

# 10

## **Uso de Sentencias DDL para Crear y Gestionar Tablas**

# Objetivos

Al finalizar esta lección, debería estar capacitado para lo siguiente:

- Clasificar los principales objetos de base de datos
- Revisar la estructura de la tabla
- Mostrar los tipos de dato disponibles para columnas
- Crear una tabla simple
- Explicar cómo crear restricciones en el momento de la creación de la tabla
- Describir el funcionamiento de los objetos de esquema

# Agenda

- Objetos de base de datos
  - Reglas de nomenclatura
- Sentencia `CREATE TABLE`:
  - Acceso a otras tablas de usuario
  - Opción `DEFAULT`
- Tipos de dato
- Visión general de restricciones: Restricciones `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK`
- Creación de una tabla mediante una subconsulta
- `ALTER TABLE`
  - Tablas de sólo lectura
- Sentencia `DROP TABLE`

# Objetos de Base de Datos

Objeto	Descripción
Tabla	Unidad básica de almacenamiento; compuesta por filas.
Vista	Representa de forma lógica subconjuntos de datos de una o más tablas.
Secuencia	Genera valores numéricos.
Índice	Mejora el rendimiento de algunas consultas.
Sinónimo	Ofrece nombres alternativos para un objeto

# Reglas de Nomenclatura

Los nombres de tabla y de columna deben:

- Empezar por una letra
- Tener entre 1 y 30 caracteres
- Contener sólo A–Z, a–z, 0–9, \_, \$ y #
- No deben ser un duplicado de otro nombre de objeto propiedad del mismo usuario
- No debe ser una palabra reservada del servidor de Oracle



# Agenda

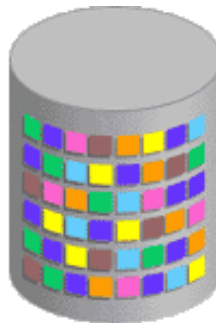
- Objetos de base de datos
  - Reglas de nomenclatura
- **Sentencia CREATE TABLE:**
  - Acceso a otras tablas de usuario
  - Opción DEFAULT
- Tipos de dato
- Visión general de restricciones: Restricciones NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK
- Creación de una tabla mediante una subconsulta
- ALTER TABLE
  - Tablas de sólo lectura
- Sentencia DROP TABLE

# Sentencia CREATE TABLE

- Debe tener:
  - El privilegio CREATE TABLE
  - Un área de almacenamiento

```
CREATE TABLE [schema.]table  
      (column datatype [DEFAULT expr] [, ...]);
```

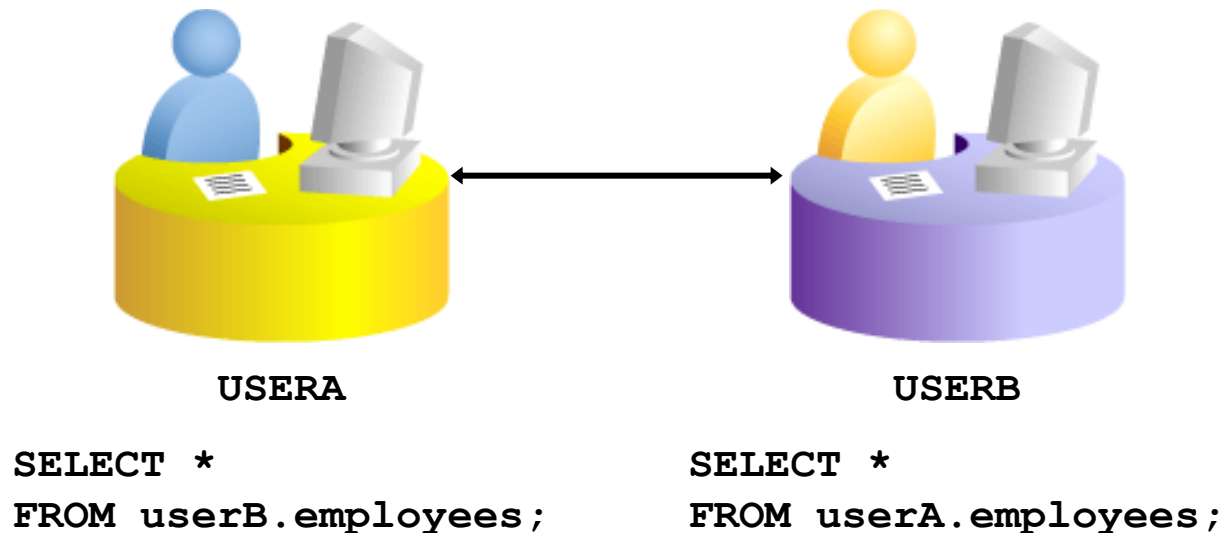
- Debe especificar:
  - El nombre de tabla
  - El nombre de columna, tipo de dato de columna y tamaño de columna





