Perfecto, Bryan. Aquí tienes la teoría del documento \*\*"BASE DE DATOS OBJETO RELACIONALES2.docx"\*\* organizada y resumida de forma clara para que la uses como guía de estudio y para hacer ejercicios.

---

# 🧬 \*\*Bases de Datos Objeto-Relacionales en Oracle\*\*

## 🎯 ¿Qué es una Base de Datos Objeto-Relacional?

Oracle permite extender el modelo relacional clásico incluyendo \*\*conceptos de la orientación a objetos\*\*, como:

- Tipos definidos por el usuario (UDT)

- Métodos

- Herencia

- Agregación

- Objetos persistentes

Estas extensiones convierten a Oracle en una \*\*base de datos objeto-relacional\*\*, ideal para modelar estructuras más complejas.

---

## 🧱 \*\*Tipos definidos por el usuario (UDT)\*\*

### 🔹 Crear un tipo de objeto:

```sql

CREATE TYPE NombreTipo AS OBJECT (

atributo1 tipo,

atributo2 tipo,

...

);

```

### 🧪 Ejemplo:

```sql

CREATE TYPE Persona AS OBJECT (

dni VARCHAR2(9),

nombre VARCHAR2(50),

edad NUMBER

);

```

---

## 📋 \*\*Tablas de objetos\*\*

### 🔹 Crear una tabla basada en un tipo:

```sql

CREATE TABLE nombre\_tabla OF NombreTipo;

```

### 🧪 Ejemplo:

```sql

CREATE TABLE personas OF Persona;

```

Esto permite guardar objetos completos como filas.

---

## 🧭 \*\*Acceso a atributos de objetos\*\*

Para acceder a los atributos:

```sql

SELECT p.nombre FROM personas p WHERE p.edad > 30;

```

---

## 🧩 \*\*Tipos anidados y colecciones\*\*

### 🔹 Crear colección tipo tabla:

```sql

CREATE TYPE ListaNumeros AS TABLE OF NUMBER;

```

### 🔹 Crear colección tipo VARRAY:

```sql

CREATE TYPE ListaNombres AS VARRAY(10) OF VARCHAR2(30);

```

Estas estructuras permiten \*\*almacenar listas de valores\*\* dentro de un solo campo.

---

## 🔁 \*\*Uso de métodos\*\*

### 🔹 Crear tipo con métodos:

```sql

CREATE TYPE Empleado AS OBJECT (

nombre VARCHAR2(50),

sueldo NUMBER,

MEMBER FUNCTION calcular\_bono RETURN NUMBER

);

```

### 🔹 Luego se implementan con:

```sql

CREATE TYPE BODY Empleado AS

MEMBER FUNCTION calcular\_bono RETURN NUMBER IS

BEGIN

RETURN sueldo \* 0.1;

END;

END;

```

---

## 🌳 \*\*Herencia (subtipos)\*\*

### 🔹 Crear subtipo con `UNDER`:

```sql

CREATE TYPE Estudiante UNDER Persona (

curso VARCHAR2(30)

);

```

Esto permite \*\*especializar tipos\*\* heredando sus atributos.

---

## ⚙️ \*\*Otros conceptos importantes\*\*

- \*\*REF\*\*: puntero o referencia a un objeto almacenado.

```sql

SELECT REF(p) FROM personas p;

```

- \*\*DEREF\*\*: acceder al objeto al que apunta una REF.

- \*\*TABLE()\*\*: acceder a una colección como tabla para hacer `SELECT`.

---

## 🧠 Consejo de estudio:

Agrupa los temas así:

1. Crear tipos (`CREATE TYPE`)

2. Crear tablas de tipos (`CREATE TABLE OF`)

3. Métodos (`MEMBER FUNCTION`)

4. Herencia (`UNDER`)

5. Colecciones (`VARRAY`, `TABLE OF`)

6. REF/DEREF y acceso

---

¿Te gustaría que te prepare también una hoja visual con esto o te hago una serie de ejercicios con soluciones paso a paso para practicar esta parte?

