ADMINISTRACIÓN DE BASE DE DATOS.

Tema Nº2:Diccionario de Datos.

Indicador de logro Nº2:Diseña modelos de datos para el uso del diccionario de datos, mediante SQL developer.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº2:**

Diccionario de Datos.

**Subtema 2.1:**

Presentar modelos de datos que incorpore reglas o restricciones.

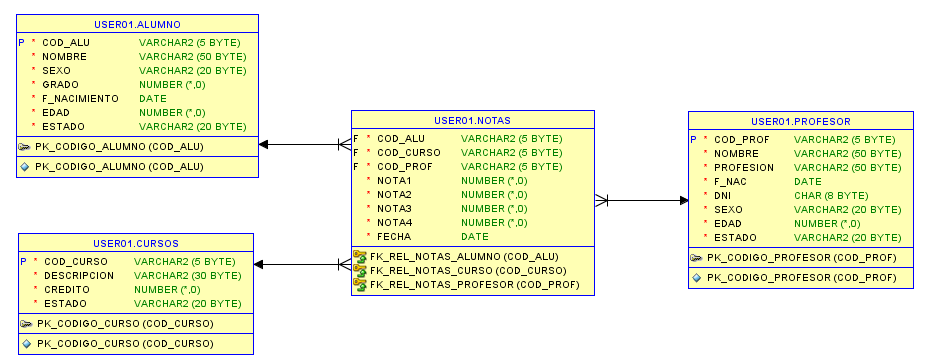
**Restricciones**

Las restricciones de los datos son reglas que se imponen para asegurarnos que los datos cumplen con una serie de condiciones predefinidas para cada tabla. Estas restricciones ayudan a conseguir la integridad referencial: todas las referencias dentro de una Base de Datos son válidas y todas las restricciones se cumplen.

Las restricciones se van a definir acompañadas por un nombre, lo que permitirá activarlas o desactivarlas según sea el caso; o también mezcladas en las definiciones de las columnas de la tabla. A continuación, vamos a describir cada una de las restricciones mencionadas.

|  |  |
| --- | --- |
| **Restricción** | **Descripción** |
| **DEFAULT** | Establece el valor por defecto para esa columna, si no se le asigna ninguno. |
| **CHECK** | Comprueba que se cumpla una condición determinada al rellenar esa columna. Esta condición sólo debe estar construida con columnas de esta misma tabla. |
| **FOREIGN KEY** | El contenido de esta columna será uno de los valores contenidos en una columna de otra tabla maestra. Esta columna marcada como clave foránea o ajena puede ser NULL. No hay límite en el número de claves foráneas. La clave ajena puede ser otra columna de la misma tabla. Se puede forzar cuando una fila de la tabla maestra sea borrada, todas las filas de la tabla detalle cuya clave ajena coincida con la clave borrada se borren también (borrado en cascada). Esto se consigue añadiendo la cláusula ON DELETE CASCADE en la definición de la clave ajena. |
| **NOT NULL** | Establece la obligatoriedad de que esta columna tenga un valor no nulo. Se debe especificar junto a la columna a la que afecta. Los valores nulos no ocupan espacio, y son distintos a 0 y al espacio en blanco. Hay que tener cuidado con los valores nulos en las operaciones, ya que e1 \* NULL es igual a NULL. |
| **PRIMARY KEY** | Conjunto de columnas que forman la clave primaria de esa tabla. Se comporta como única y obligatoria sin necesidad de explicitarlo. Sólo puede existir una clave primaria por tabla. Puede ser referenciada como clave ajena por otras tablas. Crea un índice automáticamente cuando se habilita o se crea esta restricción. |
| **UNIQUE** | Evita valores repetidos en una columna, admitiendo valores nulos. Oracle crea un índice automáticamente cuando se habilita esta restricción y lo borra al deshabilitarse. |
| **NULL** | Cuyo significado es nulidad (Sin Valor), es decir, no tener valor, ejemplo: un billete de 20 soles falso, ya que este no tiene valor. |
| **NOT NULL** | Cuyo significado es (Con valor) es decir, este dato deberá valer algo obligatoriamente al momento de acceder datos a la tabla. Ejemplo: un billete de 20 soles verdadero (ya que tiene valor) |

**Ejemplo: Se tiene el siguiente modelo relacional.**



Se pide crear el Script de base de datos, usando el comando CREATE TABLE, cada tabla tiene sus restricciones y son las siguientes:

**En la Tabla ALUMNO considere las siguientes restricciones CONSTRAINT a nivel de tabla:**

1. Establecer el Primary Key en la mencionada tabla.
2. Nombres de alumnos en Mayúsculas.
3. En el campo Sexo o Género se debe considerar únicamente ‘M’ o ‘F’.
4. En el Campo Grado considerar del 1 hasta el 6.
5. En el campo Edad, considere alumnos entre 10 a 22 años.
6. El estado del alumno puede ser: 'ACTIVO', 'BLOQUEADO', 'SUSPENDIDO', 'RETIRADO'.

