

Carátula

Universidad Nacional Autónoma de México (UNAM)

- Facultad de Ingeniería
- División de Ingeniería Eléctrica
- Ingeniería en Computación

Profesor: Jorge Rodriguez Campos

Materia: Bases de Datos Avanzadas

Alumnos: Navarrete Zamora Aldo Yael, Nuñez Hernandez Diego Ignacio

Fecha: <fecha>

Proyecto: <nombre-del-proyecto>

Semestre: 2025-1

Tabla de contenido

1. Creación de la base de datos

- Configuraciones iniciales para crear la nueva base de datos
- Módulos del sistema
- Esquemas por módulo
- Esquema de indexado
- Diseño de tablespaces
 - Tablespaces comunes
 - Tablespaces por módulo
 - Asignación de tablespace por objeto y módulo
- Generación del código DDL para el modelo relacional
- Habilitar la FRA
- Modo archivelog
- Planeación del esquema de respaldos
- Simulación de la carga diaria

1.3 Creación de la base de datos

- Comenzar con la creación de la base de datos una vez que se haya revisado y verificado el diseño lógico realizado en la sección anterior.
- Todas las configuraciones y código deberán ser guardados en scripts SQL o archivos shell. Emplear la notación `s-nn-<descripcion-corta>.sql` o `s-nn-<descripcion-corta>.sh`. `nn` se refiere al número de script iniciando en 01, `<descripcion-corta>` representa 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 encabezado eligo es el siguiente para todos los scripts:

```
--@Autores: Navarrete Zamora Aldo Yael y Nuñez Hernandez Diego Ignacio
--@Fecha: <fecha-de-creacion>
--@Descripción: <descripcion-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.

Nombre del script	Descripción
e-00-crea-contenedor.sh	Creación del contenedor base Docker.
e-01-crea-loop-devices-host-root.sh	Creación de dispositivos de loop en el host.
e-02-crea-pwdfile-oracle.sh	Creación del archivo de contraseñas para Oracle.
e-03-crea-pfile-oracle.sh	Creación del archivo PFILE con configuración básica.
e-04-crea-spfile-ordinario.sql	Creación del archivo SPFILE a partir del PFILE.
e-05-crea-directorios-root.sh	Configuración de directorios para data files y redo logs.
e-06-crea-bd-ordinario.sql	Creación de la base de datos Oracle.
e-07-crea-diccionario-datos-oracle.sh	Creación del diccionario de datos Oracle.
s-08-crea-pdb-ordinario.sql	Creación de la base de datos pluggable (PDB).

1.3.2. Configuraciones iniciales para crear la nueva base de datos

- Crear una nueva base de datos empleando como nombre `<iniciales>proy<iniciales>` (para nuestro caso, sería `naproynu`).
- 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. Se recomienda contar con 10 GB de espacio para evitar problemas durante el desarrollo del proyecto.

Proponer una configuración inicial y llenar la siguiente tabla:

Configuración	Descripción y/o configuración
Número y ubicación de los archivos de control	Los archivos de control no deberían ubicarse en los mismos discos donde se encuentran los Redo Logs y data files. Los guardaremos en la FRA.
Propuesta de grupos de REDO	Un miembro de cada grupo deberá ubicarse en la FRA. No olvidar: Los data files no deberían ubicarse en los mismos discos donde se encuentran los Redo Logs y archivos de control.
Propuesta de juego de caracteres	El juego de caracteres deberá ser AL32UTF8 . Debido a que se trata de un proyecto de base de datos para un sistema de pedidos de comida, es importante que se soporten caracteres especiales.
Tamaño del bloque de datos	Se recomienda emplear un tamaño de bloque de 8 KB.
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.

1.3.3. 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
-------------------	-------------	---------

1.3.4. Esquemas por módulo

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

Nombre de la tabla	Nombre del módulo
--------------------	-------------------

1.3.5. 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, índices basados en funciones, etc. Llenar la siguiente tabla

Módulo	Nombre de la tabla	Nombre del índice	Tipo	Propósito
--------	--------------------	-------------------	------	-----------

1.3.6. Diseño de tablespaces

Con base a los requerimientos del caso de estudio, proponer un diseño físico en el que se describen los tablespaces que serán creados para soportar la operación de la base de datos para cada uno de los módulos.

- Considerar el almacenamiento de objetos CLOB, BLOB en discos diferentes. Considerar el uso de Big File tablespaces.
- Realizar un análisis y determinar las tablas que pudieran tener una gran cantidad de registros con el paso del tiempo, proponer un esquema simple de particionamiento. Considerar almacenar los datos de cada partición en un disco diferente.
- Considerar el almacenamiento de los índices en un tablespace separado.
- No olvidar que la distribución propuesta debe cuidar en todo momento que si los tablespaces de un módulo se detienen o fallan, el otro módulo debería seguir operando sin mayor problema.
- Tener presente y procurar en todo momento evitar posibles problemas de contención, por ejemplo, considerar crear tablespaces con más de un data file donde cada archivo se almacena en discos diferentes.

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

1.3.6.1 Tablespaces comunes

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

1.3.6.2 Tablespaces por módulo

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

1.3.6.3. Asignación de tablespace por objeto y módulo

Con base al diseño de tablespaces propuesto, cada módulo estará formado por varios tablespaces. En cada módulo existirán segmentos de diferente tipo (tablas, índices, particiones, objetos blob/clob) que requieren la asignación de su correspondiente tablespace. En esta tabla se podrá realizar una propuesta de distribución de los diferentes segmentos empleando los tablespaces definidos anteriormente.

Módulo	Tipo de segmento	Nombre del segmento	Nombre del tablespace
--------	------------------	---------------------	-----------------------

1.3.8. Generación del código DDL para el modelo relacional

A partir del modelo relacional generado 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 de Constraint	Convención de nombrado
UNIQUE	<nombre_tabla>_<nombre_columna>_uk
PRIMARY KEY	<nombre_tabla>_<nombre_columna>_pk
FOREIGN KEY	<nombre_tabla_hija>_<nombre_columna>_fk
CHECK	<nombre_tabla>_<nombre_columna>_<chk>

- 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.

1.3.10. 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).

Cálculo del espacio requerido para la FRA TODO: Calcular el espacio requerido para la FRA
$$\left(\sum_{k=1}^n a_k b_k \right)^2 \leq \left(\sum_{k=1}^n a_k^2 \right) \left(\sum_{k=1}^n b_k^2 \right)$$

1.3.11. Modo archivolog

- La ubicación propuesta para las dos ubicaciones es
 - La ubicación del disco dedicado que es **/unam/bda/proyecto-final/archiveLogs/FREE/disk_a**
 - FRA **/unam/bda/proyecto-final/fast-recovery-area**
- Tip: Los discos donde se almacene la copia que no está en la FRA debería ser dedicado.

1.3.12. Planeación del esquema de respaldos.

Diseñar 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.
- Por la naturaleza de las pdbc en las que estamos desarrollando el proyecto y se parece a un ambiente de Uber Eats, hemos propuesto realizar backups incrementales de las pdbc.
- Para la frecuencia de repetición hemos decidido realizar backups incrementales diarios de las pdbc. Esto con el fin de tener un respaldo diario de los datos que se generan en la base de datos.
- Para la ubicación de los respaldos hemos decidido almacenarlos en la FRA. Esto con el fin de tener un respaldo en caso de que se pierda la información de la base de datos.
- Para la política de retención de backups hemos decidido mantener los backups incrementales diarios por 7 días. Recordando que una política de retención de backups es aquella que define cuánto tiempo se mantendrán los backups en el sistema. Nosotros

