

TEMA 7 PARTE I. –EJERCICIOS EN CLASE
CREACIÓN, MODIFICACIÓN Y ELIMINACIÓN DE OBJETOS CON SQL

1.1. SPOOL

Activar el spool para almacenar toda la interacción y comandos ejecutados durante la sesión. Esta actividad se deberá realizar al inicio de cada clase, crear un archivo nuevo por cada clase con el siguiente formato: bd-<grupo>-<iniciales>-<mm>-<dd>.txt Los valores corresponden al número de grupo, primera letra del nombre y apellidos del alumno, mes y día respectivamente. Emplear 2 dígitos. Ejemplo: bd-03-jrc-10-21.txt

1.2. CREACIÓN DEL USUARIO PARA EJEMPLOS.

Crear un usuario <nombre><grupo>07 empleando el nombre de cada alumno y el número de grupo (2 dígitos) que será el empleado para ejecutar los ejercicios. Asignarle cuota ilimitada en el tablespace users, asignarle privilegios para crear sesión y tablas

1.3. CREACIÓN DE UNA TABLA

- Crear la siguiente tabla. Todos los modelos lógicos se realizaron con tipos de datos estándar, se deberá especificar el equivalente o recomendado para Oracle.
- Generar una consulta para mostrar las tablas que ha creado el usuario
- Formatear la salida de SQL *Plus para un mejor entendimiento.

EMPLEADO_SIMPLE

EMPLEADO_ID	NUMERIC(10,0)	NOT NULL
NOMBRE	VARCHAR(40)	NOT NULL

1.4. TABLAS EXTERNAS.

- Considerar el archivo empleado-ext.sql y el archivo empleado-ext.csv que se encuentran en la carpeta compartida.
- Crear un archivo empleado-ext2.sql a partir del archivo empleado-ext.sql para crear una tabla nueva llamada EMPLEADO_EXT2 que contenga los siguientes campos adicionales: email, sueldo_mensual y comisión.
- Crear un archivo empleado-ext2.csv basado en empleado-ext.csv El archivo deberá contener los siguientes renglones adicionales a los del archivo original:
 - El tercer renglón cuenta con todos los valores de las columnas.
 - El cuarto renglón no cuenta con email
 - El quinto renglón no cuenta con sueldo mensual
 - El sexto renglón no cuenta con comisión.
- Ejecutar el archivo para verificar los resultados. Formatear columnas en caso de ser necesario. Revisar los apuntes del tema 7 para mayores detalles.

Tips:

- En una columna sin valor, las comas aparecen juntas. Ejemplo: <valor1>,,<valor3>
- No olvidar actualizar los nombres de los usuarios, nombres de tablas y los nombres de los archivos log que se especifican en el archivo empleado-ext.sql según corresponda.

EMPLEADO_EXT2

NUM_EMPLEADO	NUMERIC(10,0)	NULL
NOMBRE	VARCHAR(40)	NULL
AP_PATERNAL	VARCHAR(40)	NULL
AP_MATERNAL	VARCHAR(40)	NULL
FECHA_NACIMIENTO	DATE	NULL
EMAIL	VARCHAR(100)	NULL
SUELDO_MENSUAL	NUMERIC(8,2)	NULL
COMISION	NUMERIC(5,2)	NULL

1.5. CREACIÓN DE UNA TABLA TEMPORAL

- Crear la siguiente tabla temporal.
- Insertar un registro.
- Verificar los resultados:

```
select *
from empleado_temp;
```
- Abrir una sesión nueva en otra terminal.
- Consultar nuevamente los datos de la tabla, verificar resultados ¿Qué sucedió?
- Regresar a la sesión original.
- Consultar nuevamente los datos de la tabla, comprobar que siguen presentes.
- Hacer commit.
- Consultar nuevamente los datos de la tabla. ¿Qué sucedió?

EMPLEADO_TEMP

EMPLEADO_ID	NUMERIC(10,0)	NOT NULL
NOMBRE	VARCHAR(40)	NOT NULL

1.6. BREVE INTRODUCCIÓN AL MANEJO TRANSACCIONAL.

- ¿Qué pasa cuando se inicia SQL plus con respecto a las transacciones?
- Insertar un nuevo registro en empleado.
- Consultar la tabla para verificar la existencia del nuevo registro.
- Cerrar la terminal (salida **anormal** de SQL plus)

- E. Volver a entrar a SQL plus, consultar nuevamente la tabla. ¿qué sucedió?
- F. Insertar nuevamente un registro
- G. Salir de SQL plus de forma normal o hacer `commit`.
- H. Entrar nuevamente para comprobar que los datos son permanentes.
- I. Insertar un nuevo registro.
- J. Abrir una nueva sesión en otra terminal, consultar los datos de la tabla, ¿qué sucedió?

1.7. CREACIÓN DE TABLAS CON COLUMNAS VIRTUALES.

- A. Crear la siguiente tabla de tal forma que no exista posibilidad de inconsistencia con los atributos derivados. El IVA se calcula aplicando el 16% del sueldo mensual.
- B. Insertar un registro con la información mínima necesaria.
- C. Consultar resultados ¿Qué se observa?

