INTRODUCCIÓN SQL

(Structured Query Language)

ORACLE

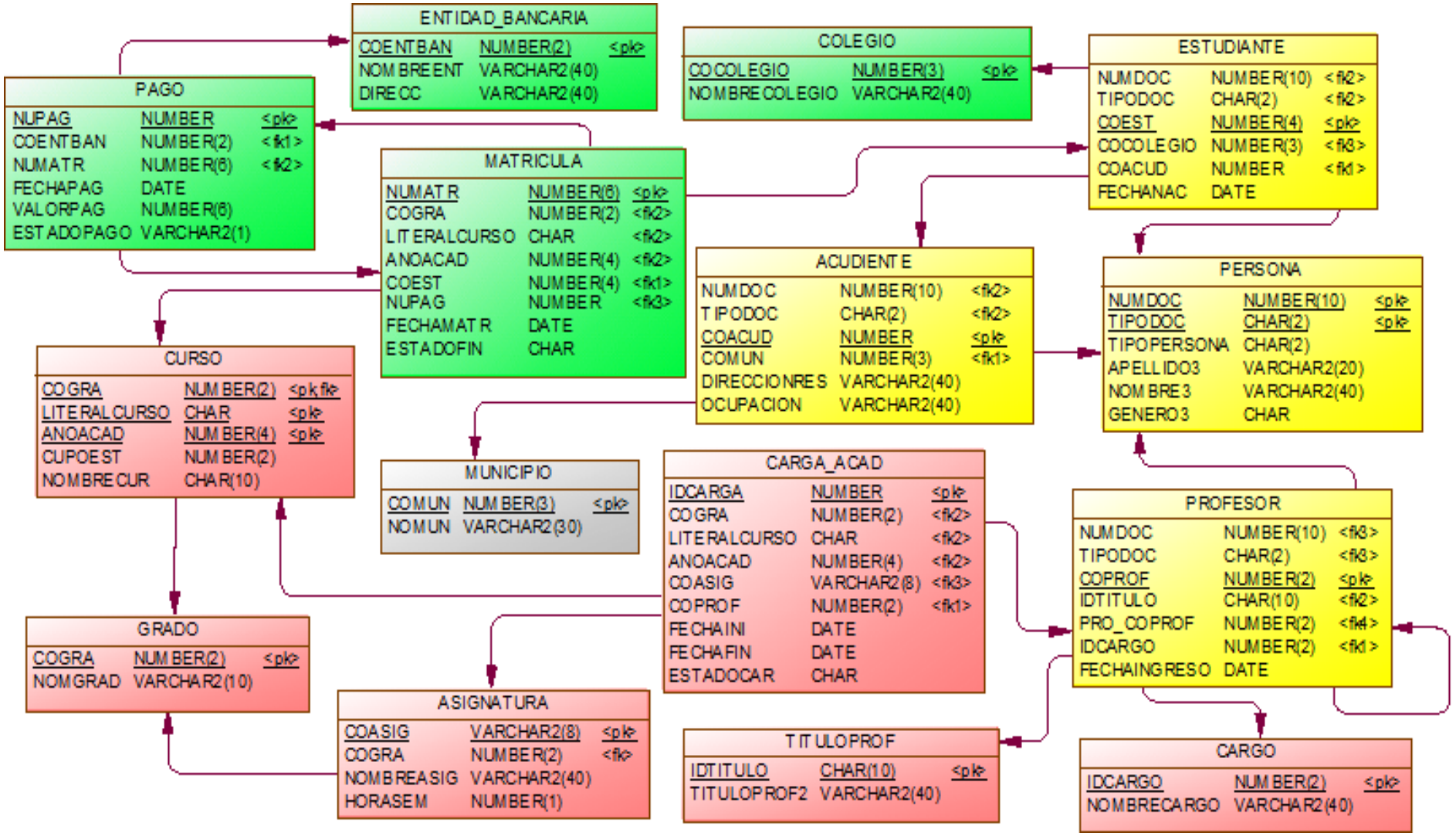
Carlos A. Mendez Reyes
Cel. 3143763009
Mail. alberto_reyes_77@hotmail.com

Tratar de hacer todo bien, desde el principio hasta el final, cuidando los detalles.
Siempre pensando en reutilizar.