# Referencia a Tablas de Otro Usuario

- Las tablas que pertenecen a otros usuarios no se incluyen en el esquema del usuario.
- Debe utilizar el nombre del propietario como prefijo de dichas tablas.



# Opción DEFAULT

- Especificar un valor por defecto para una columna durante una inserción.

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Los valores literales, expresiones o funciones SQL son valores válidos.
- El nombre de otra columna o una pseudocolumna son valores no válidos.
- El tipo de dato por defecto debe coincidir con el tipo de dato de la columna.

```
CREATE TABLE hire_dates  
  (id          NUMBER(8) ,  
   hire date DATE DEFAULT SYSDATE);
```

```
CREATE TABLE succeeded.
```

# Creación de Tablas

- Crear una tabla:

```
CREATE TABLE dept
      (deptno      NUMBER(2) ,
        dname      VARCHAR2(14) ,
        loc        VARCHAR2(13) ,
        create_date DATE DEFAULT SYSDATE) ;
```

```
CREATE TABLE succeeded.
```

- Confirmar la creación de la tabla:

```
DESCRIBE dept
```

NAME	Null	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)
CREATE_DATE		DATE

4 rows selected

# Agenda

- Objetos de base de datos
  - Reglas de nomenclatura
- Sentencia CREATE TABLE:
  - Acceso a otras tablas de usuario
  - Opción DEFAULT
- Tipos de dato
- Visión general de restricciones: Restricciones NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK
- Creación de una tabla mediante una subconsulta
- ALTER TABLE
  - Tablas de sólo lectura
- Sentencia DROP TABLE

# Tipos de Dato

Tipo de Dato	Descripción
VARCHAR2 ( <i>size</i> )	Datos de caracteres de longitud variable
CHAR ( <i>size</i> )	Datos de caracteres de longitud fija
NUMBER ( <i>p</i> , <i>s</i> )	Datos numéricos de longitud variable
DATE	Valores de fecha y hora
LONG	Datos de caracteres de longitud variable (hasta 2 GB).
CLOB	Datos binarios (hasta 4 GB).
RAW and LONG RAW	Datos binarios raw
BLOB	Datos binarios (hasta 4 GB)
BFILE	Datos binarios almacenados en un archivo externo (hasta 4 GB)
ROWID	Sistema numérico de base -64 que representa la dirección única de una fila en su tabla correspondiente.

# Tipos de Dato de Fecha y Hora

Puede utilizar varios tipos de dato de fecha y hora:

Tipo de Dato	Descripción
TIMESTAMP	Fecha con segundos fraccionarios
INTERVAL YEAR TO MONTH	Almacenados como un intervalo de años y meses
INTERVAL DAY TO SECOND	Almacenado como un intervalo de días, horas, minutos y segundos



# Agenda

- Objetos de base de datos
  - Reglas de nomenclatura
- Sentencia `CREATE TABLE`:
  - Acceso a otras tablas de usuario
  - Opción `DEFAULT`
- Tipos de dato
- **Visión general de restricciones:** Restricciones `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK`
- Creación de una Tabla mediante una Subconsulta
- `ALTER TABLE`
  - Tablas de sólo lectura
- Sentencia `DROP TABLE`

# Inclusión de Restricciones

- Las restricciones aplican reglas a nivel de tabla.
- Las restricciones impiden la supresión de una tabla si hay dependencias.
- Los siguientes tipos de restricciones son válidos:
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK





# Instrucciones de Restricción

- Puede asignar un nombre a una restricción o el servidor de Oracle genera un nombre con el formato `SYS_Cn`.
- Crear una restricción en uno de los siguientes momentos:
  - En el momento de la creación de la tabla
  - Después de la creación de la tabla
- Definir una restricción a nivel de columna o de tabla.
- Ver una restricción en el diccionario de datos.