**En la Tabla CURSOS considere las siguientes restricciones:**

1. Establecer el Primary Key en la mencionada tabla.
2. Nombre de las asignaturas en Mayúsculas.
3. Los Créditos se establecen en: 2, 3, 4, 6.
4. Los Estados se establecen como: 'ACTIVO', 'BLOQUEADO'

**En la Tabla PROFESOR considere las siguientes restricciones CONSTRAINT a nivel de tabla:**

1. Establecer el Primary Key en la mencionada tabla.
2. Nombre del Profesor en mayúsculas.
3. Profesión del Profesor en Mayúsculas.
4. El Sexo del Profesor se establece en: 'Masculino', 'Femenino'.
5. La Edad del Profesor debe estar comprendida entre 25 y 60 años
6. El Estado del Profesor se establece en: 'ACTIVO', 'BLOQUEADO', 'SUSPENDIDO', 'RETIRADO'.

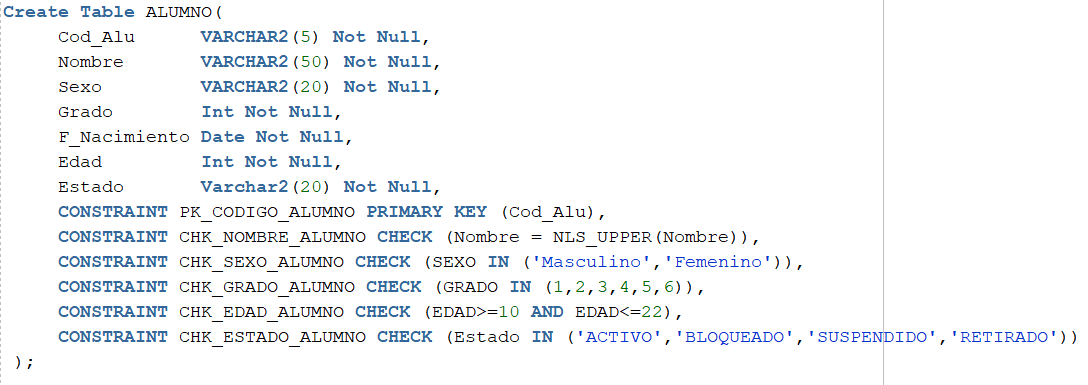
**En la Tabla NOTAS considere las siguientes restricciones CONSTRAINT a nivel de tabla:**

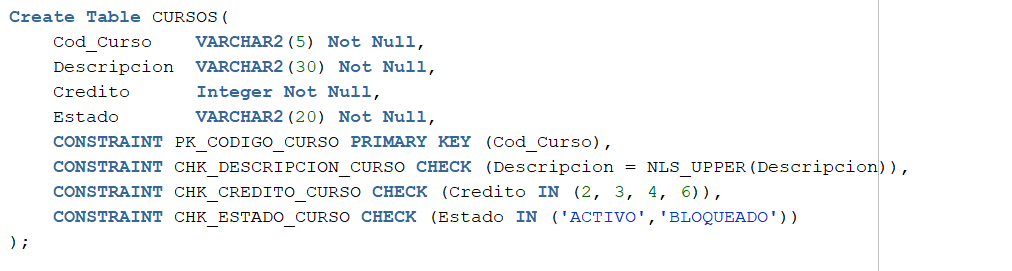
1. Todas las notas deben estar comprendidas entre 0 y 20.
2. Procurar generar las relaciones con las tablas de origen (Padre)

