

## **CUADERNO PL/SQL**

Este cuaderno de PL/SQL tiene como objetivo reforzar la programación PL/SQL con una serie de ejercicios diseñados para ser aplicados en un entorno de desarrollo Oracle. Además de mejorar las habilidades de programación, se busca potenciar la memoria y el desarrollo lógico en la creación y manipulación de datos en bases de datos estructuradas, así como en la programación y el desarrollo dentro del entorno PL/SQL. Este material también proporcionará una oportunidad para consolidar los conocimientos en la gestión de bases de datos en distintos niveles, abarcando desde los conceptos fundamentales hasta niveles más avanzados. El cuaderno debe mantenerse al alcance y completarse a lo largo del curso, el cual se extiende durante 16 semanas según lo establecido en el plan de estudio.

El cuaderno comprende un total de 100 ejercicios diseñados para abordar diversas áreas del aprendizaje. Es esencial que cada estudiante lo lleve de manera individual y que se realice una revisión durante cada clase para verificar el progreso en los ejercicios de la semana. El seguimiento y registro del cuaderno serán elementos fundamentales que contribuirán a la evaluación académica del estudiante. Es importante destacar la relevancia de este ejercicio académico como una herramienta integral para fortalecer las competencias adquiridas en clase y en todo el proceso de aprendizaje relacionado con la temática del curso.

### **METODOLOGÍA:**

- Cada estudiante debe crear un repositorio de GitHub con el nombre de Bases de Datos.
- Se debe crear el link en el Dashboard en la sección de PL/SQL
- Se debe crear un código SQL por cada Ejercicio, realizando la explicación de cómo funciona el código y que resultados se generaron.
- En caso de que una sentencia no genere ningún resultado, explicar la razón del comportamiento de esa sentencia

### **ESTRUCTURA DE LA BASE DE DATOS:**

-- Tabla de clientes

```
CREATE TABLE ClientePLSQL (  
  id_cliente NUMBER PRIMARY KEY,
```

```
nombre VARCHAR2(50),
direccion VARCHAR2(100),
telefono VARCHAR2(15)
);
```

-- Tabla de autos

```
CREATE TABLE AutoPLSQL (
  id_auto NUMBER PRIMARY KEY,
  marca VARCHAR2(50),
  modelo VARCHAR2(50),
  ano NUMBER
);
```

-- Tabla de alquileres

```
CREATE TABLE AlquilerPLSQL (
  id_alquiler NUMBER PRIMARY KEY,
  id_cliente NUMBER,
  id_auto NUMBER,
  fecha_inicio DATE,
  fecha_fin DATE,
  id_reserva NUMBER,
  FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),
  FOREIGN KEY (id_auto) REFERENCES Auto(id_auto),
  FOREIGN KEY (id_reserva) REFERENCES Reserva(id_reserva)
);
```

-- Tabla de sucursales

```
CREATE TABLE SucursalPLSQL (
  id_sucursal NUMBER PRIMARY KEY,
  nombre VARCHAR2(50),
  ciudad VARCHAR2(50),
  pais VARCHAR2(50)
);
```

-- Tabla de reservas

```
CREATE TABLE ReservaPLSQL (
  id_reserva NUMBER PRIMARY KEY,
  id_cliente NUMBER,
  id_sucursal NUMBER,
  fecha_reserva DATE,
  FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),
  FOREIGN KEY (id_sucursal) REFERENCES Sucursal(id_sucursal)
);
```

- Cliente: Almacena información sobre los clientes, como su nombre, dirección y número de teléfono.
- Auto: Almacena información sobre los autos, como su marca, modelo y

año.

- Alquiler: Almacena información sobre los alquileres, como la fecha de inicio, la fecha de finalización y el auto alquilado.
- Sucursal: Almacena información sobre las sucursales, como su nombre, ciudad y país.
- Reserva: Almacena información sobre las reservas, como la fecha de la reserva y la sucursal en la que se realizó la reserva.

## **EJERCICIOS PRIMER CICLO (1-30):**

### **1. Consultas Básicas:**

- ☐ Mostrar todos los clientes en la tabla "Cliente".
- ☐ Mostrar todos los autos en la tabla "Auto".
- ☐ Mostrar todos los alquileres en la tabla "Alquiler".
- ☐ Mostrar todas las sucursales en la tabla "Sucursal".
- ☐ Mostrar todas las reservas en la tabla "Reserva".

### **2. Filtros y Ordenamiento:**

- ☐ Mostrar los clientes que se llaman "Juan".
- ☐ Mostrar los autos de marca "Toyota".
- ☐ Mostrar los alquileres que ocurrieron después de una fecha específica.
- ☐ Mostrar las sucursales ubicadas en "Madrid".
- ☐ Mostrar las reservas realizadas por un cliente específico.

