

PRACTICA COMPLEMENTARIA 7
SQL *PLUS Y LENGUAJE DE DEFINICIÓN DE DATOS (DDL)

Esta práctica se deberá entregar de forma individual.

1.1. OBJETIVO:

Poner en práctica los conceptos adquiridos del Lenguaje de Definición de DATOS (DDL) como parte fundamental del estudio del lenguaje SQL enfocado a la administración de la estructura de una Base de datos. Adicionalmente, en esta práctica se comprenderá el uso básico de la herramienta SQL *Plus para interactuar con una instancia Oracle.

1.2. ACTIVIDADES PREVIAS.

Antes de realizar los ejercicios de la práctica, **realizar** las actividades del archivo practica7-previo.pdf

No olvidar: Todos los archivos SQL que se solicitan en la práctica y posteriores deberán llevar el siguiente encabezado:

```
--@Autor(es):      <Nombre de los integrantes>
--@Fecha creación:  dd/mm/aaaa
--@Descripción:    <breve descripción del contenido y propósito del archivo>
```

1.3. CREACIÓN DE USUARIOS

Crear un archivo SQL con el nombre s-01-usuarios.sql El archivo deberá contener lo siguiente:

- Crear un usuario con el nombre <iniciales>_p0703_admin donde <iniciales> representa a una cadena de 3 caracteres formada por la primera letra del nombre, la primera letra del apellido paterno y la primera del apellido materno del alumno. Asignarle una cuota de 1 GB y activar la cláusula password expire. Otorgar privilegios para crear sesiones.
- Crear un usuario invitado con el nombre <iniciales>_p0703_invitado. Asignarle el password "invitado", el usuario no deberá tener permisos para almacenar objetos en disco. Otorgar privilegios únicamente para crear sesiones.
- Crear un rol llamado p0703_admin_rol. Este rol deberá permitir crear los siguientes objetos: tablas, vistas, sinónimos, secuencias, triggers y procedimientos almacenados.
- Asignarle el rol p0703_admin_rol al usuario <iniciales>_p0703_admin creado anteriormente.
- Ejecutar el script empleando el usuario SYS.

Ejemplo:

(tratar de seguir el mismo estilo para los demás scripts)

```
--@Autor(es):      Jorge Rodríguez
--@Fecha creación:  dd/mm/yyyy
--@Descripción:    Creación de usuarios y roles
```

```
prompt Proporcionar el password del usuario sys:
```

```
connect sys as sysdba
```

```
prompt creando usuario jrc_p0703_admin
```

```
--completar
```

```
Prompt creando al usuario jrc_p0703_invitado
```

```
--completar
```

```
Prompt creando roles
```

```
--Completar
```

```
Prompt Asignar el rol p0703_admin_rol a jrc_p0703_admin
```

```
--Completar
```

```
Prompt Listo!
```

```
disconnect;
```

1.4. CREACIÓN DE OBJETOS

Crear un archivo SQL con el nombre s-02-admin-ddl.sql.

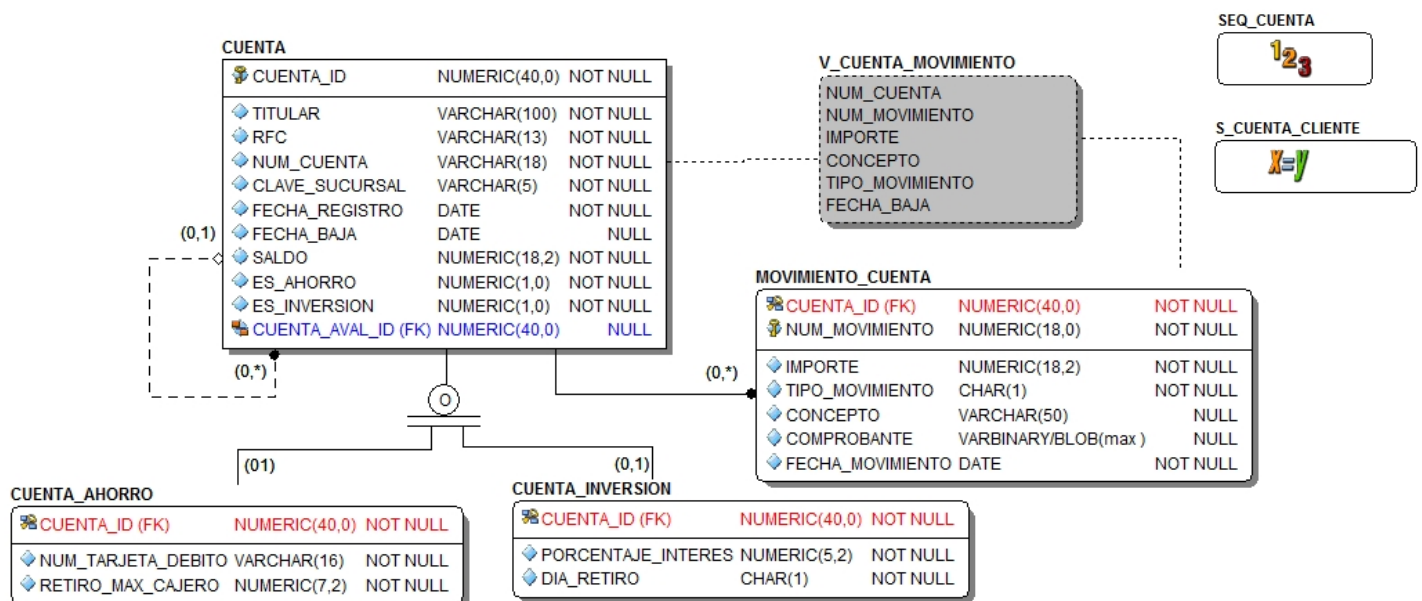
- El archivo deberá contener las instrucciones necesarias para conectarse a la instancia empleando el usuario <iniciales>_p0703_admin. Observar que se deberá actualizar el password debido a la opción password expire.
- El script deberá contener la definición de todos los objetos que se describen en el siguiente modelo relacional excepto el **sinónimo** s_cuenta_bancaria. Considerar las reglas de negocio que se muestran a continuación. Implementar todos los requerimientos solicitantes haciendo uso de los distintos tipos de constraints.
- Los nombres de las restricciones deben seguir la siguiente convención (Revisar el archivo BD/teoria/tema7/tema7-partel.pdf para mayores detalles).

