

APLICACIÓN JAVA CON ORACLE Y PL/SQL

Objetivo

Desarrollar el diseño e implementación de una base de datos Oracle para una aplicación Java, donde la lógica de negocio se ejecute mediante PL/SQL (procedimientos, funciones, paquetes, etc.).

Los alumnos deberán simular el proceso de análisis de requisitos con el cliente, diseñar la BD y crear todos los objetos necesarios para su funcionamiento.

Fases del Proyecto

1. Análisis de Requisitos (Cliente Ficticio)

Se deberá simular reuniones con el cliente para especificar los requisitos de la app y definir:

- Entidades principales (ej: usuarios, productos, citas, etc.).
- Relaciones entre entidades.
- Operaciones CRUD básicas (altas, bajas, modificaciones, consultas).
- Consultas complejas (ej: informes con filtros, búsquedas avanzadas).

2. Diseño Conceptual

- Crear el diagrama Entidad-Relación (ER) que refleje: entidades, atributos y relaciones, cardinalidades....

3. Diseño Lógico

- Transformar el modelo ER en un esquema relacional:
- Definir tablas, claves primarias y foráneas.

4. Implementación en Oracle

A. Usuarios y Permisos.

Crear 2 usuarios:

- DBA_USER: Con permisos de administrador (CREATE, ALTER, DROP, etc.).
- APP_USER: Solo con permisos DML (SELECT, INSERT, UPDATE, DELETE) y ejecución de procedimientos.

B. Estructura de la BD.

Script SQL para crear:

- Tablas con restricciones (PK, FK, CHECK, NOT NULL, etc.).
- Secuencias para IDs autoincrementales.
- Índices para campos frecuentemente consultados.

C. Programación PL/SQL

- Tablas
- Secuencias
- Índices
- Paquetes (al menos 1 por entidad importante):
- Procedimientos/funciones para CRUD.
- Procedimientos que devuelvan cursores para consultas elaboradas.
- Procedimientos con implementación, recorrido de estructuras y operaciones DML.
- Funciones para validaciones (ej: comprobar stock, calcular totales).
- Triggers (ej: auditoría de cambios, validación de datos).
 - **Validación de datos:** usar un disparador para controlar/avisar sobre algún evento. Añadir implementación adicional al triggers (no solo una excepción).
 - **Auditoría:** añadir al menos una tabla de auditoría asociada a otra tabla original, y un trigger para registrar los cambios que se producen. Se añadirán al menos 3 nuevos campos:
 - Fecha: **fecha y hora** en la que se produce la manipulación de los datos
 - Acción: tipo de sentencia DML (Insert, Update, Delete).
 - Usuario: usuario que lanza la sentencia.

Ejemplo:

En la tabla empleados inicialmente se produjo la carga inicial (20250410). Más tarde se actualizó el apellido del empleado 100 (20250420).

- 100, king, 20250410 23:20:20, insert, user_app, carga_inicial
- 100, bruce, 20250420 14:15:35, update, user_app, update_app

D. Ejemplo de Caso de Uso.

Incluir un caso donde se demuestre una transacción compleja (ej: realizar una venta = actualizar stock + registrar pedido + generar factura).

Entrega

- Documento PDF con:

- Descripción del cliente y requisitos.
 - Diagrama ER y esquema relacional.
 - Ejemplos de uso.
- Scripts SQL completos (creación de objetos, PL/SQL, permisos).

Extra: Opcionalmente, pueden adjuntar un pequeño programa Java que muestre la conexión y uso de los procedimientos.

Evaluación

Se valorará:

- ✓ Claridad del diseño conceptual y lógico.
- ✓ Correcta implementación de objetos PL/SQL.
- ✓ Organización del código y comentarios.
- ✓ Manejo de permisos y usuarios.