### **3. Join y Relaciones:**

- ☐ Mostrar los alquileres con los nombres de los clientes y las marcas de los autos.
- ☐ Mostrar los clientes que han realizado reservas en una sucursal específica.
- ☐ Mostrar los autos que han sido alquilados junto con los nombres de los clientes.
- ☐ Mostrar los detalles de las reservas con los nombres de los clientes y las ciudades de las sucursales.
- ☐ Mostrar los clientes que no han realizado ninguna reserva.

### **4. Agregación y Agrupamiento:**

- ☐ Contar cuántos autos hay de cada marca en la tabla "Auto".
- ☐ Calcular la duración promedio de los alquileres.
- ☐ Mostrar el número total de reservas realizadas en cada sucursal.
- ☐ Encontrar el cliente que ha realizado la mayor cantidad de alquileres.
- ☐ Calcular el promedio de años de los autos en la tabla "Auto".

### **5. Subconsultas:**

- ☐ Mostrar los clientes que han realizado al menos una reserva.
- ☐ Mostrar los autos que no han sido alquilados aún.
- ☐ Encontrar los clientes que han alquilado el mismo auto más de una vez.

- ☐ Mostrar los clientes que han realizado alquileres en la misma ciudad en la que viven.
- ☐ Encontrar los autos que han sido alquilados en la misma sucursal donde se realizó una reserva.

#### 6. Actualizaciones y Eliminaciones:

- ☐ Actualizar la dirección de un cliente específico.
- ☐ Eliminar un auto de la tabla "Auto".
- ☐ Marcar una reserva como completada actualizando la fecha de fin.
- ☐ Eliminar todas las reservas realizadas por un cliente específico.

Actualizar el año de un auto en la tabla "Auto".

