



ESTUDIANTES: Guachamin Shiguango Josué Andres

Rodríguez Romero Jhon Wilmer

Vallejo Rengifo John David

Villacis Silva Stefano Mateo

Yugcha Quishpe Erick Geomar

NIVEL: Quinto

PARALELO: “A”

FECHA: 22/09/2025

TEMA: Prueba Practica

DOCENTE: Ing. José Caiza, Mg.

CARRERA: Tecnologías de la Información

ASIGNATURA: Sistemas de Bases de Datos Distribuidas

PRUEBA PRACTICA PRIMER PARCIAL: FRAGMENTACIÓN VERTICAL + HORIZONTAL (MIXTA) CON VISTA GLOBAL

Objetivo

Diseñar e implementar, desde cero, una BD que aplique **fragmentación mixta** (horizontal + vertical) sobre la tabla Alumnos, simulando varios sitios en **una sola instancia**, y exponer una **vista global** que reconstruya los datos de forma transparente (union de fragmentos horizontales y join de fragmentos verticales).

Requisitos previos

- SQL Server instalado. Se recomienda trabajar en una **nueva instancia** (p. ej., SITIO_A) según la guía de APE 1 (instancias independientes y funcionales).

Escenario

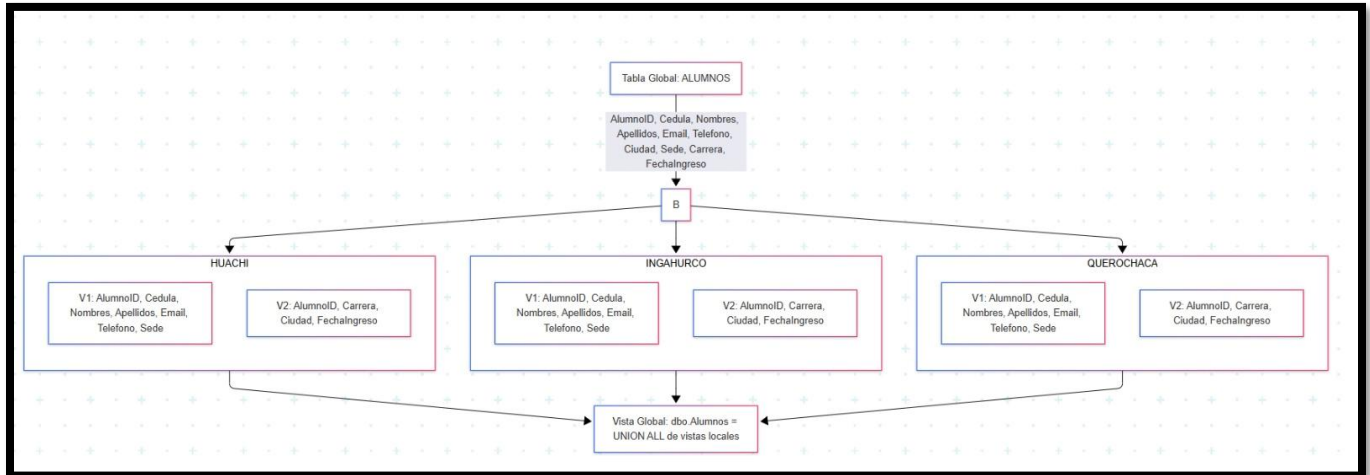
Simula tres “sedes” como **esquemas**: HUACHI, INGAHURCO, QUEROCHACA.

Tabla global conceptual: Alumnos(AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Ciudad, Sede, Carrera, FechaIngreso).

- **Fragmentación horizontal** por Sede → cada sede almacena solo sus alumnos.
- **Fragmentación vertical** dentro de cada sede:
 - Fragmento V1 (identificación/contacto): AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede
 - Fragmento V2 (académico): AlumnoID, Carrera, Ciudad, FechaIngreso(Ambos fragmentos **incluyen AlumnoID** para permitir JOIN).

Entregables del estudiante (un solo PDF + .sql)

1. Diseño de fragmentación (1 página): diagrama simple que muestre *horizontal* por sede y *vertical* en V1/V2 (justifica por qué tu fragmentación **preserva reconstrucción** con JOIN y UNION ALL).



JUSTIFICACIÓN: La tabla global conceptual Alumnos fue fragmentada horizontalmente por el atributo Sede, de manera que cada sede (HUACHI, INGAHURCO y QUEROCHACA) contiene únicamente los registros de sus estudiantes. Dentro de cada sede, se aplicó fragmentación vertical, dividiendo los atributos en dos fragmentos:

- V1 (identificación y contacto): AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede.
- V2 (académico): AlumnoID, Carrera, Ciudad, FechaIngreso.

