados lanzados no eran los esperados. En conclusión, los objetivos de la práctica fueron logrados porque utilicé y practiqué las consultas básicas y funciones de agregación.