Profesor
M. en I.A. Erick Orlando Matla Cruz
Ayudantes
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

Práctica 07

Lenguaje de Manipulación de Datos (DML) Paginación y Composición

1. Objetivo General

Profundizar en el uso de paginación y consultas compuestas.

2. Objetivos Secundarios

- Crear scripts (código en SQL) de mayor nivel para el caso de inserciones, actualizaciones y consultas.
- Incrementar la probabilidad de dar respuesta a diversos problemas a través de una base de datos.

3. Introducción

Sabemos que uno de los objetivos de las bases de datos es ofrecer soluciones a través del correcto y oportuno manejo de información. Sin embargo, en muchas ocasiones el entorno del problema o la naturaleza de éste incrementan su complejidad, lo que obliga a los usuarios de las base de datos a proveer soluciones eficaces.

Para crear estas soluciones mediante una base de datos y un SMBD que maneje la solución, utilizaremos el lenguaje SQL. Este lenguaje nos permite manipulaciones básicas como lo es la inserción, actualización, consulta y eliminación de los datos, tal como se trabajó en la práctica anterior.

Construir consultas de mayor nivel requiere de algunas destrezas que se adquieren a través del estudio del problema y la experiencia personal en el manejo de éste lenguaje. Algunas de estas destrezas son:

1. *Entendimiento del problema*
   El primer paso para construir una solución es entender cuál es el problema, en qué contexto se encuentra inmerso y cuáles son los beneficios deseados por los involucrados.

2. *Planteamiento de posibles soluciones*
   Teniendo claro el punto anterior, se pueden construir secuencias de pasos lógicos que nos lleven a resolverlo. El conjunto de estas secuencias formará un conjunto de posibles soluciones.

3. *Traducción del planteamiento a código SQL*
   Los pasos lógicos identificados se traducen formalmente a código SQL como instrucciones precisas dirigidas al SMBD para que las ejecute.

4. *Análisis de resultados*
   Los resultados que arroja el código en forma de tuplas pueden mostrar diferencias debido a los diversos caminos que se siguieron. Es por eso que es de suma importancia validar los resultados obtenidos contra los resultados esperados.

3.1. Funciones de paginación

La paginación es útil para restringir los registros que se retornan en una consulta SELECT.

En este apartado veremos la utilidad de ORDER BY, OFFSET, LIMIT y FETCH NEXT.

Si se desea consultar registros a partir de cierta posición en la tabla, es decir, omitir registros de la tabla y regresar una cierta cantidad de tuplas, se tiene las sentencias OFFSET y LIMIT o lo que es equivalente a OFFSET y FETCH NEXT n ROWS ONLY. Además, si se desea complementar estos resultados de forma ordenada, se tiene el comando ORDER BY.

A continuación, se describe su uso.

```
SELECT column1,column2 FROM Tabla ORDER BY column1
OFFSET n FETCH NEXT m ROWS ONLY;
```

Figura A.1-Sintaxís de Paginación y Orden.

```
SELECT column1,column2 FROM Tabla ORDER BY column1
OFFSET n LIMIT m;
```

Figura A.2-Paginación con LIMIT.

En esta sintaxis encontramos tanto en A.1 y A.2 lo siguiente:

1. SELECT
   Palabra reservada para indicar que atributos se mostrarán como resultado de la consulta.
2. column1, column2
   Es el nombre de la columna, o columnas separadas por comas, que queremos consultar.
3. FROM
   Palabra reservada para indicarle al SMBD la tabla de la cual obtener la columna que necesitamos

4. Tabla
   Nombre de la tabla necesarias para obtener las columnas que se definieron en la cláusula SELECT.
5. ORDER BY
   Palabra reservada para indicar que se ordenara la tabla por medio de la column1.
6. OFFSET n
   Palabra reservada para indicar la paginación a partir del registro n de la Tabla.
7. FETCH NEXT m ROWS ONLY
   Palabras reservadas que indica el límite de m registros a regresar en la consulta.
8. LIMIT m
   Palabra reservada que indica el límite de m registros a regresar en la consulta. Como se ha indicado LIMIT es una versión más corta de FETCH NEXT m ROWS ONLY.
9. ;
   Indica la terminación de la instrucción.

