

# Oracle Express 11g y DDL

INF 324 Bases de datos – Ing. Civil Informática 2020

Laboratorio N1 | 09-09-2020

Daniel Morán Faúndez

# Oracle Express 11G

- **Links de descarga:**

- Oracle (cuenta requerida): <https://www.oracle.com/database/technologies/xe-prior-releases.html>
- Google Drive: <https://drive.google.com/uc?export=download&id=1YgGp6t9T0Ti7tWtYQq-JLI-sxYEgldKx>

- **Tutorial de instalación:**

- <https://www.youtube.com/watch?v=IlnOZ5U8rqU&feature=youtu.be>

- **Recordar la contraseña que pongan durante la instalación para el usuario SYSTEM o SYS.**

# Oracle Express 11G

- 1 instalación por ordenador
- Limitado a una CPU
- Los datos no pueden exceder 11 Gb de datos. Si se supera este tamaño nos aparecerá un error ORA-12592
- Utiliza 1 GB de RAM como máximo.
- No trae soporte para HTTPS.
- **Puerto de conexión 1521**

# Lenguaje SQL

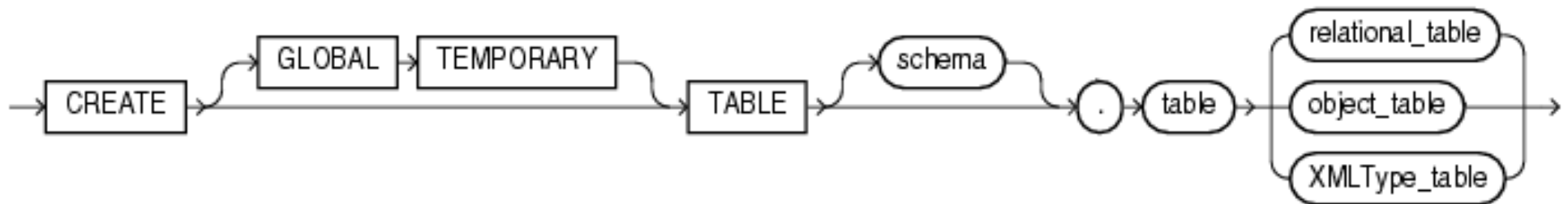
- Podemos dividir las sentencias SQL en dos categorías:
  - Lenguaje de definición de datos (DDL, Data Definition Language):

Sentencias que permiten crear y modificar la estructuras de la base de datos, ya sean las tablas u objetos. Ejemplos: [CREATE](#), [ALTER](#), [DROP](#), [TRUNCATE](#) ...
  - Lenguaje de manipulación de datos (DML, Data Manipulation Language):

Sentencias que permiten gestionar los datos dentro de los esquemas. Ejemplos: [SELECT](#), [INSERT](#), [UPDATE](#), [DELETE](#)...

# CREATE TABLE - DDL

- Creación de una tabla en un esquema



Que no cunda el pánico

# CREATE TABLE - DDL

```
CREATE TABLE NOMBRE_TABLA (  
    COLUMNA1 TIPO_DE_DATO(TAMAÑO),  
    COLUMNA2 TIPO_DE_DATO(TAMAÑO),  
    COLUMNA3 TIPO_DE_DATO(TAMAÑO),  
);
```

- Tipos de datos

```
-- Tipos numéricos  
NUMBER  
INT  
FLOAT  
LONG  
  
-- Tipos caracteres  
CHAR  
VARCHAR2  
  
-- Fechas  
DATE
```

- Limites:  
[https://docs.oracle.com/cd/B28359\\_01/server.111/b28320/limits001.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28320/limits001.htm)

# CREATE TABLE - DDL

- Ejemplo

```
CREATE TABLE USUARIO(  
    COD_USUARIO NUMBER NOT NULL,  
    RUT INT NOT NULL,  
    NOMBRE VARCHAR2(100) NOT NULL,  
    APELLIDO VARCHAR2(100),  
    CORREO VARCHAR2(60),  
    FECHA_NACIMIENTO DATE,  
    DIRECCION VARCHAR2(200),  
    PESO NUMBER CHECK(PESO > 0)  
);
```

# ALTER TABLE - DDL

- Modificar la estructura de una tabla ya creada en el esquema

```
-- Adicionar columna TELEFONO a la tabla USUARIO  
ALTER TABLE USUARIO ADD TELEFONO INT;  
  
-- Eliminar columna CORREO de la tabla USUARIO  
ALTER TABLE USUARIO DROP COLUMN CORREO;  
  
-- Modificar el tipo de dato de PESO en la tabla USUARIO  
ALTER TABLE USUARIO MODIFY PESO INT;
```



# DROP TABLE - DDL

- Eliminar una tabla ya creada en el esquema
  - Elimina los datos que contenga
  - No hay posibilidad de recuperar la tabla una vez eliminada