El atributo AlumnoID se preserva en ambos fragmentos para permitir la reconstrucción mediante un JOIN. Luego, cada sede expone una vista local que une sus fragmentos verticales ($V1 \bowtie V2$). Finalmente, la vista global **dbo.Alumnos** integra todas las vistas locales con un UNION ALL, garantizando transparencia y preservación de la tabla original sin pérdida de información.

2. Script .sql con todo lo ejecutado (creación BD, esquemas, tablas, datos, vistas).

USE master;

IF DB_ID('BD_ALUMNOS') IS NOT NULL

BEGIN

ALTER DATABASE BD_ALUMNOS SET SINGLE_USER WITH ROLLBACK IMMEDIATE;

DROP DATABASE BD_ALUMNOS;

END;

GO

CREATE DATABASE BD_ALUMNOS;

GO

USE BD_ALUMNOS;

GO

CREATE SCHEMA HUACHI

GO

CREATE SCHEMA INGAHURCO

GO

CREATE SCHEMA QUEROCHACA

GO

CREATE SEQUENCE HUACHI.SeqAlumnoID AS INT START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE INGAHURCO.SeqAlumnoID AS INT START WITH 1001 INCREMENT BY 1;

CREATE SEQUENCE QUEROCHACA.SeqAlumnoID AS INT START WITH 2001 INCREMENT BY 1;

GO

-- HUACHI

CREATE TABLE HUACHI.Alumnos_V1 (

AlumnoID INT NOT NULL PRIMARY KEY,

Cedula CHAR(10) NOT NULL,

Nombres NVARCHAR(50) NOT NULL,

Apellidos NVARCHAR(50) NOT NULL,

Email NVARCHAR(100) NOT NULL,

Telefono NVARCHAR(20) NULL,

Sede NVARCHAR(20) NOT NULL

CONSTRAINT DF_HUACHI_V1_Sede DEFAULT (N'HUACHI'),

```

        CONSTRAINT CK_HUACHI_V1_Sede CHECK (Sede = N'HUACHI'),
        CONSTRAINT UQ_HUACHI_Cedula UNIQUE (Cedula)
    );

CREATE TABLE HUACHI.Alumnos_V2 (
    AlumnoID INT NOT NULL PRIMARY KEY,
    Carrera NVARCHAR(60) NOT NULL,
    Ciudad NVARCHAR(60) NOT NULL,
    FechaIngreso DATE NOT NULL,
    CONSTRAINT FK_HUACHI_V2_V1 FOREIGN KEY (AlumnoID)
        REFERENCES HUACHI.Alumnos_V1 (AlumnoID) ON DELETE CASCADE
    );

-- Índices de consulta
CREATE INDEX IX_HUACHI_V1_Sede ON HUACHI.Alumnos_V1 (Sede) INCLUDE (Nombres,
Apellidos);
CREATE INDEX IX_HUACHI_V2_Carrera ON HUACHI.Alumnos_V2 (Carrera) INCLUDE
(Ciudad, FechaIngreso);
GO

-----
-- INGAHURCO
-----

CREATE TABLE INGAHURCO.Alumnos_V1 (
    AlumnoID INT NOT NULL PRIMARY KEY,
    Cedula CHAR(10) NOT NULL,
    Nombres NVARCHAR(50) NOT NULL,
    Apellidos NVARCHAR(50) NOT NULL,
    Email NVARCHAR(100) NOT NULL,
    Telefono NVARCHAR(20) NULL,
    Sede NVARCHAR(20) NOT NULL
    CONSTRAINT DF_INGA_V1_Sede DEFAULT (N'INGAHURCO'),
    CONSTRAINT CK_INGA_V1_Sede CHECK (Sede = N'INGAHURCO'),
    CONSTRAINT UQ_INGA_Cedula UNIQUE (Cedula)
    );

```

```

CREATE TABLE INGAHURCO.Alumnos_V2 (
    AlumnoID    INT          NOT NULL PRIMARY KEY,
    Carrera     NVARCHAR(60) NOT NULL,
    Ciudad      NVARCHAR(60) NOT NULL,
    FechaIngreso DATE        NOT NULL,
    CONSTRAINT FK_INGA_V2_V1 FOREIGN KEY (AlumnoID)
        REFERENCES INGAHURCO.Alumnos_V1 (AlumnoID) ON DELETE CASCADE
);

CREATE INDEX IX_INGA_V1_Sede ON INGAHURCO.Alumnos_V1 (Sede) INCLUDE (Nombres,
Apellidos);
CREATE INDEX IX_INGA_V2_Carrera ON INGAHURCO.Alumnos_V2 (Carrera) INCLUDE
(Ciudad, FechaIngreso);
GO

