

Asignatura

Acceso a datos



**UNIVERSAE**  
Instituto Superior de FP

Asignatura

Acceso a datos

## UNIDAD 5

Base de datos de objetos y objeto-relacionales



UNIVERSAE  
Instituto Superior de FP

# Base de datos de objetos

## Acercarse al lenguaje de programación

- Tratan directamente con objetos del lenguaje de programación
- No existen tablas, filas ni columnas como en las base de datos relacionales
- Operan con Java, C#, Visual Basic y C++

## Ventajas

- Se adaptan al lenguaje de programación
- Menos tiempo de desarrollo y mantenimiento
- Mejor rendimiento

## Inconvenientes

- Pocos sistemas que utilices este tipo de base de datos
- Oferta muy limitada

## Ejemplos

- Db4o
- Objectivity/DB
- Objectdb





# Características base de datos objeto



## Soporte de objetos complejos

- Atributos, referencias a otros objetos, colecciones, etc.



## Identidad de objetos

- Identificar un objeto unequivocamente.
- Igualdad vs Identidad



## Tipos o clases

- Tratar tipos primitivos y clases.



## Extensibilidad

- Definir nuevos tipos



## Complejidad computacional

- Disponer de herramientas de manipulación de datos y cálculos



## Consultas

- Eficientes e independientes de la base de datos y lenguaje de programación



# Estándar ODMG

## Object Data Management Group (ODMG)

- Comunidad para estandarizar las bases de datos orientadas a objeto
- Fundada en 1991
- Última versión ODMG 3.0 del año 2000
- Actualmente este grupo está disuelto.

## Especificaciones del estándar

- Modelo de datos OMG (Object Management Group)
- Lenguaje de definición de objetos (ODL)
- Lenguaje de consultas de objetos (OQL)
- Vinculaciones con el lenguaje de programación (Language bindings)

# ODL

## Lenguaje de definición de objetos

- Se aplica directamente con el lenguaje de programación

## Características

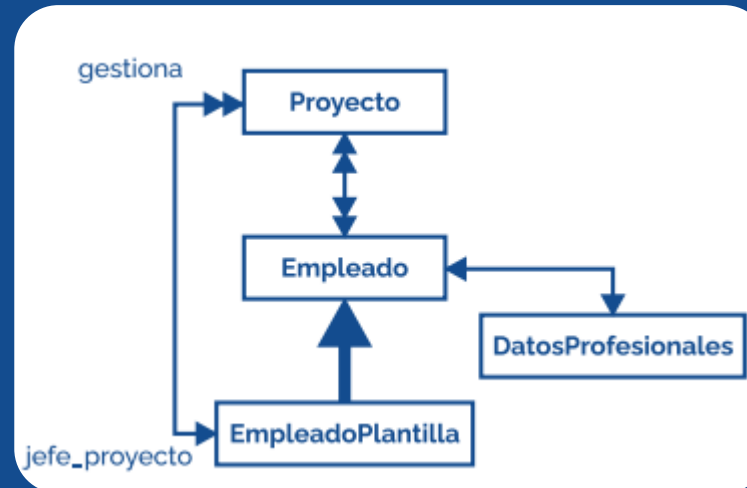
- El estado de las propiedades de un objeto puede cambiar
- Las propiedades puede ser atributos de un objeto o de una relación
- Un objeto es una instancia de una clase
- Un objeto puede tener un conjunto de operaciones
- Por cada propiedad debe de existir sus métodos get y set

## Clases e Interfaces

- Clases representación abstracta de una entidad real
  - Nuevo concepto **extent** para permitir el acceso a todas las instancias de una clase y **key** para identificar un objeto inequívocamente
- Interfaz representación abstracta de operaciones
  - Las interfaces no se pueden instanciar

## Relaciones

- La definen los atributos de una clase
- Si la cardinalidad es 1. Solo hay un atributo con sus métodos get/set
- Si la cardinalidad es N. El atributo es una colección con métodos añadir, modificar y eliminar
- Se puede aplicar el concepto de relación inversa





