

PROYECTO FINAL
PARTE 2.

Antes de comenzar con el desarrollo del presente documento es **importante** leerlo hasta el final, y evitar así actualizaciones o correcciones en pasos previos.

1.1. CREACIÓN DE LA BASE DE DATOS.

- Todas las configuraciones y código deberá ser guardado en scripts SQL. Emplear la notación `s-nn-<descripcion-corta>.sql` nn se refiere al número de script iniciando en 01, <descripcion-corta> Representa a una cadena separada por guiones medios que indica una descripción corta en cuanto al propósito del script. Ejemplo: `s-01-creacion-bd.sql`
- Todos los scripts deberán contener su encabezado que incluye integrantes, fecha de creación, descripción del script.
- El código deberá estar correctamente formateado.
- Una vez que el proyecto haya sido concluido, generar una tabla con el resumen de todos los scripts creados

Num. Script	Nombre del script	Descripción

1.1.1. Planeación para crear la nueva base de datos.

- Crear una nueva base de datos empleando como nombre <iniciales>proy. <Iniciales> corresponden con los 2 primeros caracteres del apellido paterno de cada integrante. Si el proyecto se realiza de forma individual, emplear las primeras 2 letras del apellido paterno y las 2 primeras letras del apellido materno.
- Si el espacio en disco resulta ser un impedimento para crear esta nueva base de datos, eliminar la base de datos 1 creada al inicio del semestre.
- Todos los archivos de la nueva base de datos incluidos los backups, FRA, Redo Logs, data files, archive redo logs, etc., deberán estar ubicados dentro de la ruta `$ORACLE_BASE/oradata/$ORACLE_SID`.
- Los archivos multiplexados también deberán estar dentro de esta ruta. Crear carpetas con el prefijo `disk_n` para simular un punto de montaje que pertenece a discos duros independientes.
- Proponer una configuración inicial y llenar la siguiente tabla:

Configuración	Descripción y/o configuración
Número y ubicación de los archivos de control.	
Propuesta de grupos de REDO	
Propuesta de juego de caracteres	
Tamaño del bloque de datos.	
Lista de parámetros que serán configurados al crear la base de datos.	Especificar nombre y valor.
Archivo de passwords	Indicar los usuarios que contendrá este archivo de forma inicial. Como requisito indispensable, deberá existir un usuario diferente a sys que será el encargado de realizar la administración de backups.

1.1.2. Módulos del sistema.

Con base a los requerimientos y a las características del caso de estudio, proponer una división por módulos funcionales. La idea es que estos módulos puedan ser administrados de forma independiente, en especial sus estructuras físicas de almacenamiento. Los datos de cada módulo deberán ser almacenados en tablespaces separados para poder implementar esta independencia de administración. Cada módulo deberá contar con un usuario dueño de todos los objetos. Se recomienda crear solo 2 módulos. Llenar la siguiente tabla.

Nombre del módulo	Descripción	Usuario

1.1.3. Diseño lógico de la base de datos.

Con base al modelo relacional realizado anteriormente realizar una distribución de los objetos considerando la propuesta de módulos realizada anteriormente. Llenar la siguiente tabla.

Nombre de la tabla	Nombre del módulo.

1.1.4. Esquema de indexado.

Con base a las reglas de negocio del caso de estudio asignado generar una lista de índices que serían considerados como necesarios para implementar reglas de negocio que requieran valores únicos, o para mejorar desempeño. Por ejemplo, indexar FKs, etc. Llenar la siguiente tabla:

Nombre de la tabla	Nombre del índice	tipo	Propósito.

1.2. DISEÑO FÍSICO BÁSICO DE LA BD.

Con base a los requerimientos del caso de estudio, proponer un diseño físico en el que se describan los tablespaces que serán creados para soportar la operación de la base de datos para cada uno de los módulos. Especificar tanto los tablespaces requeridos y recomendados así como los tablespaces creados para almacenar los datos del módulo. Considerar que los datos CLOB/BLOB se deberán almacenar en tablespaces dedicados. Considerar que los índices también deberán almacenarse en tablespaces diferentes (uno por módulo) Llenar la siguientes tablas para cada uno de los módulos.

