Bases de Datos Estructuradas

2024



Oracle Database es un sistema de gestión de bases de datos relacionales (RDBMS, por sus siglas en inglés) de Oracle, el fabricante estadounidense de software y hardware. Como software de bases de datos, Oracle Database optimiza la gestión y seguridad de los conjuntos de datos creando esquemas estructurados a los que solo pueden acceder administradores autorizados.



Herramientas importantes de Oracle Database
Para desarrollar y ampliar las Oracle Databases, Oracle proporciona varias herramientas de desarrollo y gestión.

- SQL*Plus
- Oracle SQL Developer
- Oracle Data Modeler
- Oracle Enterprise Manager Database Control
- Oracle Enterprise Manager Grid Control
- Oracle JDeveloper

Tipos de datos que admite **Oracle Database**



XML, se trata de la forma estándar de identificar y describir datos en la web.



LOB, este tipo de datos permiten almacenar y manipular grandes volúmenes de datos.



Oracle Text, permite búsqueda de texto.



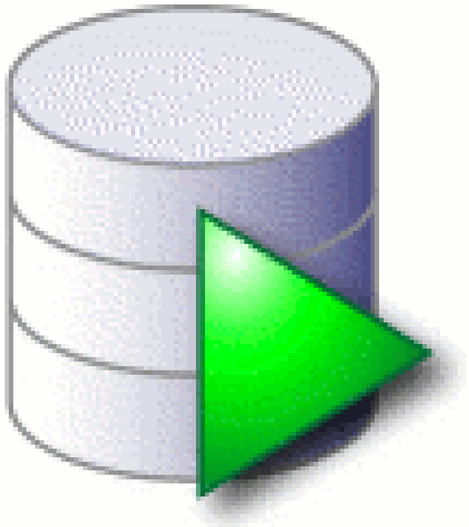
Oracle InterMedia, con servicios para desarrollar e implementar aplicaciones



Oracle espacial, con funciones para **almacenar** y administrar contenido de ubicación.