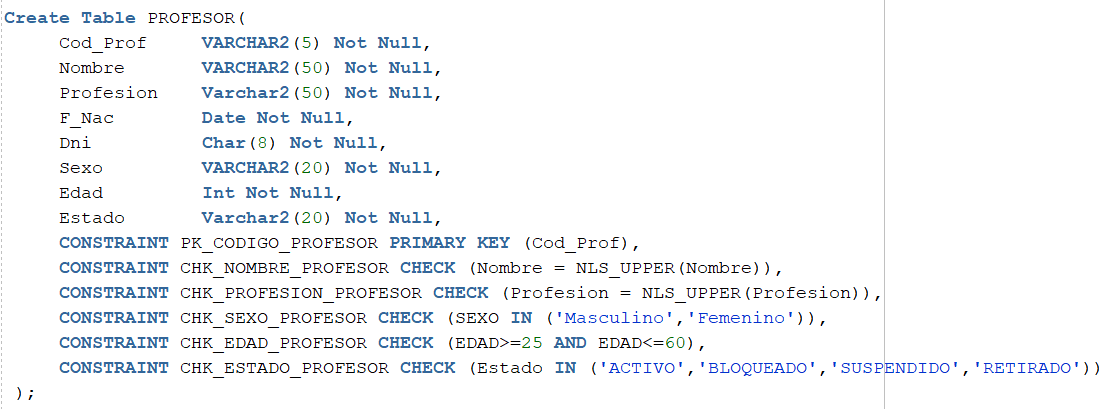
**SOLUCION:**

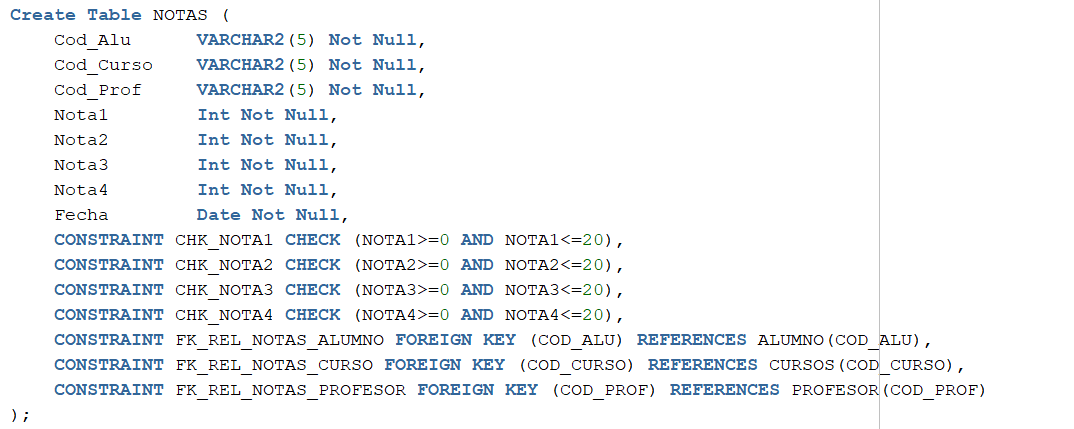
**PASO 1:**

Ingrese a la herramienta Cliente SQL Developer, usando como usuario a: USER01 y contraseña: 123456 Y redacte el siguiente código, luego grábelo con el nombre: **Script\_Validaciones.sql**.



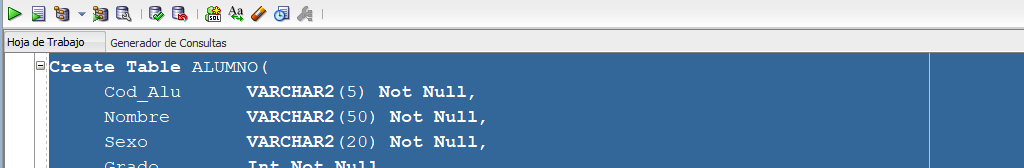






**PASO 2:**

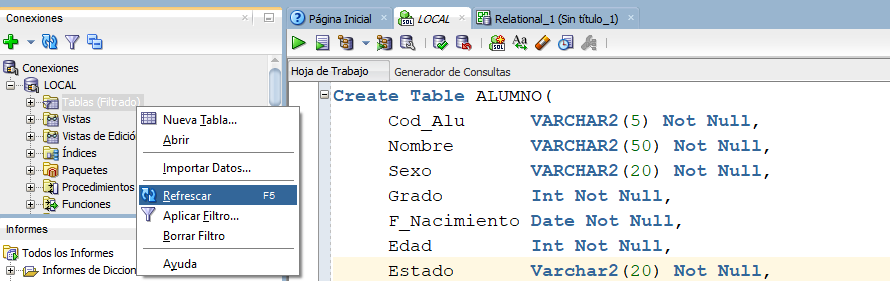
Seleccione cada bloque código de creación de tabla, ejecútelo usando el botón de color verde el cual tiene la forma de un PLAY.

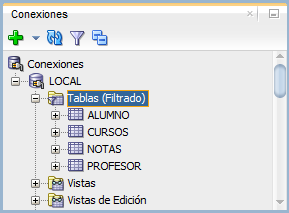




**PASO 3:**

Refrescar la carpeta Tablas (filtrado), hacer Click derecho sobre la opción “Tablas” y seleccionar **Refrescar**. Esto se hace para mostrar las tablas creadas.





**FIN DEL PROCESO DE RESTRICCIONES.**

**CONSTRAINT.**

**Subtema 2.2:**

Implementar Índices, Secuencias y Sinónimos en las tablas creadas con un usuario de Oracle

**INDICES:**

Los índices son elementos que ayudan a las tablas, a obtener acceso más rápido a los datos contenidos en una tabla.

Las consultas que se harán mediante un índice. Sobre una tabla, reduce notablemente la búsqueda de registros aumentando el rendimiento.