```

```

-----
-- QUEROCHACA
-----

```

```

CREATE TABLE QUEROCHACA.Alumnos_V1 (
    AlumnoID    INT          NOT NULL PRIMARY KEY,
    Cedula      CHAR(10)     NOT NULL,
    Nombres     NVARCHAR(50) NOT NULL,
    Apellidos   NVARCHAR(50) NOT NULL,
    Email       NVARCHAR(100) NOT NULL,
    Telefono    NVARCHAR(20) NULL,
    Sede        NVARCHAR(20) NOT NULL
    CONSTRAINT DF_QUERO_V1_Sede DEFAULT (N'QUEROCHACA'),
    CONSTRAINT CK_QUERO_V1_Sede CHECK (Sede = N'QUEROCHACA'),
    CONSTRAINT UQ_QUERO_Cedula UNIQUE (Cedula)
);

```

```

CREATE TABLE QUEROCHACA.Alumnos_V2 (
    AlumnoID    INT          NOT NULL PRIMARY KEY,
    Carrera     NVARCHAR(60) NOT NULL,
    Ciudad      NVARCHAR(60) NOT NULL,
    FechaIngreso DATE        NOT NULL,

```

CONSTRAINT FK_QUERO_V2_V1 FOREIGN KEY (AlumnoID)

REFERENCES QUEROCHACA.Alumnos_V1 (AlumnoID) ON DELETE CASCADE

);

CREATE INDEX IX_QUERO_V1_Sede ON QUEROCHACA.Alumnos_V1 (Sede) INCLUDE (Nombres, Apellidos);

CREATE INDEX IX_QUERO_V2_Carrera ON QUEROCHACA.Alumnos_V2 (Carrera) INCLUDE (Ciudad, FechaIngreso);

GO

-- HUACHI (IDs 1..)

INSERT INTO HUACHI.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede) VALUES

(1, '0102030405', N'Ana', N'Pérez', N'ana.perez@uni.edu.ec', N'0987654321', N'HUACHI'),

(2, '0102030406', N'Luis', N'García', N'luis.garcia@uni.edu.ec', N'0981112223', N'HUACHI'),

(3, '0102030407', N'María', N'Vera', N'maria.vera@uni.edu.ec', N'0983334445', N'HUACHI');

INSERT INTO HUACHI.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES

(1, N'Ingeniería en Sistemas', N'Ambato', '2023-03-15'),

(2, N'Contabilidad', N'Ambato', '2022-10-10'),

(3, N'Marketing', N'Latacunga', '2024-05-05');

-- INGAHURCO (IDs 1001..)

INSERT INTO INGAHURCO.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede) VALUES

(1001, '1102030405', N'Carla', N'Salas', N'carla.salas@uni.edu.ec', N'0991112233', N'INGAHURCO'),

(1002, '1102030406', N'Pedro', N'Mora', N'pedro.mora@uni.edu.ec', N'0992223344', N'INGAHURCO'),

(1003, '1102030407', N'Andrea', N'Chavez', N'andrea.chavez@uni.edu.ec', N'0993334455', N'INGAHURCO');

INSERT INTO INGAHURCO.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES

(1001, N'Ingeniería Industrial', N'Ambato', '2021-09-01'),

(1002, N'Enfermería', N'Baños', '2022-04-20'),

(1003, N'Administración', N'Pelileo', '2023-11-02');

-- QUEROCHACA (IDs 2001..)

INSERT INTO QUEROCHACA.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede) VALUES

(2001, '1202030405', N'José', N'Caiza', N'jose.caiza@uni.edu.ec', N'0971112233', N'QUEROCHACA'),

(2002, '1202030406', N'Lucía', N'Paredes', N'lucia.paredes@uni.edu.ec', N'0972223344', N'QUEROCHACA');

INSERT INTO QUEROCHACA.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES

(2001, N'Sistemas', N'Quero', '2020-01-10'),

(2002, N'Agroindustria', N'Cevallos', '2024-03-12');

GO

-- 5) VISTAS LOCALES (reconstrucción VERTICAL en cada sede)

CREATE VIEW HUACHI.Alumnos AS

SELECT

V1.AlumnoID, V1.Cedula, V1.Nombres, V1.Apellidos, V1.Email, V1.Telefono, V1.Sede,

V2.Carrera, V2.Ciudad, V2.FechaIngreso

FROM HUACHI.Alumnos_V1 AS V1

JOIN HUACHI.Alumnos_V2 AS V2

ON V1.AlumnoID = V2.AlumnoID;

GO

CREATE VIEW INGAHURCO.Alumnos AS

SELECT

V1.AlumnoID, V1.Cedula, V1.Nombres, V1.Apellidos, V1.Email, V1.Telefono, V1.Sede,

V2.Carrera, V2.Ciudad, V2.FechaIngreso

FROM INGAHURCO.Alumnos_V1 AS V1

JOIN INGAHURCO.Alumnos_V2 AS V2

ON V1.AlumnoID = V2.AlumnoID;