Oracle SQL Developer es el IDE de Oracle Database

Oracle SQL Developer

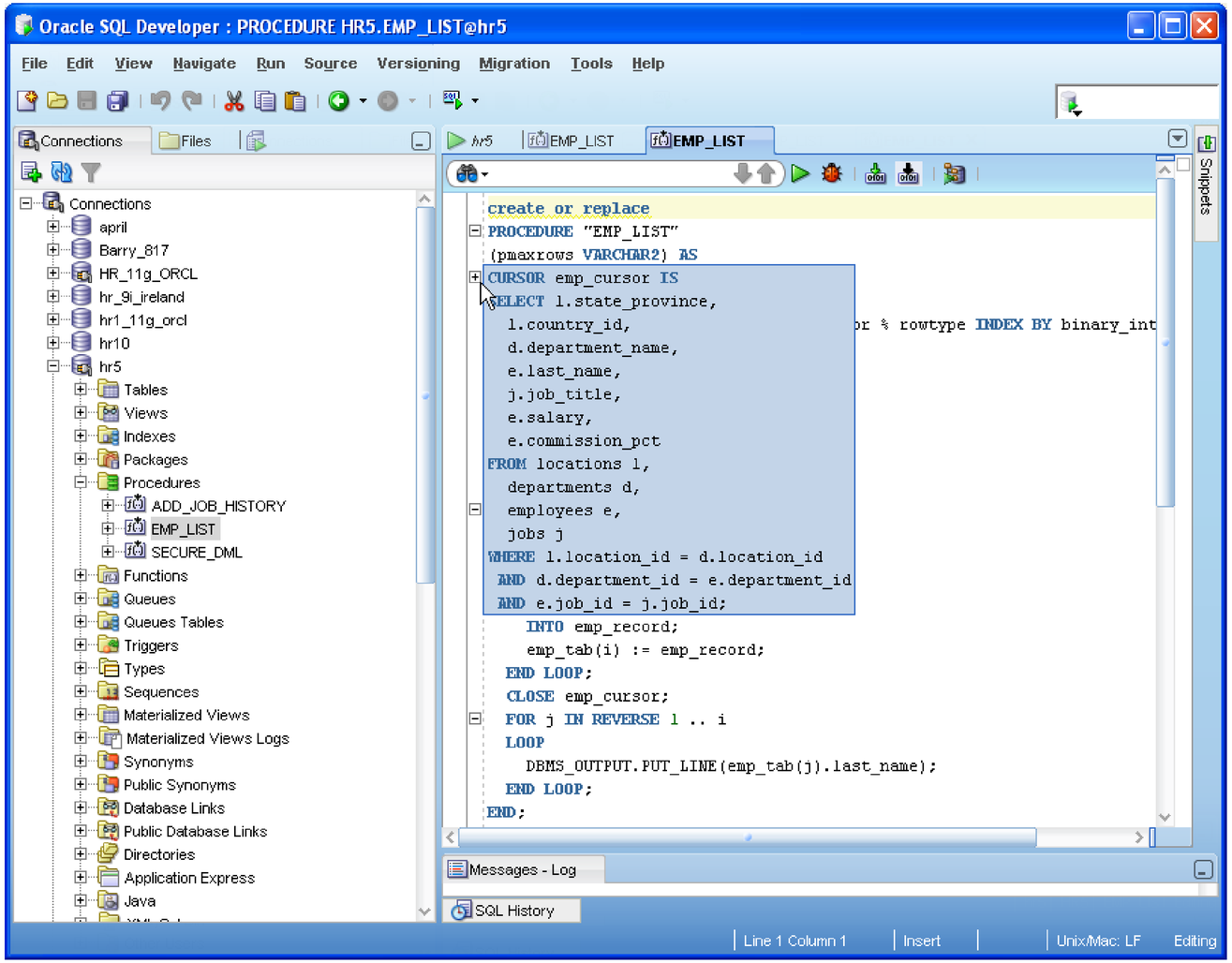


ORACLE

Oracle SQL Developer es una interfaz gráfica de usuario gratuita que permite a los usuarios y administradores de bases de datos realizar sus tareas con menos clics y pulsaciones de teclas.