Tener en cuenta que ORDER BY, OFFSET, LIMIT y/o FECTH NEXT m ROWS ONLY se pueden usar por separado.

Ejemplos:

```
SELECT Id_Orden, Cliente FROM Cliente ORDER BY Id_Orden
OFFSET 2 FETCH NEXT 4 ROWS ONLY;
```

Figura A.3-Paginación ordenada de clientes con FETCH NEXT.

```
SELECT Id_Orden, Cliente FROM Cliente ORDER BY Id_Orden
OFFSET 2 LIMIT 4;
```

Figura A.4-Paginación ordenada de clientes con LIMIT.

Tanto la consulta de la figura A.3 y figura A.4 regresan las mismas tuplas, omitiendo los primeros 2 Clientes (101 y 102), ordenados por Id_orden y solo 4 registros, ver Tabla B.

Tabla B Tuplas de consulta de A.3 o A.4

Id_orden	Cliente
103	Sara
104	Norma
105	Jorge
106	Carlos

3.2. Funciones de Composición

De no existir los datos en una misma tabla, existen operadores que permiten integrar la información de dos o más tablas. A estos operadores se les conoce como JOINs. En la Figura 8.8 se muestra un ejemplo de un NATURAL JOIN entre dos tablas.

```
SELECT Id_Orden, Id_Cliente, Total
FROM Orden NATURAL JOIN Orden_por_Cliente;
```

Figura 8.8 - Sintaxis de NATURAL JOIN entre tablas.

En esta sintaxis encontramos:

1. Orden
   Nombre de una de las tablas necesarias para obtener las columnas que se definieron en la cláusula SELECT.
2. NATURAL JOIN
   Palabra reservada para indicar que se realizará una composición entre dos tablas.
3. Orden_por_Cliente
   Nombre de una de las tablas necesarias para obtener las columnas que se definieron en la cláusula SELECT.

Esta consulta arrojará como resultado lo definido en la Tabla 8.10.

Tabla 8.10 - Resultado de la consulta de la Figura 8.8.

Id_Orden	***Id_Cliente***	***Total***
20501	109	347.90
20502	106	1014.30
20503	108	2325.54
20504	108	486.08
20505	110	568.40
20506	101	2840.04
20507	102	62.72
20508	110	1229.90
20509	107	3350.62
20510	107	1123.08
20511	103	292.04
20512	104	2380.42
20513	110	899.64
20514	105	846.72
20515	109	911.40
20516	105	1865.92
20517	102	583.10

Existen otros tipos de conectores de tablas como lo son LEFT OUTER JOIN, RIGHT OUTER JOIN, INNER JOIN y CROSS JOIN, mismos que serán descritos a continuación.

- **INNER JOIN**
  Permite especificar una condición para realizar la composición de registros. Para determinar ésta condición se utiliza la palabra reservada **ON**. Es importante notar que en la siguiente sintaxis en la condición **ON**, se encuentra el nombre de la tabla que contiene la columna a la cual queremos hacer referencia separados por un punto (Costo.Id_Articulo). Esto se hace para evitar ambigüedad en los nombres de las columnas que se encuentran en ambas tablas. Ver Figura 8.9.

```
SELECT *
FROM Costo INNER JOIN Detalle_de_Orden
ON Costo.Id_Articulo = Detalle_de_Orden.Id_Articulo
```

Figura 8.9 - Sintaxis de INNER JOIN entre tablas.

