

# CAPÍTULO 4: PROGRAMACIÓN PERSISTENTE

## 4.1 VISTAS

### 4.1.1 Fundamentos Teóricos

#### Definición

Una Vista (View) es un objeto de base de datos que representa una tabla virtual basada en el resultado de una consulta SELECT. A diferencia de las tablas base, las vistas no almacenan datos físicamente (excepto las vistas materializadas); en su lugar, almacenan la definición de la consulta y generan los datos dinámicamente cuando se accede a ellas.

#### Características Principales

1. **Tabla Virtual:** La vista actúa como una tabla, pero no contiene datos propios
2. **Consulta Almacenada:** Guarda la definición SQL, no los resultados
3. **Actualización Dinámica:** Refleja automáticamente los cambios en las tablas base
4. **Capa de Abstracción:** Oculta la complejidad de las consultas subyacentes
5. **Seguridad:** Permite restringir el acceso a columnas o filas específicas

#### Tipos de Vistas

##### 1. Vistas Simples

- Basadas en una sola tabla
- Pueden ser actualizables (INSERT, UPDATE, DELETE)
- No contienen funciones de grupo, GROUP BY, DISTINCT

##### 2. Vistas Complejas

- Basadas en múltiples tablas (JOIN)
- Contienen funciones de grupo, expresiones calculadas
- Generalmente no son actualizables

##### 3. Vistas Materializadas (Oracle)

- Almacenan físicamente los datos resultantes
- Mejoran el rendimiento en consultas complejas
- Requieren estrategias de refresco (REFRESH)

## 4.1.2 Sintaxis y Creación

### Oracle Database

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW nombre_vista
[(alias_columna1, alias_columna2, ...)]
AS
    SELECT ...
[WITH CHECK OPTION [CONSTRAINT nombre_constraint]]
[WITH READ ONLY [CONSTRAINT nombre_constraint]];
```

#### Parámetros:

- **OR REPLACE:** Reemplaza la vista si ya existe
- **FORCE:** Crea la vista aunque las tablas base no existan
- **NOFORCE:** Solo crea si las tablas base existen (por defecto)
- **WITH CHECK OPTION:** Impide modificaciones que violen la condición WHERE
- **WITH READ ONLY:** Hace la vista no actualizable

### SQL Server

```
CREATE [OR ALTER] VIEW nombre_vista
[(alias_columna1, alias_columna2, ...)]
[WITH ENCRYPTION | SCHEMABINDING | VIEW_METADATA]
AS
    SELECT ...
[WITH CHECK OPTION];
```

#### Parámetros:

- **OR ALTER:** Modifica la vista si existe (SQL Server 2016+)
- **WITH ENCRYPTION:** Encripta la definición de la vista
- **SCHEMABINDING:** Liga la vista a las tablas base
- **VIEW\_METADATA:** Retorna metadata de la vista, no de las tablas base

### 4.1.3 Ejemplos Prácticos

**Contexto:** Sistema de Universidad con las siguientes tablas:

```
-- ORACLE / SQL SERVER
```

```
CREATE TABLE ESTUDIANTES (  
    est_id NUMBER(6) PRIMARY KEY,  
    est_nombre VARCHAR2(50),  
    est_apellido VARCHAR2(50),  
    est_carrera VARCHAR2(30),  
    est_creditos NUMBER(3),  
    est_promedio NUMBER(4,2)  
);
```

```
CREATE TABLE CURSOS (  
    curso_id NUMBER(4) PRIMARY KEY,  
    curso_nombre VARCHAR2(60),  
    curso_creditos NUMBER(2),  
    curso_departamento VARCHAR2(30)  
);
```

```
CREATE TABLE MATRICULAS (  
    mat_id NUMBER(8) PRIMARY KEY,  
    est_id NUMBER(6),  
    curso_id NUMBER(4),  
    mat_nota NUMBER(4,2),  
    mat_periodo VARCHAR2(10),  
    FOREIGN KEY (est_id) REFERENCES ESTUDIANTES(est_id),  
    FOREIGN KEY (curso_id) REFERENCES CURSOS(curso_id)  
);
```

## EJEMPLO 1: Vista Simple - Estudiantes de Sistemas

### Oracle:

```
CREATE OR REPLACE VIEW V_ESTUDIANTES_SISTEMAS AS
SELECT est_id,
       est_nombre || ' ' || est_apellido AS nombre_completo,
       est_creditos,
       est_promedio
FROM ESTUDIANTES
WHERE est_carrera = 'Sistemas'
WITH READ ONLY;
```

```
-- Consultar la vista
SELECT * FROM V_ESTUDIANTES_SISTEMAS
WHERE est_promedio >= 80;
```

### SQL Server:

```
CREATE OR ALTER VIEW V_ESTUDIANTES_SISTEMAS AS
SELECT est_id,
       est_nombre + ' ' + est_apellido AS nombre_completo,
       est_creditos,
       est_promedio
FROM ESTUDIANTES
WHERE est_carrera = 'Sistemas';
```

## EJEMPLO 2: Vista Compleja - Reporte de Matriculaciones

### Oracle:

```
CREATE OR REPLACE VIEW V_REPORTER_MATRICULAS AS
SELECT
    e.est_id,
    e.est_nombre || ' ' || e.est_apellido AS estudiante,
    c.curso_nombre,
    c.curso_creditos,
    m.mat_nota,
    m.mat_periodo,
    CASE
        WHEN m.mat_nota >= 70 THEN 'APROBADO'
        WHEN m.mat_nota < 70 AND m.mat_nota >= 60 THEN 'SUPLETORIO'
        ELSE 'REPROBADO'
    END AS estado
FROM ESTUDIANTES e
INNER JOIN MATRICULAS m ON e.est_id = m.est_id
INNER JOIN CURSOS c ON m.curso_id = c.curso_id;
```

```
-- Uso de la vista
SELECT estudiante, curso_nombre, mat_nota, estado
FROM V_REPORTER_MATRICULAS
WHERE mat_periodo = '2024-1' AND estado = 'APROBADO';
```

### EJEMPLO 3: Vista con Agregación - Estadísticas por Carrera

#### Oracle:

```
CREATE OR REPLACE VIEW V_ESTADISTICAS_CARRERA AS
SELECT
    est_carrera AS carrera,
    COUNT(*) AS total_estudiantes,
    ROUND(AVG(est_promedio), 2) AS promedio_general,
    MAX(est_promedio) AS promedio_maximo,
    MIN(est_promedio) AS promedio_minimo,
    ROUND(AVG(est_creditos), 0) AS creditos_promedio
FROM ESTUDIANTES
GROUP BY est_carrera;
```

```
-- Consulta
SELECT * FROM V_ESTADISTICAS_CARRERA
ORDER BY promedio_general DESC;
```

## EJEMPLO 4: Vista con WITH CHECK OPTION

### Oracle:

```
CREATE OR REPLACE VIEW V_ESTUDIANTES_DESTACADOS AS
SELECT est_id, est_nombre, est_apellido, est_promedio
FROM ESTUDIANTES
WHERE est_promedio >= 85
WITH CHECK OPTION CONSTRAINT chk_promedio_destacado;
```

```
-- Esto funcionará
INSERT INTO V_ESTUDIANTES_DESTACADOS
VALUES (1001, 'María', 'González', 90.5);
```

```
-- Esto generará ERROR (viola WITH CHECK OPTION)
INSERT INTO V_ESTUDIANTES_DESTACADOS
VALUES (1002, 'Juan', 'Pérez', 75.0);
```

## 4.1.4 Operaciones DML sobre Vistas

### Vistas Actualizables - Reglas

Una vista es actualizable si cumple:

1. No contiene funciones de agregación (SUM, AVG, COUNT, etc.)
2. No contiene GROUP BY, HAVING, DISTINCT
3. No contiene operadores de conjunto (UNION, INTERSECT, MINUS)
4. No contiene subconsultas correlacionadas
5. Todas las columnas NOT NULL de la tabla base están en la vista

### Ejemplo de Vista Actualizable:

```
-- Oracle
CREATE OR REPLACE VIEW V_ESTUDIANTES_EDITABLE AS
SELECT est_id, est_nombre, est_apellido, est_carrera
FROM ESTUDIANTES;
```

```
-- Operaciones permitidas
UPDATE V_ESTUDIANTES_EDITABLE
SET est_carrera = 'Electrónica'
WHERE est_id = 1001;
```

```
DELETE FROM V_ESTUDIANTES_EDITABLE
WHERE est_id = 1002;
```



### 4.1.5 Vistas Materializadas (Oracle)

```
-- Creación de Vista Materializada
CREATE MATERIALIZED VIEW MV_RESUMEN_CURSOS
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS
SELECT
    c.curso_departamento,
    COUNT(*) AS total_cursos,
    SUM(c.curso_creditos) AS creditos_totales,
    COUNT(DISTINCT m.est_id) AS estudiantes_unicos
FROM CURSOS c
LEFT JOIN MATRICULAS m ON c.curso_id = m.curso_id
GROUP BY c.curso_departamento;
```

```
-- Refrescar manualmente
EXEC DBMS_MVIEW.REFRESH('MV_RESUMEN_CURSOS');
```

## 4.1.6 Gestión de Vistas

### Consultar Vistas Existentes

#### Oracle:

```
-- Ver todas las vistas del usuario
SELECT view_name, text
FROM USER_VIEWS;
```

```
-- Ver definición de una vista específica
SELECT text FROM USER_VIEWS
WHERE view_name = 'V_ESTUDIANTES_SISTEMAS';
```

#### SQL Server:

```
-- Ver todas las vistas
SELECT name, create_date, modify_date
FROM sys.views;
```

```
-- Ver definición de una vista
EXEC sp_helptext 'V_ESTUDIANTES_SISTEMAS';
```

### Eliminar Vistas

```
-- Oracle y SQL Server
DROP VIEW V_ESTUDIANTES_SISTEMAS;
```

## 4.1.7 Ventajas y Desventajas

### Ventajas

- **Seguridad:** Restringe acceso a datos sensibles
- **Simplificación:** Oculta consultas complejas
- **Independencia lógica:** Cambios en tablas no afectan aplicaciones
- **Reutilización:** Lógica de negocio centralizada
- **Personalización:** Diferentes perspectivas de los datos

### Desventajas

- **Rendimiento:** Pueden ser más lentas que tablas
- **Limitaciones DML:** No todas son actualizables
- **Dependencias:** Cambios en tablas base pueden invalidarlas
- **Complejidad:** Vistas sobre vistas pueden ser difíciles de mantener

### 4.1.8 Ejercicios Propuestos

#### **EJERCICIO 1: Crear las siguientes vistas simples**

- a) Vista que muestre estudiantes con promedio superior a 80
- b) Vista de cursos del departamento de 'Computación'
- c) Vista de estudiantes que han completado más de 100 créditos

#### **EJERCICIO 2: Crear vistas complejas**

- d) Vista que muestre el top 10 de estudiantes con mejor promedio y el número total de cursos matriculados
- e) Vista que liste cursos con su tasa de aprobación (porcentaje de estudiantes aprobados)
- f) Vista que muestre estudiantes y el total de créditos matriculados en el periodo actual

#### **EJERCICIO 3: Implementar WITH CHECK OPTION**

Crear una vista de cursos de 3 créditos que impida insertar cursos con diferente cantidad de créditos.

#### **EJERCICIO 4: Operaciones DML**

- g) Intentar actualizar datos a través de la vista V\_REPORTE\_MATRICULAS  
¿Qué sucede?
- h) Crear una vista actualizable y realizar INSERT, UPDATE, DELETE
- i) Explicar por qué algunas operaciones fallan

#### **EJERCICIO 5: Vista Materializada (Oracle)**

Crear una vista materializada que calcule estadísticas mensuales de matriculación con refresco automático diario.