1.2.1.1. Definición de tablespaces comunes a los módulos

- En esta tabla se documentan los tablespaces comunes a los módulos.

Nombre del tablespace	Configuración
	Especificar: Big File o múltiple data files, tamaño, tipo de administración de segmentos y extensiones, ubicación de sus data files.

1.2.1.2. Definición de tablespaces por módulo

Nombre del módulo:

Nombre del tablespace	Objetivo / Beneficio	Tipo

1.2.1.3. Configuración de tablespaces por módulo

Nombre del módulo:

Nombre del tablespace	Configuración
	Especificar: Big File o múltiple data files, tamaño, tipo de administración de segmentos y extensiones, ubicación de sus data files.

1.2.1.4. Asignación de tablespace para tablas de cada módulo

Nombre del módulo:

Nombre de la tabla	Nombre del tablespace

1.2.1.1. Asignación de tablespace para índices de cada módulo.

Nombre del módulo:

Nombre del índice	Tipo de índice	Nombre de la tabla	Nombre de la columna	Nombre del tablespace

1.2.1.2. Asignación de tablespaces para columnas CLOB/BLOB de cada módulo.

Nombre del módulo:

Nombre de la columna CLOB/BLOB	Nombre de índice asociado a la columna CLOB/BLOB	Nombre de la tabla	Nombre del tablespace para la columna CLOB/BLOB	Nombre del tablespace para el índice de la columna CLOB/BLOB

1.3. CREACIÓN DE USUARIOS.

Con base a los usuarios propuestos anteriores, generar las sentencias SQL necesarias para realizar su creación. **Importante:** Los usuarios solo deberán contar con los privilegios mínimos indispensables. No emplear sentencias SQL que otorguen privilegios generales, no hacer uso de `any`, `admin option`, etc. Cada uno de estos usuarios deberá tener cuota ilimitada en los tablespaces donde se almacenarán sus objetos. Su tablespace por default será el tablespace donde se almacenarán las tablas de su respectivo módulo.

1.4. GENERACIÓN DEL CÓDIGO DDL.

A partir del modelo relacional realizado anteriormente realizar las siguientes acciones en ER -Studio

- Crear un nuevo modelo lógico por cada uno de los módulos propuestos anteriormente.
- Incluir en cada modelo lógico las tablas que le corresponden.
- crear un modelo físico a partir del modelo lógico para Oracle a partir de cada uno de los módulos (modelos lógicos creados en el punto anterior).
- Los constraints deben ser creados como parte de la instrucción `create table`. Evitar el uso de `alter table` para crear constraints.
- Emplear las siguientes convenciones para realizar el nombrado de los constraints. Si el nombre es demasiado largo, pueden aplicar algunas abreviaturas que sean lo más claras posible.

Tipo Constraint	Convención de nombrado
<code>unique</code>	<code><nombre_tabla>_<nombre_columna>_uk</code>
<code>primary key</code>	<code><nombre_tabla>_pk</code>
<code>references, foreign key</code>	<code><nombre_tabla_hija>_<nombre_columna>_fk</code>
<code>check</code>	<code><nombre_tabla>_<nombre_columna>_chk</code>

- Revisar el documento `BD/practicas/practica04/practica04-previo.pdf` para configurar la herramienta e implementar fácilmente estos requerimientos.
- Generar el código SQL empleando ER-Studio.
- Editar el script generado para realizar las asignaciones de tablespaces tanto de tablas como para índices, PKs, índices tipo LOB.

1.5. MODOS DE CONEXIÓN.

- Realizar las configuraciones necesarias de tal forma que un usuario pueda conectarse a la instancia ya sea en modo compartido o en modo dedicado. La configuración por default deberá ser modo dedicado.

