

# INTRODUCCIÓN SQL

(Structured Query Language)

## ORACLE

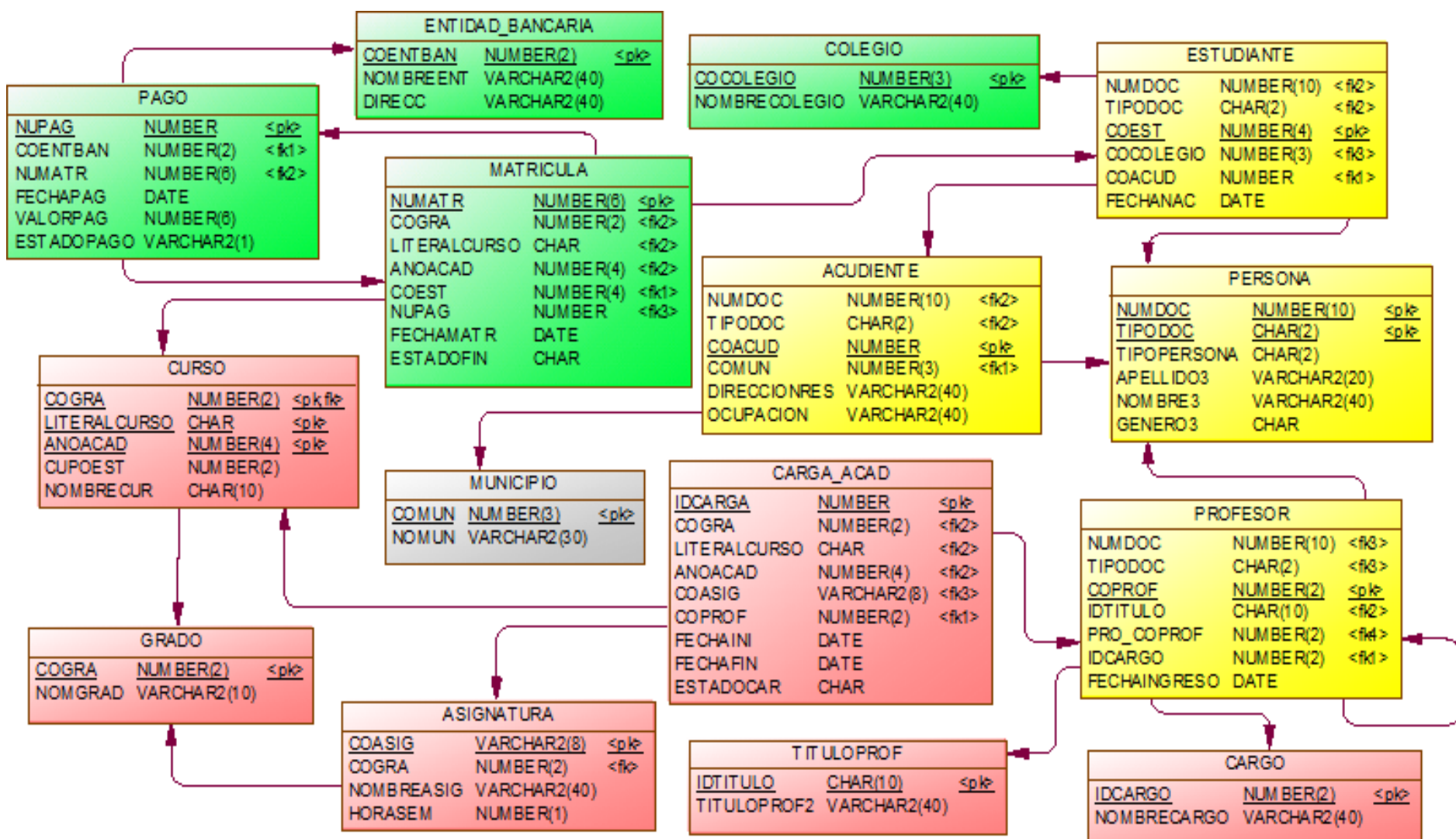


*Tratar de hacer todo bien,  
desde el principio hasta el  
final, cuidando los detalles.  
Siempre pensando en reutilizar.*



- 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). ➡

Sentencias: CREATE, DROP, ALTER, **TRUNCATE**

TABLE

INDEX

SEQUENCE

SYNONYM

VIEW

TABLESPACE

USER

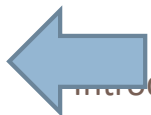
PROFILE

ROLE

FUNCTION

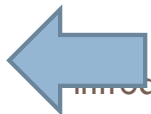
PROCEDURE

PACKAGE .....



## 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



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** )