



SQL

2024

Profesores del curso



ÍNDICE

1. Lenguaje SQL
2. SQL Data Definition Language
3. Conclusiones
4. Referencias

Saberes previos

- El Modelo Relacional:
 - Estructura y operaciones relacionales
 - Reglas de integridad
- Técnicas de modelamiento de datos
 - IDEF1X
 - Tipos de datos
 - Modelamiento y convenciones
 - Notación Barker



Objetivos



Objetivos

- Conocer las principales características del lenguaje SQL
- Explicar los principales conceptos de definición de datos (DDL) de una base de datos tales como tablas, vistas e índices.
- Conocer la sintaxis de los comandos SQL para la creación, modificación y eliminación de tablas, vistas e índices.



Lenguaje SQL



Principales características de SQL

- Primera versión, SQL-86 en 1986. Versión más reciente en 2016 (SQL-2016)
- Aceptado por el American National Standards Institute (ANSI) en 1986 y por la Organización Internacional de Normalización (ISO) en 1987
- Cada proveedor ofrece su propia implementación (dialecto SQL) de SQL
- Orientado a conjuntos y declarativo
- Lenguaje de forma libre
- No distingue entre mayúsculas y minúsculas
- Puede utilizarse de forma interactiva desde un símbolo del sistema o ejecutarse mediante un programa



Definición de Lenguaje SQL

- Las siglas SQL significan *Structured Query Language*, que traducido al castellano significa lenguaje de consulta estructurado.
- SQL está conformado por un conjunto de comandos o mandatos con las que todos los programas y usuarios acceden a los datos en una base de datos Oracle.

Mandato	Tipo
Create	DDL: Data Definition Language
Alter	
Drop	
Select	DML: Data Manipulation Language
Insert	
Delete	
Update	
Grant	DCL: Data Control Language
Revoke	



SQL Data Definition Language



Conceptos clave DDL

SQL utiliza los términos tabla, fila, y columna para los términos relación , tupla y atributo del modelo relacional. El principal comando de SQL para definir datos es la sentencia CREATE, que se utiliza para crear esquemas, tablas (relaciones) y dominios (así como otras estructuras, como vistas, aserciones y *triggers*)



Conceptos clave DDL

- Un **esquema SQL** se identifica con un nombre de esquema e incluye un identificador de autorización. Los elementos del esquema son las tablas y otros objetos de la base de datos, como las vistas, los dominios, restricciones e índices, que lógicamente están relacionados entre sí.

Ejemplo:

```
CREATE SCHEMA EMPRESA AUTHORIZATION Jperez;
```

- La tabla SQL implementa una relación del modelo relacional.

```
CREATE TABLE PRODUCT ... (esquema implícito)
```

```
CREATE TABLE EMPRESA.PRODUCT ... (esquema explícito)
```



Conceptos clave DDL

Definición de esquema

- **CREATE TABLE:** crea una tabla, definiendo sus columnas, restricciones y ubicaciones de almacenamiento.
- **ALTER TABLE:** modifica (redefine) o añade columnas, restricciones o ubicaciones de almacenamiento.
- **DROP:** elimina (remueve) objetos de la base de datos: tablas, vistas, procedimientos, disparadores, índices, paquetes, funciones, usuarios, etc.



Conceptos clave DDL

Tipos de datos*

Data Type	Description
CHAR(n)	Holds a fixed length string with size n
VARCHAR(n)	Holds a variable length string with maximum size n
SMALLINT	Small integer (no decimal) between -32768 to 32767
INT	Integer (no decimal) between -2147483648 to 2147483647
FLOAT(n,d)	Small number with a floating decimal point. The total maximum number of digits is n with a maximum of d digits to the right of the decimal point.
DOUBLE(n,d)	Large number with a floating decimal point. The total maximum number of digits is n with a maximum of d digits to the right of the decimal point.

Los tipos de datos* pueden variar según el SGBD*



Conceptos clave DDL

Tipos de datos*

Data Type	Description
DATE	Date in format YYYY-MM-DD
DATETIME	Date and time in format YYYY-MM-DD HH:MI:SS
TIME	Time in format HH:MI:SS
BOOLEAN	True or False
BLOB	Binary Large Object (e.g. image, audio, video)

Los tipos de datos* pueden variar según el SGBD*



Conceptos clave DDL

CREATE TABLE

La sentencia CREATE TABLE se utiliza para crear una tabla, definiendo sus columnas, restricciones y ubicación de almacenamiento.

- Primero se especifica el nombre de la tabla y el nombre del esquema al que pertenece la nueva tabla.
- Luego se define las columnas de la tabla con sus respectivos nombres, tipos de datos y **restricciones**.
- Finalmente se agregan las restricciones de tabla según corresponda: clave primaria, clave foránea y verificaciones.