**Otros Conceptos:**

* Es un objeto de esquema.
* Oracle Server lo utiliza para acelerar la recuperación de filas utilizando un puntero.
* Puede reducir la E/S de disco mediante un método de ruta de acceso rápido para encontrar datos.
* Es independiente de su tabla indexada.
* Oracle Server lo utiliza y mantiene automáticamente.

**¿Cómo se crean los índices?**

* **Automáticamente**: Un índice único se crea automáticamente al definir una restricción UNIQUE o PRIMARY KEY en una definición de tabla.
* **Manualmente**: Los usuarios pueden crear índices no únicos en columnas para acelerar el acceso a las filas.

SINTAXIS:

**CREATE INDEX** Nombre\_index **ON <Tabla> (column [, column]...);**

EJEMPLO:

Mejore la velocidad de acceso de consulta a la columna **Nombre** de la tabla CLIENTE.

**CREATE INDEX** Nombre\_Cliente\_idx **ON Cliente (Nombre);**

**Index created.**

**Debe crear un índice si:**

* Una columna contiene un amplio rango de valores.
* Una columna contiene un gran número de valores nulos.
* Una o más columnas se utilizan juntas frecuentemente en una cláusula WHERE o en una condición de unión.
* La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2-4 por ciento de las filas.

**Normalmente no merece la pena crear un índice si:**

* La tabla es pequeña.
* Las columnas no se suelen utilizar como condición en la consulta.
* Se espera que la mayoría de las consultas recuperen más del 2-4 por ciento de las filas de la tabla.
* La tabla se actualiza frecuentemente.
* Se hace referencia a las columnas indexadas como parte de una expresión.

**Eliminación de un índice:**

* Elimine un índice del diccionario de datos utilizando el comando DROP INDEX.

**DROP INDEX** Nombre\_index**;**

* Elimine el índice **Nombre\_Cliente\_idx** del diccionario de datos.

**DROP INDEX** Nombre\_Cliente\_idx**;**

**Index dropped.**

Para borrar un índice, debe ser el propietario del mismo o tener el privilegio DROP ANY INDEX.

**SECUENCIAS:**

**Una secuencia:**

* Genera automáticamente números únicos.
* Es un objeto compartible.
* Normalmente se utiliza para crear un valor de clave primaria.
* Sustituye al código de aplicación.
* Acelera la eficacia del acceso a los valores de secuencia al almacenarse en memoria caché.

**SINTAXIS:**

**CREATE SEQUENCE** Nombre\_sequence

[**INCREMENT BY** n]

[**START WITH** n]

[**{MAXVALUE n | NOMAXVALUE}**]

[**{MINVALUE n | NOMINVALUE}**]

[**{CYCLE | NOCYCLE}**]

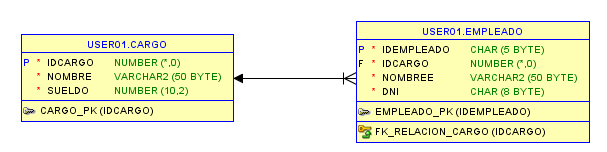
[**{CACHE n | NOCACHE}**];

**PARAMETROS DE SECUENCIAS:**

* **MINVALUE** es el valor mínimo que tendrá la secuencia
* **MAXVALUE** es el valor máximo que tendrá la secuencia (se puede obviar y por defecto se le asignará 999999999999999999999999999)
* **START WITH** es el valor con el que empezará la secuencia
* **INCREMENT BY** es el valor con el que se incrementará la secuencia
* **CACHE / NOCACHE** el uso de CACHE permite indicar cuantos valores queremos que sean guardados en memoria para un acceso más rápido. No se recomienda su uso ya que podría generar inconsistencia.
* **CYCLE** indica que, cuando la secuencia llegue a máximo valor (valor de "maxvalue") se reinicie, comenzando con el mínimo valor ("minvalue") nuevamente, es decir, la secuencia vuelve a utilizar los números.

**EJEMPLO DE LA CREACION DE SECUENCIA:**

Crear una secuencia llamada CARGO\_ID\_SEQ para utilizarla en la clave primaria de la tabla **CARGO**. Esta incrementa de 10 en 10, inicia con un valor de 280, con un valor mínimo de 0, y máximo de 1000, No utilice la opción CYCLE para valores de clave primaria.



**SOLUCIÓN:**

**CREATE SEQUENCE CARGO\_ID\_SEQ**

**INCREMENT BY 10**

**START WITH 280**

**MINVALUE 0**

**MAXVALUE 1000**

**NOCACHE**

**NOCYCLE;**

**Sequence created.**

**Pseudocolumnas NEXTVAL y CURRVAL.**

* **NEXTVAL** devuelve el siguiente valor de secuencia disponible. Devuelve un valor único cada vez que se hace referencia a ella, incluso para usuarios distintos.
* **CURRVAL** obtiene el valor de la secuencia actual.