# OQL

## Lenguaje de consultas

- OQL (Object Query Language)
- Muy parecido al SELECT de SQL
- Cláusulas de agrupamiento y ordenación (GROUP BY, HAVING, ORDER BY)
- Operaciones de resumen (MIN, MAX, COUNT)
- No hay operaciones INSERT, UPDATE o DELETE

## Procedimiento

- Definir punto de entrada
- Sentencia SQL
- El resultado puede ser:
  - Una colección de objetos con elementos repetidos <bag>
  - Un colección de objetos sin elemento repetidos <set>  
(Para este resultado hay que utilizar DISTINCT)

```
class Empleado (extent empleados key dni)

SELECT emp.nombre
FROM empleados emp
WHERE emp.fecha_alta > date '2020-12-01';
```

```
class Empleado (extent empleados key dni)

SELECT struct(nombre: emp.nombre, alta: emp.fecha_alta)
FROM empleados emp
WHERE emp.fecha_alta > date '2020-12-01';
```

# Base de datos de objeto-relacional

## BDOR (Base de datos objeto-relacional)

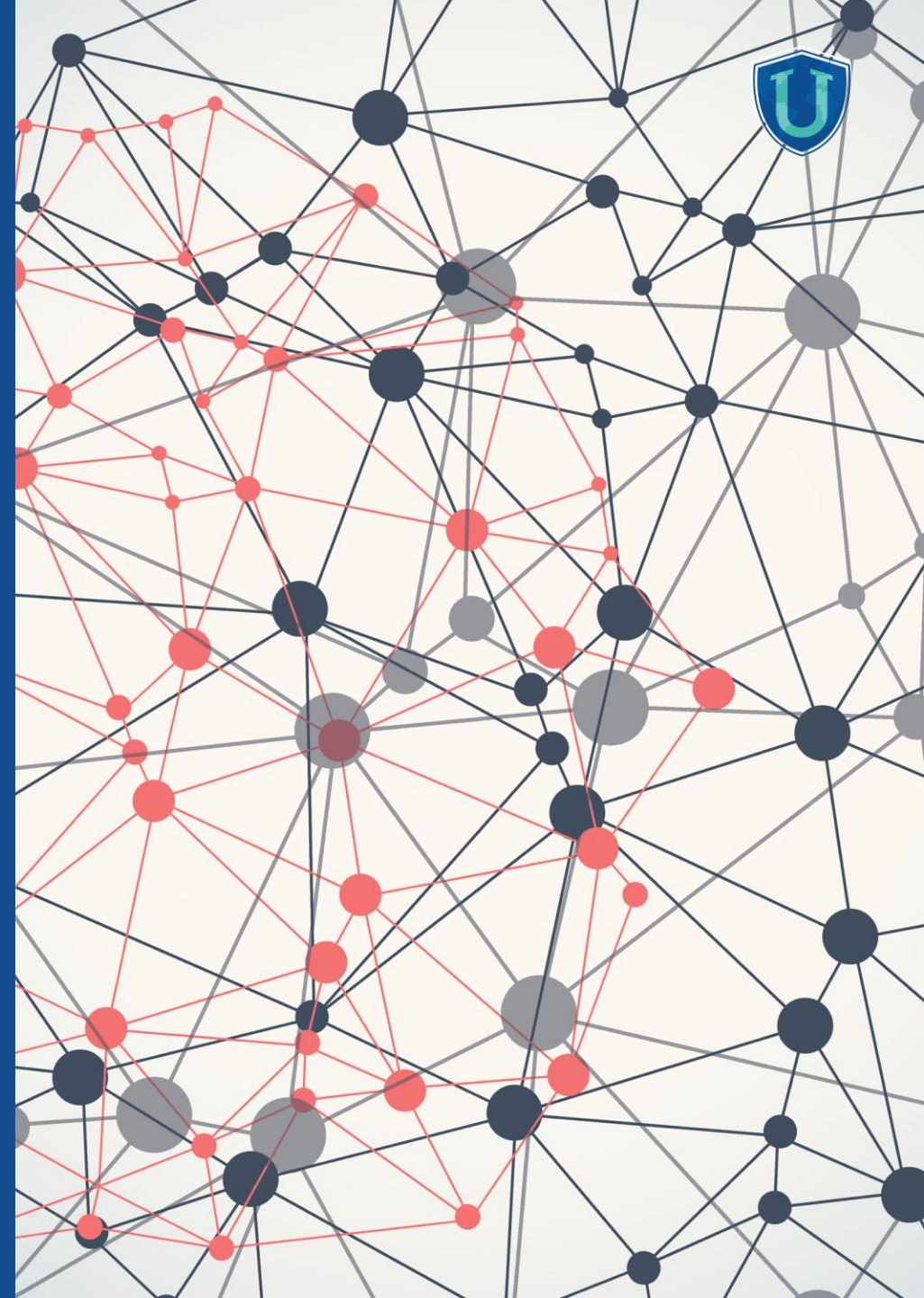
- Surgen a partir de la revisión SQL:99 o SQL3
- Son base de datos relacionales
- Utilizan una capa intermedia para usar objetos
- Definición de tipos estructurados de datos

