Nota: la creación y consulta de algunas tablas tiene señalizada el esquema, debido a que al encontrarme logueado como SYSDBA en el Toad, inicialmente me creaba las tablas por defecto en el esquema SYS.

Capitulo 8

Tiempo máximo estimado para esta práctica: 20 minutos.

1. ¿A qué se le conoce como DML y que instrucciones están incluidas dentro de este?

Es el Lenguaje de Manipulación de Datos, consiste en una serie de instrucciones para hacer modificaciones en los datos que se encuentran contenidos dentro de una tabla de datos. Las instrucciones que lo componen son INSERT, UPDATE, DELETE, TRUNCATE (aunque este es más de tipo DDL).

1. ¿Es posible eliminar registros de tablas con llaves foráneas? ¿Por qué ?

No. Esto se explica porque afecta la integridad de la base de datos, Oracle lo registra como una violación a dicha integridad.

1. Define que es una transacción

Es un grupo de instrucciones que se ejecuta en la base de datos y consta de una sentencia DDL con la que se realiza el cambio en el dato y se garantiza la integridad de la base de datos, una sentencia DDL y por último una sentencia DCL.

1. ¿Qué comandos hacen que una transacción finalice satisfactoriamente?

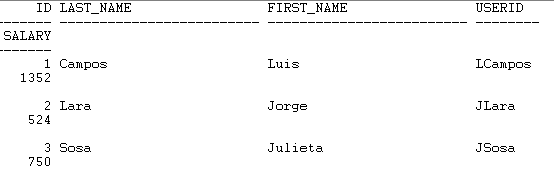
Los comandos COMMIT o ROLLBACK ejecutan una transacción de forma exitosa. Por otro lado un comando DDL o DCL también ejecuta implícitamente el comando COMMIT.

1. Corre el script llamado lab\_08\_05.sql
2. Con el script anterior se creo una tabla llamada MY\_EMPLOYEE, agregar a dicha tablas los siguientes datos:

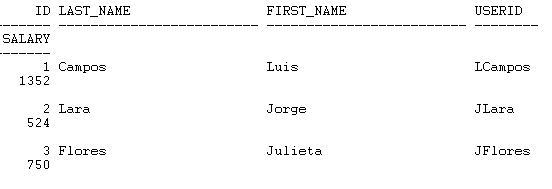
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Last\_name | First\_name | Userid | salary |
| 1 | Campos | Luis | Lcampos | 1352 |
| 2 | Lara | Jorge | Jlara | 524 |
| 3 | Sosa | Julieta | Jsosa | 750 |

Nota: Observa que el userid se compone de la primera letra del first name concatenada con el last name, hacer mediante las funciones ya conocidas. Hacer dinámica la sentencia para ingresar dichos registros variables (&).

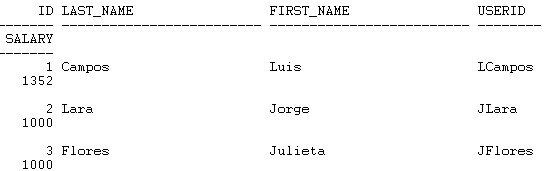
1. Confirma tu transacción de insertar
2. Hacer permanente dichos cambios en la tabla MY\_EMPLOYEE.



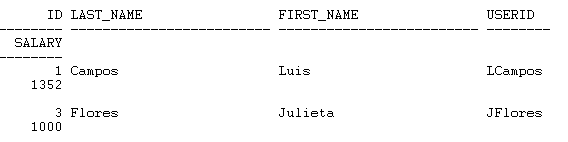
1. Cambia el last name del empleado 3 a Flores



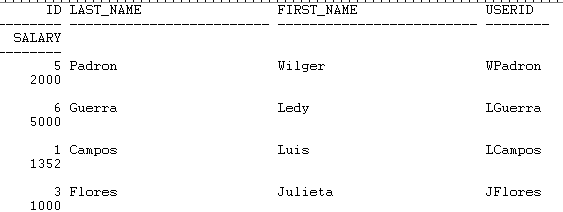
1. Cambia el salario a 1000 para todos los empleados que tenga salario menor a 800



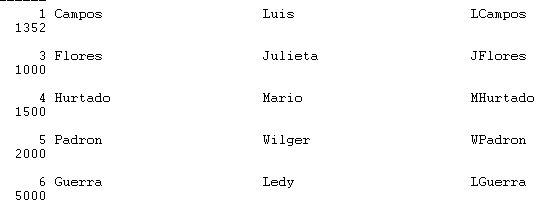
1. Borra a Jorge de la tabla.
2. Verifica y confirma tus cambios.