**NEXTVAL** se debe emitir para dicha secuencia antes de que CURRVAL contenga un valor.

**Eliminando la Secuencia.**

* Elimine una secuencia del diccionario de datos utilizando la sentencia DROP SEQUENCE. Una vez que se ha eliminado, ya no se puede hacer referencia a la secuencia.

**SINTAXIS:**

**DROP SEQUENCE** Nombre\_Secuencia**;**

EJEMPLO: Elimine la secuencia **CARGO\_ID\_SEQ**.

**DROP SEQUENCE** CARGO\_ID\_SEQ**;**

**Sequence dropped.**

**SINÓNIMOS:**

Simplifique el acceso a los objetos mediante la creación de un sinónimo (un objeto con otro nombre). Con los sinónimos, puede:

* Facilitar la referencia a una tabla propiedad de otro usuario
* Reducir nombres largos de objeto.

**SINTAXIS:**

**CREATE [PUBLIC] SYNONYM** Nombre\_sinónimo **FOR Object;**

**EJEMPLO:**

Crear un sinónimo el cual permita abreviar el uso de la tabla CLIENTE, use, por ejemplo: CLI.

**CREATE SYNONYM** CLI **FOR** CLIENTE**;**

**ELIMINAR UN SINÓNIMO:**

Use la sentencia: DROP SYNONYM

**DROP SYNONYM** Nombre\_sinónimo**;**

**EJEMPLO:**

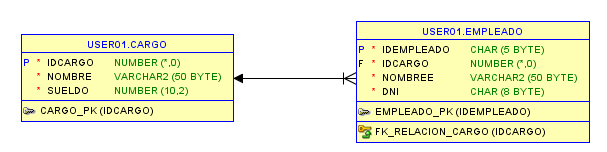
Eliminar el sinónimo CLI de nuestro diccionario de datos.

**DROP SYNONYM** CLI**;**

**Synonym dropped.**

**EJEMPLO GENERAL DE INDICES, SECUENCIAS Y SINONIMOS:**

Se tiene el siguiente Modelo Relacional:



**Implementar:**

1. Un índice para el campo nombre de la tabla Empleado.
2. Genere una secuencia para la tabla Cargo, específicamente sobre la columna **IdCargo**, dado que es una columna numérica, este debe incrementarse 1 en 1, valor mínimo de 100, máximo 10000, dado que es una clave no puede volver a generarse los valores (NO USAR CYCLE).
3. Asigne sinónimos para las tablas CARGO🡪CAR y EMPLEADO 🡪 EMP.

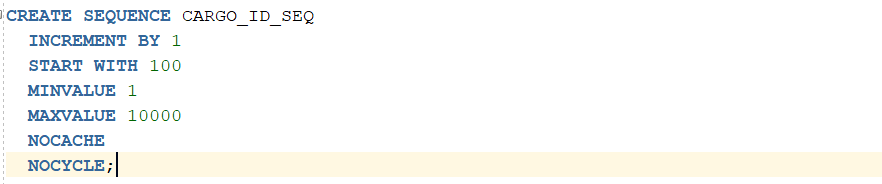
**SOLUCIÓN:**

Los comandos deben ser ejecutados en bloques.

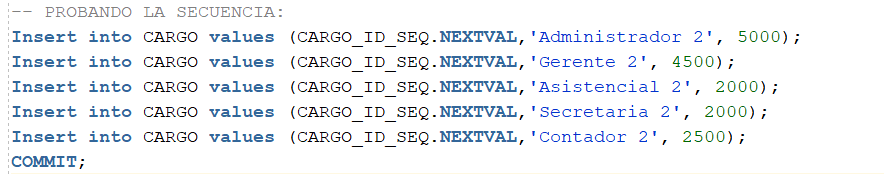
Creando un índice en la tabla Empleado.



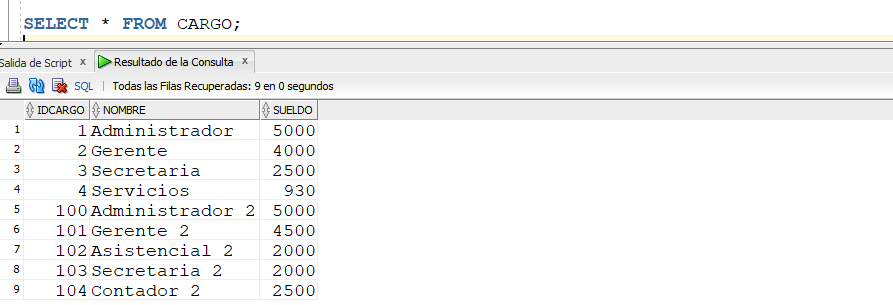
Creando la Secuencia para la tabla Cargo.