# Definición de Restricciones

- Sintaxis:

```
CREATE TABLE [schema.]table
    (column datatype [DEFAULT expr]
     [column_constraint],
     ...
     [table_constraint] [, ...] );
```

- Sintaxis de restricción a nivel de columna:

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Sintaxis de restricción a nivel de tabla:

```
columna, ...
    [CONSTRAINT constraint_name] constraint_type
    (column, ...),
```

# Definición de Restricciones

- Ejemplo de una restricción a nivel de columna:

```
CREATE TABLE employees(  
  employee_id  NUMBER(6)  
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,  
  first_name   VARCHAR2(20) ,  
  ...);
```

1

- Ejemplo de una restricción a nivel de tabla:

```
CREATE TABLE employees(  
  employee_id  NUMBER(6) ,  
  first_name   VARCHAR2(20) ,  
  ...  
  job_id       VARCHAR2(10) NOT NULL,  
  CONSTRAINT emp_emp_id_pk  
    PRIMARY KEY (EMPLOYEE_ID) );
```

2

# Restricción NOT NULL

Garantiza que no se permiten los valores nulos para la columna:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	24000	(null)	90	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	17000	(null)	90	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	17000	(null)	90	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	9000	(null)	60	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	6000	(null)	60	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	4200	(null)	60	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	5800	(null)	50	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	3500	(null)	50	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	3100	(null)	50	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	2600	(null)	50	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	2500	(null)	50	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	10500	0.2	80	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	11000	0.3	80	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	8600	0.2	80	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	7000	0.15	(null)	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	4400	(null)	10	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	13000	(null)	20	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	6000	(null)	20	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	12000	(null)	110	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	8300	(null)	110	WGIEZT	515.123.8181	07-JUN-94

**Restricción NOT NULL**  
(La clave primaria aplica la restricción NOT NULL.)


**NOT NULL**  
restricción




**Ausencia de la restricción NOT NULL**  
(Cualquier fila puede contener un valor nulo para esta columna.)

# Restricción UNIQUE

## EMPLOYEES

Restricción UNIQUE



		EMPLOYEE_ID		LAST_NAME		EMAIL
1		100		King		SKING
2		101		Kochhar		NKOCHHAR
3		102		De Haan		LDEHAAN
4		103		Hunold		AHUNOLD
5		104		Ernst		BERNST
6		107		Lorentz		DLORENTZ

...



INSERT INTO

208	SMITH	JSMITH
-----	-------	--------

← Permitido

209	SMITH	JSMITH
-----	-------	--------

← No permitido: ya existe





# Restricción UNIQUE

Definida a nivel de tabla o de columna:

```
CREATE TABLE employees(  
    employee_id  NUMBER(6),  
    last_name    VARCHAR2(25) NOT NULL,  
    email        VARCHAR2(25),  
    salary       NUMBER(8,2),  
    commission_pct  NUMBER(2,2),  
    hire_date    DATE NOT NULL,  
    ...  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# Restricción PRIMARY KEY

**DEPARTMENTS**  **PRIMARY KEY**

	 DEPARTMENT_ID	 DEPARTMENT_NAME	 MANAGER_ID	 LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	(null)	1700

**No permitido  
(valor nulo)** 

 **INSERT INTO**


(null)	Public Accounting	124	2500
50	Finance	124	1500





**No permitido  
(ya existen 50)** 

# Restricción PRIMARY KEY

## DEPARTMENTS


**PRIMARY  
KEY**






	 DEPARTMENT_ID	 DEPARTMENT_NAME	 MANAGER_ID	 LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500

...

## EMPLOYEES



	 EMPLOYEE_ID	 LAST_NAME	 DEPARTMENT_ID
1	200	Whalen	10
2	201	Hartstein	20
3	202	Fay	20
4	205	Higgins	110
5	206	Gietz	110

**FOREIGN  
KEY**

...



**INSERT INTO**

200	Ford	9
200	Ford	60

**No permitido  
(9 no existen)**

**Permitido**



# Restricción PRIMARY KEY

Definida a nivel de tabla o de columna:

```
CREATE TABLE employees(  
    employee_id  NUMBER(6),  
    last_name    VARCHAR2(25) NOT NULL,  
    email        VARCHAR2(25),  
    salary       NUMBER(8,2),  
    commission_pct  NUMBER(2,2),  
    hire_date    DATE NOT NULL,  
    ...  
    department_id  NUMBER(4),  
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
        REFERENCES departments(department_id),  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# Restricción FOREIGN KEY: Palabras Clave

- FOREIGN KEY: define la columna en la tabla secundaria a nivel de restricción de tabla
- REFERENCES: identifica la tabla y la columna en la tabla principal
- ON DELETE CASCADE: suprime las filas dependientes de la tabla secundaria cuando se suprime una fila de la tabla principal
- ON DELETE SET NULL: convierte los valores de clave ajena dependiente en nulos.

# Restricción CHECK

- Define una condición que debe cumplir cada fila.
- Las siguientes expresiones no están permitidas:
  - Referencias a las pseudocolumnas CURRVAL, NEXTVAL, LEVEL y ROWNUM
  - Llamadas a las funciones SYSDATE, UID, USER y USERENV
  - Consultas que hagan referencia a otros valores en otras filas

```
..., salary  NUMBER(2)  
      CONSTRAINT emp_salary_min  
      CHECK (salary > 0), ...
```

# Sentencia CREATE TABLE: Ejemplo

```
CREATE TABLE employees
( employee_id      NUMBER(6)
  CONSTRAINT emp_employee_id PRIMARY KEY
, first_name       VARCHAR2(20)
, last_name        VARCHAR2(25)
  CONSTRAINT emp_last_name_nn NOT NULL
, email            VARCHAR2(25)
  CONSTRAINT emp_email_nn     NOT NULL
  CONSTRAINT emp_email_uk     UNIQUE
, phone_number     VARCHAR2(20)
, hire_date        DATE
  CONSTRAINT emp_hire_date_nn NOT NULL
, job_id           VARCHAR2(10)
  CONSTRAINT emp_job_nn      NOT NULL
, salary           NUMBER(8,2)
  CONSTRAINT emp_salary_ck   CHECK (salary>0)
, commission_pct   NUMBER(2,2)
, manager_id       NUMBER(6)
  CONSTRAINT emp_manager_fk REFERENCES
    employees (employee_id)
, department_id    NUMBER(4)
  CONSTRAINT emp_dept_fk     REFERENCES
    departments (department_id));
```

# Violación de Restricciones

El departamento 55 no existe.

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110;
```

Error starting at line 1 in command:

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110
```

Error report:

```
SQL Error: ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:      A foreign key value has no matching primary key value.
```

# Violación de Restricciones

No puede suprimir una fila que contenga una clave primaria que se utilice como clave ajena en otra tabla.

```
DELETE FROM departments
WHERE department_id = 60;
```

Error starting at line 1 in command:

```
DELETE FROM departments
WHERE department_id = 60
```

Error report:

SQL Error: ORA-02292: integrity constraint (ORA1.JHIST\_DEPT\_FK) violated - child record found  
02292. 00000 - "integrity constraint (%s.%s) violated - child record found"

\*Cause: attempted to delete a parent key value that had a foreign  
dependency.

\*Action: delete dependencies first then parent or disable constraint.