1. Agrega tres empleados cuales tu quieras verificando que no se repitan



1. Marca un punto llamado SAVEPOINT prueba.
2. Borra todo de la tabla y verifica que este vacía.
3. Finalmente da un rollback al savepoint prueba y observa lo que hay en la tabla MY\_EMPLOYEE.



1. **Práctica**

Con el desarrollo de esta práctica se alcanzarán los siguientes objetivos:

* Crear nuevas tablas
* Verificar que las tablas existen
* Borrar tablas

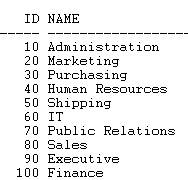
Tiempo máximo estimado para esta práctica: 15 minutos.

1. Crear una tabla llamada DEPT basándose en:

|  |  |  |
| --- | --- | --- |
| Nombre de columna | ID | NAME |
| Llave primaria | Primary key |  |
| Null/unique |  |  |
| Llave foranea tabla |  |  |
| Llave foranea columna |  |  |
| Tipo de dato | NUMBER | VARCHAR2 |
| Longitud | 7 | 25 |

Guardar el script en lab\_09\_01.sql y confirmar la creación.

1. Llenar la tabla DEPT con los datos de la tabla DEPARTMENTS, incluye solo las columnas que necesitas



1. Crea otra tabla llamada EMP en base a:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre de columna | ID | Last\_name | First\_name | Dep.\_id |
| Llave primaria |  |  |  |  |
| Null/unique |  |  |  |  |
| Llave foranea tabla |  |  |  | DEPT |
| Llave foranea columna |  |  |  | ID |
| Tipo de dato | NUMBER | VARCHAR2 | VARCHAR2 | NUMBER |
| Longitud | 7 | 25 | 25 | 7 |

Guardar el script en lab\_09\_03.sql y confirmar la creación.



1. Crear una tabla llamada EMPLOYEES2 basada en la tabla EMPLOYEES incluir solo el numero, nombre, salario del empleado así como el numero de departamento. Nombrar las columnas como ID, FIRST\_NAME, LAST\_NAME, SALARY, y DEPT\_ID respectivamente.



1. Borrar la tabla EMP



1. **Práctica**

Con el desarrollo de esta práctica se alcanzarán los siguientes objetivos:

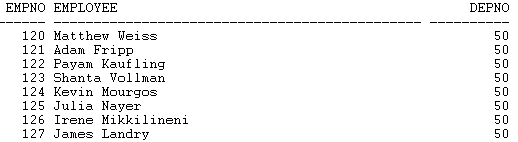
* Crear vistas simples y complejas
* Crear vistas con un constraint check
* Crear secuencias
* Usar secuencias
* Crear índices no únicos
* Crear sinónimos

Tiempo máximo estimado para esta práctica: 20 minutos.

1. Crear una vista llamada EMPLOYEES\_VU basada en el número de empleado, nombre del empleado y número de departamento de la tabla EMPLOYEES. En el nombre del empleado el encabezado debe ser EMPLOYEE, Desplegar el contenido de la vista creada. Guardar en lab\_10\_01.sql.
2. Usando la vista anterior, escribir un query que muestre el nombre del empleado y el número de departamento. Ojo usar la vista no la tabla.



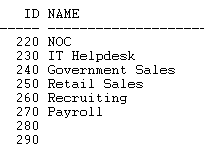
1. Crear una vista llamada DEPT50 que contenga el número de empleado, su nombre y el número de departamento, exclusivamente para los empleados del departamento 50 renombrar las columnas como EMPNO, EMPLOYEE y DEPTNO.
2. Desplegar la estructura de la vista y correr la vista



1. Recuerdas la tabla DEPT de la practica pasada, ahora se necesita una secuencia que se utilizará como llave primaria de esa tabla, le secuencia debe empezar en 200 y como valor máximo 1000, La secuencia debe incrementarse de 10 en 10 se debe llamar DEPT\_ID\_SEG. Guardar en lab\_10\_05.sql.

NOTA: AQUÍ MODIFIQUE EL COMIENZO DE LA SECUENCIA, DEBIDO A QUE MI LLAVE PRIMARIA TERMINA EN 270, POR LO TANTO NO ME DEJABA INSERTAR DOS REGISTROS EN LA TABLA DEPT POR TEMAS DE INTEGRIDAD. LA MODIFICACIÓN LA HICE EN UN SCRIPT APARTE.