GO

CREATE VIEW QUEROCHACA.Alumnos AS

SELECT

```

V1.AlumnoID, V1.Cedula, V1.Nombres, V1.Apellidos, V1.Email, V1.Telefono, V1.Sede,
V2.Carrera, V2.Ciudad, V2.FechaIngreso
FROM QUEROCHACA.Alumnos_V1 AS V1
JOIN QUEROCHACA.Alumnos_V2 AS V2
ON V1.AlumnoID = V2.AlumnoID;
GO

-----

-- 6) VISTA GLOBAL (reconstrucción HORIZONTAL con UNION ALL)

-----

CREATE VIEW dbo.Alumnos AS
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede, Carrera, Ciudad,
FechaIngreso
FROM HUACHI.Alumnos
UNION ALL
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede, Carrera, Ciudad,
FechaIngreso
FROM INGAHURCO.Alumnos
UNION ALL
SELECT AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono, Sede, Carrera, Ciudad,
FechaIngreso
FROM QUEROCHACA.Alumnos;
GO

-----

-- 7) VERIFICACIÓN / EVIDENCIAS

--

-----

-- a) Esquemas y tablas creadas
-- (Explorador de objetos) o:
SELECT s.name AS Esquema, t.name AS Tabla
FROM sys.schemas s
JOIN sys.tables t ON t.schema_id = s.schema_id
WHERE s.name IN (N'HUACHI', N'INGAHURCO', N'QUEROCHACA')
ORDER BY s.name, t.name;

-- b) Conteo GLOBAL
SELECT COUNT(*) AS TotalGlobal FROM dbo.Alumnos;

```


-- c) Conteo por sede

```
SELECT Sede, COUNT(*) AS TotalPorSede  
FROM dbo.Alumnos  
GROUP BY Sede  
ORDER BY Sede;
```

-- d) Filtros por sede/carrera sobre la vista GLOBAL (transparencia)

```
SELECT * FROM dbo.Alumnos  
WHERE Sede = N'INGAHURCO';
```

```
SELECT * FROM dbo.Alumnos  
WHERE Carrera LIKE N'Ingeniería';
```

-- e) Consultas por ciudad y por año de ingreso

```
SELECT Ciudad, COUNT(*) AS TotalPorCiudad  
FROM dbo.Alumnos  
GROUP BY Ciudad  
ORDER BY TotalPorCiudad DESC;
```

```
SELECT YEAR(FechaIngreso) AS AnioIngreso, COUNT(*) AS Total  
FROM dbo.Alumnos  
GROUP BY YEAR(FechaIngreso)  
ORDER BY AnioIngreso;
```

-- f) Conteo por sede y carrera (mixto)

```
SELECT Sede, Carrera, COUNT(*) AS Total  
FROM dbo.Alumnos  
GROUP BY Sede, Carrera  
ORDER BY Sede, Total DESC;
```

3. Evidencias (capturas):

- Esquemas y tablas creadas.

The screenshot shows the SQL Server Enterprise Manager interface on the left, displaying the database structure for 'DESKTOP-720VGT9\MULTI_SEXTO (16)'. The 'BD_ALUMNOS' database is selected, showing tables like 'HUACHI.Alumnos_V1', 'HUACHI.Alumnos_V2', 'INGAHURCO.Alumnos_V1', 'INGAHURCO.Alumnos_V2', 'QUEROCHACA.Alumnos_V1', and 'QUEROCHACA.Alumnos_V2'. The right pane shows the SQL Query window with the following code:

```
GO

-- HUACHI
CREATE TABLE HUACHI.Alumnos_V1 (
    AlumnoID INT NOT NULL PRIMARY KEY,
    Cedula CHAR(10) NOT NULL,
    Nombres NVARCHAR(50) NOT NULL,
    Apellidos NVARCHAR(50) NOT NULL,
    Email NVARCHAR(100) NOT NULL,
    Telefono NVARCHAR(20) NULL,
    Sede NVARCHAR(20) NOT NULL
    CONSTRAINT DF_HUACHI_V1_Sede DEFAULT (N'HUACHI'),
    CONSTRAINT CK_HUACHI_V1_Sede CHECK (Sede = N'HUACHI'),
    CONSTRAINT UQ_HUACHI_Cedula UNIQUE (Cedula)
);

CREATE TABLE HUACHI.Alumnos_V2 (
    AlumnoID INT NOT NULL PRIMARY KEY,
    Carrera NVARCHAR(60) NOT NULL,
    Ciudad NVARCHAR(60) NOT NULL,
    FechaIngreso DATE NOT NULL,
    CONSTRAINT FK_HUACHI_V2_V1 FOREIGN KEY (AlumnoID)
        REFERENCES HUACHI.Alumnos_V1 (AlumnoID) ON DELETE CASCADE
);

-- Índices útiles de consulta
CREATE INDEX IX_HUACHI_V1_Sede ON HUACHI.Alumnos_V1 (Sede) INCLUDE (Nombres, Apellidos, Email, Telefono);
CREATE INDEX IX_HUACHI_V2_Carrera ON HUACHI.Alumnos_V2 (Carrera) INCLUDE (Ciudad, FechaIngreso);

GO

-- INGAHURCO
CREATE TABLE INGAHURCO.Alumnos_V1 (
    AlumnoID INT NOT NULL PRIMARY KEY,
    Cedula CHAR(10) NOT NULL,
    Nombres NVARCHAR(50) NOT NULL,
    Apellidos NVARCHAR(50) NOT NULL,
    Email NVARCHAR(100) NOT NULL,
    Telefono NVARCHAR(20) NULL,
    Sede NVARCHAR(20) NOT NULL
    CONSTRAINT DF_INGA_V1_Sede DEFAULT (N'INGAHURCO'),
    CONSTRAINT CK_INGA_V1_Sede CHECK (Sede = N'INGAHURCO'),
    CONSTRAINT UQ_INGA_Cedula UNIQUE (Cedula)
);

CREATE TABLE INGAHURCO.Alumnos_V2 (
    AlumnoID INT NOT NULL PRIMARY KEY,
    Carrera NVARCHAR(60) NOT NULL,
    Ciudad NVARCHAR(60) NOT NULL,
    FechaIngreso DATE NOT NULL
);
```

