

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 crear los siguientes objetos:

Tipo de Objeto	Nombre	Requerimientos
Rol	p0701_admin_rol	Este rol deberá permitir crear los siguientes objetos: sesiones, tablas, vistas.
Rol	p0701_basic_rol	Este rol deberá permitir crear los siguientes objetos: sesiones y tablas.
Rol	p0701_guest_rol	Este rol deberá permitir crear los siguientes objetos: sesiones y sinónimos privados.
Usuario	<iniciales>_p0701_admin	Asignarle una cuota de 1 GB, asignarle el rol p0701_admin_rol
Usuario	<iniciales>_p0701_oper	Asignarle una cuota de 500 MB, asignarle el rol p0701_basic_rol
Usuario	<iniciales>_p0701_invitado	El usuario no deberá tener permisos para almacenar objetos en disco. Asignarle el rol p0701_guest_rol
Privilegios de sistema	create procedure, create sequence	Estos 2 privilegios deberán ser otorgados a los 3 usuarios anteriores.

- <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.
- El valor del password es a libre elección.
- Ejecutar el script empleando el usuario SYS.

Ejemplo:

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

```
--@Autor(es):      Jorge Rodríguez
--@Fecha creación: 06/05/2017
--@Descripción:    Creación de usuarios y roles
```

prompt Proporcionar el password del usuario sys:

connect sys as sysdba

prompt creando usuario jrc p0701 admin  
--completar

Prompt creando al usuario jrc\_p0701\_invitado  
--completar

Prompt creando roles  
--Completar

Prompt Asignando roles  
--Completar

Prompt Listo!  
disconnect;

#### 1.4. CREACIÓN DE OBJETOS

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

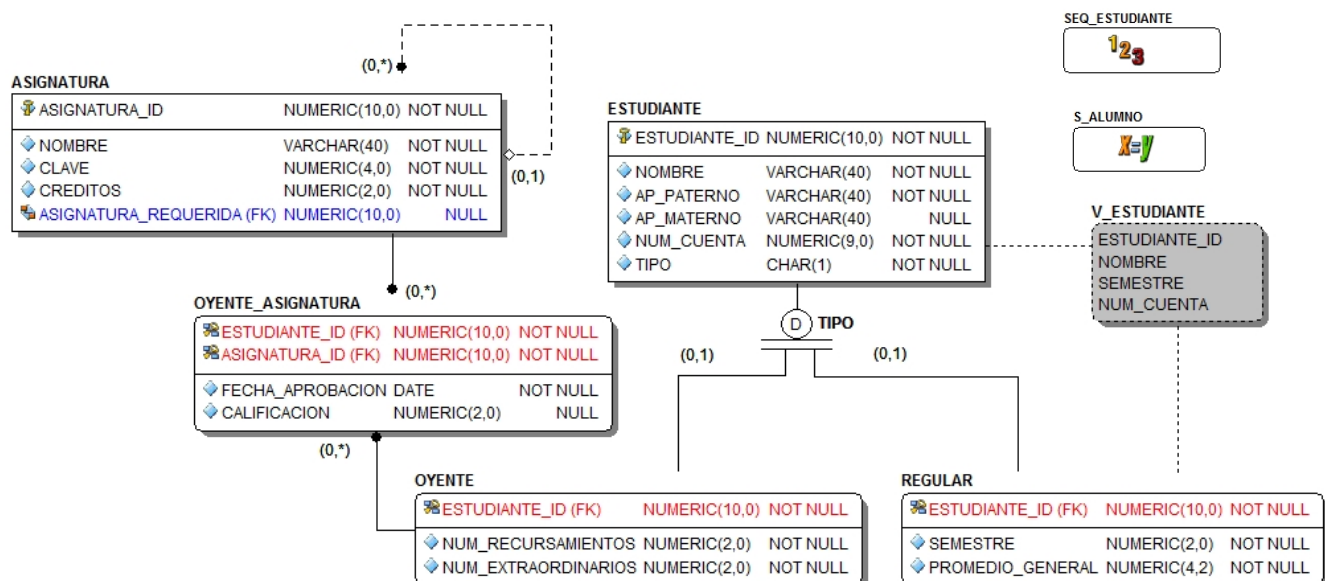
- El archivo deberá contener las instrucciones necesarias para conectarse a la instancia empleando el usuario <iniciales>\_p0701\_admin.
- El script deberá contener la definición de todos los objetos que se describen en el siguiente modelo relacional excepto el **sinónimo** S\_ALUMNO. Considerar las 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: <nombre_tabla>_<nombre_columna>_nn
unique	<nombre_tabla>_<nombre_columna>_uk
primary key	<nombre_tabla>_pk
references, foreign key	<nombre_tabla_hija>_<nombre_columna>_fk
check	<nombre_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	<nombre_tabla>_<nombre_campo>_ix Si en índice es compuesto, usar <nombre_grupo> en lugar de <nombre_campo>
Unique	<nombre_tabla>_<nombre_campo>_iuk Si en índice es compuesto, usar <nombre_grupo> en lugar de <nombre_campo>

