

JSON EN PL/SQL

JSON en PL/SQL

❑ REPASO JSON EN PL/SQL.

- ❑ Se usan en general las mismas funciones que en SQL
- ❑ También se pueden usar objetos PL/SQL para gestionar los datos JSON:
 - ❑ JSON_ELEMENT_T : Super Tipo del que heredan algunos de los otros objetos no se suele utilizar directamente
 - ❑ JSON_OBJECT_T : un objeto que representa un documento JSON.
 - ❑ JSON_ARRAY_T : Un objeto que representa un array JSON.
 - ❑ JSON_SCALAR_T : un valor escalar asociado a una clave. puede ser un una una cadena, un número, boolean or NULL.
 - ❑ JSON_KEY_LIST : Un array de claves.

JSON en ORACLE

□ Ejemplos

-- Trabajar con un objeto JSON

```
json1 := JSON_OBJECT_T('{“codigo_emple”:9999}');
```

```
json1.put('nombre', 'pepe');
```

```
Json1.remove(“nombre”);
```

```
json1.put('telefonos', JSON_ARRAY_T(['11111111','222222222','33333333']));
```

-- Trabajar con un array JSON

```
array1 := JSON_ARRAY_T(['rojo', " amarillo", " verde"]);
```

```
array1.append('azul');
```

JSON en PL/SQL

- ❑ Estos tipos JSON se almacenan en memoria en forma de árbol jerárquico y son una representación de la información que está almacenada dentro de algún campo de la base de datos.
- ❑ Por tanto podemos manipular la información guardada en la base de datos a través de estos objetos y sus procedimientos y funciones asociadas.
- ❑ Es importante recordar que estos objetos son temporales y por lo tanto deben ser almacenados dentro de la base de datos antes de cerrar la conexión
- ❑ Normalmente lo que hacemos una vez que hemos recuperado la información en el objeto JSON es serializarlo para convertirlo a algún dato de tipo carácter como VARCHAR2 ,CLOB o BLOB.

JSON en PL/SQL

❑ Constructores. Construyendo el objeto JSON. Se puede hacer de 3 formas

❑ Constructor vacío

```
DECLARE
    l_object    json_object_t;
BEGIN
    l_object := json_object_t();
END;
```

❑ Pasando contenido JSON

```
DECLARE
    l_object    json_object_t;
BEGIN
    l_object := json_object_t('{\"nombre\":\"alberto\"}');
END;
```

JSON en PL/SQL

- ❑ Constructores. Construyendo el objeto JSON. Se puede hacer de 3 formas
 - ❑ Con el método PARSE

```
DECLARE
    l_object    json_object_t;
BEGIN
    l_object := json_object_t.parse ('{"Nombre":"Alberto"}');
END;
```

JSON en PL/SQL

❑ Serialización

- ❑ Es lo contrario de lo anterior. es decir convierte algún objeto de tipo json en en tipos carácter como por ejemplo VARCHAR2.

```
MEMBER FUNCTION STRINGIFY RETURNS VARCHAR2  
MEMBER FUNCTION TO_STRING RETURNS VARCHAR2  
MEMBER FUNCTION TO_BOOLEAN RETURNS BOOLEAN  
MEMBER FUNCTION TO_NUMBER RETURNS NUMBER  
MEMBER FUNCTION TO_DATE RETURNS DATE  
MEMBER FUNCTION TO_TIMESTAMP RETURNS TIMESTAMP
```

JSON en PL/SQL

❑ Instrospección

- ❑ Permite identificar el contenido que puede tener el objeto json .
de esta forma podemos averiguar por ejemplo si es una cadena,
un número, etc .

```
MEMBER FUNCTION IS_OBJECT RETURNS BOOLEAN  
MEMBER FUNCTION IS_ARRAY RETURNS BOOLEAN  
MEMBER FUNCTION IS_SCALAR RETURNS BOOLEAN  
MEMBER FUNCTION IS_STRING RETURNS BOOLEAN  
MEMBER FUNCTION IS_NUMBER RETURNS BOOLEAN  
MEMBER FUNCTION IS_BOOLEAN RETURNS BOOLEAN  
MEMBER FUNCTION IS_TRUE RETURNS BOOLEAN  
MEMBER FUNCTION IS_FALSE RETURNS BOOLEAN  
MEMBER FUNCTION IS_NULL RETURNS BOOLEAN  
MEMBER FUNCTION IS_DATE RETURNS BOOLEAN  
MEMBER FUNCTION IS_TIMESTAMP RETURNS BOOLEAN  
MEMBER FUNCTION GET_SIZE RETURNS NUMBER
```


JSON en PL/SQL

□ Getters y Setters

□ Nos permiten recuperar o modificar los valores de un objeto json .

GET
GET_OBJECT
GET_ARRAY
GET_* : Por ejemplo GET_STRING, GET_CLOB ,GET_BLOB.
CLONE

PUT
PUT_NULL
REMOVE
RENAME_KEY
APPEND