Probando la secuencia:

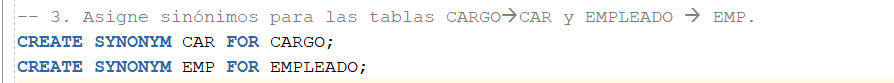


Verificando el contenido de la tabla Cargo mediante una consulta.



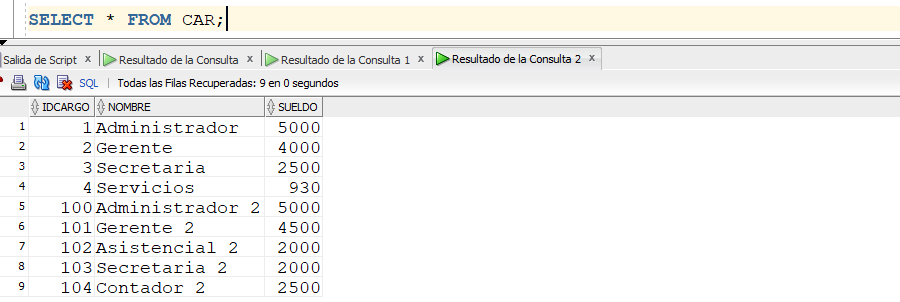
Los códigos se generan desde 100, de uno en uno.

Asignando Sinónimos a las tablas CARGO y EMPLEADO.

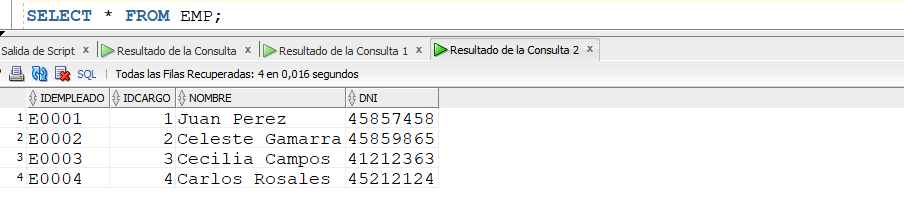


**Probando los sinónimos:**

Mediante una consulta al sinónimo CAR que representa a la tabla Cargo.

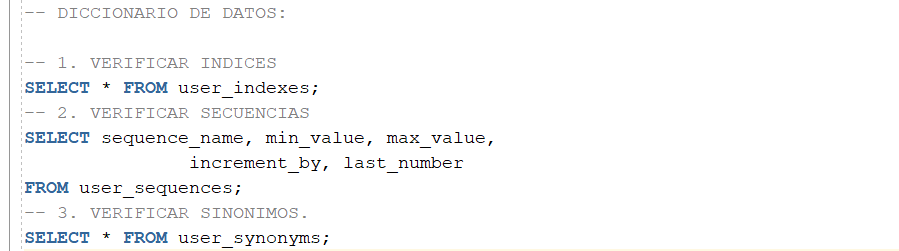


Mediante una consulta al sinónimo EMP que representa a la tabla Empleado.



DICCIONARIO DE DATOS:

Si ejecuta estas instrucciones, podrá observar todas las restricciones, índices, secuencia, sinónimos, creados en esta sesión de clase.



FIN DEL EJERCICIO.

**Subtema 2.3:**

Utiliza el gestor de base de datos ORACLE para seleccionar información a través de los objetos del diccionario de datos.

**Consulta del diccionario de Datos.**

* Es fundamental para cualquier base de datos
* Describe la base de datos y sus objetos
* Contiene tablas y vistas de sólo lectura
* Es propiedad del usuario SYS
* Lo mantiene Oracle Server
* Se accede a él con SELECT

**Información en Diccionario de Datos.**

* Estructuras lógicas y físicas de la base de datos
* Usuarios registrados en la base de datos
* Definiciones e integridad de todos los esquemas
* Información de Auditoría
* Información de espacio utilizado por los objetos
* Roles y Privilegios de los usuarios

**Estructura del Diccionario.**

**Tablas Base**

* Almacena información de la base de datos
* Solo Oracle Server puede leer y escribir estas tablas.
* Se crean al momento de crear la base de datos

**Vistas**

* Utilizadas para mostrar de forma resumida la información contenida en las tablas base.
* Se acceden a ellas a través de sinónimos públicos.
* Ver todas las vistas del diccionario de datos

**Sql> SELECT \* FROM Dictionary;**

* Ver las columnas de las vistas del diccionario de datos

**Sql> SELECT \* FROM Dict\_columns;**

**Ejemplos:**

* Ver objetos propiedad del usuario

**Sql> SELECT \* FROM User\_Objects;**

* Ver tablespaces de la base de datos

**Sql> SELECT \* FROM Dba\_tablespaces;**

* Vea tipos de objeto distintos propiedad del usuario

**Sql> SELECT table\_name FROM user\_tables;**

* Vea tablas, vistas, sinónimos y secuencias propiedad del usuario.