CARGA DE DATOS DE LA BD

The screenshot shows the SQL Query window with the following code for loading data into the database:

```
GO

-- HUACHI (IDs 1..)
INSERT INTO HUACHI.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono) VALUES
(1, '0102030405', N'Ana', N'Pérez', N'ana.perez@uni.edu.ec', N'0987654321'),
(2, '0102030406', N'Luis', N'García', N'luis.garcia@uni.edu.ec', N'0981112223'),
(3, '0102030407', N'Maria', N'Vera', N'maria.vera@uni.edu.ec', N'0983334445');

INSERT INTO HUACHI.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES
(1, N'Ingeniería en Sistemas', N'Ambato', '2023-03-15'),
(2, N'Contabilidad', N'Ambato', '2022-10-10'),
(3, N'Marketing', N'Latacunga', '2024-05-05');

-- INGAHURCO (IDs 1001..)
INSERT INTO INGAHURCO.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono) VALUES
(1001, '1102030405', N'Carla', N'Salas', N'carla.salas@uni.edu.ec', N'0991111222'),
(1002, '1102030406', N'Pedro', N'Mora', N'pedro.mora@uni.edu.ec', N'0992222333'),
(1003, '1102030407', N'Andrea', N'Chávez', N'andrea.chavez@uni.edu.ec', N'0993333444');

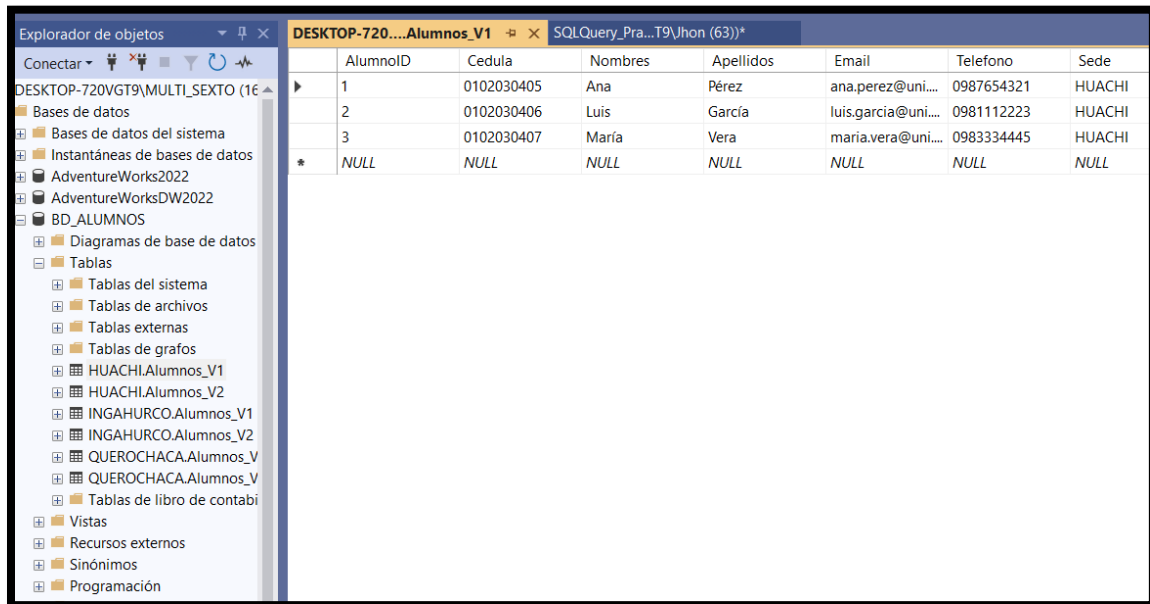
INSERT INTO INGAHURCO.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES
(1001, N'Ingeniería Industrial', N'Ambato', '2021-09-01'),
(1002, N'Enfermería', N'Baños', '2022-04-20'),
(1003, N'Administración', N'Pelileo', '2023-11-02');

-- QUEROCHACA (IDs 2001..)
INSERT INTO QUEROCHACA.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Telefono) VALUES
(2001, '1202030405', N'José', N'Caiza', N'jose.caiza@uni.edu.ec', N'0971111222'),
(2002, '1202030406', N'Lucía', N'Paredes', N'lucia.paredes@uni.edu.ec', N'0972222333');

INSERT INTO QUEROCHACA.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES
(2001, N'Sistemas', N'Quero', '2020-01-10'),
(2002, N'Agroindustria', N'Cevallos', '2024-03-12');

GO
```