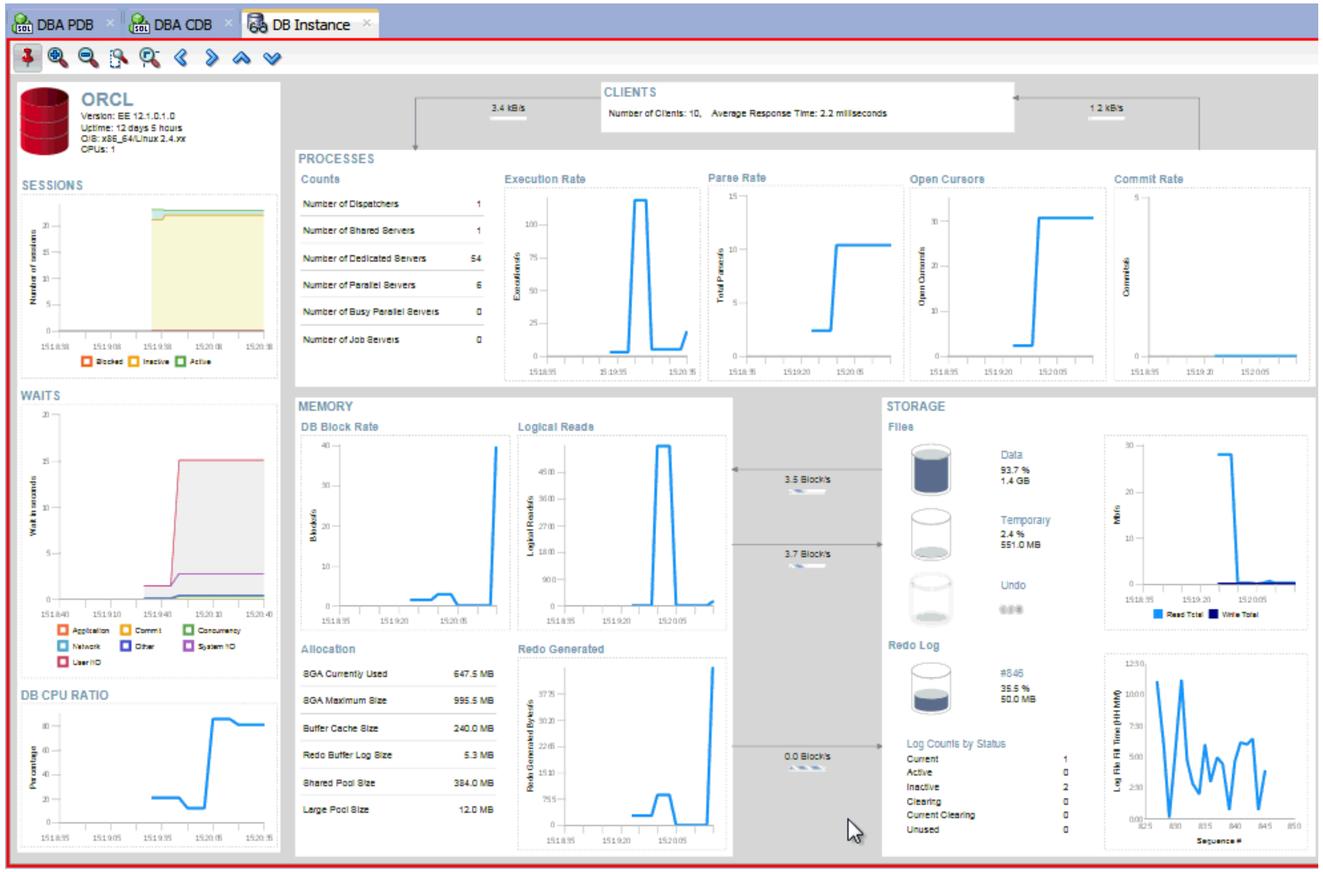
SQL Developer es una herramienta de productividad cuyo objetivo principal es ayudar al usuario final a ahorrar tiempo y maximizar el retorno de la inversión en el paquete de tecnología de Oracle Database.