1.6. HABILITAR LA FRA.

- Habilitar la FRA, realizar un cálculo estimado de su tamaño con base a la cantidad de datos que se pretenden almacenar (ver siguientes secciones).
- Una de las copias del archivo de control deberá almacenarse en la FRA
- Uno de los grupos de Online Redo Logs deberá ubicarse en la FRA.
- Habilitar el uso de Flashback logs, proponer un periodo de retención de estos logs para garantizar las funcionalidades asociadas con el uso de flashback.

1.7. MODO ARCHIVELOG

- La base de datos deberá estar en modo archivelog.
- Proponer 2 ubicaciones. Una de ellas deberá almacenarse en la FRA.

1.8. PLANEACIÓN DEL ESQUEMA DE RESPALDOS.

Describir la estrategia que se empleará para realizar los respaldos de la base de datos. Esta estrategia deberá incluir:

- Tipos de backups a realizar
- Frecuencia de repetición
- Ubicaciones de respaldo (FRA)
- Política de retención de backups.
- Tamaño total en espacio en disco disponible para realizar backups.

Una vez que esta estrategia ha sido decidida y documentada, realizar las configuraciones necesarias para poder implementar esta estrategia.

1.9. CARGA DE DATOS.

- Se recomienda emplear un generador de datos para realizar esta actividad. Tratar de poblar la base de datos con una buena cantidad de datos para similar un ambiente real. Para las tablas que potencialmente pudieran tener grandes cantidades de registros, se recomienda generar por lo menos 2000 registros.
- No es necesario generar la misma cantidad de datos BLOB, pueden ser solo unos cuantos.
- Debido a que se trata de la carga inicial, no se requiere generar datos REDO. Deshabilitar la generación de REDO mientras se realiza la carga inicial.
- Posterior a la ejecución de la carga inicial habilitar nuevamente la generación de REDO.

1.10. RESPALDO INICIAL.

Una vez que los datos hayan sido cargados, ejecutar una primera iteración de la estrategia de respaldos programada. Revisar el espacio requerido

1.11. SIMULACIÓN DE LA CARGA DIARIA.

- Generar Scripts que simulen la generación de datos de REDO los cuales representarán la carga diaria de una base productiva. Se recomienda tomar como base los scripts proporcionados en temas anteriores. Como mínimo se deberán generar aproximadamente 30 MB de datos REDO. Este valor también deberá ser considerado para decidir el tamaño de los grupos de REDO al momento de crear la base de datos.
- Realizar algunos ciclos de generación de datos REDO, y posteriormente hacer respaldos para comprobar su correcto funcionamiento.
- Ejecutar los comandos necesarios para liberar espacio en disco considerando archivos obsoletos.
- Llenar la siguiente tabla:

Programación de respaldos.

Fecha y hora	Datos REDO producidos (MB)	Fecha de Respaldo	Tipo de backup	Espacio requerido por el backup

1.12. SIMULAR UN PROCESO DE INSTANCE RECOVERY.

- Ejecutar una simulación de datos de REDO de varios días.
- Hacer un shutdown abort y posteriormente iniciar para provocar un instance recovery.
- Revisar el tiempo requerido que se ocupó para realizar la recuperación de los datos.
- Tratar de ajustar el valor del parámetro `fast_start_mttr_target` de tal forma que el tiempo requerido para recuperar disminuya.

1.13. SIMULAR UN PROCESO DE COMPLETE MEDIA RECOVERY.

- Dañar o eliminar uno de los data files de los módulos anteriores.
- Realizar un proceso de complete media recovery tanto manual como a través del uso del DRA.

1.14. ENTREGA DEL PROYECTO.

- No será necesario generar reporte. Únicamente se deberá presentar los modelos relacionales y las tablas con las configuraciones solicitadas en este documento.
- El día de la entrega los integrantes del equipo deberán explicar el contenido del proyecto y se les solicitará hacer algunas consultas en el diccionario de datos para validar los resultados.