**Sql> SELECT DISTINCT object\_type FROM user\_objects;**

* Vea los nombres de tablas propiedad del usuario.

**Sql> SELECT \* FROM user\_catalog;**

* Se pueden ver los archivos control file

**Sql> SELECT \* FROM V$CONTROLFILE;**

* Se pueden ver los constraints definidos para una tabla específica en la tabla

**Sql> SELECT \* FROM USER\_CONSTRAINTS;**

* Tablespace agrupada por tipo de segmento y ordenada por tablespace.

**Sql> Select segment\_type, tablespace\_name, count(\*) from dba\_segments**

**group by segment\_type,tablespace\_name order by 2 desc**

**Las Vistas tienen Categorías:**

* DBA: Contiene información de todos los objetos que hay en todos los esquemas de la base de datos.
* ALL: Contiene información de todos los objetos a los que puede acceder el usuario en sesión.
* USER: Contiene información de objetos de propiedad del usuario en sesión.

**Uso del Diccionario de Datos:**

* Oracle consulta el diccionario de datos para hallar la información acerca de los usuarios, esquemas y estructuras almacenadas.
* Oracle modifica el diccionario de datos cada vez que una instrucción DDL es usada.
* Cualquier usuario puede utilizar el diccionario de datos.
* Vista maestra que es el punto de partida para conocer los objetos de la BD.
* Campos:
  + OWNER: Dueño del objeto
  + OBJECT\_NAME: el nombre del objeto
  + OBJETC\_TYPE: el tipo de objeto: tabla, índice, package, etc.
  + CREATED: fecha de creación del objeto.

**Ejemplos:**

* Para las tablas del usuario scott
* SELECT TABLE\_NAME FROM ALL\_TABLES WHERE OWNER ='SCOTT';
* Usuarios conectados en este momento
* SELECT COUNT(\*), USERNAME FROM V$SESSION GROUP BY USERNAME;

|  |  |
| --- | --- |
| INFORMACION | VISTAS CORRESPONDIENTES |
| Errores de compilación | all\_errors, dba\_errors, user\_errors |
| Trabajos en la BD | all\_jobs, dba\_jobs,user\_jobs |
| Usuarios | all\_users, dba\_users, user\_users |
| Objetos de la BD: Tablas, programas, vistas secuencias, índices, etc. | all\_objects, dba\_objects, user\_objects |
| Dependencias entre los objetos (PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY) | all\_dependencies, dba\_dependencies, user\_dependencies |
| Tablas de la base de datos | All\_tables, dba\_tables, user\_tables |
| Vistas en la BD | All\_views, dba\_views, user\_views |
| Secuencias | All\_sequences, dba\_sequences, user\_sequences |
| Nombres de tablas, vistas, secuencias, sinónimos | All\_catalog, dba\_catalog, user\_catalog |
| Columnas de las tablas y vistas | All\_tab\_columns, dba\_tabs\_colunms, user\_tab\_colunms |
| Comentarios sobre tablas o columnas | All\_tab\_comments, dba\_tab\_comments, user\_tab\_comments, All\_col\_comments, dba\_col\_comments, user\_col\_comments |

|  |  |
| --- | --- |
| INFORMACION | VISTAS CORRESPONDIENTES |
| Restricciones de integridad | All\_constraints, dba\_constraints, user\_constrains, all\_cons\_columns, dba\_cons\_columns, user\_cons\_columns |
| Triggers en la BD | All\_triggers, dba\_triggers, user\_triggers |
| Uso de las columnas en un trigger | All\_trigger\_cols, dba\_trigger\_cols, user\_trigger\_cols |

**Tablas y vistas:**

* ALL\_ALL\_TABLES
* DBA\_ALL\_TABLES
* USER\_ALL\_TABLES
* ALL\_COL\_COMMENTS
* DBA\_COL\_COMMENTS
* USER\_COL\_COMMENTS
* ALL\_PARTIAL\_DROP\_TABS
* DBA\_PARTIAL\_DROP\_TABS
* USER\_PARTIAL\_DROP\_TABS
* ALL\_REFS
* DBA\_REFS
* USER\_REFS
* ALL\_TAB\_COLUMNS
* DBA\_TAB\_COLUMNS
* USER\_TAB\_COLUMNS
* ALL\_TAB\_COMMENTS
* DBA\_TAB\_COMMENTS
* USER\_TAB\_COMMENTS
* ALL\_TABLES
* DBA\_TABLES
* USER\_TABLES
* ALL\_UNUSED\_COL\_TABS
* DBA\_UNUSED\_COL\_TABS
* USER\_UNUSED\_COL\_TABS
* ALL\_UPDATABLE\_COLUMNS
* DBA\_UPDATABLE\_COLUMNS
* USER\_UPDATABLE\_COLUMNS
* ALL\_VIEWS
* DBA\_VIEWS
* USER\_VIEWS

