

Ventajas de AMDP

1. **Alto rendimiento:** Ejecución directa en la base de datos
2. **Reutilización:** Pueden ser llamados desde múltiples programas ABAP
3. **Mantenibilidad:** Código centralizado y versionado
4. **Seguridad:** Integrado con el concepto de transporte de SAP
5. **Compatibilidad:** Soporte para múltiples bases de datos (HANA, Oracle, etc.)

Estructura Básica de un AMDP

```
CLASS zcl_amdp_example DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_amdp_marker_hdb.

    CLASS-METHODS:
      get_customer_data
        IMPORTING VALUE(iv_client) TYPE mandt
                  VALUE(iv_country) TYPE land1
        EXPORTING VALUE(et_data) TYPE ztt_customer_data
        RAISING cx_amdp_error.
  ENDClass.

CLASS zcl_amdp_example IMPLEMENTATION.
  METHOD get_customer_data BY DATABASE PROCEDURE
    FOR HDB
    LANGUAGE SQLSCRIPT
    OPTIONS READ-ONLY
    USING kna1.

    et_data = SELECT mandt, kunnr, name1, ort01, land1
      FROM kna1
      WHERE mandt = :iv_client
        AND land1 = :iv_country;

  ENDMETHOD.
ENDCLASS.
```

Ejemplo Real: Procesamiento Complejo de Datos

```

CLASS zcl_amdp_sales_analysis IMPLEMENTATION.
  METHOD calculate_sales_kpi BY DATABASE PROCEDURE
    FOR HDB
    LANGUAGE SQLSCRIPT
    USING vbak vbap kna1.

    -- Declaración de variables locales
    DECLARE lv_total_sales DECIMAL(15,2);
    DECLARE lv_avg_order_value DECIMAL(15,2);

    -- Tabla temporal para resultados intermedios
    lt_sales_data = SELECT
      k.mandt, k.kunnr, k.name1,
      v.vkorg, v.netwr, v.waerk
    FROM vbak AS v
    JOIN vbap AS p ON v.mandt = p.mandt
      AND v.vbeln = p.vbeln
    JOIN kna1 AS k ON v.mandt = k.mandt
      AND v.kunnr = k.kunnr
    WHERE v.gjahr = :iv_year
      AND v.quarter = :iv_quarter;

    -- Cálculos complejos
    lv_total_sales = SELECT SUM(netwr) FROM :lt_sales_data;
    lv_avg_order_value = SELECT AVG(netwr) FROM :lt_sales_data;

    -- Resultados finales
    et_kpi_results = SELECT
      mandt, vkorg,
      SUM(netwr) AS org_sales,
      COUNT(DISTINCT kunnr) AS customer_count,
      :lv_total_sales AS total_sales,
      :lv_avg_order_value AS avg_order_value
    FROM :lt_sales_data
    GROUP BY mandt, vkorg;

  ENDMETHOD.
ENDCLASS.

```

Comparación entre AMDP, CDS y ABAP Clásico

Característica	AMDP	CDS (Core Data Services)	ABAP Clásico
Lugar de ejecución	Base de datos	Base de datos	Servidor de aplicaciones
Lenguaje	SQLScript (HANA)	SQL extendido	ABAP
Rendimiento	Muy alto	Alto	Medio/Bajo
Complejidad lógica	Alta (procedimientos)	Media (consultas)	Alta (cualquier lógica)
Reutilización	Media (desde ABAP)	Alta (vistas)	Alta
Uso recomendado	Procesamiento complejo	Modelado de datos	Lógica de aplicación
Mantenibilidad	Buena (centralizado)	Excelente	Depende de implementación

Tips para Implementación con AMDP

1. Optimización para HANA:

- Aprovecha las tablas temporales (DECLARE TABLE) para procesamiento intermedio
- Usa paralelización con CE_COLUMN_TABLE cuando sea posible

2. Buenas prácticas:

- Documenta siempre los parámetros y lógica interna
- Divide procedimientos complejos en varios métodos más pequeños
- Maneja errores con excepciones específicas

3. Integración con ABAP:

- Usa tipos de datos ABAP para los parámetros de entrada/salida
- Considera encapsular llamadas AMDP en métodos de fachada ABAP

4. Pruebas:

- Desarrolla pruebas unitarias que cubran los casos límite
- Verifica el rendimiento con volúmenes reales de datos

Ejemplo Avanzado: Unión de AMDP y CDS

```
-- CDS View (zcds_sales_data)
@AbapCatalog.sqlViewName: 'ZCDS_SALES'
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Sales data for analysis'
define view zcds_sales_data as select from vbak
    join vbap on vbak.mandt = vbap.mandt
                and vbak.vbeln = vbap.vbeln {
    key vbak.mandt,
    key vbak.vbeln,
    vbak.erdat,
    vbak.kunnr,
    vbap.matnr,
    vbap.werks,
    vbap.netwr
}

-- AMDP que usa la CDS View
CLASS zcl_amdp_enhanced_analysis IMPLEMENTATION.
    METHOD analyze_sales BY DATABASE PROCEDURE
                        FOR HDB
                        LANGUAGE SQLSCRIPT
                        USING zcds_sales_data.

        et_results = SELECT
                        matnr, werks,
                        SUM(netwr) AS total_sales,
                        COUNT(DISTINCT vbeln) AS order_count
        FROM zcds_sales_data
        GROUP BY matnr, werks
        ORDER BY total_sales DESC;

    ENDMETHOD.
ENDCLASS.
```

Conclusión y Recomendaciones

Cuando usar AMDP:

- Procesamiento complejo de grandes volúmenes de datos
- Cálculos intensivos que benefician de ejecución en BD
- Operaciones que requieren SQL avanzado o específico de HANA

Cuando preferir CDS:

- Modelado de datos para consumo de aplicaciones UI5/Fiori
- Consultas relativamente simples con necesidad de reutilización
- Cuando se necesita exposición OData directa

Cuando mantener ABAP clásico:

- Lógica de aplicación compleja con múltiples pasos
- Integración con otros módulos de SAP
- Cuando se necesita flexibilidad completa

Recomendación final: Combina estas tecnologías según las necesidades específicas de cada escenario para obtener el mejor balance entre rendimiento y mantenibilidad.