ALISTAMIENTO

- Descargar e Instalar Versión Express Oracle (Ver Tutorial).
- Descargar, Ejecutar y Conectar SqlDeveloper con Versión Express Oracle.
- Crear y cargar esquema. matricula.(Ver Script)

MODELO GUÍA – TALLER MATRICULA



CARACTERÍSTICAS SQL

- DDL, Lenguaje de definición de datos (Create, Drop, Alter y Truncate).
- DCL, Lenguaje de Control de datos (Grant, Revock).
- TCL, Lenguaje de Control de transacciones (Commit, Rollback).
- DML, Lenguaje de Manipulación de datos (Select, Insert, Delete, Update).

DDL - APLICACIONES

Sentencias: CREATE, DROP, ALTER, TRUNCATE

TABLE	USER
INDEX	PROFILE
SEQUENCE	ROLE
SYNONYM	FUNCTION
VIEW	PROCEDURE
TABLESPACE	PACKAGE ……

DCL - APLICACIONES

Sentencias: GRANT, REVOKE

- GRANT (to) y REVOKE (from): Asignación y Revocación de Permisos y privilegios, sobre roles, profiles, tablas y demás objetos de la BD

TCL - APLICACIONES

Sentencias: COMMIT, ROLLBACK, SAVEPOINT

- COMMIT, ROLLBACK y SAVEPOINT : Control de confirmación o no, de transacciones.

DML: INSERT

Agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Formato:

INSERT INTO <tabla> [(<columna1>, <columna2>, ...)]

{ VALUES (<valor columna1>, <valor columna2>, ...) }

Ejemplos:

```
INSERT INTO colegio(cocolegio, nombrecolegio)
    VALUES (100, 'Colegio Americano');

INSERT INTO colegio(cocolegio, nombrecolegio)
    VALUES (200, 'Colegio Ingles');
```

DML: INSERT SELECT

Agregar registros a una tabla a partir del resultado de una consulta.

Formato:

INSERT INTO <tabla> [(<columna1>, <columna2>, ...)]

{ SELECT <columna1>, <columna2>, ... FROM <tabla>}

Ejemplo:

```
INSERT INTO colegio_aux(cocolegio, nombrecolegio)
  SELECT cocolegio, nombrecolegio
   FROM colegio;
```

DML: INSERT – FORMATOS - SECUENCIAS

Insertar campos de tipo Date

Ejemplo:

```
INSERT INTO pago (nupag,coentban,numatr,fechapag)
 VALUES (100, 1,1, to_date('01-01-2001 01:01:01', 'dd-mm-yyyy hh24:mi:ss') );
```

Insertar campos asociando una secuencia

Ejemplo:

```
INSERT INTO pago (nupag,coentban,numatr,fechapag)
VALUES( sq_nombre.nextval , 1,1, to_date('01-01-2001 01:01:01', 'dd-mm-yyyy hh24:mi:ss'));
```

DML: UPDATE

Permite modificar datos de columnas en filas existentes.

Formato:

```
UPDATE <tabla>
    SET <columna1> = <expresion1> [, <columna2> = <expresion2>, ...]
[ WHERE <predicado> ]
```

Ejemplo:

```
UPDATE colegio
    SET nombrecolegio= 'Colegio Los Samanes'
WHERE  cocolegio= 100;
```

DML: UPDATE POR SUBQUERY Y ROWID

Actualizar múltiples campos a partir del resultado de una consulta.

Formato:

```
UPDATE <tabla>
    SET (<columna1>,<columna2>) =  (SELECT <columna1> , <columna2>
                                    FROM tabla
                                    WHERE <condicion>)
```

Actualizar campos por ROWID

Formato:

```
UPDATE <tabla>
    SET (<columna1> = valor_columna1)
WHERE  ROWID = sbROWID
```

DML: DELETE

Permite Eliminar una o mas filas de una tabla.

Formato:

DELETE FROM <tabla>

(WHERE <condición>)

O

DELETE <tabla>

(WHERE <condición>)

Ejemplo:

```
DELETE colegio
 WHERE cocolegio= 200;
```

DML: SELECT Y COMPONENTES

La sentencia SELECT permite seleccionar u obtener datos de una o de varias tablas:

Formato:

SELECT [DISTINCT | ALL] {* | <expr1>[, <expr2>] ...}

FROM <tabla1>[, <tabla2>, ...]