## ¿Base de datos relacionales o objeto-relacional?

- Por defecto es una base de datos relacional
- Cuando se utiliza las características de objetos, se pasa a objeto-relacional

## Ejemplos

- Oracle
- Postgre-SQL







# Características base de datos objeto-relacional (Oracle)



## Tipos de objetos

- Clases



## Relación entre objetos

- Herencia



## Tablas de objetos

- Cada fila representa un objeto



## Tablas con columnas de objetos

- La columna guarda un objeto



## Tablas con columnas de arrays

- La columna guarda una colección de elementos



## Tablas con columnas de tablas

- La columna guarda una tabla anidada

# Tipos de objetos



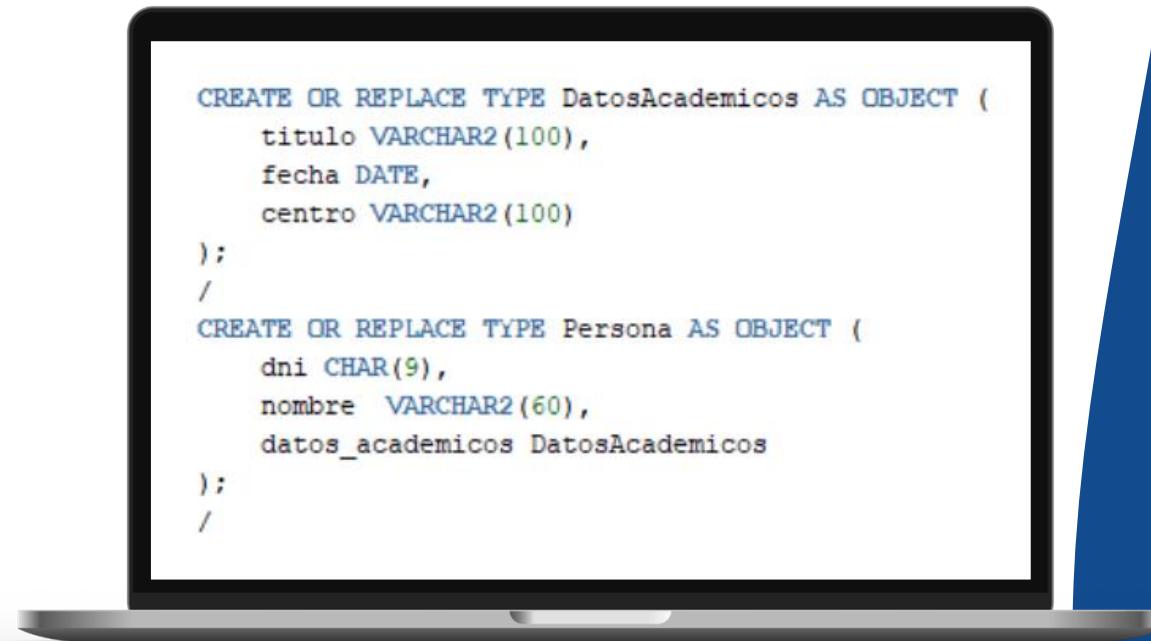
## Los objetos

- Es la base de la creación de objetos
- Es una estructura muy simple, definida por un nombre del tipo y luego puede contener atributos y métodos
- Se suele utilizar en PL/SQL



## Sintaxis

```
CREATE TYPE nombre AS OBJECT (  
    Atributo Tipo,  
    ...  
)
```



# Herencia



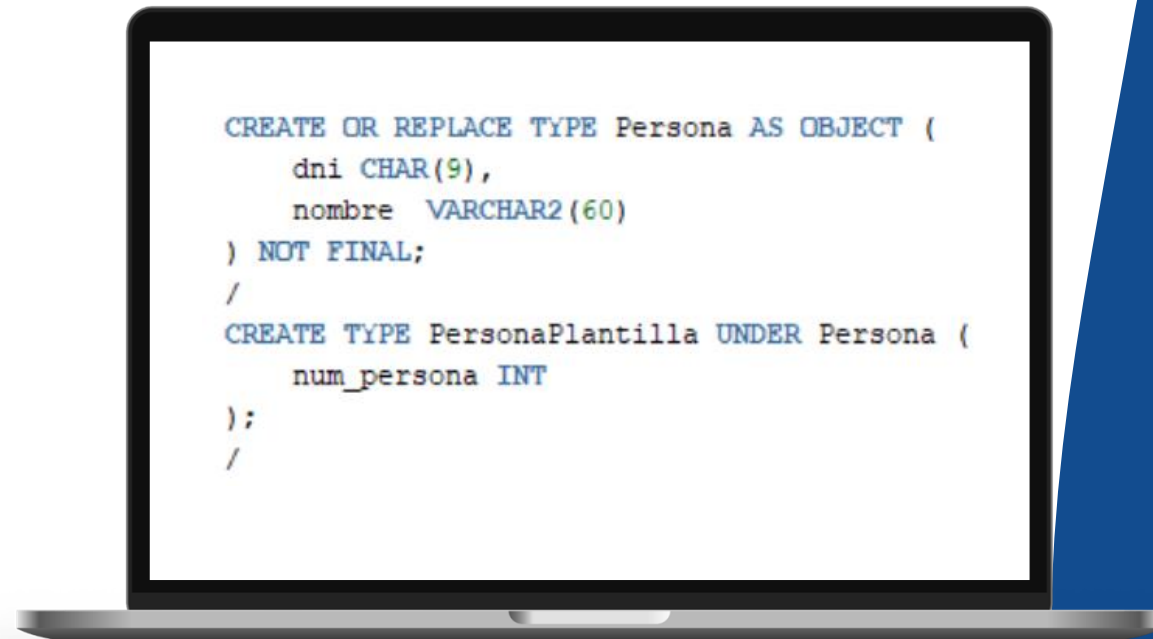
## La Herencia

- Es herencia simple de tipos
- Misma utilidad que orientado a objetos
- Es necesario definir un objeto padre



## Sintaxis

- El objeto padre debe de incluir NOT FINAL
- Los objetos hijos deben de utilizar UNDER



# Objetos por filas y por columnas

## Por filas (Row object)

- Cada fila representa un objeto
- Sintaxis: CREATE TABLE nombre OF nombre objeto

## Por columnas (Object column)

- Cada columna representa un objeto
- Sintaxis: en cada atributo especificar como tipo el nombre del objeto

```
CREATE TABLE personas OF Persona (  
    dni PRIMARY KEY,  
    nombre NOT NULL  
);  
  
INSERT INTO personas VALUES (Persona('11223344X','Pepe'));
```

```
CREATE TABLE personas (  
    referencia INT PRIMARY KEY,  
    persona Persona not null  
);  
  
INSERT INTO personas VALUES (1, Persona('11223344X','Pepe'));
```





# Referencias de objetos

## Las referencias

- Cada objeto tiene una referencia (OID)
- Permite que ese objeto pueda ser usado en diferentes tablas.
- Si se modifica en una tabla, se modifica en todas.