VERIFICACIÓN DE LA CARGA DE DATOS



Explorador de objetos

DESKTOP-720VGT9\MULTI_SEXTO (16)

Bases de datos

Instantáneas de bases de datos

AdventureWorks2022

AdventureWorksDW2022

BD_ALUMNOS

Diagramas de base de datos

Tablas

Tablas del sistema

Tablas de archivos

Tablas externas

Tablas de grafos

HUACHI.Alumnos_V1

HUACHI.Alumnos_V2

INGAHURCO.Alumnos_V1

INGAHURCO.Alumnos_V2

QUEROCHACA.Alumnos_V

QUEROCHACA.Alumnos_V

Tablas de libro de contabi

Vistas

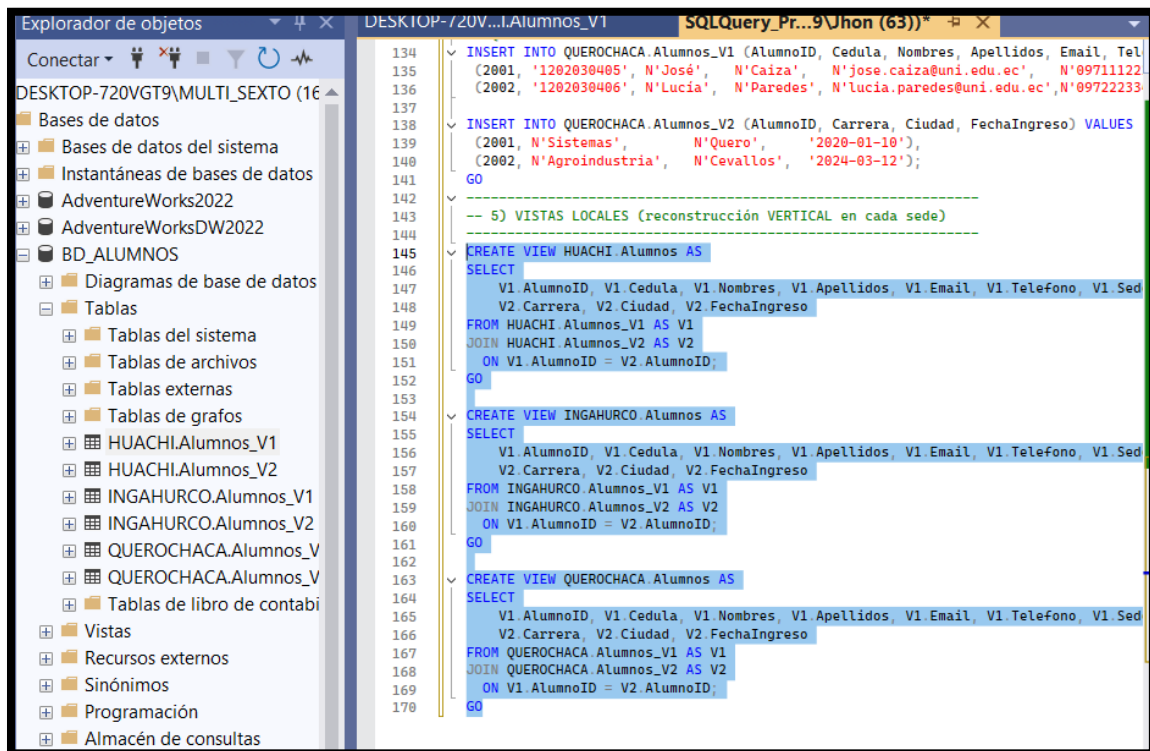
Recursos externos

Sinónimos

Programación

AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede
1	0102030405	Ana	Pérez	ana.perez@uni....	0987654321	HUACHI
2	0102030406	Luis	García	luis.garcia@uni....	0981112223	HUACHI
3	0102030407	María	Vera	maria.vera@uni....	0983334445	HUACHI
*	NULL	NULL	NULL	NULL	NULL	NULL