- **LEFT OUTER JOIN** y **RIGHT OUTER JOIN**
  Realiza la composición entre dos tablas, es decir, regresa un registro de la primera tabla por cada match que haga con la segunda tabla, asignando NULL a todas las columnas de aquellos registros que no tengan correspondencia con la segunda tabla, ver Figura 8.10.

```
SELECT *
FROM Cliente LEFT JOIN Orden_por_Cliente
ON Cliente.Id_Cliente = Orden_por_Cliente.Id_Cliente
```

Figura 8.10 - Sintaxis de NATURAL JOIN entre tablas.

Para el caso de LEFT los valores NULL serían asignados a los atributos sin correspondencia de la segunda tabla. Ver Tabla 8.11

Tabla 8.11 - Resultado aplicar un LEFT OUTER JOIN.

Id_Cliente	***Nombre***	***Ap_Materno***	***Ap_Paterno***	***Id_Orden***	***Id_Cliente***
101	David	Avila	Martínez	20506	101
102	Genaro	López	Villegas	20517	102
102	Genaro	López	Villegas	20507	102
103	Sara	Monroy	Vega	20511	103
104	Norma	Ponce	Noriega	20512	104
105	Jorge	Arzola	Cervantes	20516	105
105	Jorge	Arzola	Cervantes	20514	105
106	Carlos	Lozano	Morales	20502	106
107	Ricardo	Navarro	Tapia	20509	107
107	Ricardo	Navarro	Tapia	20510	107
108	Isidro	Juárez	Altamirano	20503	108
108	Isidro	Juárez	Altamirano	20504	108
109	Estefanía	González	Leyva	20515	109
109	Estefanía	González	Leyva	20501	109
110	María	Meza	Segura	20513	110
110	María	Meza	Segura	20508	110
110	María	Meza	Segura	20505	110
100	Edna	Pérez	León		

Para la tabla anterior se puede apreciar que el valor del último registro para las columnas Id_Orden y la segunda de Id_Cliente aparecen en blanco (NULL para PostgreSQL). Esto quiere decir que el cliente con id = 100 no tiene ordenes expedidas.

Mientras que para RIGHT JOIN, los valores NULL se asignarán a los atributos sin correspondencia de la primera tabla. Ver Tabla 8.12.

Tabla 8.12 - Resultado de aplicar un RIGHT OUTER JOIN.

Profesor
M. en I.A. Erick Orlando Matla Cruz
Ayudantes
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

Id_Orden	Id_Articulo	Cantidad	Total_por_Art	Id_Articulo	Costo_Unitario
20508	A-3	7	418.46	A-3	58.78
20511	A-3	1	59.78	A-3	58.78
20515	A-4	15	911.4	A-4	60.76
20505	A-4	5	303.8	A-4	60.76
20512	B-3	11	689.92	B-3	62.72
20507	B-3	1	62.72	B-3	62.72
20501	B-3	1	62.72	B-3	62.72
20510	C-2	9	582.12	C-2	64.68
20512	B-5	25	1690.5	B-5	67.62
20502	B-5	5	338.1	B-5	67.62
20514	B-5	4	270.48	B-5	67.62
20506	C-4	42	2840.04	C-4	67.62
20508	C-4	12	811.44	C-4	67.62
20502	C-4	10	676.2	C-4	67.62
20510	C-4	8	540.96	C-4	67.62
20501	C-4	3	202.86	C-4	67.62
20504	C-4	2	135.24	C-4	67.62
20511	C-4	1	67.62	C-4	67.62
20516	C-5	4	301.48	C-5	75.46
20504	C-5	2	150.92	C-5	75.46
20503	C-5	2	150.92	C-5	75.46
20516	A-5	19	1564.08	A-5	82.32
20514	A-5	7	576.24	A-5	82.32
20511	A-5	2	164.64	A-5	82.32
20505	A-5	2	164.64	A-5	82.32
20501	A-5	1	82.32	A-5	82.32
20517	C-6	7	582.12	C-6	83.3
20509	A-6	14	1358.28	A-6	97.02
20503	A-6	7	679.14	A-6	97.02
20513	B-4	9	899.64	B-4	99.96
20504	B-4	2	199.92	B-4	99.96
20505	B-4	1	99.96	B-4	99.96
20509	B-2	19	1992.34	B-2	104.86
20503	A-2	14	1495.48	A-2	106.82
				Z-99	666

- **CROSS JOIN** es una unión entre LEFT y RIGHT OUTER JOIN, regresando el producto cartesiano de los registros de ambas tablas. Es decir, si la primera tabla almacena 5 registros y la segunda almacena 7 registros, el total de registros será 5 x 7 = 35. Ver Tabla 8.13

Tabla 8.13 - Resultado de aplicar un CROSS JOIN.