##### 1.4.1. Caso de estudio: Control escolar.



- La secuencia `seq_estudiante` debe iniciar en 10, incrementarse de 3 en 3, no cíclica con 5 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.
- El tipo del estudiante solo puede tener los valores O, R
- La `fecha_aprobacion` que se asigna por defecto, es la fecha actual del sistema.
- En caso de existir, la `calificación` debe encontrarse en el rango [5,10].
- El atributo `num_cuenta` debe contar con el siguiente formato: Los números de cuenta de un alumno regular deben iniciar en 31. Cuando se inserta o se actualiza el tipo de estudiante a Oyente, su número de cuenta debe ser modificado, debe iniciar con 30.
- La clave de la asignatura no debe duplicarse.
- Para los estudiantes Oyentes, la suma del número de extraordinarios y el número de veces que ha recurso la asignatura no debe exceder a 10
- Las búsquedas de los alumnos se realizan empleando el número de cuenta con una frecuencia muy alta. Las consultas no consideran todos los caracteres del número de cuenta. Solo se consideran del tercer al octavo carácter. Se ha decidido crear un índice basado en funciones para que dichas consultas sean lo más eficiente posible.
- Se desea indexar el campo `asignatura_requerida_id` para mejorar desempeño en consultas.
- `v_estudiante` se refiere a la definición de una vista que contiene atributos de las tablas `estudiante` y `regular`.
- **C3. Incluir en el reporte** únicamente el código DDL que se empleó para crear la tabla `oyente_asignatura` y `estudiante`.

Ejecutar el script.

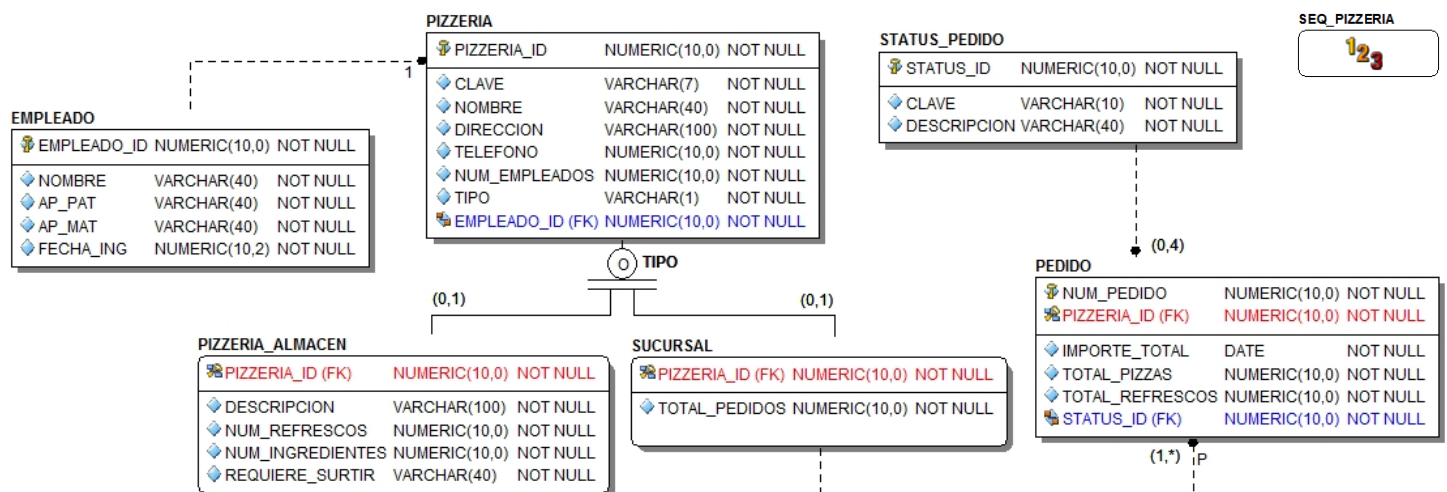
### 1.4.2. Creación de sinónimos.