## Procedimiento

- Usar REF nombreObjeto para especificar que se guardara como referencia como campo de un tipo
- Usar SCOPE IS para especificar el ámbito de donde obtenerlo

```
CREATE OR REPLACE TYPE Actividad AS OBJECT (  
    nombre VARCHAR2(100),  
    descripcion VARCHAR2(100),  
    supervisor REF Persona  
);  
/  
CREATE TABLE actividades OF Actividad (  
    nombre NOT NULL,  
    descripcion NOT NULL,  
    supervisor SCOPE IS personas  
);
```

## Consulta de un campo con referencias y sus valores

- Con REF podemos ver el OID del objeto
- Con Deref podemos ver los valores del objeto

```
SELECT REF(a)  
FROM actividades a;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0,001 segundos

REF(A)
SYSTEM.ACTIVIDAD('Buceo', 'Iniciacion al buceo en piscina', 'oracle.sql.REF@d0df48e2')

```
SELECT Deref(a.supervisor)  
FROM actividades a;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0,004

Deref(A.SUPERVISOR)
1 SYSTEM.PERSONA('11223344X', 'Pepe')



# Arrays y tablas anidadas

## Arrays

- Uso igual que en los lenguajes de programación.
- Sintaxis: `VARRAY(tamaño) OF tipo`

```
CREATE OR REPLACE TYPE Telefono AS VARRAY(2) OF CHAR(9);
/
CREATE OR REPLACE TYPE Persona AS OBJECT (
    dni CHAR(9),
    nombre VARCHAR2(60),
    telefonos Telefono
) NOT FINAL;
/
CREATE TABLE personas OF Persona (
    dni PRIMARY KEY,
    nombre NOT NULL
);
```

```
INSERT INTO personas
VALUES (Persona('22334455Y',
               'Maria',
               Telefono('600111222', '700111222')));
```

## Tablas anidadas

- Una columna con una tabla dentro
- Procedimiento:
  1. Crear un tipo con referencia a una tabla
  2. Crear una tabla con una columna con el tipo objeto tabla

```
CREATE OR REPLACE TYPE Alumno AS TABLE OF REF Persona;
/
CREATE OR REPLACE TYPE Curso AS OBJECT (
    nombre VARCHAR2(50),
    fecha DATE,
    alumnos Alumno
);
/
CREATE TABLE cursos OF Curso (
    nombre PRIMARY KEY
) NESTED TABLE alumnos STORE AS alumnos_tab;
```

```
INSERT INTO cursos
VALUES (Curso('Informatica',
            null,
            Alumno((SELECT REF(e)
                    FROM personas e))
            )
);
```



# Resumen

1. Base de datos objetos
2. Características base de datos objetos
3. Estándar ODMG
4. ODL
5. OQL
6. Base de datos objeto-relacional
7. Características base de datos Objeto-relacional (Oracle)
8. Tipos de objetos
9. Herencia
10. Objetos por filas y por columnas
11. Referencias de objetos
12. Arrays y tablas anidadas

The background is a solid blue color. Overlaid on this are several faint, light-blue geometric patterns. A prominent feature is a grid of small squares that forms a stylized world map, centered behind the text. Scattered across the entire background are numerous small, light-blue arrows of varying sizes and orientations, some pointing towards the center and others away from it, creating a sense of movement and direction.

# UNIVERSAE

— CHANGE YOUR WAY —