SQL Developer admite Oracle Database 10g, 11g y 12c, y se ejecuta en cualquier sistema operativo compatible con Java.



Para el desarrollador

SQL Developer proporciona potentes editores para trabajar con SQL, PL/SQL, procedimientos almacenados de Java y XML. Con SQL Developer, puede ejecutar consultas, generar planes de ejecución, exportar datos al formato deseado (XML, Excel, HTML, PDF, etc.), ejecutar, depurar, probar y documentar sus programas de base de datos, y mucho más.

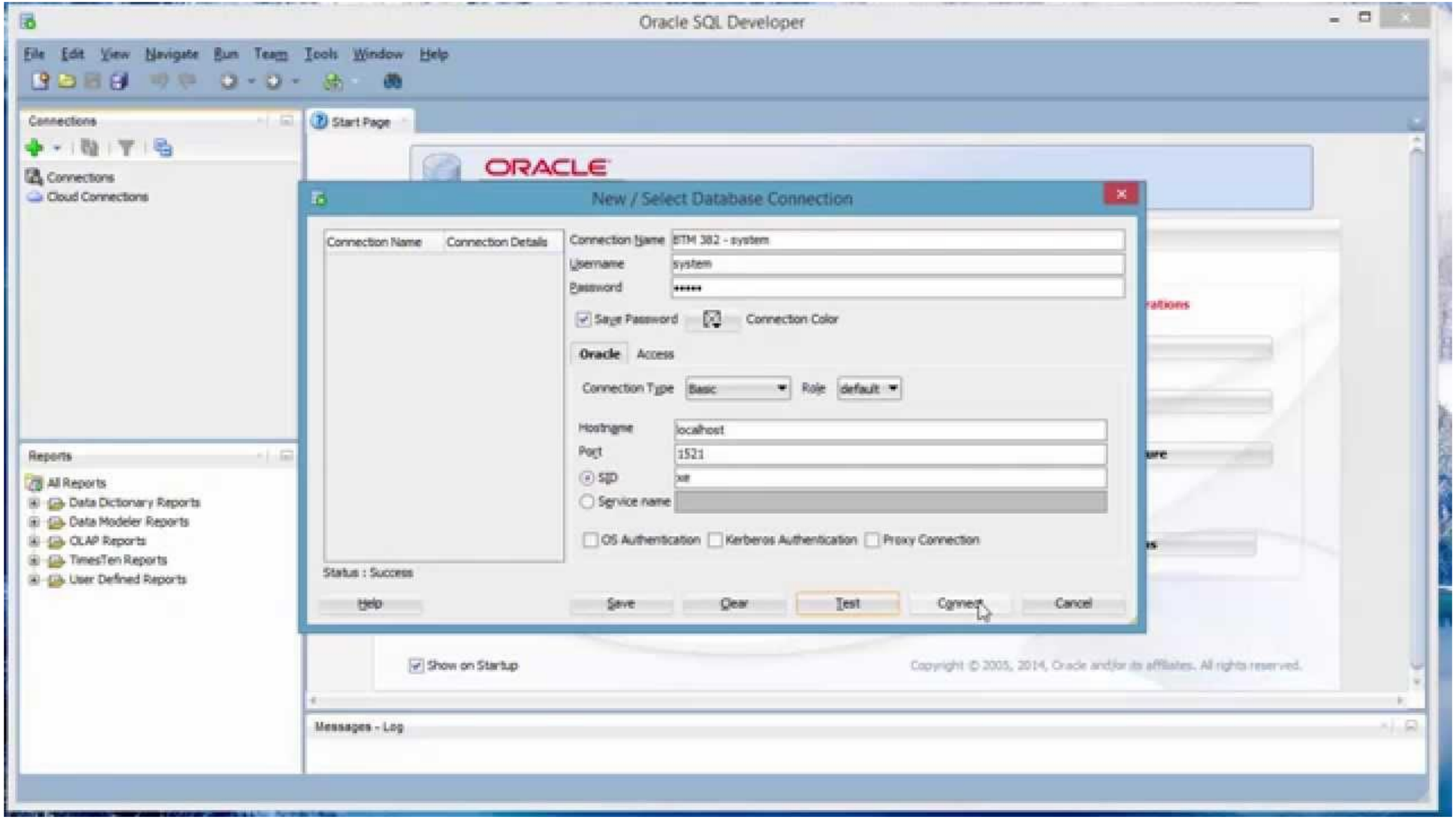


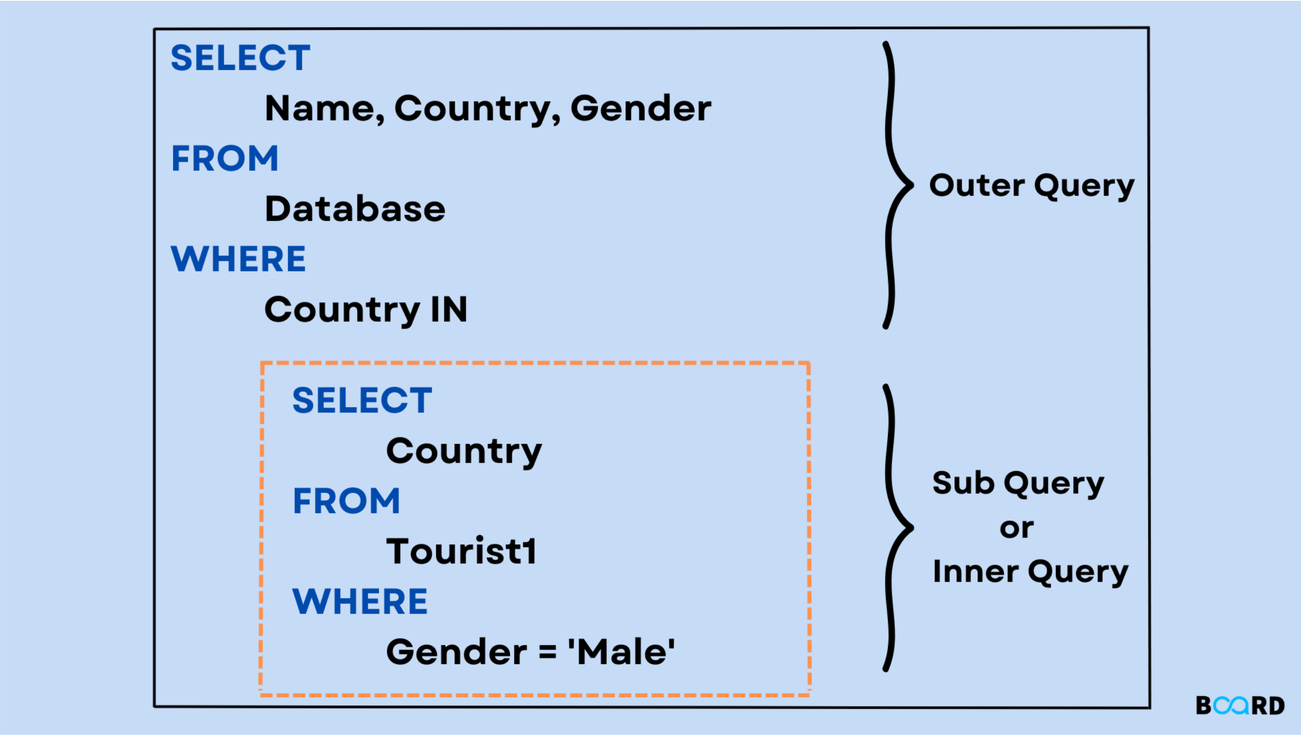
Para el DBA

SQL Developer no es solo para desarrolladores. Desde la versión 3.0, el panel DBA (disponible en el menú Ver) ha proporcionado a los administradores de bases de datos un conjunto de interfaces para sus tareas esenciales. Seguiremos incorporando características para el DBA y mejorándolas en SQL. Actualmente, el panel DBA de SQL Developer ofrece:

- Bomba de datos
- Gestor de recuperación (RMAN)
- Auditoría Oracle
- Gestión de usuarios y roles
- Gestión del almacenamiento, incluida la capacidad de agregar espacio a los espacios de tabla
- Gestor de recursos
- Características del paquete de diagnóstico:
    - Instantáneas
    - Valores de referencia
    - ADDM
    - ASH
    - AWR

Abrir Oracle SQL Developer – (esta en el desktop)






```
SELECT ProductID,
       Name,
       ListPrice
FROM   production.Product
WHERE  ListPrice > (SELECT AVG(ListPrice)
                    FROM   Production.Product)
```

subquery

SUBQUERY

Una subconsulta, también conocida como **SUBQUERY** o **SUBSELECT**, es una consulta incrustada dentro de otra consulta, de forma anidada, pasando los resultados de la consulta más interna a la consulta más externa a través de una cláusula **WHERE** o de la cláusula **HAVING**.