[WHERE <condicion_where>]

[GROUP BY <group_expr1>[, <group_expr2>, ...]

[HAVING <condicion_having>]

[ORDER BY <expr_orderby1 [ASC | DESC]>[, ...]]

DML: SELECT – ALIAS DE TABLA Y DE CAMPO

Ejemplos :

```
SELECT colegio.cocolegio, colegio.nombrecolegio
  FROM colegio;
```

```
SELECT colegio.cocolegio, colegio.nombrecolegio
  FROM colegio
WHERE colegio.cocolegio = 1;
```

```
SELECT c.*
  FROM colegio c
WHERE c.cocolegio = 1;
```

```
SELECT c.cocolegio as "Codigo Colegio",
           c.nombrecolegio as "Nombre Colegio"
  FROM colegio c
WHERE c.cocolegio = 1;
```

DML: SELECT OPERADORES

- Operadores Aritméticos (+, -, x, /)
- Condiciones de Comparación (>, >=, <, <=, <>, BETWEEN AND, IN, LIKE, IS NULL)
- Condiciones Lógicas (AND, OR, NOT)
- Operador de Concatenación (||)
- Funciones con caracteres (Substr, Lower, Upper, Initcap)
- Manejo de Valores Nulos (NVL , IS NOT NULL)

DML: SELECT APLICACIÓN OPERADORES

Ejemplo 1:

```
select nupag, numatr, valorpag, ( valorpag * 19 / 100 ) Iva
  from pago
 where valorpag > 10000 and valorpag <= 45000;
```

Ejemplo 2:

```
select nupag, numatr, estadopago, fechapag, nvl(fechapag,sysdate) fecha_ajustada
  from pago
 where estadopago = 'P' or fechapag is not null
   or trunc(fechapag) <= to_date('10-06-2015','dd-mm-yyyy');
```

Ejemplo 3:

```
select tipodoc || ' - ' || numdoc codigocliente, lower(nombre3) nomlower,
    initcap(nombre3) initcapnom, upper(nombre3) nomlower, substr(nombre3,1,3)
sub3
  from persona
 where nombre3 like '%A%';
```

DML: SELECT – EJERCICIOS

Ejemplos

Listar de las personas el nombre y apellido , genero y tipo de documento de todos aquellas que pertenezcan al genero masculino, y que su tipo de documento sea cedula de ciudadanía.

	NOMBRES	GENERO	TIPODOCUMENTO
1	JULIAN RODRÍGUEZ	M	cc
2	RAMON AYALA	M	cc
3	JORGE PACHECO CASADIEGO	M	cc

Listar de las personas el nombre y apellido , genero y tipo de documento de todos aquellas que pertenezcan al genero femenino, y que en alguna parte de su nombre o apellido tengan la palabra 'ez' o 'es', se deben contemplar las variaciones entre minúscula y mayúscula.

	NOMBRES	GENERO	TIPODOCUMENTO
1	LILIANA MARTÍNEZ	F	ti
2	MARINA RODRÍGUEZ	F	ti
3	ESTELLA RUBIO	F	cc
4	Yisel Del Carmen Saez Castro	F	cc

Listar de los pagos el numero, fecha, valor, y valor del pago descontando el IVA del 19%, de los pagos que su valor de IVA sea mayor a 48,000 y que el pago se hubiera registrado con fecha mayor o igual al 27 de mayo de 2020.

	NUPAG	FECHAPAG	VALORPAG	VALPAGO_MENOSIVA
1	14	27/05/2020 9:23:58 p. m.	260000	210600
2	19	28/05/2020 9:23:58 p. m.	255000	206550
3	20	28/05/2020 9:23:58 p. m.	255000	206550

DML: SELECT

- Consultas MultiTablas
- Funciones de Agrupamiento (MAX, MIN, AVG, COUNT, SUM).
- Criterios de Ordenamiento
- Restricción Having

DML: SELECT

```
SELECT p.numdoc, p.nombre3
  FROM persona  p
WHERE genero3 = 'M'
ORDER BY p.nombre3 asc
```

```
SELECT eb.coentban, eb.nombreent, p.valorpag, p.fechapag
FROM entidad_bancaria eb, pago p
WHERE eb.coentban = p.coentban
ORDER BY eb.nombreent asc, p.fechapag desc
```

```
SELECT max(valorpag), min(valorpag), avg(valorpag), count(nupag),sum(valorpag)
FROM pago
```

```
SELECT eb.nombreent, max(valorpag), min(valorpag), avg(valorpag),count(valorpag)
  FROM entidad_bancaria eb, pago p
WHERE eb.coentban = p.coentban
GROUP BY eb.nombreent
```