- Se desea que el usuario invitado pueda leer el contenido de la tabla `estudiante`. Para ello el usuario invitado deberá crear el sinónimo privado 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.
- Emplear SQL \*Plus, conectarse empleando el usuario invitado y ejecutar el script

### 1.5. MODIFICACIÓN DE LA ESTRUCTURA DE UNA BASE DE DATOS.

- De la carpeta compartida obtener el script `s-04-oper-ddl.sql`
- Editar el script para modificar los datos del usuario `oper`, emplear SQL \*Plus para ejecutar el script.

Este script crea los objetos mostrados en el siguiente modelo relacional que corresponden a una BD de una cadena de pizzerías. Como se puede apreciar en el modelo, el diseñador cometió varios errores y se desea corregirlos. Para ello, considerar los siguientes puntos:



- No se permite la eliminación de tablas ya que hay referencias entre ellas.
- Se deberán generar las instrucciones `alter` necesarias para corregir los errores.
- Una pizzería puede tener el rol de almacén (`es_almacen`) y sucursal (`es_sucursal`) de forma simultánea.
- El atributo `requiere_surtir` de la tabla `pizzeria_almacen` sólo puede tener valores de 0 ó 1 para indicar si el almacén requiere ser surtido. Además, modificar el tipo de dato para que sea el adecuado.
- El campo `ap_pat` y `ap_mat` deberá renombrarse a `ap_paterno` y `ap_materno` respectivamente. Éste último deberá ser opcional.

- El campo `status_id` en `status_pedido` debe llamarse `status_pedido_id`
- La FK `empleado_id` en `pizzeria` está invertida.
- El `importe_total` de la tabla `pedido`, al igual que la `fecha_ing` de la tabla `empleado`, no tienen el formato adecuado.
- La `fecha_apertura` no fue indicada en la tabla `sucursal` donde el valor por defecto es la fecha actual del sistema.
- Los 3 primeros caracteres de la clave de cada pizzería debe ser `PIZ`, seguido de un guión ( - ). El siguiente carácter será 'A' si se trata de un almacén, el carácter 'S' si se trata de una sucursal o los caracteres 'AS' si cuenta con ambos roles. El siguiente carácter es nuevamente un guión ( - ) y finalmente se concatena una secuencia de tres dígitos, que aumenta de uno en uno iniciando en 001. Ejemplo:
  - Si la primera pizzería a registrar es almacén entonces el formato de la clave es la siguiente: `PIZ-A-001`
  - Si la segunda pizzería a registrar es sucursal entonces el formato de la clave es la siguiente: `PIZ-S-002`
  - Si la tercera pizzería a registrar es almacén y sucursal entonces el formato de la clave es la siguiente: `PIZ-AS-003`
  - Considerar que con esta regla de negocios, este campo debe ser de 10 caracteres.
- Se desea que `pedido` tenga una PK artificial y que el índice compuesto implícito se conserve. Además de que el atributo `num_pedido` sea renombrado a `pedido_id` y este sea la pk artificial.
- El nombre del campo para almacenar la fecha de ingreso del empleado debe llamarse `fecha_ingreso`.
- La longitud máxima del número telefónico de cada pizzería debe ser de 10 dígitos.
- La tabla `pizzeria_almacen` debe renombrarse a `almacen`
- La secuencia `seq_pizzeria` no debe ser cíclica, debe contar con 7 elementos en el caché, su valor mínimo debe ser 7, incrementable cada 4 valores.

Posterior a la aplicación de los cambios, las siguientes sentencias SQL deberán ser ejecutadas sin error alguno.

```
--1
insert into pizzeria(pizzeria_id,clave,nombre,direccion,
telefono,num_empleados,es_almacen,es_sucursal)
values(1,'PIZ-AS-001','Pizzas Cachos','Av. Juarez 512',5510028938,
200,1,1);
--2
insert into pizzeria(pizzeria_id,clave,nombre,direccion,
telefono,num_empleados,es_almacen,es_sucursal)
values(2,'PIZ-S-002','Pizza Planeta','Av. Constitucion de 1917 10',5513523978,
400,0,1);
--3
insert into empleado(empleado_id,nombre,ap_paterno,ap_materno,
fecha_ingreso,pizzeria_id)
values(1,'Daniel','Lopez',null,sysdate,1);
--4
insert into almacen(pizzeria_id,descripcion,num_refrescos,
num_ingredientes,requiere_surtir)
values(1,'Almacen para pizza planeta',80,100,0);
--5
insert into sucursal(pizzeria_id,fecha_apertura,total_pedidos)
values(2,sysdate,500);

--6
insert into status_pedido(status_pedido_id,clave,descripcion)
values(1,'ACTIVO','SOLICITADO');
--7
insert into pedido(pedido_id,importe_total,total_pizzas,total_refrescos,
pizzeria_id,status_pedido_id)
values(1,145.50,1,2,2,1);
```