Conceptos clave DDL

CREATE TABLE

```
CREATE TABLE schema_name.table_name (  
    column_1 data_type column_constraint,  
    column_2 data_type column_constraint,  
    ...  
    table_constraint  
);
```



Conceptos clave DDL

Restricciones de columna para las tablas

- La restricción **PRIMARY KEY** define la clave primaria de la tabla
- La restricción **FOREIGN KEY** define una clave externa de una tabla
- La restricción **UNIQUE** define una clave alternativa de una tabla
- La restricción **NOT NULL** prohíbe valores **nulos** para una columna
- La restricción **DEFAULT** establece un valor por defecto para una columna
- La restricción **CHECK** define una restricción sobre los valores de la columna



Conceptos clave DDL

Ejemplo de uso de CREATE TABLE

```
CREATE TABLE scott.emp
(empno NUMBER CONSTRAINT pk_emp PRIMARY KEY,
ename VARCHAR2(10) CONSTRAINT nn_name NOT NULL
                        CONSTRAINT upper_name CHECK(ename=UPPER(ename)),
job VARCHAR2(9),
mgr NUMBER CONSTRAINT fk_mgr REFERENCES scott.emp(empno),
hiredate DATE DEFAULT SYSDATE,
sal NUMBER(10,2) CONSTRAINT ck_sal CHECK (sal > 500),
comm NUMBER(9,0) DEFAULT NULL,
deptno NUMBER(2) CONSTRAINT nn_deptno NOT NULL
                        CONSTRAINT fk_deptno REFERENCES scott.dept(deptno))
PCTFREE 5 PCTUSED 75;
```

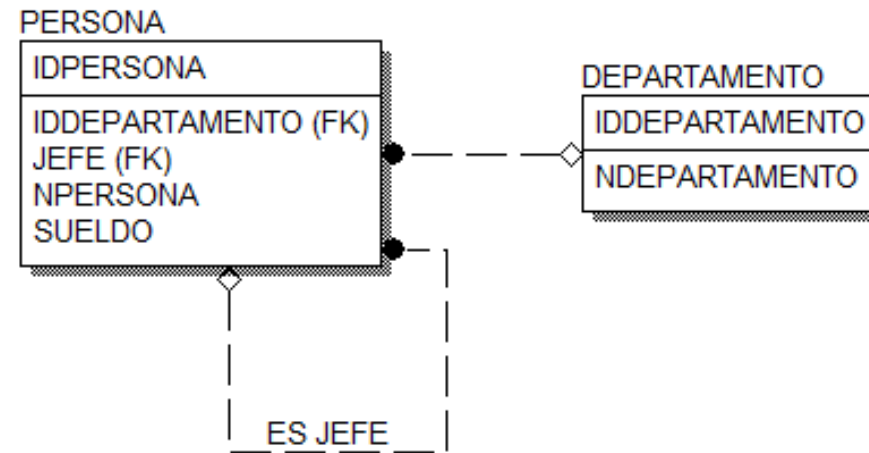


Conceptos clave DDL

1. Crear la tabla Persona considerando la relación recursiva "es jefe de"

```
CREATE TABLE Persona  
(IdPersona CHAR(8) PRIMARY KEY,  
IdDepartamento CHAR(2) REFERENCES Departamento(IdDepartamento),  
Npersona VARCHAR2(40) NOT NULL,  
Jefe CHAR(8) REFERENCES Persona(IdPersona),  
Sueldo NUMBER (6,2) DEFAULT 930);
```

```
CREATE TABLE Departamento(  
IdDepartamento NUMBER(3) PRIMARY KEY,  
NDepartamento VARCHAR2(60) NOT NULL );
```



Conceptos clave DDL

ALTER TABLE

La sentencia ALTER TABLE se utiliza para:

- Agregar una nueva columna
- Modificar una columna existente
- Definir un valor DEFAULT para una columna
- Borrar una columna

Puede agregar o modificar una columna de una tabla, pero no puede especificar donde aparece la columna.