Notas:

- No se puede reutilizar una columna virtual en otra expresión
- No se requiere especificar el tipo de dato
- La palabra `generated always` es opcional, solo se agrega por claridad.

CALCULO_SUELDO

FECHA_CALCULO	DATE	NOT NULL
SUELDO_MENSUAL	NUMERIC(10,2)	NOT NULL
SUELDO_QUINCENAL	NUMERIC(10,2)	NOT NULL
IVA	NUMERIC(10,2)	NOT NULL

1.8. USO DE DEFAULT:

- A. Crear la siguiente tabla considerando:
 - La fecha de registro si no se especifica se toma la del sistema.
 - El tipo de incidencia es 'N' en caso de no especificarse valor
 - El descuento a aplicar es de 5.5 por default.
- B. Insertar un registro.
- C. Consultar el resultado ¿qué se observa?

INCIDENCIA

DESCRIPCION	VARCHAR(40)	NOT NULL
FECHA_REGISTRO	DATE	NOT NULL
TIPO	CHAR(1)	NOT NULL
DESCUENTO	NUMERIC(5,2)	NOT NULL

1.9. DEFINICIÓN DE CONSTRAINTS NIVEL COLUMNA

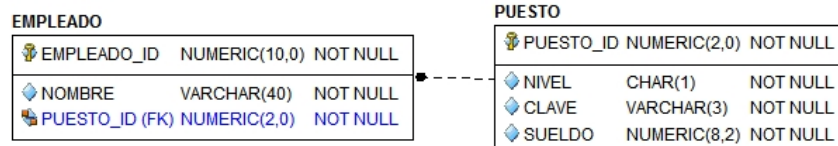
- A. Crear una tabla con las siguientes especificaciones, emplear column constraints.
 - Observar que ya se especifica el atributo que actuará como PK.
 - El nivel del puesto solo debe tener valores 'a', 'b', o 'c'
 - La clave no debe duplicarse
 - El sueldo no debe exceder los 100000

PUESTO

PUESTO_ID	NUMERIC(2,0)	NOT NULL
NIVEL	CHAR(1)	NOT NULL
CLAVE	VARCHAR(3)	NOT NULL
SUELDO	NUMERIC(8,2)	NOT NULL

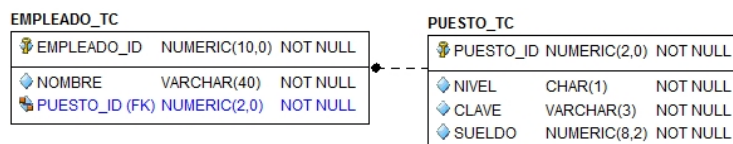
1.10. CREACIÓN DE UNA LLAVE FORÁNEA

- A. Crear la tabla Empleado considerando la FK `puesto_id`, emplear column constraints.



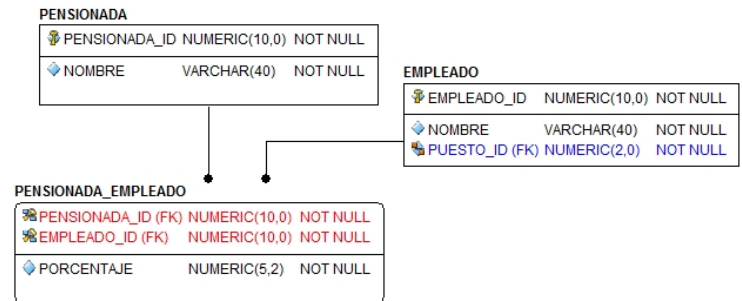
1.11. TABLE CONSTRAINTS (TC)

- A. Aplicando la técnica de Table Constraints, reescribir el código anterior, modificar el nombre de los constraints y de las tablas con base al siguiente diagrama:

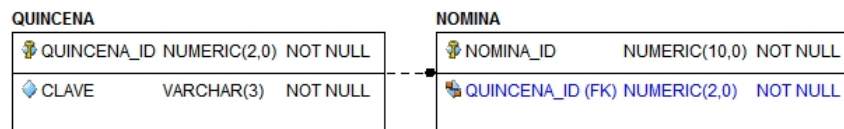


1.12. CREACIÓN DE UNA TABLA CON PK COMPUESTA.

Crear las tablas `pensionada` Y `pensionada_empleado`, reutilizar `empleado`. ¿Es posible emplear cualquier estrategia para crear el constraint de llave primaria?