Tipo Constraint	Convención de nombrado
not null	Para este tipo de constraint, generalmente se emplea la forma corta y no se especifica un nombre: <pre>create table empleado(num_empleado number not null);</pre> Si se desea emplear la sintaxis antes mencionada, la convención es: <prefijo_tabla>_<nombre_columna>_nn
unique	<prefijo_tabla>_<nombre_columna>_uk
primary key	<nombre_tabla>_pk
references, foreign key	<prefijo_tabla_hija>_<nombre_columna>_fk
check	<prefijo_tabla>_<nombre_columna>_chk

- Los nombres de los índices deben seguir la siguiente convención (Revisar el archivo BD/teoria/tema7/tema7-partel.pdf para mayores detalles).

Tipo de índice	Convención
Non unique	<prefijo_tabla>_<nombre_campo>_ix Si en índice es compuesto, usar <nombre_grupo> en lugar de <nombre_campo>
Unique	<prefijo_tabla>_<nombre_campo>_iuk Si en índice es compuesto, usar <nombre_grupo> en lugar de <nombre_campo>

1.4.1. Caso de estudio: Registro de cuentas bancarias.



- La secuencia seq_cuenta debe iniciar en 1001, incrementarse de 2 en 2, no cíclica con 9 valores en cache.
- Incluir en el script una instrucción que obtenga el primer valor de la secuencia y otra instrucción que muestre dicho valor sin incrementarlo.

1.4.1.1. Requerimientos para garantizar consistencia de datos.

- Las combinaciones válidas para los atributos `es_ahorro`, `es_inversion` son: (1,1),(0,1),(1,0) , es decir, se permite que una Cuenta adquiera los 2 roles, uno de ellos , pero debe contar con al menos un rol. Se debe validar que no se intente guardar o modificar un valor (0,0). Tip: Emplear un 'Table constraint' para implementar esta regla.
- El número de cuenta no debe duplicarse y debe iniciar con el prefijo 'AH' si es de ahorro y 'IN' si es de inversión. Si la cuenta tiene los 2 roles asignados, su prefijo debe ser 'INAH'.
- El saldo de la cuenta no debe ser menor a los \$1,500.
- En caso de no haberse especificado la fecha de registro se deberá asignar la fecha del sistema.
- En caso de existir, la fecha de baja debe ser al menos 1 mes después con respecto a la fecha de registro.
- El número de tarjeta de débito no debe duplicarse.
- El monto máximo para retirar de cajero es \$12,500
- El día de retiro para una cuenta de inversión se refiere al día del mes en el que el titular puede hacer retiro de su cuenta. Este valor debe estar entre el quinto día del mes y el día 25.
- La combinación de valores, `rfc`, `titular` no debe duplicarse. Se decide emplear un índice para validar esta regla.

1.4.1.2. Requerimientos para mejorar desempeño

- Frecuentemente se realizan búsquedas empleando las primeras 5 letras del titular sin importar si el dato está capturado en minúsculas o mayúsculas. Se decide crear un índice para cuidar desempeño.
- De forma similar, se consultan los movimientos de las cuentas con base al campo `fecha_movimiento`. Las consultas hacen un uso muy frecuente del mes y año en el formato `mm/yyyy`. Se ha decidido indexar el campo considerando este formato de fechas.
- Otra consulta de uso frecuente son las operaciones `join` empleando la FK `cuenta_aval_id`. Se ha decidido indexar este campo.
- C3. Incluir en el reporte** únicamente el código DDL que se empleó para crear la tabla `cuenta`

Ejecutar el script.