**Secuencias**

* ALL\_SEQUENCES
* DBA\_SEQUENCES
* USER\_SEQUENCES

**Sinónimos**

* ALL\_SYNONYMS
* DBA\_ SYNONYMS
* USER\_ SYNONYMS

**Índices**

* ALL\_IND\_COLUMNS
* DBA\_IND\_COLUMNS
* USER\_IND\_COLUMNS
* ALL\_IND\_EXPRESSIONS
* DBA\_IND\_EXPRESSIONS
* USER\_IND\_EXPRESSIONS
* ALL\_INDEXES
* DBA\_INDEXES
* USER\_INDEXES
* ALL\_INDEXTYPES
* DBA\_INDEXTYPES
* USER\_INDEXTYPES
* ALL\_INDEXTYPE\_OPERATORS
* DBA\_INDEXTYPE\_OPERATORS
* USER\_NDEXTYPE\_OPERATORS

**Sub Programas**

* ALL\_ARGUMENTS
* USER\_ARGUMENTS

**Disparadores**

* ALL\_INTERNALS\_TRIGGERS
* DBA\_INTERNAL\_TRIGGERS
* USER: INTERNAL\_TRIGGERS
* ALL\_TRIGGERS
* DBA\_TRIGGERS
* USER\_TRIGGERS
* ALL\_TRIGGER\_COLS
* DBA\_TRIGGER\_COLS
* USER\_TRIGGER\_COLS
* Código fuente y errores de compilación
* ALL\_ERRORS
* DBA\_ERRORS
* USER\_ERRORS
* ALL\_SOURCE
* DBA\_SOURCE
* USER\_SOURCE

**Restricciones**

* ALL\_CONS\_COLUMNS
* DBA\_CONS\_COLUMNS
* USER\_CONS\_COLUMNS
* ALL\_CONSTRAINTS
* DBA\_CONSTRAINTS
* USER\_CONSTRAINTS

**Privilegios y concesiones**

* ALL\_COL\_PRIVS
* DBA\_COL\_PRIVS
* USER\_COL\_PRIVS
* ALL\_COL\_PRIVS\_MADE
* USER\_COL\_PRIVS\_MADE
* ALL\_COL\_PRIVS\_RECD
* USER\_COL\_PRIVS\_RECD

**Privilegios de Tablas**

* ALL\_TABS\_PRIVS
* DBA\_TABS\_PRIVS
* USER\_TABS\_PRIVS
* ALL\_TABS\_PRIVS\_MADE
* DBA\_TABS\_PRIVS\_MADE
* USER\_TABS\_PRIVS\_MADE

**Privilegios del sistema**

* DBA\_SYS\_PRIVS
* USER\_SYS\_PRIVS

**Dependencias**

* ALL\_DEPENDENCIES
* DBA\_DEPENDENCIES
* USER\_DEPEDENCIES

**Estructuras de a base de datos**

* V$TABLESPACE
* USER\_TABLESPACES
* DBA\_DATA\_FILES
* V$DATAFILE

**CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA**

El objetivo de este capítulo es comprender las potencialidades de oracle al registrar cada acción que el usuario realiza, es decir desde que se crean tablas, constructores (validaciones), índices, secuencias, sinónimos, entre otros y llevar un control de los componentes recientemente creados por el usuario de manera exitosa.

El estudiante maneja la información de la base de datos Oracle y configura componentes que quedaran automáticamente registrados en un diccionario de la base de datos, mediante las herramientas SQL \*PLUS y PL/SQL, gestionando una instancia de base de datos.

**ACTIVIDAD VIRTUAL**

Revisar y analizar el material presentado del Tema 02.

De acuerdo al material presentado del Tema 02, Responda a las siguientes Preguntas:

1. ¿Qué es un Modelo Relacional?
2. ¿Qué es una Restricción?
3. Defina: CONSTRAINT, UNIQUE, CHECK, IN, DEFAULT
4. Qué entiende Ud. ¿Por Diccionario de datos?
5. ¿Es importante un Diccionario de datos?
6. Mencione Ud. Algunas tablas del diccionario de datos, las más importantes
7. Mediante el comando SELECT Muestre los Índices de una determinada tabla.
8. Mediante el comando SELECT Muestre las Secuencias de una determinada tabla.
9. Mediante el comando SELECT Muestre los comentarios a nivel de columna de una determinada tabla.
10. Mediante el comando SELECT Muestre los TableSpace de la base de datos.
11. Mediante el comando SELECT ver tablas, vistas, sinónimos y secuencias propiedad del usuario.