1. Para probar la secuencia, escribir un query que inserte dos registros a la tabla DEPT, el nombre de dicho script será lab\_10\_06.sql. Asegurarse de que la secuencia que creaste se utilizo.



1. Crea un índice no único sobre la columna NAME de la tabla DEPT.



1. Crea un sinónimo para la tabla de EMPLOYEES llámala EMP.



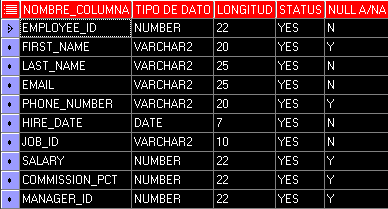
1. **Práctica**

Con el desarrollo de esta práctica se alcanzarán los siguientes objetivos:

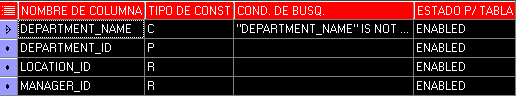
* Realizar queries sobre las vistas del diccionario de datos para obtener información de tablas y columnas.
* Obtener información sobre los constraints.
* Obtener información sobre vistas, secuencias.
* Agregar comentarios a una tabla.

Tiempo máximo estimado para esta práctica: 15 minutos.

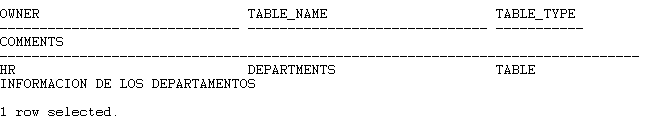
1. Para una tabla especifica, crear un script que reporte el nombre de las columnas, su tipo de dato y la longitud del tipo de dato, el estado de la columna, así como una bandera que nos indique si los valores nulos son permitidos o no, el nombre de la tabla debe ser ingresado desde el prompt de usuario, dar el apropiado alias a las columnas. Salvarlo como lab\_11\_01.sql.



1. Crear un script que reporte el nombre de la columna, nombre del constraint, tipo de constraint, la búsqueda de condición y el estado para esa tabla en específico. Para obtener toda esta información debes hacer un join entre las tablas de USER\_CONSTRAINTS y USER\_CONS\_COLUMNS. El nombre de la tabla debe ser proporcionado por el usuario. Guardar como lab\_11\_02.sql.



1. Agrega comentarios a la tabla de DEPARTMENTS, Después realiza una consulta a la vista de USER\_TAB\_COMMENTS para verificar que el comentario esta presente.



1. Se necesita determinar los nombres y definiciones de todas las vistas de tu esquema. Crea un reporte que despliegue el nombre de la vista, texto de la vista de USER\_VIEWS del diccionario de datos.





1. Encuentra los nombres de tus secuencias, escribe un query que despliegue la siguiente información sobre las secuencias: nombre, valor máximo, incremento, y el último número. Sálvalo como lab\_11\_05.sql.



* 1. **Consulta de información del diccionario de datos**

Los datos existentes en las tablas del diccionario de datos como los datos de las tablas del usuario pueden ser obtenidos mediante declaraciones SQL, dependiendo de los privilegios que tenga cada usuario para las vistas del diccionario.

El diccionario es consultado por los usuarios valiéndose de sus vistas, para lo cual existen 4 categorías de vistas las cuales son:

*USER\_* Contiene objetos cuyo dueño es el usuario. Por ejemplo las vistas con este prefijo permite al usuario desplegar información acerca de las tablas creadas por el usuario y los privilegios asignados por el usuario.

*ALL\_* Permite mostrar información de los objetos en los cuales el usuario tiene privilegios asignados, además de los suyos propios.

*DBA\_* Permite a usuarios con privilegios de BDA accesar algún objeto en la base de datos.

*V$* Despliega información del funcionamiento del servidor, de bloqueos, etc….inicialmente solamente permitido a DBA´s.

Algunas vistas del diccionario de datos no usan los prefijos listados arriba. Estas incluyen sinónimos para vistas con nombre grandes.

*DICTIONARY*  Lista todas las tablas, vistas y sinónimos.

*TABLE\_PRIVILEGES* Privilegios sobre objetos para el cual el usuario es asignador, el asignado o el propietario.

*IND* Es un sinónimo para USER\_INDEXES.