

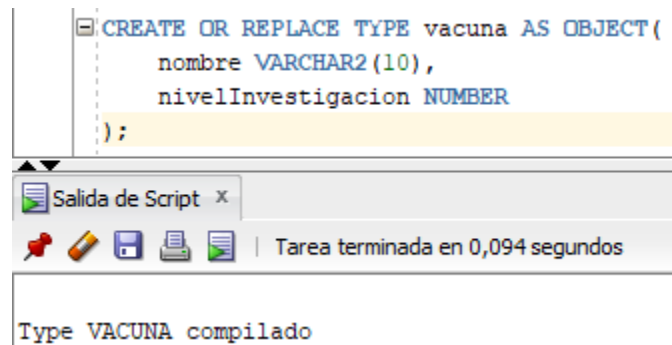
# CONTROL DE PARTIDAS PANDEMIC

## TABLA, OBJETOS Y VARRAYS

### OBJETO VACUNA

Primero vamos a crear un objeto para guardar las vacunas donde guardaremos el nombre de la vacuna y el nivel de investigación.

```
CREATE OR REPLACE TYPE vacuna AS OBJECT(  
    nombre VARCHAR2(10),  
    nivelInvestigacion NUMBER  
);
```

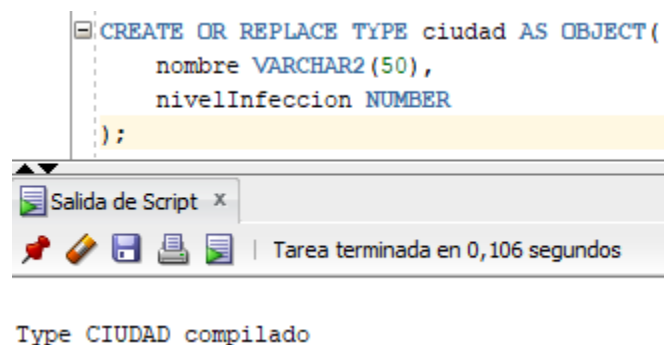


The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL script for creating the 'vacuna' object type. The bottom pane, titled 'Salida de Script', shows the message 'Tarea terminada en 0,094 segundos' and 'Type VACUNA compilado'.

### OBJETO CIUDAD

Luego crearemos otro objeto para guardar las ciudades donde vamos a guardar el nombre de la ciudad y el nivel de infección.

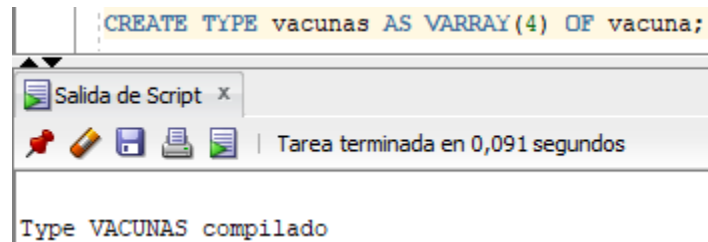
```
CREATE OR REPLACE TYPE ciudad AS OBJECT(  
    nombre VARCHAR2(50),  
    nivelInfeccion NUMBER  
);
```



The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL script for creating the 'ciudad' object type. The bottom pane, titled 'Salida de Script', shows the message 'Tarea terminada en 0,106 segundos' and 'Type CIUDAD compilado'.

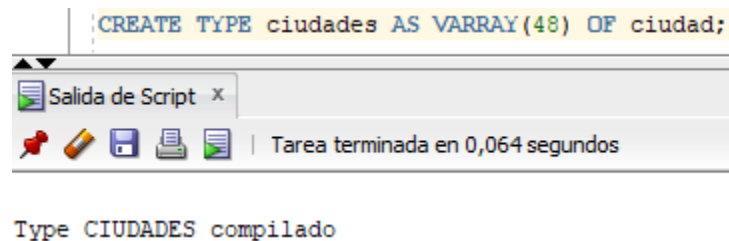
## VARRAY VACUNAS

Una vez creados los dos objetos pasaremos a crear un varray indicando su tamaño y el tipo de objeto que va a guardar. En este caso creamos varray vacunas donde vamos a guardar 4 vacunas.



## VARRAY CIUDADES

Ahora crearemos otro varray para las ciudades, donde vamos a guardar 48 objetos de tipo ciudad en este caso las 48 ciudades.



## TABLA PARTIDA

**PARTIDA**(idPartida, usuario, fecha, dificultad, estado, ronda, acciones, enfermedadesActivas, brotes, investigacionVacunas, investigacionCiudades)

idPartida	NUMBER	PK
usuario	VARCHAR2(35)	NN
fecha	DATE	NN
dificultad	VARCHAR2(10)	NN
estado	CHAR(1)	NN Valor por defecto: 'E' En este campo se va a guardar el estado de la partida, 'E': empezada, 'G': ganada, 'P': perdida
ronda	NUMBER	NN
acciones	NUMBER	NN
enfermedadesActivas	NUMBER	NN
brotes	NUMBER	NN
investigacionVacunas	vacunas	NN
infeccionCiudades	ciudades	NN

## SENTENCIA DE CREACIÓN DE LA TABLA

Esta es la sentencia para la creación de la tabla.

```
CREATE TABLE PARTIDA(  
    idPartida NUMBER PRIMARY KEY,  
    usuario VARCHAR2(35) NOT NULL,  
    fecha DATE NOT NULL,  
    dificultad VARCHAR2(10) NOT NULL,  
    estado CHAR(1) DEFAULT 'E' NOT NULL,  
    ronda NUMBER NOT NULL,  
    acciones NUMBER NOT NULL,  
    enfermedadesActivas NUMBER NOT NULL,  
    brotes NUMBER NOT NULL,  
    investigacionVacunas vacunas,  
    infeccionCiudades ciudades  
);
```

Salida de Script x

Tarea terminada en 0,143 segundos

Table PARTIDA creado.