**DROP TABLE USUARIO;**

# PRIMARY KEY – DDL

- Asignar la clave primaria a un atributo de una tabla
  - El atributo debe ser no nulo

```
CREATE TABLE USUARIO(  
    COD_USUARIO NUMBER NOT NULL,  
    RUT INT NOT NULL,  
    NOMBRE VARCHAR2(100) NOT NULL,  
    APELLIDO VARCHAR2(100),  
    CORREO VARCHAR2(60),  
    FECHA_NACIMIENTO DATE,  
    DIRECCION VARCHAR2(200),  
    PESO NUMBER CHECK(PESO > 0),  
    PRIMARY KEY(COD_USUARIO)  
);
```

```
CREATE TABLE USUARIO(  
    COD_USUARIO NUMBER NOT NULL,  
    RUT INT NOT NULL,  
    NOMBRE VARCHAR2(100) NOT NULL,  
    APELLIDO VARCHAR2(100),  
    CORREO VARCHAR2(60),  
    FECHA_NACIMIENTO DATE,  
    DIRECCION VARCHAR2(200),  
    PESO NUMBER CHECK(PESO > 0),  
    CONSTRAINT PK_USUARIO PRIMARY KEY(COD_USUARIO)  
);
```

Al utilizar CONSTRAINT, asignará un nombre para la clave primaria.

# PRIMARY KEY – DDL

- **OJO:** Si se utiliza la palabra UNIQUE en un atributo de la tabla, esta pasará a ser la clave primaria automáticamente.

```
CREATE TABLE USUARIO(  
    COD_USUARIO NUMBER UNIQUE NOT NULL,  
    RUT INT NOT NULL,  
    NOMBRE VARCHAR2(100) NOT NULL,  
    APELLIDO VARCHAR2(100),  
    CORREO VARCHAR2(60),  
    FECHA_NACIMIENTO DATE,  
    DIRECCION VARCHAR2(200),  
    PESO NUMBER CHECK(PESO > 0),  
    CONSTRAINT PK_USUARIO PRIMARY KEY(COD_USUARIO)  
);
```

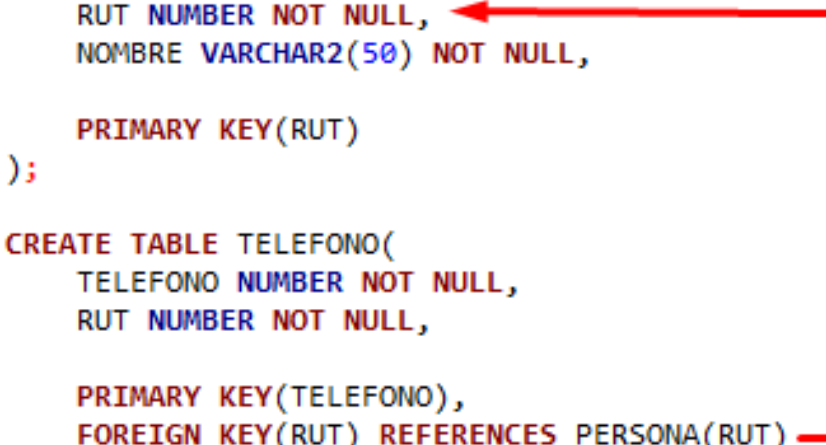


SQL Error [2261] [42000]: ORA-02261: such unique or primary key already exists in the table

# FOREIGN KEY – DDL

- Asignar la clave foránea de una tabla hacia otra, para establecer la relación. Recuerden que cada clave foránea hace referencia a una clave primaria de otra tabla.

```
CREATE TABLE PERSONA(  
  RUT NUMBER NOT NULL,  
  NOMBRE VARCHAR2(50) NOT NULL,  
  
  PRIMARY KEY(RUT)  
);  
  
CREATE TABLE TELEFONO(  
  TELEFONO NUMBER NOT NULL,  
  RUT NUMBER NOT NULL,  
  
  PRIMARY KEY(TELEFONO),  
  FOREIGN KEY(RUT) REFERENCES PERSONA(RUT)  
);
```



- Al eliminar las tablas, estas deben ser eliminadas en orden, es decir, borrar la tabla TELEFONO y luego PERSONA, ya que si se elimina PERSONA primero, se perdería la referencia de la tabla TELEFONO.

# KEYS con ALTER TABLE

- Se pueden asignar tanto la clave primaria como la clave foránea utilizando ALTER TABLE.

```
CREATE TABLE PERSONA(  
  RUT NUMBER NOT NULL,  
  NOMBRE VARCHAR2(50) NOT NULL  
);
```

```
CREATE TABLE TELEFONO(  
  TELEFONO NUMBER NOT NULL,  
  RUT NUMBER NOT NULL  
);
```

```
-- Asignar la clave primaria en la tabla PERSONA  
ALTER TABLE PERSONA ADD CONSTRAINT PK_PERSONA PRIMARY KEY(RUT);
```

```
-- Asignar la clave primaria en la tabla TELEFONO  
ALTER TABLE TELEFONO ADD CONSTRAINT PK_TELEFONO PRIMARY KEY(TELEFONO);
```

```
-- Relacionar las tablas mediante el atributo RUT  
ALTER TABLE TELEFONO ADD CONSTRAINT FK_RUT_TELEFONO FOREIGN KEY(RUT)  
REFERENCES PERSONA(RUT);
```