```
configure retention policy.
```

TODO: Para nuestros respaldos hemos decidido...

- Verificar los scripts en nuestras carpetas

1.3.15. 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
--------------	----------------------------	-------------------	----------------	---------------------------------

8.4.1.6. Respalda archive redo logs El siguiente comando realiza un respaldo completo de la base de datos, realiza switch de los Redo logs e incluye los archived redo logs en el backup. `backup database plus archivelog;` 8.4.1.7. Respalda tablespaces `backup device type disk tablespace users, tools;` 8.4.1.8. Respalda datafiles `backup device type sbt datafile 1,2,3,4 datafilecopy'/tmp/system01.dbf';` 8.4.1.9. Respalda el archivo de control En caso de que la configuración configure `controlfile autobackup` no esté habilitada, se deben emplear alguna de las siguientes instrucciones para incluirlo en un backup. `backup current controlfile` Jorge A. Rodríguez C. jorgerdc@gmail.com 24 Material de apoyo FI UNAM `backup device type sbt tablespace users include current controlfile;` `backup as copy current controlfile format'/tmp/control01.ctl';` `backup as copy current controlfile format'/tmp/control01.ctl';` `backup device type sbt controlfilecopy'/tmp/control01.ctl';` 8.4.1.10. `backup device type sbt spfile;`