# Agenda

- Objetos de base de datos
  - Reglas de nomenclatura
- Sentencia `CREATE TABLE`:
  - Acceso a otras tablas de usuario
  - Opción `DEFAULT`
- Tipos de dato
- Visión general de restricciones: Restricciones `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK`
- **Creación de una Tabla mediante una Subconsulta**
- `ALTER TABLE`
  - Tablas de sólo lectura
- Sentencia `DROP TABLE`

# Creación de una Tabla mediante una Subconsulta

- Crear una tabla e insertar filas combinando la sentencia `CREATE TABLE` y la opción `AS subquery`.

```
CREATE TABLE table  
            [ (column, column...) ]  
AS subquery;
```

- Hacer coincidir el número de columnas especificadas con el número de columnas de la subconsulta.
- Definir columnas con nombres de columna y valores por defecto.



# Creación de una Tabla mediante una Subconsulta

```
CREATE TABLE dept80
AS
SELECT  employee_id, last_name,
        salary*12 ANNSAL,
        hire_date
FROM    employees
WHERE   department id = 80;
```

CREATE TABLE succeeded.

```
DESCRIBE dept80
```

Name	Null	Type
-----	-----	-----
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

# Agenda

- Objetos de base de datos
  - Reglas de nomenclatura
- Sentencia CREATE TABLE:
  - Acceso a otras tablas de usuario
  - Opción DEFAULT
- Tipos de dato
- Visión general de restricciones: Restricciones NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK
- Creación de una tabla mediante una subconsulta
- ALTER TABLE
  - Tablas de sólo lectura
- Sentencia DROP TABLE

# Sentencia ALTER TABLE

Utilizar la sentencia ALTER TABLE para:

- Agregar una nueva columna
- Modificar una definición de columna existente
- Definir un valor por defecto para la nueva columna
- Borrar una columna
- Cambiar el nombre de una columna
- Cambiar la tabla al estado de sólo lectura

# Tablas de Sólo Lectura

Puede utilizar la sentencia `ALTER TABLE` para:

- Definir una tabla en modo de sólo lectura, que evita que DDL o DML cambie durante el mantenimiento de la tabla
- Volver a definir la tabla en modo de lectura/escritura

```
ALTER TABLE employees READ ONLY;
```

```
-- perform table maintenance and then
```

```
-- return table back to read/write mode
```

```
ALTER TABLE employees READ WRITE;
```

# Agenda

- Objetos de base de datos
  - Reglas de nomenclatura
- Sentencia CREATE TABLE:
  - Acceso a otras tablas de usuario
  - Opción DEFAULT
- Tipos de dato
- Visión general de restricciones: Restricciones NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK
- Creación de una tabla mediante una subconsulta
- ALTER TABLE
  - Tablas de sólo lectura
- Sentencia DROP TABLE

# Borrado de una Tabla

- Mueve una tabla a la papelera de reciclaje
- Elimina la tabla y todos sus datos completamente si se especifica la cláusula `PURGE`
- Invalida objetos dependientes y elimina privilegios de objeto en la tabla

```
DROP TABLE dept80;
```

```
DROP TABLE dept80 succeeded.
```

# Prueba

Puede utilizar restricciones para realizar lo siguiente:

1. Aplicar reglas a los datos de la tabla cuando se inserta, actualiza o suprime una fila de dicha tabla.
2. Evitar la supresión de una tabla.
3. Evitar la creación de una tabla.
4. Evitar la creación de datos en una tabla.

# Resumen

En esta lección, debe haber aprendido a utilizar la sentencia `CREATE TABLE` para crear una tabla e incluir restricciones.

- Clasificación de los Principales Objetos de Base de Datos
- Revisar la estructura de la tabla
- Mostrar los tipos de dato disponibles para columnas
- Crear una tabla simple
- Explicar cómo crear restricciones en el momento de la creación de la tabla
- Describir el funcionamiento de los objetos de esquema



# Práctica 10: Visión General

En esta práctica se abordan los siguientes temas:

- Creación de nuevas tablas
- Creación de una nueva tabla mediante la sintaxis `CREATE TABLE AS`
- Verificación de la existencia de las tablas
- Definición de una tabla al estado de sólo lectura
- Borrado de tablas