¡Claro, Bryan! Aquí tienes la teoría de **Bases de Datos Objeto-Relacionales en Oracle** con **definiciones de conceptos clave** y una sección especial con **reglas de oro** que te ayudarán a no cometer errores comunes y a tener claras las ideas importantes al estudiar y programar.

**🧬 Oracle Objeto-Relacional: Definiciones y Reglas de Oro**

**🧠 CONCEPTOS CLAVE**

**🟦 1. Tipo de objeto (OBJECT)**

Estructura que define un conjunto de atributos (como una clase en POO).

🔹 **Sintaxis:**

CREATE TYPE TipoNombre AS OBJECT (

atributo1 tipo,

atributo2 tipo

);

📝 **Ejemplo:**

CREATE TYPE Persona AS OBJECT (

dni VARCHAR2(9),

nombre VARCHAR2(50),

edad NUMBER

);

**🟦 2. Tabla de objetos**

Tabla basada en un tipo de objeto; cada fila es una instancia del objeto.

🔹 **Sintaxis:**

CREATE TABLE tablaNombre OF TipoObjeto;

📝 **Ejemplo:**

CREATE TABLE personas OF Persona;

**🟦 3. Acceso a atributos de objetos**

Usamos alias y punto para acceder a los atributos.

📝 **Ejemplo:**

SELECT p.nombre FROM personas p WHERE p.edad > 25;

**🟦 4. Colecciones**

Tipos que almacenan múltiples valores (como arrays o listas).

**a) VARRAY**

Lista con límite de elementos.

CREATE TYPE ListaNombres AS VARRAY(10) OF VARCHAR2(30);

**b) TABLE OF**

Lista sin límite fijo (tipo tabla).

CREATE TYPE ListaNumeros AS TABLE OF NUMBER;

**🟦 5. Métodos en objetos**

Funciones o procedimientos asociados a objetos, como en la POO.

CREATE TYPE Empleado AS OBJECT (

nombre VARCHAR2(50),

sueldo NUMBER,

MEMBER FUNCTION calcular\_bono RETURN NUMBER

);

🔹 **Implementación:**

CREATE TYPE BODY Empleado AS

MEMBER FUNCTION calcular\_bono RETURN NUMBER IS

BEGIN

RETURN sueldo \* 0.1;

END;

END;

**🟦 6. Herencia de tipos (UNDER)**

Crear subtipos que extienden otros tipos (herencia).

CREATE TYPE Estudiante UNDER Persona (

curso VARCHAR2(30)

);

**🟦 7. REF y DEREF**

REF es un puntero a un objeto, y DEREF accede al contenido apuntado.

SELECT REF(p) FROM personas p;

-- REF devuelve un puntero al objeto

**🟦 8. TABLE()**

Permite trabajar con colecciones como si fueran tablas en un SELECT.

**⭐ REGLAS DE ORO**

1. ✅ **Primero defines el tipo, luego la tabla.**
   * No puedes crear una tabla de un tipo que no existe aún.
2. ⚠️ **No confundas objetos con registros normales.**
   * Una tabla OF Tipo no se maneja igual que una tabla relacional clásica.
3. ✅ **Usa alias en SELECT para acceder a atributos del objeto.**
4. SELECT p.edad FROM personas p;
5. ⚠️ **Las colecciones TABLE OF no se pueden usar directamente como columnas simples.**
   * Necesitas crear una tabla anidada o usar TABLE() en un SELECT.
6. ✅ **REF y DEREF se usan solo con tablas de objetos.**
   * No tienen sentido en tablas relacionales normales.
7. ⚠️ **Cuando crees métodos, define primero el TYPE, luego el TYPE BODY.**
   * El cuerpo (BODY) implementa la lógica de las funciones.
8. ✅ **Las VARRAYs tienen límite de tamaño, los TABLE OF no.**
   * Elige según el caso de uso.
9. ✅ **UNDER permite herencia, pero los atributos deben ser compatibles.**
   * No puedes cambiar tipos al heredar.

¿Quieres que prepare una tabla resumen visual con esta teoría y reglas, o que cree ejercicios tipo examen sobre estos temas?