VISTAS LOCALES (reconstrucción VERTICAL en cada sede)



Explorador de objetos

DESKTOP-720VGT9\MULTI_SEXTO (16)

Bases de datos

Instantáneas de bases de datos

AdventureWorks2022

AdventureWorksDW2022

BD_ALUMNOS

Diagramas de base de datos

Tablas

Tablas del sistema

Tablas de archivos

Tablas externas

Tablas de grafos

HUACHI.Alumnos_V1

HUACHI.Alumnos_V2

INGAHURCO.Alumnos_V1

INGAHURCO.Alumnos_V2

QUEROCHACA.Alumnos_V

QUEROCHACA.Alumnos_V

Tablas de libro de contabi

Vistas

Recursos externos

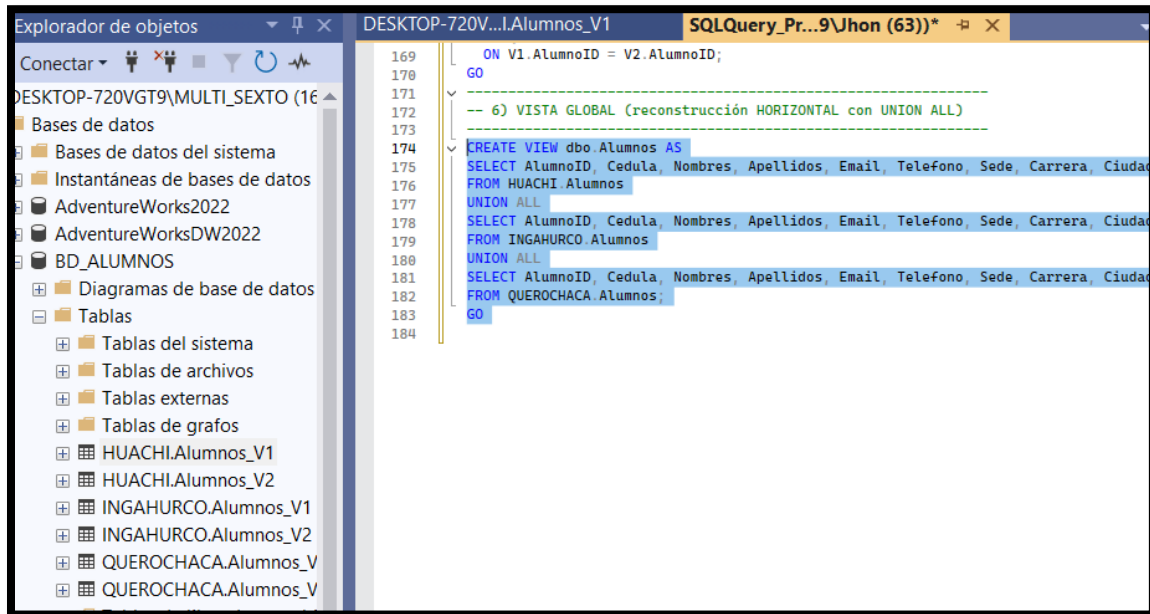
Sinónimos

Programación

Almacén de consultas

```
134 INSERT INTO QUEROCHACA.Alumnos_V1 (AlumnoID, Cedula, Nombres, Apellidos, Email, Tel
135 (2001, '1202030405', N'José', N'Caiza', N'jose.caiza@uni.edu.ec', N'09711122
136 (2002, '1202030406', N'Lucía', N'Paredes', N'Lucia.paredes@uni.edu.ec', N'09722233
137
138
139 INSERT INTO QUEROCHACA.Alumnos_V2 (AlumnoID, Carrera, Ciudad, FechaIngreso) VALUES
140 (2001, N'Sistemas', N'Quero', '2020-01-10'),
141 (2002, N'Agroindustria', N'Cevallos', '2024-03-12');
142
143 -- 5) VISTAS LOCALES (reconstrucción VERTICAL en cada sede)
144
145 CREATE VIEW HUACHI.Alumnos AS
146 SELECT
147 V1.AlumnoID, V1.Cedula, V1.Nombres, V1.Apellidos, V1.Email, V1.Telefono, V1.Sed
148 V2.Carrera, V2.Ciudad, V2.FechaIngreso
149 FROM HUACHI.Alumnos_V1 AS V1
150 JOIN HUACHI.Alumnos_V2 AS V2
151 ON V1.AlumnoID = V2.AlumnoID;
152 GO
153
154 CREATE VIEW INGAHURCO.Alumnos AS
155 SELECT
156 V1.AlumnoID, V1.Cedula, V1.Nombres, V1.Apellidos, V1.Email, V1.Telefono, V1.Sed
157 V2.Carrera, V2.Ciudad, V2.FechaIngreso
158 FROM INGAHURCO.Alumnos_V1 AS V1
159 JOIN INGAHURCO.Alumnos_V2 AS V2
160 ON V1.AlumnoID = V2.AlumnoID;
161 GO
162
163 CREATE VIEW QUEROCHACA.Alumnos AS
164 SELECT
165 V1.AlumnoID, V1.Cedula, V1.Nombres, V1.Apellidos, V1.Email, V1.Telefono, V1.Sed
166 V2.Carrera, V2.Ciudad, V2.FechaIngreso
167 FROM QUEROCHACA.Alumnos_V1 AS V1
168 JOIN QUEROCHACA.Alumnos_V2 AS V2
169 ON V1.AlumnoID = V2.AlumnoID;
170 GO
```

