



**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**

**Laboratorio de Bases de Datos**

**Práctica 08**

**LENGUAJE DE MANIPULACIÓN DE DATOS Y  
TRANSACCIONES**

**Semestre 2019-1**

**Integrantes:**

**Valderrama Navarro Armando**

## Introducción

En las bases de datos uno de los temas mas importantes a saber es el manejo del lenguaje DML, que ayuda a poblar de registros nuestra base de datos, pues sin estos la base no tendría razón de ser, por esa razón durante el desarrollo de esta practica se explica mediante ejemplos el correcto uso de este lenguaje dentro de las bases de datos.

## Objetivos

Comprender, practicar y verificar la instalación de herramientas asociadas a la base de datos Oracle en Linux. En este caso, se instalará TORA, una herramienta gráfica empleada para administrar una instancia de base de datos que será empleada en prácticas posteriores. Se explicará y comprenderá el uso de SQL Developer, herramienta incluida en la instalación de Oracle. Adicionalmente, reafirmar conocimientos y práctica en las instrucciones SQL asociadas con DML (Data Manipulation Language).

## Previo.

### C1. Conceptos previos.

Cmake: herramienta plataforma de generación o automatización de código.

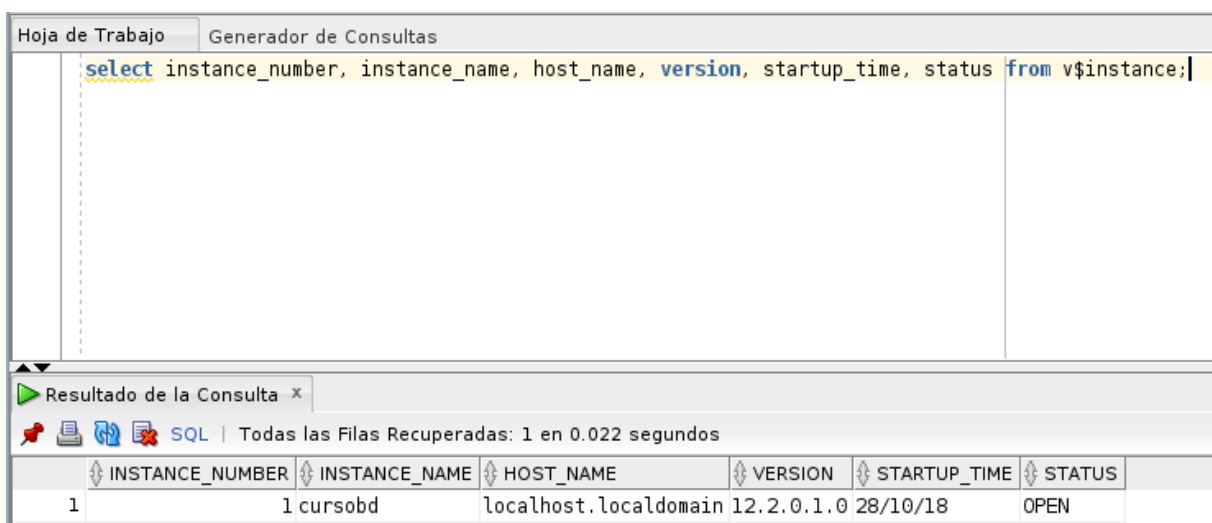
&: Se utiliza para ejecutar un proceso en segundo plano regresando el PID del proceso.

tnsnames.ora: Archivo **ubicado en los clientes**, contiene los nombres de servicio de red, asignados a descriptores a través de los cuales se nos permite acceder

### C2. Tora.

Debido a que yo uso la distribución de CentOS me fue imposible encontrar todos lo paquetes necesarios para la instalación de Tora, pues los paquetes presentados en los manuales no coincidían con ninguno de los disponibles en el gestor de CentOS.

### C3. SQL Developer



The screenshot shows the SQL Developer interface. The top pane, titled 'Hoja de Trabajo' and 'Generador de Consultas', contains the following SQL query:

```
select instance_number, instance_name, host_name, version, startup_time, status from v$instance;
```

The bottom pane, titled 'Resultado de la Consulta', displays the query results in a table. The status bar indicates 'Todas las Filas Recuperadas: 1 en 0.022 segundos'.

INSTANCE_NUMBER	INSTANCE_NAME	HOST_NAME	VERSION	STARTUP_TIME	STATUS
1	1 cursobd	localhost.localdomain	12.2.0.1.0	28/10/18	OPEN

## **Practica laboratorio.**

Nota: Todas las instrucciones DML crean una transacción y esa transacción seguirá siempre y cuando las instrucciones siguientes sean DML también.  
Las instrucciones DDL terminan las transacciones.