DML: SELECT

```
SELECT eb.nombreent, p.fechapag, sum(valorpag)
FROM entidad_bancaria eb, pago p
WHERE eb.coentban = p.coentban
GROUP BY eb.nombreent, p.fechapag
ORDER BY 1 desc
```

```
SELECT eb.nombreent, p.fechapag, sum(valorpag)
  FROM entidad_bancaria eb, pago p
 WHERE eb.coentban = p.coentban
GROUP BY eb.nombreent, p.fechapag
HAVING sum(valorpag) >= 50000
ORDER BY 1 desc
```

```
SELECT eb.nombreent, p.fechapag, sum(valorpag), sum( valorpag*(19/100) ) Iva_Pagos
  FROM entidad_bancaria eb, pago p
 WHERE eb.coentban = p.coentban
GROUP BY eb.nombreent, p.fechapag
ORDER BY 1 desc
```

DML: SELECT - SUBCONSULTA

Se presenta cuando es necesario utilizar el resultado de una consulta para resolver otra.

```
SELECT x.campo1, x.campo2
  FROM tabla1 x
WHERE x.campo1IN (SELECT  y.campo1
                  FROM tabla2 y);
```

```
SELECT x.campo1, x.campo2
  FROM tabla1 x
WHERE x.campo1NOT IN (SELECT  y.campo1
                      FROM tabla2 y);
```

```
SELECT x.campo1, x.campo2
  FROM tabla1 x
WHERE  EXISTS (SELECT 0
               FROM tabla2 y
               WHERE y.campo1 = x.campo1);
```

```
SELECT x.campo1, x.campo2
  FROM tabla1 x
WHERE  NOT EXISTS (SELECT 0
                   FROM tabla2 y
                   WHERE y.campo1 = x.campo1);
```

DML: SELECT - SUBCONSULTA

Ejemplo:

```
select nupag, fechapag, valorpag
   from pago
 where valorpag < ( select avg(valorpag)
                      from pago
                    where to_number(to_char(fechapag,'yyyy')) = 2020 );
```

Ejercicio

Listar nombres y apellidos de todos los estudiantes que pertenezcan al mismo colegio donde estudia el estudiante mas joven registrado en la base de datos.

	NOMBRE	APELLIDO
1	MARINA	RODRÍGUEZ
2	GUSTAVO	PEREZ

DML: SELECT – CASE - DECODE

El CASE y el DECODE permiten manejar estructura condicional dentro de una consulta

```
SELECT a.campo1, a.campo2,
        DECODE( a.campo3, <opcion1>, mostrar1, <opcion2>, mostrar2, mostradefault) )
FROM tabla a;
```

```
SELECT a.campo1, a.campo2,
  CASE a.campo3
    WHEN <opcion1> THEN mostrar1
    WHEN <opcion2> THEN mostrar2
    ELSE mostrardefault
  END
FROM tabla a;
```

DML: SELECT – CASE - DECODE

Ejemplos :

```
select Numdoc NumeroDocumento,
  decode(lower(tipodoc) , 'cc‘ , 'Cedula' , 'ti' , 'Tarjeta Identidad' , 'No definida') TipoDoc
 from persona;
```

```
SELECT Numdoc NumeroDocumento,
     CASE lower(tipodoc)
          WHEN 'cc' THEN 'Cedula'
          WHEN 'ti'  THEN 'Tarjeta Identidad'
          ELSE 'No definida'
      END TipoDoc
FROM persona a;
```

DML: SELECT – CASE - DECODE

Listar nombre y apellidos de los estudiantes, código y nombre del colegio al que pertenecen, valor de los pagos por matricula, año académico de la matricula, y valor del pago descontando el IVA aplicando los siguientes criterios:

- Si el pago se realizo en el banco BANCOLOMBIA código 1 el valor del IVA a aplicar es del 19%.
- Si el pago se realizo en el banco CITIBANK código 4 el valor del IVA a aplicar es del 18%.
- Si el pago se realizo en el banco BANCO OCCIDENTE código 3 el valor del IVA a aplicar es del 0%.
- Para cualquier pago realizado en otro banco el valor del IVA a aplicar será del 14%.