Ejercicio 1: Consulta con Subconsulta

Enunciado: Escribe una consulta para encontrar los nombres de los empleados que ganan más que el salario promedio de su propio departamento.

Ejercicio 1: Consulta con Subconsulta

Enunciado: Escribe una consulta para encontrar los nombres de los empleados que ganan más que el salario promedio de su propio departamento.

```
SELECT FirstName, LastName, Salary
FROM Employees e
WHERE Salary > (SELECT AVG(Salary)
        FROM Employees
        WHERE DepartmentID = e.DepartmentID);
```

Ejercicio 2: Consulta con JOIN

Enunciado: Escribe una consulta que muestre los nombres completos de los empleados junto con el nombre de su departamento.

Ejercicio 2: Consulta con JOIN

Enunciado: Escribe una consulta que muestre los nombres completos de los empleados junto con el nombre de su departamento.

```
SELECT e.FirstName || ' ' || e.LastName AS FullName, d.DepartmentName
FROM Employees e
JOIN Departments d ON e.DepartmentID = d.DepartmentID;
```

Ejercicio 3: Creación de una Vista
Enunciado: Crea una vista llamada VistaEmpleadosSinDepartamento que muestre todos los empleados que no tienen un departamento asignado.

Ejercicio 3: Creación de una Vista
Enunciado: Crea una vista llamada VistaEmpleadosSinDepartamento que muestre todos los empleados que no tienen un departamento asignado.

```
CREATE OR REPLACE VIEW VistaEmpleadosSinDepartamento AS
SELECT FirstName, LastName, Email
FROM Employees
WHERE DepartmentID IS NULL;
```

Ejercicio 4: Uso de Funciones de Agregación

Enunciado: Escribe una consulta que muestre el salario total, el salario promedio, el salario más alto y el más bajo por cada departamento.