#### **EJERCICIOS SEGUNDO CICLO (31-80):**

- `SELECT * FROM ClientePLSQL;`
- `SELECT * FROM AutoPLSQL;`
- `SELECT * FROM AlquilerPLSQL;`
- `SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente;`
- `SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto;`
- `SELECT * FROM AlquilerPLSQL WHERE id_cliente = 1;`
- `SELECT * FROM AlquilerPLSQL WHERE id_auto = 1;`
- `SELECT * FROM AlquilerPLSQL WHERE id_sucursal = 1;`
- `SELECT * FROM AlquilerPLSQL WHERE fecha_inicio = '2023-09-27';`
- `SELECT COUNT(*) FROM AlquilerPLSQL;`
- `SELECT c.nombre FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente JOIN SucursalPLSQL s ON a.id_sucursal = s.id_sucursal WHERE s.nombre = 'Sucursal Central';`
- `SELECT a.marca, a.modelo FROM AutoPLSQL a`
- `JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto WHERE al.id_cliente = 1 AND al.fecha_inicio = '2023-09-27';`
- `SELECT * FROM AlquilerPLSQL WHERE fecha_fin - fecha_inicio > 7;`
- `SELECT c.nombre, COUNT(*) AS numero_alquileres FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id_cliente = a.id_cliente GROUP BY c.nombre ORDER BY numero_alquileres DESC LIMIT 1;`
- `SELECT a.marca, a.modelo, COUNT(*) AS numero_alquileres FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id_auto = al.id_auto GROUP BY a.marca, a.modelo ORDER BY numero_alquileres DESC LIMIT 1;`
- `SELECT s.nombre, COUNT(*) AS numero_alquileres FROM SucursalPLSQL s JOIN AlquilerPLSQL al ON s.id_sucursal = al.id_sucursal GROUP BY s.nombre ORDER BY numero_alquileres DESC LIMIT 1;`
- `SELECT EXTRACT(MONTH FROM fecha_inicio) AS mes, COUNT(*) AS numero_alquileres FROM AlquilerPLSQL GROUP BY EXTRACT(MONTH FROM fecha_inicio) ORDER BY numero_alquileres DESC LIMIT 1;`
- `SELECT EXTRACT(DAYOFWEEK FROM fecha_inicio) AS dia_semana, COUNT(*) AS numero_alquileres FROM AlquilerPLSQL GROUP BY`

- ```
EXTRACT(DAYOFWEEK FROM fecha_inicio) ORDER BY numero_alquileres
DESC LIMIT 1;
```
- SELECT \* FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;
  - SELECT \* FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1;
  - SELECT \* FROM ClientePLSQL WHERE nombre LIKE '%Juan%'; SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a WHERE precio < 10000;
  - SELECT \* FROM AlquilerPLSQL WHERE fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30';
  - SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id\_cliente = a.id\_cliente WHERE c.direccion LIKE '%Bogotá%';
  - SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id\_auto = al.id\_auto WHERE al.id\_reserva = 1;
  - SELECT \* FROM AlquilerPLSQL WHERE id\_cliente IN (1, 2, 3); SELECT \* FROM AlquilerPLSQL WHERE id\_auto IN (1, 2, 3);
  - SELECT \* FROM AlquilerPLSQL WHERE id\_sucursal IN (1, 2, 3);
  - SELECT \* FROM AlquilerPLSQL WHERE fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30' AND id\_cliente IN (1, 2, 3);
  - SELECT \* FROM AlquilerPLSQL WHERE fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30' AND id\_auto IN (1, 2, 3);
  - SELECT \* FROM AlquilerPLSQL WHERE fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30' AND id\_sucursal IN (1, 2, 3); SELECT c.nombre, COUNT(\*) AS numero\_alquileres FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id\_cliente = a.id\_cliente GROUP BY c.nombre ORDER BY numero\_alquileres DESC LIMIT 1;
  - SELECT a.marca, a.modelo, COUNT(\*) AS numero\_alquileres FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id\_auto = al.id\_auto GROUP BY a.marca, a.modelo ORDER BY numero\_alquileres DESC LIMIT 1;
  - SELECT s.nombre, COUNT(\*) AS numero\_alquileres FROM SucursalPLSQL s JOIN AlquilerPLSQL al ON s.id\_sucursal = al.id\_sucursal GROUP BY s.nombre ORDER BY numero\_alquileres DESC LIMIT 1;
  - SELECT EXTRACT(MONTH FROM fecha\_inicio) AS mes, COUNT(\*) AS numero\_alquileres FROM AlquilerPLSQL GROUP BY EXTRACT(MONTH FROM fecha\_inicio) ORDER BY numero\_alquileres DESC LIMIT 1;
  - SELECT EXTRACT(DAYOFWEEK FROM fecha\_inicio) AS dia\_semana, COUNT(\*) AS numero\_alquileres FROM AlquilerPLSQL GROUP BY EXTRACT(DAYOFWEEK FROM fecha\_inicio) ORDER BY numero\_alquileres DESC LIMIT 1;
  - SELECT \* FROM AlquilerPLSQL ORDER BY precio DESC LIMIT 1;
  - SELECT \* FROM AlquilerPLSQL ORDER BY precio ASC LIMIT 1;
  - SELECT \* FROM ClientePLSQL WHERE nombre LIKE '%Juan%' AND fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30'; SELECT a.marca, a.modelo, a.ano FROM AutoPLSQL a WHERE precio < 10000 AND fecha\_inicio BETWEEN '2023-09-01' AND '2023-09-30';

### **EJERCICIOS TERCER CICLO (81-90):**

- CREATE VIEW vista\_clientes\_alquilados\_sucursal AS SELECT c.nombre, a.marca, a.modelo FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id\_cliente = a.id\_cliente JOIN SucursalPLSQL s ON a.id\_sucursal = s.id\_sucursal WHERE s.nombre = 'Sucursal Central';
- CREATE VIEW vista\_autos\_alquilados\_cliente\_fecha AS SELECT a.marca, a.modelo FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id\_auto = al.id\_auto WHERE al.id\_cliente = 1 AND al.fecha\_inicio = '2023-09-27';
- CREATE VIEW vista\_alquileres\_mas\_7dias AS SELECT \* FROM AlquilerPLSQL WHERE fecha\_fin - fecha\_inicio > 7; CREATE VIEW vista\_clientes\_mas\_alquileres AS SELECT c.nombre, COUNT(\*) AS numero\_alquileres FROM ClientePLSQL c JOIN AlquilerPLSQL a ON c.id\_cliente = a.id\_cliente GROUP BY c.nombre ORDER BY numero\_alquileres DESC;
- CREATE VIEW vista\_autos\_mas\_alquileres AS SELECT a.marca, a.modelo, COUNT(\*) AS numero\_alquileres FROM AutoPLSQL a JOIN AlquilerPLSQL al ON a.id\_auto = al.id\_auto GROUP BY a.marca, a.modelo ORDER BY numero\_alquileres DESC; CREATE VIEW vista\_sucursales\_mas\_alquileres AS SELECT s.nombre, COUNT(\*) AS numero\_alquileres FROM SucursalPLSQL s JOIN AlquilerPLSQL al ON s.id\_sucursal = al.id\_sucursal GROUP BY s.nombre ORDER BY numero\_alquileres DESC;
- CREATE VIEW vista\_meses\_mas\_alquileres AS SELECT EXTRACT(MONTH FROM fecha\_inicio) AS mes, COUNT(\*) AS numero\_alquileres FROM AlquilerPLSQL GROUP BY EXTRACT(MONTH FROM fecha\_inicio) ORDER BY numero\_alquileres DESC;
- CREATE VIEW vista\_dias\_semana\_mas\_alquileres AS SELECT EXTRACT(DAYOFWEEK FROM fecha\_inicio) AS dia\_semana, COUNT(\*) AS numero\_alquileres FROM AlquilerPLSQL GROUP BY EXTRACT(DAYOFWEEK FROM fecha\_inicio) ORDER BY numero\_alquileres DESC;
- CREATE VIEW vista\_alquileres\_mas\_caros AS SELECT \* FROM AlquilerPLSQL ORDER BY precio DESC;
- CREATE VIEW vista\_alquileres\_mas\_baratos AS SELECT \* FROM AlquilerPLSQL ORDER BY precio ASC;