TALLER 1

- Crear una tabla de nombre persona_mujer con la estructura necesaria para almacenar en un mismo campo el nombre y apellido de la persona, tipo documento, y numero documento. Esta tabla se debe llenar con los registros de la tabla persona que pertenezcan al genero femenino.

- Crear una tabla de nombre pagos_2020 con la estructura necesaria para almacenar el nombre de la entidad donde se realizo el pago, fecha que se realizo el pago, y el valor total de los pagos realizados para esa fecha. Esta se debe llenar solo con los pagos realizados en el año 2020.

TALLER 1

- Actualizar la dirección de las entidades bancarias, pasando la dirección a minúscula, solo si en el nombre de la entidad bancaria existe la palabra 'BANCO', se deben contemplar las variaciones entre minúscula y mayúscula.

- Actualizar sobre la tabla pagos_2020 el valor del pago aumentándolo en un 40%, solo si el valor del pago es mayor a 240000, y si el pago le corresponde a un estudiante del "colegio UCC" código 99 .

TALLER 1

- Eliminar registros de la tabla persona_mujer, donde el tipo de documento sea diferente a Cedula de Ciudadanía.

- Eliminar registros de la tabla pagos_2020, donde el nombre de la entidad bancaria tenga mas de 10 caracteres (Utilizar función length(cadena) para extraer la cantidad de caracteres de una cadena).

TALLER 1

Utilizando las tablas reales del diseño, se deben realizar las siguiente consultas:

- Consultar las asignaturas definidas para un grado, de estas desplegar el Código, Nombre, Código Grado, y Horas Semana.
- Consultar las asignaturas a cargo de un profesor, desplegar el numero documento, nombre y apellidos del profesor, y nombre de la asignaturas asignadas, solo de los asignaturas correspondientes al año académico del 2020.
- El sistema deberá generar un listado de profesores, desplegar nombre y apellidos de los profesores, descripción del cargo y el titulo profesional asignado.
- Consultar listado de estudiantes matriculados, desplegar código del estudiante, nombres y apellidos, código del grado y año académico, solo de las matriculas correspondientes al año académico del 2020.

OTRAS FUNCIONES SQL

LPAD (cad1, n[,cad2])=Añade caracteres a la izquierda de la cadena hasta que tiene una cierta longitud.

RPAD (cad1, n[,cad2])=Añade caracteres a la derecha de la cadena hasta que tiene una cierta longitud.

LTRIM (cad [,set])=Suprime un conjunto de caracteres a la izquierda de la cadena.

RTRIM (cad [,set])=Suprime un conjunto de caracteres a la derecha de la cadena.

REPLACE (cad, cadena_busqueda [, cadena_sustitucion])=Sustituye un carácter o caracteres de una cadena con 0 o mas caracteres.

OTRAS FUNCIONES SQL

ABS(n)=Devuelve el valor absoluto de (n).

CEIL(n)=Obtiene el valor entero inmediatamente superior o igual a "n".

FLOOT(n) =Devuelve el valor entero inmediatamente inferior o igual a "n".

MOD (m, n)=Devuelve el resto resultante de dividir "m" entre "n".

POWER (m, exponente)=Calcula la potencia de un numero.

ROUND (numero [, m])=Redondea números con el numero de dígitos de precisión indicados.

GENERALIDADES DE RENDIMIENTO - SQL

- Al realizar operación de insert utilizar el nombrado de los campos.
- Para las consultas multitabla utilizar los alias de tabla, ayuda al motor a resolver mas eficientemente.
- Evitar el uso de búsquedas con el criterio NULL o NOT NULL, se inactivan los index incrementa el costo de la consulta.
- Tratar de no utilizar conversiones implícitas de tipos de datos.
- No utilizar el distinct y el group by en una misma consulta ya que realizan el mismo filtro.
- Evitar el uso de las consultas con select *.

Para ejecutar el plan de ejecución se antepone a la consulta la sentencia:
EXPLAIN PLAN FOR

Ejemplo:

```
SQL> EXPLAIN PLAN FOR
Select *
  From articulo a, tipo_articulo t
Where a.tipart = t.tipart;
```

Verificar Resultado en la tabla PLAN_TABLE;