## INSTRUCCIONES PARA INSERTAR DATOS

Se hará un insert al elegir la opción guardar y salir o al finalizar la partida. La sentencia que se va a usar es la siguiente.

```
INSERT INTO PARTIDA VALUES(pf_seqIdPartida.NEXTVAL, 'pelofregona', SYSDATE, 'Facil', 'E', 1, 4, 9, 0,
vacunas(vacuna('Alfa', 0), vacuna('Beta', 0), vacuna('Gama', 0), vacuna('Delta', 0)),
ciudades(ciudad('San Francisco', 0), ciudad('Chicago', 0), ciudad('Atlanta', 0),
ciudad('Montreal', 1), ciudad('Nueva York', 2), ciudad('Washington', 1), ciudad('Londres', 0),
ciudad('Madrid', 0), ciudad('Paris', 0), ciudad('Essen', 0), ciudad('Milan', 0),
ciudad('San Petersburgo', 0), ciudad('Los Angeles', 0), ciudad('Miami', 0), ciudad('Mexico', 0),
ciudad('Bogota', 0), ciudad('Lima', 0), ciudad('Santiago de Chile', 0), ciudad('Buenos Aires', 0),
ciudad('Sao Paulo', 0), ciudad('Lagos', 0), ciudad('Kinsasa', 0), ciudad('Jartum', 2),
ciudad('Johannesburgo', 0), ciudad('Argel', 0), ciudad('El Cairo', 1), ciudad('Riad', 0),
ciudad('Estambul', 0), ciudad('Bagdad', 0), ciudad('Moscu', 0), ciudad('Teheran', 0),
ciudad('Karachi', 0), ciudad('Nueva Delhi', 0), ciudad('Calcuta', 0), ciudad('Madras', 0),
ciudad('Bombay', 0), ciudad('Yakarta', 1), ciudad('Bangkok', 0), ciudad('Hong Kong', 1),
ciudad('Shanghai', 0), ciudad('Pekin', 0), ciudad('Seul', 0), ciudad('Tokio', 0),
ciudad('Osaka', 0), ciudad('Taipei', 0), ciudad('Ho Chi Minh', 0), ciudad('Manila', 0),
ciudad('Sidney', 0)))
```

Salida de Script x

Tarea terminada en 0,152 segundos

1 fila insertadas.

## INSTRUCCIONES PARA ACTUALIZAR DATOS

En caso de querer actualizar algún dato usaremos la siguiente sentencia.

```
UPDATE PARTIDA
SET brotes = 3
WHERE idPartida = 4;
```

Salida de Script x

Tarea terminada

1 fila actualizadas.



## INSTRUCCIONES PARA BORRAR DATOS

En caso de querer borrar alguna partida usaremos la siguiente sentencia.

```
DELETE FROM PARTIDA
WHERE idPartida = 4;
```

Salida de Script x

Tarea terminada

1 fila eliminado

## INSTRUCCION PARA RECUPERAR INFORMACION DE TODAS LAS PARTIDAS QUE TENEMOS EMPEZADAS

La sentencia que usaremos para recuperar información que hemos empezado es la siguiente.

```
SELECT idPartida, fecha, dificultad
FROM PARTIDA
WHERE estado LIKE 'E'
AND usuario LIKE 'usuariol';
```

Salida de Script x

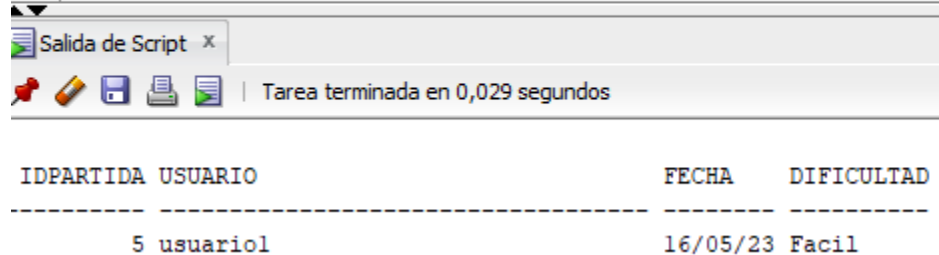
Tarea terminada en 0,382 segundos

IDPARTIDA	FECHA	DIFICULTAD
8	16/05/23	Facil
5	16/05/23	Facil
6	16/05/23	Facil

## INSTRUCCION PARA RECUPERAR INFORMACION DE UNA UNICA PARTIDA

La sentencia para recuperar información de una única partida es la siguiente.

```
SELECT idPartida, usuario, fecha, dificultad
FROM PARTIDA
WHERE idPartida = 5;
```



Salida de Script x

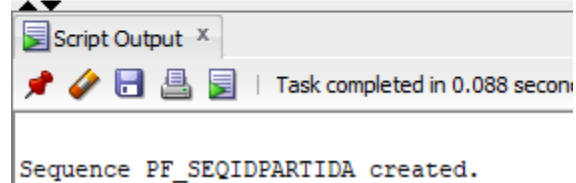
Tarea terminada en 0,029 segundos

IDPARTIDA	USUARIO	FECHA	DIFICULTAD
5	usuariol	16/05/23	Facil

## SECUENCIA ID PARTIDA

Esta secuencia se encarga de asignar un id de partida automáticamente a cada partida.

```
CREATE SEQUENCE pf_seqIdPartida
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE;
```



Script Output x

Task completed in 0.088 seconds

Sequence PF\_SEQIDPARTIDA created.