### **EJERCICIOS TERCER CICLO (91-100):**

```
CREATE TRIGGER trg_insert_auto
BEFORE INSERT ON AutoPLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de autos disponibles
    UPDATE AutoPLSQL
    SET numero_disponibles = numero_disponibles + 1
    WHERE id_auto = NEW.id_auto;
END;

CREATE TRIGGER trg_delete_auto
```

```

BEFORE DELETE ON AutoPLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de autos disponibles
    UPDATE AutoPLSQL
        SET numero_disponibles = numero_disponibles - 1
        WHERE id_auto = OLD.id_auto;
END;

CREATE TRIGGER trg_update_auto
BEFORE UPDATE ON AutoPLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de autos disponibles
    IF NEW.numero_disponibles != OLD.numero_disponibles THEN
        UPDATE AutoPLSQL
            SET numero_disponibles = NEW.numero_disponibles
            WHERE id_auto = NEW.id_auto;
    END IF;
END;

CREATE TRIGGER trg_insert_cliente
BEFORE INSERT ON ClientePLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de clientes
    UPDATE ClientePLSQL
        SET numero_clientes = numero_clientes + 1;
END;

CREATE TRIGGER trg_delete_cliente
BEFORE DELETE ON ClientePLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de clientes
    UPDATE ClientePLSQL
        SET numero_clientes = numero_clientes - 1;
END;

CREATE TRIGGER trg_update_cliente
BEFORE UPDATE ON ClientePLSQL
FOR EACH ROW
BEGIN
    -- Actualizar el número de clientes
    IF NEW.numero_alquileres != OLD.numero_alquileres THEN
        UPDATE ClientePLSQL
            SET numero_alquileres = NEW.numero_alquileres
            WHERE id_cliente = NEW.id_cliente;
    END IF;
END;

CREATE PROCEDURE proc_calcular_precio_alquiler
(
    IN id_alquiler INT,
    IN id_auto INT,
    IN fecha_inicio DATE,

```

```

    IN fecha_fin DATE
)
AS
BEGIN
    -- Calcular el precio del alquiler
    DECLARE
        precio_base NUMERIC(10, 2);
        dias_alquiler INT;
    BEGIN
        precio_base := (SELECT precio FROM AutoPLSQL WHERE id_auto =
id_auto);
        dias_alquiler := (fecha_fin - fecha_inicio) + 1;
        SET NEW.precio = precio_base * dias_alquiler;
    END;
END;

CREATE PROCEDURE proc_listar_alquileres_cliente
(
    IN id_cliente INT
)
AS
BEGIN
    -- Listar los alquileres del cliente
    SELECT *
    FROM AlquilerPLSQL
    WHERE id_cliente = id_cliente;
END;

CREATE PROCEDURE proc_listar_autos_sucursal
(
    IN id_sucursal INT
)
AS
BEGIN
    -- Listar los autos de la sucursal
    SELECT *
    FROM AutoPLSQL
    WHERE id_sucursal = id_sucursal;
END;

CREATE PROCEDURE proc_agregar_auto
(
    IN marca VARCHAR(255),
    IN modelo VARCHAR(255),
    IN ano INT,
    IN numero_disponibles INT
)
AS
BEGIN
    -- Insertar un nuevo auto
    INSERT INTO AutoPLSQL (marca, modelo, ano, numero_disponibles)
    VALUES (marca, modelo, ano, numero_disponibles);
END;

CREATE PROCEDURE proc_eliminar_auto
(
    IN id_auto INT

```



```
)  
AS  
BEGIN  
    -- Eliminar un auto  
    DELETE FROM AutoPLSQL  
    WHERE id_auto = id_auto;  
END;
```