- Generar un script `s-05-oper-alter.sql` con los cambios requeridos.
- Emplear SQL \*Plus, conectarse empleando el usuario `oper` y ejecutar el script.

## 1.6. VALIDACIÓN DE RESULTADOS.

### 1.6.1. Verificación de creación de objetos.

- De la carpeta compartida en drive correspondiente a esta práctica complementaria, obtener todos los archivos `sql` y `plb`
- Guardarlos en el mismo directorio donde se encuentran todos los scripts de la práctica.
- Abrir el archivo `s-06-validador-variables-usuarios.sql` para configurar los usuarios y passwords empleados en esta práctica (leer los comentarios).
- Entrar a `Sql *Plus` y ejecutar el archivo `s-06-validador-main.sql`

Ejemplo:

```
sqlplus /nolog  
-SQL> start s-06-validador-main.sql
```

- 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.

#### 1.7. 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.

**PRÁCTICA 7**  
**Rúbrica para grupo del laboratorio**

Contenido	Penalizaciones mínimas	Puntaje máximo	Observaciones
Carátula *	-5P	5P	
objetivos e Introducción *	-5P	5P	
<b>Actividades Previas de la práctica.</b>			
<b>C1.</b> Resultados de la actividad 1. (Inciso N).	-10P Resultados incorrectos o no realizados	5P	
<b>C2.</b> Resultado de ejecutar la actividad 2 (Terminal 2).	-10P Resultados incorrectos o no realizados	5P	
<b>Práctica complementaria.</b>			
<b>C3.</b> Código DDL de la tabla solicitada (Script con header, correctamente formateado e indentado), código válido, compilable y funcional).	-10P	10P	
<b>C4.</b> Resultado de validador.	-10 P por error	60P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador, si se altera o si se encuentran 3 o más errores.
Conclusiones, comentarios, recomendaciones *	-5P	5P	
Bibliografía *	-5P	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.

**PRÁCTICA 7**  
**Rúbrica para grupo de teoría plan 2010**

Contenido	Penalizaciones mínimas	Puntaje máximo	Observaciones
Carátula *	-5P	5P	
objetivos e Introducción *	-5P	5P	
<b>Actividades Previas de la práctica.</b>			
<b>C1.</b> Resultados de la actividad 1. (Inciso N).	-10P Resultados incorrectos o no realizados	5P	
<b>C2.</b> Resultado de ejecutar la actividad 2 (Terminal 2).	-10P Resultados incorrectos o no realizados	5P	
<b>Práctica complementaria.</b>			
<b>C3.</b> Código DDL de la tabla solicitada (Script con header, correctamente formateado e indentado), código válido, compilable y funcional).	-10P	10P	
<b>C4.</b> Resultado de validador.	-10 P por error	60P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador, si se altera o si se encuentran 3 o más errores.
Conclusiones, comentarios, recomendaciones *	-5P	5P	
Bibliografía *	-5P	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.

## PRÁCTICA 7

## Rúbrica para grupo de teoría plan 2016 inscritos en otro grupo de laboratorio

Contenido	Penalizaciones mínimas	Puntaje máximo	Observaciones
Carátula *	-5P	5P	
<b>Actividades Previas de la práctica.</b>			
<b>C1.</b> Resultados de la actividad 1. (Inciso N).	-10P Resultados incorrectos o no realizados	5P	
<b>C2.</b> Resultado de ejecutar la actividad 2 (Terminal 2).	-10P Resultados incorrectos o no realizados	5P	
<b>Práctica complementaria.</b>			
<b>C3.</b> Código DDL de la tabla solicitada (Script con header, correctamente formateado e indentado), código válido, compilable y funcional).	-10P	10P	
<b>C4.</b> Resultado de validador.	-10 P por error	80P Sin Errores	La práctica no se evalúa si no se incluye la salida del validador, si se altera o si se encuentran 3 o más errores.
Conclusiones, comentarios, recomendaciones *	-5P	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.