Profesor
M. en I.A. Erick Orlando Matla Cruz
Ayudantes
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

Id_Cliente	*Nombre*	*Ap_Materno*	*Ap_Paterno*	*Id_Orden*	*Id_Cliente*
108	Isidro	Juárez	Altamirano	20506	101
105	Jorge	Arzola	Cervantes	20506	101
100	Edna	Pérez	León	20506	101
109	Estefanía	González	Leyva	20506	101
101	David	Avila	Martínez	20506	101
106	Carlos	Lozano	Morales	20506	101
104	Norma	Ponce	Noriega	20506	101
110	María	Meza	Segura	20506	101
107	Ricardo	Navarro	Tapia	20506	101
103	Sara	Monroy	Vega	20506	101
102	Genaro	López	Villegas	20506	101
108	Isidro	Juárez	Altamirano	20517	102
105	Jorge	Arzola	Cervantes	20517	102
100	Edna	Pérez	León	20517	102
109	Estefanía	González	Leyva	20517	102
101	David	Avila	Martínez	20517	102
106	Carlos	Lozano	Morales	20517	102
104	Norma	Ponce	Noriega	20517	102
110	María	Meza	Segura	20517	102
107	Ricardo	Navarro	Tapia	20517	102
103	Sara	Monroy	Vega	20517	102
102	Genaro	López	Villegas	20517	102
⋮	⋮	⋮	⋮	⋮	⋮
108	Isidro	Juárez	Altamirano	20508	110
105	Jorge	Arzola	Cervantes	20508	110
100	Edna	Pérez	León	20508	110
109	Estefanía	González	Leyva	20508	110
101	David	Avila	Martínez	20508	110
106	Carlos	Lozano	Morales	20508	110
104	Norma	Ponce	Noriega	20508	110
110	María	Meza	Segura	20508	110
107	Ricardo	Navarro	Tapia	20508	110
103	Sara	Monroy	Vega	20508	110
102	Genaro	López	Villegas	20508	110
108	Isidro	Juárez	Altamirano	20505	110
105	Jorge	Arzola	Cervantes	20505	110
100	Edna	Pérez	León	20505	110
109	Estefanía	González	Leyva	20505	110
101	David	Avila	Martínez	20505	110
106	Carlos	Lozano	Morales	20505	110
104	Norma	Ponce	Noriega	20505	110
110	María	Meza	Segura	20505	110
107	Ricardo	Navarro	Tapia	20505	110
103	Sara	Monroy	Vega	20505	110
102	Genaro	López	Villegas	20505	110

Profesor
M. en I.A. Erick Orlando Matla Cruz
Ayudantes
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

4. Ejercicios

(NOTA: Resuelve los siguientes ejercicios en relación al proyecto que realizarás durante el curso, en dado caso que no tengas un proyecto, utiliza la información en el apéndice Seguros parte 08 al final de esta práctica para realizarlos)

1. Realiza 5 consultas a diferentes tablas donde utilices funciones de paginación.
2. Realiza 5 consultas que conecten diferentes tablas donde utilices funciones de composición (utiliza diferentes tipos de JOIN).
3. Realiza 2 consultas donde utilices JOIN con tablas renombradas.

Entregables requeridos para prácticas subsecuentes, un archivo en formato SQL:

- 5 consultas con JOIN y funciones de paginación.
- 5 consultas con JOIN y funciones de composición.
- 2 consultas con JOIN y tablas renombradas.