**C1:**

- A. 4 transacciones.
- B. deshace todos los cambios hasta el savepoint sv1.
- C. commit (implícito).
- D. realiza un commit implícito de la misma manera que el inciso anterior.

**C2:**

- A. 10
- B. 10
- C. 20
- D. Lo que pasaría es que la consulta leería un valor que ya no existe y esto generaría problemas de inconsistencia.

**C3:**

- A. 10
- B. 10
- C. Al realizar la segunda consulta, esta regresaría el valor actualizado por Txn1

**C4:**

- A. La sentencia se ejecuta sin problema alguno.
- B. Txn2 se bloquearía ya que el mismo registro fue modificado por Txn1 y Txn1 aún no termina.
- C. 30
- D. 30
- F. Se perdió, debido a que Txn2 se bloqueó mientras Txn1 realizaba el update, sin embargo al desbloquear Txn2 el valor se actualiza y el update de Txn1 se pierde.

Por default algunos clientes de bases de datos no permiten la visualización de modificaciones de datos de otros usuarios hasta que ese cliente realice un commit.

## **Práctica complementaria.**

**C1.**

```

Funcion creada.

No hay errores.
----- Resultados -----
total estatus          4
total pasajeros        2
registros t_aereo      1
registros t_terrestre  1
total medio transporte  2
registros lugar        2
total lugar            2
registros viaje         0
total viaje            1
registros historico     1
total historico         1
total ----- 12

Procedimiento PL/SQL terminado correctamente.

```

## C2.

--F sentencia 1

```

select VIAJE_ID from viaje
  where estatus_viaje_id =
        (select estatus_viaje_id from estatus_viaje where clave = 'PROGRAMADO')
        and MEDIO_TRANSPORTE_ID = (select MEDIO_TRANSPORTE_ID from
MEDIO_AEREO where NUMERO_AVION='MXA903');

```

--F sentencia 1

```

delete from VIAJE_DETALLE where VIAJE_ID = 1003;

```

--F sentencia 2

```

delete from HISTORICO_ESTATUS_VIAJE where VIAJE_ID = 1003;

```

--F sentencia 3

```

select MEDIO_TRANSPORTE_ID from MEDIO_AEREO where NUMERO_AVION = 'MXA903';

```

--F sentencia 4

```

delete from VIAJE where MEDIO_TRANSPORTE_ID=14 AND ESTATUS_VIAJE_ID=1;

```

--F sentencia 5

```

delete from MEDIO_AEREO where MEDIO_TRANSPORTE_ID = 14;

```

--F sentencia 6

```

delete from MEDIO_TRANSPORTE where MEDIO_TRANSPORTE_ID = 14;

```

### C3.

```
No hay errores.
----- Resultados -----
num pasajeros          2
total pasajeros        4
registros t_aereo      1
total medio_transporte 3
registros lugar        4
total lugar            6
registros viaje        0
total viaje            2
registros historico    1
total historico        2
update datos avion    1
update datos viaje     1
registros detalle viaje 3
total detalle viaje    3
delete transporte      0
total ----- 25

Procedimiento PL/SQL terminado correctamente.
```

### C4.

--@Autor(es): Armando Valderrama

--@Fecha creación: 22/10/2018

--@Descripción: Ejercicio 5.

prompt Conectando con el usuario vald\_p0801\_viajes  
connect vald\_p0801\_viajes

```
create table PASAJERO_TEMP as
  select * from PASAJERO;
```

```
insert into PASAJERO_TEMP (pasajero_id, nombre, apellido_paterno, apellido_materno, edad,
  num_identificacion)
  values(3, 'MARIANA', 'JIMENEZ', 'LOPEZ', 20, 'J4U58IWSLFFNEIRX8R4');
```

```
merge into PASAJERO_TEMP pt using PASAJERO p on
  (pt.PASAJERO_ID = p.PASAJERO_ID)
  when not matched then insert
    (pt.PASAJERO_ID, pt.NOMBRE, pt.APELLIDO_PATERNO,
pt.APELLIDO_MATERNO, pt.EDAD, pt.TELEFONO, pt.NUM_IDENTIFICACION) values
    (p.PASAJERO_ID, p.NOMBRE, p.APELLIDO_PATERNO,
p.APELLIDO_MATERNO, p.EDAD, p.TELEFONO, p.NUM_IDENTIFICACION);
```

commit;

```
1 fila creada.

3 filas fusionadas.

Confirmacion terminada.
```

### Conclusiones.

Se logro comprender de mejor manera el uso de las transacciones, así como el lenguaje de definición de datos.

Se logro poner en practica el uso de merge, insert y update para los registros de una tabla.