**1.13. CONSTRAINT ON DELETE.**

- A. Crear las siguientes tablas, de tal forma que, al eliminar una quincena, en automático se eliminen las nóminas asociadas.
 B. En lugar de eliminar las nóminas establecer el valor `null`

**1.14. CREACIÓN DE SECUENCIAS**

- A. Crear una secuencia `MY_SEQ` con las siguientes características:
- Debe iniciar en 100
 - Debe existir una diferencia de 10 entre cada valor generado
 - Al llegar al valor máximo de 200, el siguiente valor a mostrar debe ser el 50
 - Guardar en caché 5 valores
- B. Generar una tabla llamada `BOLETO_IDENTITY`. La tabla contendrá un atributo llamado `CLAVE` de tipo `identity`.
- La tabla debe ser lo más flexible posible, de tal forma que permita sentencias `insert` con valores personalizados e inclusive permita especificar `null` como valor del campo clave.
 - La columna debe iniciar en 1
 - Debe existir una diferencia de 3 entre cada valor.
 - Al llegar al valor máximo de 1000, ya no se deberá permitir la generación de nuevos valores.
 - Guardar en caché 4 valores.
 - Ejecutar el siguiente código PL/SQL para comprobar los resultados:



```

begin
  for i in 1..1000 loop
    insert into boleto_identity (clave) values (null);
    commit;
  end loop;
end;
/
  
```

- Este Código generará un error al alcanzar el valor máximo:

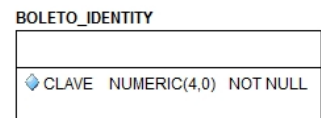
```

ERROR at line 1:
ORA-08004: secuencia ISEQ$$_93374.NEXTVAL exceeds MAXVALUE y no se puede
instanciar
ORA-06512: en línea 3
  
```

- Ejecutar la siguiente sentencia para mostrar el último registro insertado.

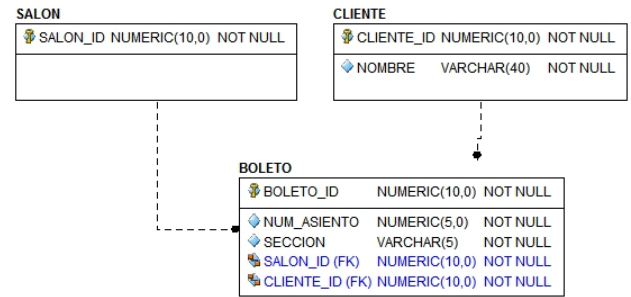
```

select max(clave)
from boleto_identity;
  
```

**1.15. ÍNDICES**

A. Crear las siguientes tablas considerando:

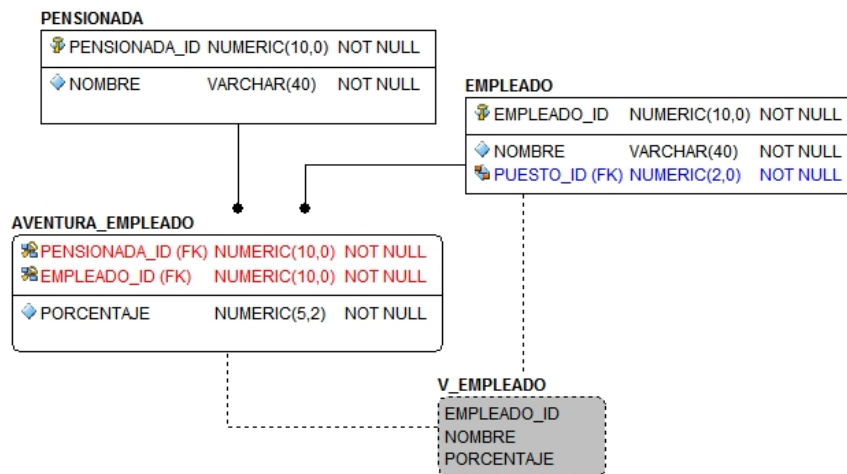
- La combinación salón –sección –número de asiento debe ser única para cada cliente que compra un boleto.
 - Se desea que al hacer búsquedas de los boletos que compra cada cliente sea lo más eficiente posible.
 - Se desea hacer adicionalmente búsquedas por el nombre del cliente. El problema es que algunos nombres están almacenados en minúsculas y otros en mayúsculas. La búsqueda debe ignorar este detalle y debe ejecutarse lo más eficiente posible.
- B. ¿Cuántos índices se crearán después de crear todas las tablas? Comprobar esta respuesta consultando el diccionario de datos.



1.16. VISTAS.

Suponer el siguiente escenario:

- A los empleados se les ha permitido consultar de la base de datos, los porcentajes que se les descontarán de su sueldo por concepto de pensiones.
 - Por seguridad el empleado no debe ver el nombre de la pensionada y solo se le permite consultar los valores de empleado_id, nombre del empleado y porcentaje.
 - Se decide construir una vista para implementar este requerimiento.
- A. Genere el código para crear la vista.



1.17. SINÓNIMOS

- A. Crear un usuario invitado, únicamente otorgarle permisos para crear sesión.
- B. El usuario invitado podrá consultar los datos de la tabla EMPLEADO del usuario dueño de dicha tabla.
- C. Empleando el usuario invitado, generar un sinónimo público llamado emp1 de tal forma que el usuario invitado pueda ejecutar sin problemas la siguiente instrucción:
- ```
select * from emp1;
```
- D. Empleando nuevamente el usuario invitado, generar un sinónimo privado llamado emp2 de tal forma que el usuario invitado pueda ejecutar sin problemas la siguiente instrucción:
- ```
select * from emp2;
```
- E. Conectarse con cualquier otro usuario diferente a invitado, verificar que efectivamente solo está disponible el sinónimo público.

EMP1

EMP2