1.4.2. Creación de sinónimos.

- Se desea que el usuario invitado pueda leer el contenido de la tabla `cuenta`. Para ello el usuario invitado deberá crear el sinónimo privado `s_cuenta_cliente` mostrado en el modelo relacional. El usuario admin deberá otorgarle permisos al usuario invitado para que pueda hacer consultas a la tabla. Por otro lado, el usuario invitado deberá contar con permisos para poder crear sinónimos privados.
- Generar un script llamado `s-03-crea-sinonimo.sql` que contenga todas las instrucciones necesarias para implementar esta funcionalidad.

1.4.3. Verificación de creación de objetos.

- De la carpeta compartida en drive correspondiente a esta práctica complementaria, obtener los siguientes archivos:
 - Archivos con extensión `.plb`
 - `s-04-validador-username.sql`
- Guardarlos en el mismo directorio donde se encuentran todos los scripts de la práctica.
- El primer archivo se encarga de validar que los ejercicios hasta el momento hayan sido resueltos correctamente. Para ejecutarlo realizar las siguientes acciones:
 - Abrir el archivo `s-04-validador-username.sql` para configurar los usuarios y passwords empleados en esta práctica (leer los comentarios).
 - Entrar a `Sql *Plus` y ejecutar el archivo `s-04-validador-main.plb`

Ejemplo:

```
sqlplus /nolog
-SQL> start s-04-validador-main.plb
```

- El script mostrará el resultado del proceso de validación. En caso de encontrar errores, revisar la descripción del error, corregir y reintentar.
- C4. Incluir en el reporte** el resultado de la ejecución de este script a partir del punto indicado.

Importante: Asegurarse que el validador no reporte error alguno antes de pasar a la siguiente sección. Posterior a la ejecución del siguiente ejercicio, ya no se podrá ejecutar nuevamente el validador ya que la estructura de los objetos va a cambiar.

1.5. MODIFICACIÓN DE LA ESTRUCTURA DE OBJETOS.

Generar un script `s-05-cambios-objetos.sql` Para cada uno de los siguientes cambios, generar una instrucción SQL.

1. Agregar un campo alfanumérico opcional de 18 caracteres CURP a cuenta opcional.
2. La BD debe validar que el porcentaje de interés de las cuentas de inversión debe tener un valor máximo del 20%.
3. Se ha modificado una regla de negocio en la que se explica que las cuentas ya no podrán ser laborales y de inversión al mismo tiempo. Ahora, una cuenta solo podrá tener un tipo asignado. Se ha decidido emplear el campo `tipo_cuenta char(1)` como discriminante. Aplicar los cambios necesarios. El tipo de cuenta solo podrá aceptar valores I, A. Tip: Para crear este campo, se deberán eliminar los campos anteriores que actuaban como discriminantes: `es_saldo`, `es_inversion`. Notar que si alguno de estos 2 campos cuenta con constraints donde participan 2 o más columnas, dichos constraints deberán ser eliminados antes de eliminar a los campos `es_saldo`, `es_inversion`. Para este caso, existen 2 constraints que se deberán eliminar. De no eliminar dichos constraints se obtendrá ERROR ORA-12991: column is referenced in a multi-column constraint
4. Se desea modificar el nombre del campo titular en cuenta a `nombre_titular`
5. Se desea renombrar la tabla `movimiento_cuenta` a `operacion_cuenta`
6. Se desea que la tabla `operacion_cuenta` tenga una llave primaria artificial llamada `operacion_cuenta_id` en lugar de una PK compuesta. El nombre del constraint de llave primaria debe llamarse `operacion_cuenta_pk`. Restricción: No se deben eliminar campos.
7. Se desea modificar a `number(20, 2)` el campo `importe` en `operacion_cuenta`.
8. Se desea agregar el campo `nombre_tutular` a la vista `v_cuenta_movimiento`. El campo en la vista debe llamarse `cuentahabiente`.
9. Debido a los cambios aplicados hasta el momento, la vista se volverá un objeto inválido debido al cambio de nombre de las tablas y/o columnas. Revisar la definición de la vista, actualizar cambios y volver a crearla. La siguiente instrucción permite verificar el status de la vista (`valid/invalid`):
`select status from user_objects where object_name = 'V_CUENTA_MOVIMIENTO';`
10. La secuencia `seq_cuenta` debe ser ahora cíclica y debe reiniciarse con el valor 15. El último valor permitido es 100,000,000.
11. Se desea renombrar el índice aplicado a los campos `rfc` y `nombre_titular` a: `cuenta_cuentahabiente_uk`

1.5.1. Verificación de cambios.

- Ejecutar el script `s-05-validador-cambios-objetos.plb` de forma similar al ejercicio anterior. En caso de obtener errores, revisar los mensajes de error, corregir y reintentar. **C5. Incluir en el reporte** el resultado de su ejecución a partir del punto indicado.

