

Nombre de la práctica	Vista			No.	1
Asignatura:	Taller de Base de datos	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	

NOMBRE DEL ALUMNO: Vanesa Hernández Martínez
GRUPO: 3501

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):
Actividades en aula de clases y en equipo personal

III. Material empleado:

- Laptop
- Navicat

Vistas

1. Vista Simple con Información de la tienda

Creación de la vista

CREATE VIEW nombre_vista AS

SELECT name, status FROM store;

Descripción: Esta línea crea una vista llamada nombre_vista, que contiene únicamente los campos name (nombre) y status (estado) de cada registro de la tabla store.

Objetivo: La vista permite obtener el nombre y el estado de las tiendas sin tener que escribir el SELECT name, status FROM store cada vez. Esto es útil si esta consulta se necesita frecuentemente.

Consulta de la vista

SELECT * FROM nombre_vista;

Descripción: Con esta consulta, se obtiene directamente el contenido de la vista nombre_vista, mostrando solo los nombres y los estados de todas las tiendas registradas en la tabla store.

Resultado: Se obtiene una lista de todas las tiendas con sus respectivos estados, proporcionando una forma rápida y ordenada de acceder a estos datos.

2. Crear y Usar una Vista Simple

Sin Vista

SELECT

a.username AS admin_name,

s.name AS store_name,

s.status AS store_status

FROM administrator AS a

JOIN store AS s ON s.id_administrator = a.id_administrator;

Descripción: Este query selecciona el nombre de usuario de un administrador, el nombre de la tienda y el estado de la tienda. Se utiliza un JOIN para combinar las tablas administrator y store mediante el campo id_administrator.

Resultado: Muestra el nombre del administrador, el nombre de la tienda y su estado para cada combinación que coincida.

Con Vista

CREATE VIEW admin_store_info AS

SELECT

a.username AS admin_name,

s.name AS store_name,

s.status AS store_status

FROM administrator AS a

JOIN store AS s ON s.id_administrator = a.id_administrator;

SELECT * FROM admin_store_info;

Descripción: Se crea una vista admin_store_info que guarda el query anterior. La vista permite acceder a la información de las tiendas y los administradores sin tener que repetir la consulta completa cada vez.

Resultado: Al ejecutar `SELECT * FROM admin_store_info;`, se obtiene el mismo resultado que en la consulta sin vista, pero de forma más simplificada.

3. Vista para Mostrar Cliente, Teléfono, Tienda y Puntos

CREATE VIEW ejercicio1 AS

SELECT c.id_client, c.phone, s.name, cp.points

FROM client AS c

JOIN card_points AS cp ON c.id_client = cp.id_client

JOIN store AS s ON cp.id_store = s.id_store;

SELECT * FROM ejercicio1;

Descripción: Esta vista ejercicio1 muestra el id_client, el phone del cliente, el name de la tienda y los points que tiene el cliente en dicha tienda. Se usan dos JOINS para combinar las tablas client, card_points y store.

Resultado: Al ejecutar `SELECT * FROM ejercicio1;`, se obtiene una lista con el ID del cliente, su teléfono, el nombre de la tienda y los puntos acumulados en cada tienda específica.

4. Vista para Contar Clientes por Tienda

CREATE VIEW tienda_numero_clientes AS

SELECT

s.name AS name_store,

(SELECT COUNT(DISTINCT cp.id_client)

FROM card_points AS cp

WHERE cp.id_store = s.id_store) AS num_client

FROM

store AS s;

SELECT * FROM tienda_numero_clientes;

Descripción: La vista tienda_numero_clientes calcula el número de clientes únicos para cada tienda. Usa un COUNT en una subconsulta para contar clientes únicos (DISTINCT cp.id_client) en la tabla card_points filtrados por cada tienda (id_store).

Resultado: Al ejecutar SELECT * FROM tienda_numero_clientes;, se obtiene una lista de tiendas con la cantidad total de clientes únicos asociados a cada una.

5. Vista para el Monto Total de Ventas por Tienda

CREATE VIEW tienda_monto AS

SELECT

s.name AS store_name,

(SELECT SUM(t.amount)

FROM transaction t

JOIN card_points cp ON t.id_card = cp.id_card

WHERE cp.id_store = s.id_store) AS total_sales

FROM

store s;

SELECT * FROM tienda_monto;

Descripción: Esta vista tienda_monto calcula el monto total de ventas generadas en cada tienda. En una subconsulta, se utiliza SUM sobre el campo amount de la tabla transaction, combinándola con card_points para identificar las transacciones correspondientes a cada tienda (id_store).

Resultado: Al ejecutar SELECT * FROM tienda_monto;, se obtiene el nombre de cada tienda junto con el total de ventas generadas en ella.

Conclusión

El uso de vistas en SQL proporciona una poderosa herramienta para simplificar consultas complejas y mejorar la eficiencia del acceso a datos en bases de datos. Al definir una vista, encapsulamos una consulta que puede ser reutilizada fácilmente, reduciendo la necesidad de escribir repetidamente código largo o intrincado. Esto no solo mejora la legibilidad y mantenimiento del código, sino que también permite a los desarrolladores y usuarios enfocarse en los datos relevantes de manera estructurada y organizada.

En casos como los ejemplos presentados, las vistas facilitan el acceso a datos específicos, como la información de tiendas, clientes y ventas, permitiendo generar reportes y análisis sin complicaciones. Además, al estandarizar ciertas consultas, las vistas contribuyen a optimizar la gestión de datos y promueven un enfoque modular en el desarrollo de aplicaciones y reportes.