Conceptos clave DDL

2. Añadir la columna FechaIngreso tipo **DATE** que represente la fecha de ingreso a laborar y además la columna Sexo tipo **CHAR** de tamaño uno con la restricción NOT NULL.

```
ALTER TABLE Persona ADD (FechaIngreso DATE, Sexo CHAR(1) not null)
```

3. Modificar la columna Sueldo a un tamaño de 7 dígitos, y con un salario por defecto de 1025 soles.

```
ALTER TABLE Persona MODIFY Sueldo NUMBER(7,2) DEFAULT (1025)
```

4. Modificar la columna FechaIngreso para que por defecto se registre la fecha actual.

```
ALTER TABLE Persona MODIFY FechaIngreso DEFAULT (sysdate)
```



Conceptos clave DDL

5. Agregar la columna país de nacimiento y modificar sueldo con la restricción del mínimo legal.

```
ALTER TABLE Persona  
ADD (PaisNacimiento NUMBER(3) REFERENCES Pais)  
MODIFY (Sueldo NUMBER (9,2) CHECK (Sueldo >= 1025 and Sueldo <=15000));
```

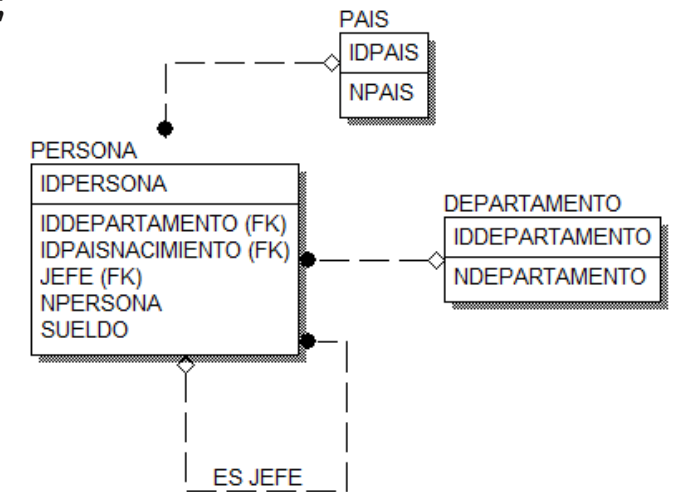
```
CREATE TABLE Pais(  
IdPais NUMBER(3) PRIMARY KEY,  
NPais VARCHAR2(60) );
```

También puede cambiar el nombre de una columna.

```
ALTER TABLE tablename
```

```
RENAME COLUMN columnname TO new_columnname;
```

```
ALTER TABLE PAIS RENAME COLUMN Nombre TO NPais;
```



Conceptos clave DDL

UNIQUE

Impide la duplicación de claves alternas (no primarias), es decir, que dos registros no puedan tener el mismo valor en un campo. Se permiten nulos.

Se pueden aplicar varias restricciones de este tipo a una misma tabla y pueden aplicarse a uno o varios campos que no sean clave primaria.

```
ALTER TABLE nombretable
```

```
ADD CONSTRAINT nombrerestriccion
```

```
UNIQUE (campo)
```

Ejemplo

```
ALTER TABLE alumnos
```

```
ADD CONSTRAINT uq_alumnos_doc
```

```
UNIQUE (documento)
```



Conceptos clave DDL

CHECK

Se utiliza para especificar reglas para el contenido de una tabla. Se especifica una **condición** de búsqueda (que es una expresión booleana) para una restricción de comprobación. Esta **condición** debe cumplirse para todas las filas de la tabla. La **condición** se aplica a cada fila que se modifica en una instrucción INSERT o UPDATE en el momento de la modificación de la fila. La instrucción completa se cancela si se viola alguna restricción de comprobación.

Ejemplo

```
ALTER TABLE horario
ADD CONSTRAINT ck_vacantes CHECK (vacantes > 0);           -- si el campo existe
ALTER TABLE horario
ADD inscritos number(3) CONSTRAINT ck_inscritos CHECK(inscritos > 0); -- si el campo no existe
```



Conceptos clave DDL

6. Agregar a la columna Npaís una restricción de no permitir nombres duplicados.

```
ALTER TABLE Pais  
ADD Npais CONSTRAINT uq_nombre_pais UNIQUE (NPais);
```

7. Agregar la columna Npoblacion con una restricción que debe ser mayor a cero.

```
ALTER TABLE Pais  
ADD Npoblacion NUMBER(10) CONSTRAINT ck_poblacion CHECK(Npoblacion > 0);
```



Conceptos clave DDL

ALTER TABLE

Al **borrar una columna** se aplican las siguientes reglas:

- Se puede borrar una columna que contenga datos
- Solo se puede borrar una columna cada vez
- No puede borrar todas las columnas de una tabla; debe permanecer al menos una columna
- Una vez borrada la columna, los valores de datos de esta no se pueden recuperar

Sintaxis:

ALTER TABLE *tablename* DROP COLUMN *column name*;

Además de colocar **constraints** o restricciones

**ALTER TABLE *tablename* ADD CONSTRAINT *fk_nombre* FOREIGN KEY
(*columnname*) REFERENCES *tabla* (*columnname*);**



Conceptos clave DDL

DROP TABLE

Permite **eliminar** la tabla nombrada.