1.6. CONTENIDO DEL REPORTE.

Para realizar la evaluación de la práctica se deberá anexar en la última página del reporte la rúbrica correspondiente:

- Rúbrica para el grupo de laboratorio.
- Rúbrica para el grupo de teoría plan 2010
- Rúbrica para grupo de teoría plan 2016 inscritos en otro grupo de laboratorio

Imprimir alguna de las siguientes páginas de este documento e incluirla en el reporte. Prácticas que no incluyan esta tabla se considerarán como prácticas no entregadas. La rúbrica permite conocer a detalle los criterios empleados para asignar la calificación final.

Importante: Los validadores que se proporcionan como archivos cifrados y algunos ejercicios de la práctica, permiten identificar fácilmente si el código y en general, el contenido del reporte corresponde a “copias” o “plagios” totales o parciales de otros reportes. En caso de confirmarse una situación de este tipo, la calificación del reporte será de **-110 puntos (negativo)**, y **todos** los alumnos involucrados deberán realizar primera y segunda vuelta de exámenes finales **completos**. Al segundo intento, los alumnos no tendrán derecho a calificación aprobatoria tanto teoría como Laboratorio. ***¡No se arriesguen, no compartan su trabajo!, ¡Compartan conocimiento!***, es decir, resuelvan dudas, expliquen lo que han aprendido. Esto aplica para cualquier práctica.

PRACTICA 7
Rubrica para grupo del laboratorio

Contenido	Puntaje Obtenido		Observaciones
Carátula *	0P	5P	
objetivos e Introducción *	0P	5P	
Actividades Previas de la práctica.			
C1. Resultados de la actividad 1. (Inciso N).	0P	5P	
C2. Resultado de ejecutar la actividad 2 (terminal 2).	0P	10P	
Práctica complementaria.			
C3. Código DDL de la tabla solicitada (Script con header, correctamente formateado e indentado), código válido, compilable y funcional.	0P	5P	
C4. Resultado del primer validador.	-20P 3 o más errores	5P 2 errores 30P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador.
C5. Resultado del segundo validador	-20P 3 o más errores	5P 2 errores 30P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador.
Conclusiones, comentarios, recomendaciones *	0P	5P	
Bibliografía *	0P	5P	

* Ver Rubrica general de prácticas para mayores detalles en cuanto a los requisitos que debe cumplir el elemento de evaluación y los puntajes asignados.

PRACTICA 7
Rubrica para grupo de teoría plan 2010

Contenido	Puntaje Obtenido		Observaciones
Carátula *	0P	5P	
objetivos e Introducción *	0P	5P	
Actividades Previas de la práctica.			
C1. Resultados de la actividad 1. (Inciso N).	0P	5P	
C2. Resultado de ejecutar la actividad 2 (terminal 2).	0P	10P	
Práctica complementaria.			
C3. Código DDL de la tabla solicitadas (Script con header, correctamente formateado e indentado), código válido, compilable y funcional.	0P	5P	
C4. Resultado del primer validador.	-20P 3 o más errores	5P 2 errores 30P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador.
C5. Resultado del segundo validador	-20P 3 o más errores	5P 2 errores 30P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador.
Conclusiones, comentarios, recomendaciones *	0P	5P	
Bibliografía *	0P	5P	

* Ver Rubrica general de prácticas para mayores detalles en cuanto a los requisitos que debe cumplir el elemento de evaluación y los puntajes asignados.

PRACTICA 7

Rubrica para grupo de teoría plan 2016 inscritos en otro grupo de laboratorio

Contenido	Puntaje Obtenido		Observaciones
Carátula *	0P	5P	
Actividades Previas de la práctica.			
C1. Resultados de la actividad 1. (Inciso N).	0P	5P	
C2. Resultado de ejecutar la actividad 2 (terminal 2).	0P	10P	
Práctica complementaria.			
C3. Código DDL de la tabla solicitada (Script con header, correctamente formateado e indentado), código válido, compilable y funcional.	0P	5P	
C4. Resultado del primer validador.	-25P 3 o más errores	5P 2 errores 35P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador.
C5. Resultado del segundo validador	-25P 3 o más errores	5P 2 errores 35P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador.
Conclusiones, comentarios, recomendaciones *	0P	5P	

* Ver Rubrica general de prácticas para mayores detalles en cuanto a los requisitos que debe cumplir el elemento de evaluación y los puntajes asignados.