

Nombre de la práctica	LISTAS			No.	7
Asignatura:	Taller de Base de Datos	Carrera:	ISIC	Duración de la práctica (Hrs)	4

**Ana Edith Hernández Hernández**

**I. Realizar consultas con views en bases de datos**

**II. Competencia(s) específica(s):**

**III. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula**

**IV. Material empleado:**

Computadora

**V. Desarrollo de la práctica:**

**1.- Crearemos una base de datos llamada puntos en la cual realizaremos 6 tablas las cuales serán: - administrador**

- tiendas
- usuarios
- clientes
- tarjetas\_puntos
- transacciones

**A continuación, se muestra el procedimiento**

```

1 CREATE TABLE Administrador (
2     id_admin INT PRIMARY KEY AUTO_INCREMENT,
3     nombre VARCHAR(50) NOT NULL,
4     email VARCHAR(100) UNIQUE NOT NULL,
5     contraseña VARCHAR(255) NOT NULL,
6     fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
7     fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
8 );
9
10
11 CREATE TABLE Tiendas (
12     id_tienda INT PRIMARY KEY AUTO_INCREMENT,
13     nombre VARCHAR(100) NOT NULL,
14     direccion VARCHAR(150),
15     telefono VARCHAR(15),
16     id_admin INT NOT NULL,
17     fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
18     fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
19     FOREIGN KEY (id_admin) REFERENCES Administrador(id_admin) ON DELETE CASCADE
20 );
21
22 CREATE TABLE Usuarios (
23     id_usuario INT PRIMARY KEY AUTO_INCREMENT,
24     nombre VARCHAR(50) NOT NULL,
25     email VARCHAR(100) UNIQUE NOT NULL,
26     contraseña VARCHAR(255) NOT NULL,
27     id_tienda INT NOT NULL,
28     fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
29     fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
30     FOREIGN KEY (id_tienda) REFERENCES Tiendas(id_tienda) ON DELETE CASCADE
31 );
32
33 CREATE TABLE Clientes (
34     id_cliente INT PRIMARY KEY AUTO_INCREMENT,
35     nombre VARCHAR(50) NOT NULL,
36     email VARCHAR(100) UNIQUE,
37     telefono VARCHAR(15),
38     direccion VARCHAR(150),
39     fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
40     fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
41 );
42

```



```
1  --
2
3  CREATE TABLE Clientes (
4      id_cliente INT PRIMARY KEY AUTO_INCREMENT,
5      nombre VARCHAR(50) NOT NULL,
6      email VARCHAR(100) UNIQUE,
7      telefono VARCHAR(15),
8      direccion VARCHAR(150),
9      fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
10     fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
11 );
12
13 CREATE TABLE Tarjetas_Puntos (
14     id_tarjeta INT PRIMARY KEY AUTO_INCREMENT,
15     id_cliente INT NOT NULL,
16     id_tienda INT NOT NULL,
17     puntos INT DEFAULT 0,
18     fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
19     fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
20     FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente) ON DELETE CASCADE,
21     FOREIGN KEY (id_tienda) REFERENCES Tiendas(id_tienda) ON DELETE CASCADE
22 );
23
24 CREATE TABLE Transacciones (
25     id_transaccion INT PRIMARY KEY AUTO_INCREMENT,
26     id_tarjeta INT NOT NULL,
27     id_usuario INT NOT NULL,
28     monto DECIMAL(10, 2) NOT NULL,
29     puntos_otorgados INT DEFAULT 0,
30     fecha_transaccion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
31     FOREIGN KEY (id_tarjeta) REFERENCES Tarjetas_Puntos(id_tarjeta) ON DELETE CASCADE,
32 );
33
34
```

2.- Comenzaremos a realizar consultas con vistas con cada una de las instrucciones que piden

```
-- Creamos una vista con el nombre y estado de la tienda

CREATE VIEW nombre_vista AS
SELECT
name,
status
FROM tiendas;
```

Ejecución:



Message		Summary	Result 1
Data		Info	
name		status	
Clark LLC		L8lpZdwqt	
Liao Toy Company Limitec		7heBUINf6	
Jialun Industrial Company		sleu1ZGxs/	
Chi Ming Limited		Jado34HpF	
Bennett's LLC		iCEM73Aji	
Wong's Logistic Limited		5qBBQyen	
Lee Brothers Software LLC		rIS1tBI3Pz	
Wang Kee Electronic Com		MqaDSW0	
Chi Ming Limited		pjZzC0ZxG	

```
-- Crear una vista que muestre los detalles de los
-- administradores y las tiendas que administran incluyendo
-- el nombre de la tienda y estado en el que se encuentra.

CREATE VIEW vista_administrador_tiendas AS
SELECT
a.id_administrador,
a.username AS username,
s.name AS nombre_tienda,
s.status AS estado_tienda
FROM administrador AS a
JOIN tiendas AS s ON s.id_administrador = a.id_administrador;
```

Ejecución:



Message

Summary

Result 1

Data

Info

id_administrador	username	nombre_tienda	estado_tienda
1239dcb0-f7a8-89b5-2fbb-	Amber West	Luo Kee Development & Ti	hxlxr6Sdnl
1239dcb0-f7a8-89b5-2fbb-	Amber West	Tiffany Engineering Inc.	Bsh9Rzpytm
1239dcb0-f7a8-89b5-2fbb-	Amber West	Aguilar Inc.	zKfXSFC9q6
1239dcb0-f7a8-89b5-2fbb-	Amber West	Liu Kee Industrial Compan	MqDgl0xiDY
1239dcb0-f7a8-89b5-2fbb-	Amber West	Lok's Telecommunication	epFgWySqe6
1239dcb0-f7a8-89b5-2fbb-	Amber West	Sheila LLC	lkC77fQ6Vb
1239dcb0-f7a8-89b5-2fbb-	Amber West	Song Engineering Compar	Ab1U8gX4eQ
2bac851e-9695-5650-9b94-	Eva Murphy	Yeung's Development & Ti	jFCIJ7CY8N

```
-- Crear una vista que muestre el id del cliente su telefono,  
-- el nombre de la tienda y cuantos puntos tiene ese cliente  
-- en esa tienda
```

```
CREATE VIEW visita_cliente AS  
SELECT  
c.id_client,  
c.phone AS telefono,  
s.name AS nombre_tienda,  
ca.points AS puntos  
FROM clientes AS c  
JOIN tarjetas_puntos AS ca ON c.id_client = ca.id_client  
JOIN tiendas AS s ON s.id_store = s.id_store;
```

Ejecución:



MessageSummaryResult 1

DataInfo

id_client	telefono	nombre_tienda	puntos
24441e01-e3b9-b745-1c7c-	614-004-0678	Yunxi Toy Company Limite	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Riku Corporation	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Wing Sze Limited	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Harada Communications (	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Shi Kee Company Limited	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Charlotte Inc.	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Feng Company Limited	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Peng Telecommunication	389.09
24441e01-e3b9-b745-1c7c-	614-004-0678	Kwan's Limited	389.09

```
-- Crea una vista que muestre los nombres de las tiendas y  
-- la cantidad de clientes que tiene cada una
```

```
CREATE VIEW vista_tiendas_clientes AS  
SELECT  
s.name AS nombre_tienda,  
COUNT(DISTINCT cp.id_client) AS cantidad_clientes  
FROM tiendas s  
JOIN tarjetas_puntos cp ON s.id_store = cp.id_store  
WHERE cp.status = 1 GROUP BY s.name;
```

Ejecución:





Message	Summary	Result 1
Data	Info	
nombre_tienda	cantidad_clientes	
Aguilar Inc.	5	
Arai Electronic Corporator	6	
Au Kee Property Limited	2	
Ayato Consultants Corpora	5	
Barry Inc.	6	
Bennett's LLC	5	
Brian LLC	3	
Cao Kee Electronic Compa	5	

```
-- crea una vista que muestre el nombre de cada tienda y el  
-- monto total de ventas generadas en ellas
```

```
CREATE VIEW vista_tienda_ventas AS  
SELECT  
s.name AS nombre_tienda,  
SUM(t.amount) AS monto_total_ventas  
FROM tiendas s  
JOIN tarjetas_puntos cp ON s.id_store = cp.id_store  
JOIN transacciones t ON cp.id_card = t.id_card  
WHERE t.amount > 0 GROUP BY s.name;
```

Ejecución:



Message	Summary	Result 1
Data	Info	
nombre_tienda	monto_total_ventas	
Hsuan's Toy Limited	414827.83	
Barry Inc.	509615.20	
Ricky Inc.	464587.77	
Arai Electronic Corporator	453790.63	
Lee Brothers Software LLC	740211.23	
Phillips's Electronic LLC	470691.23	
Ikeda Electronic Corporatic	306294.42	
Zhiyuan Communications	568447.32	
Wei Kee Engineering Com	671663.58	

```
CREATE VIEW prueba AS
SELECT first_name,last_name FROM actor
WHERE YEAR (last_update)>2020;
```

## Ejecución:

Message	Summary	Result 1
Data	Info	
first_name	last_name	
Mark Kelly	Mark Kelly	
Lei Rui	Lei Rui	
Douglas Phillip	Douglas Phillip	
Sakamoto Tsul	Sakamoto Tsu	
Deng Yunxi	Deng Yunxi	
He Jialun	He Jialun	
Cho Tsz Hin	Cho Tsz Hin	
Morita Ikki	Morita Ikki	

## VI. Conclusiones:

Las vistas en bases de datos, conocidas como **\*\*views\*\***, son herramientas potentes y versátiles que permiten simplificar consultas complejas y organizar de manera efectiva la presentación de datos. Una vista actúa como una "tabla virtual" que se construye sobre una consulta predefinida, lo cual facilita el acceso a datos específicos sin necesidad de interactuar directamente con las tablas originales. Este enfoque es particularmente útil en escenarios donde los usuarios necesitan consultar grandes volúmenes de datos, ya que las vistas permiten reducir la complejidad de las consultas al ofrecer un conjunto de datos previamente filtrado y estructurado. En términos de seguridad, las vistas son fundamentales para limitar el acceso a datos sensibles, ya que pueden configurarse para mostrar solo los campos o registros autorizados para ciertos usuarios, protegiendo así la información confidencial. Además, las vistas contribuyen a la consistencia de los datos que se presentan a los usuarios, ya que al actualizarse automáticamente cuando cambian los datos de las tablas subyacentes, siempre muestran información actualizada sin requerir mantenimiento adicional. En resumen, las vistas no solo facilitan el acceso y la gestión de la información, sino que también optimizan el rendimiento y aumentan la seguridad en la administración de bases de datos sin afectar la estructura original de las tablas.