Ejercicio 4: Uso de Funciones de Agregación

Enunciado: Escribe una consulta que muestre el salario total, el salario promedio, el salario más alto y el más bajo por cada departamento.

```
SELECT DepartmentID,
    SUM(Salary) AS TotalSalary,
    AVG(Salary) AS AvgSalary,
    MAX(Salary) AS MaxSalary,
    MIN(Salary) AS MinSalary
FROM Employees
GROUP BY DepartmentID;
```

Ejercicio 5: Creación de un Procedimiento Almacenado
Enunciado: Crea un procedimiento almacenado llamado sp_ActualizarEmail que permita actualizar el correo electrónico de un empleado dado su ID.

Ejercicio 5: Creación de un Procedimiento Almacenado
Enunciado: Crea un procedimiento almacenado llamado sp_ActualizarEmail que permita actualizar el correo electrónico de un empleado dado su ID.

```
CREATE OR REPLACE PROCEDURE sp_ActualizarEmail (
    p_EmployeeID NUMBER,
    p_NewEmail NVARCHAR2
) AS
BEGIN
    UPDATE Employees
    SET Email = p_NewEmail
    WHERE EmployeeID = p_EmployeeID;
END;
/
```

Ejercicio 6: Creación de un Trigger de Auditoría
Enunciado: Crea un trigger llamado trg_AuditInsert que registre en una tabla de auditoría cada vez que se inserte un nuevo empleado en la tabla Employees.

Ejercicio 6: Creación de un Trigger de Auditoría
Enunciado: Crea un trigger llamado trg_AuditInsert que registre en una tabla de auditoría cada vez que se inserte un nuevo empleado en la tabla Employees.

```
CREATE OR REPLACE TRIGGER trg_AuditInsert
AFTER INSERT ON Employees
FOR EACH ROW
BEGIN
    INSERT INTO AuditInsert (EmployeeID, FirstName, LastName, InsertionDate)
    VALUES (:NEW.EmployeeID, :NEW.FirstName, :NEW.LastName, SYSDATE);
END;
/
```

Ejercicio 7: Consulta con Funciones de Fecha
Enunciado: Escribe una consulta que muestre todos los empleados que han sido contratados en los últimos dos años.

Ejercicio 7: Consulta con Funciones de Fecha
Enunciado: Escribe una consulta que muestre todos los empleados que han sido contratados en los últimos dos años.

```
SELECT FirstName, LastName, HireDate
FROM Employees
WHERE HireDate >= ADD_MONTHS(SYSDATE, -24);
```

Ejercicio 9: Eliminar Empleados con Condiciones
Enunciado: Escribe un procedimiento almacenado sp_EliminarEmpleadosAntiguos que elimine empleados que hayan sido contratados hace más de 10 años.

Ejercicio 9: Eliminar Empleados con Condiciones
Enunciado: Escribe un procedimiento almacenado sp_EliminarEmpleadosAntiguos que elimine empleados que hayan sido contratados hace más de 10 años.

```
CREATE OR REPLACE PROCEDURE sp_EliminarEmpleadosAntiguos AS
BEGIN
    DELETE FROM Employees
    WHERE HireDate < ADD_MONTHS(SYSDATE, -120);
END;
/
```

Ejercicio 10: Consulta para Encontrar Empleados sin Gerente Asignado
Enunciado: Escribe una consulta que muestre todos los empleados que no son gerentes de ningún departamento.

Ejercicio 10: Consulta para Encontrar Empleados sin Gerente Asignado
Enunciado: Escribe una consulta que muestre todos los empleados que no son gerentes de ningún departamento.

```
SELECT e.FirstName, e.LastName
FROM Employees e
LEFT JOIN Departments d ON e.EmployeeID = d.ManagerID
WHERE d.ManagerID IS NULL;
```

Ejercicio 11: Uso de Subconsultas en el WHERE
Enunciado: Escribe una consulta para listar todos los departamentos que tengan más de 3 empleados.

Ejercicio 11: Uso de Subconsultas en el WHERE
Enunciado: Escribe una consulta para listar todos los departamentos que tengan más de 3 empleados.

```
SELECT DepartmentName
FROM Departments
WHERE DepartmentID IN (SELECT DepartmentID
                       FROM Employees
                       GROUP BY DepartmentID
                       HAVING COUNT(*) > 3);
```

MUCHAS GRACIAS!

inacap.cl