DROP TABLE *NombreTabla*;

DROP TABLE *NombreTabla* CASCADE CONSTRAINTS; //borra la referencia a FK si lo tuviera

Al borrar una tabla:

- Desaparecen todos los datos
- Cualquier vista y sinónimo referente a la tabla seguirá existiendo, pero ya no funcionarán (conviene eliminarlos)
- Las transacciones pendientes son aceptadas (COMMIT)

El borrado de una tabla **es irreversible**, y Oracle Server no cuestiona la decisión y borra la tabla inmediatamente.



Conceptos clave DDL

ÍNDICES

Un índice de Oracle Server es un objeto de esquema que puede acelerar la recuperación de filas mediante un puntero.

Los índices se pueden crear explícita o automáticamente. Además, debe considerar:

- Si no hay un índice en la columna seleccionada, se produce una exploración de tabla completa
- Un índice proporciona acceso directo y rápido a las filas de una tabla
- El índice lo utiliza y mantiene automáticamente Oracle Server. Una vez creado un índice, no será necesaria ninguna intervención directa por parte del usuario
- Los índices son lógicamente y físicamente independientes de la tabla que indexan
- Esto significa que se pueden crear o borrar en cualquier momento sin que afecten a las tablas base o a otros índices



Conceptos clave DDL

ÍNDICES

Se pueden definir dos tipos de índices:

- **Índice único:** Oracle Server **crea automáticamente** este índice al definir una restricción de clave PRIMARY KEY o UNIQUE en una columna de la tabla
- **Índice no único:** este es un índice que un usuario puede crear para acelerar el acceso a las filas

Para crear un índice en una o más columnas, utilice la sentencia CREATE INDEX:

```
CREATE INDEX index_name  
ON table_name (column..., column);
```

```
CREATE INDEX wf_cont_reg_id_idx  
ON wf_countries (region_id);
```



Conceptos clave DDL

ÍNDICES

Se **debe crear** un índice solo si:

- La columna contiene una amplia variedad de valores
- Una columna contiene un gran número de valores nulos
- Una o más columnas se utilizan con frecuencia en conjunto en una cláusula WHERE o una condición de unión
- La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2 al 4% de las filas



Conceptos clave DDL

ÍNDICES

Por lo general, **no merece** la pena crear un índice si:

- La tabla es pequeña
- No se suelen utilizar las columnas como condición en la consulta
- Se espera que la mayoría de las consultas recuperen más del 2 al 4% en la tabla
- La tabla se actualiza con frecuencia
- Se hace referencia a las columnas indexadas como parte de una expresión



Conceptos clave DDL

ÍNDICES COMPUESTOS

- Un índice compuesto (también denominado índice "concatenado") es un índice creado en varias columnas de una tabla
- Las columnas del índice compuesto pueden aparecer en cualquier orden y no es necesario que sean adyacentes en la tabla

```
CREATE INDEX emps_name_idx  
ON employees (first_name, last_name);
```



Conceptos clave DDL

USO DE VISTAS

Una vista, como una tabla, es un objeto de base de datos. Sin embargo, tenemos que:

- Las vistas no son tablas "reales"
- Son representaciones lógicas de tablas existentes o de otra vista
- Las vistas no contienen datos propios
- Funcionan como una ventana por la que se pueden ver o cambiar los datos de las tablas
- Las tablas en las que se basa la vista se denominan tablas "base"



Conceptos clave DDL

USO DE VISTAS

La vista es una consulta almacenada como una sentencia SELECT en el diccionario de datos.

```
CREATE VIEW view_employees
AS SELECT employee_id, first_name, last_name, email
FROM employees
WHERE employee_id BETWEEN 100 and 124;

SELECT *
FROM view_employees;
```





Conclusiones

En esta sesión, debe haber aprendido lo siguiente:

- SQL es el lenguaje utilizado para la gestión de bases de datos relacionales
- SQL DDL son comandos de SQL que se utilizan para gestionar la estructura física de la base de datos





Referencias

- R. Elmasri y S.B. Navathe. (2007). Fundamentos de Sistema de Base de Datos, 5ta edición
- Korth, H. and A. Silberschatz (1996). Fundamentos de Bases de Datos. España, McGraw Hill.
- Oracle Help Center. (5 de septiembre de 2024). *SQL Language Reference*.
<https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/Introduction-to-Oracle-SQL.html#GUID-89203F7C-527E-4C9E-B628-9AE0F955F4A7>

¡Gracias!