VISTA GLOBAL (reconstrucción HORIZONTAL con UNION ALL)



- SELECT COUNT(*) y consultas por sede/carrera sobre dbo.Alumnos.

The screenshot shows the SQL query editor with the following code:

```
198  
199 -- c) Conteo por sede  
200 SELECT Sede, COUNT(*) AS TotalPorSede  
201 FROM dbo.Alumnos  
202 GROUP BY Sede  
203 ORDER BY Sede;  
204  
205 -- d) Filtros por sede/carrera sobre la vista GLOBAL (transparencia)  
206 SELECT * FROM dbo.Alumnos
```

Below the query editor, the 'Resultados' (Results) tab is active, displaying the following table:

	Sede	TotalPorSede
1	HUACHI	3
2	INGAHURCO	3
3	QUEROCHACA	2

```

205 -- d) Filtros por sede/carrera sobre la vista GLOBAL (transparencia)
206 SELECT * FROM dbo.Alumnos
207 WHERE Sede = N'INGAHURCO';
208
209 SELECT * FROM dbo.Alumnos
210 WHERE Carrera LIKE N'Ingenier%';
211
212 -- e) Consultas por ciudad y por año de ingreso

```

85 %

19 0

↑ ↓

Resultados

Mensajes

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede	Carrera	Ciudad	FechaIngreso
1	1001	1102030405	Carla	Salas	carla.salas@uni.edu.ec	0991112233	INGAHURCO	Ingeniería Industrial	Ambato	2021-09-01
2	1002	1102030406	Pedro	Mora	pedro.mora@uni.edu.ec	0992223344	INGAHURCO	Enfermería	Baños	2022-04-20
3	1003	1102030407	Andrea	Chávez	andrea.chavez@uni.edu.ec	0993334455	INGAHURCO	Administración	Pelileo	2023-11-02

	AlumnoID	Cedula	Nombres	Apellidos	Email	Telefono	Sede	Carrera	Ciudad	FechaIngreso
1	1	0102030405	Ana	Pérez	ana.perez@uni.edu.ec	0987654321	HUACHI	Ingeniería en Sistemas	Ambato	2023-03-15
2	1001	1102030405	Carla	Salas	carla.salas@uni.edu.ec	0991112233	INGAHURCO	Ingeniería Industrial	Ambato	2021-09-01

```

212 -- e) Consultas por ciudad y por año de ingreso
213 SELECT Ciudad, COUNT(*) AS TotalPorCiudad
214 FROM dbo.Alumnos
215 GROUP BY Ciudad
216 ORDER BY TotalPorCiudad DESC;
217
218 SELECT YEAR(FechaIngreso) AS AnioIngreso, COUNT(*) AS Total
219 FROM dbo.Alumnos
220 GROUP BY YEAR(FechaIngreso)
221 ORDER BY AnioIngreso;
222
223 -- f) Conteo por sede y carrera (mixto)
224 SELECT Sede, Carrera, COUNT(*) AS Total

```

85 %

✖ 19

⚠ 0

↑

↓

Resultados

Mensajes

	Ciudad	TotalPorCiudad
1	Ambato	3
2	Baños	1
3	Cevallos	1
4	Latacunga	1
5	Pelileo	1
6	Quero	1

	AnioIngreso	Total
1	2020	1
2	2021	1
3	2022	2
4	2023	2
5	2024	2

```

222
223 -- f) Conteo por sede y carrera (mixto)
224 SELECT Sede, Carrera, COUNT(*) AS Total
225 FROM dbo.Alumnos
226 GROUP BY Sede, Carrera
227 ORDER BY Sede, Total DESC;

```

85 % 19 0

Resultados Mensajes

	Sede	Carrera	Total
1	HUACHI	Contabilidad	1
2	HUACHI	Ingeniería en Sistemas	1
3	HUACHI	Marketing	1
4	INGAHURCO	Ingeniería Industrial	1
5	INGAHURCO	Enfermería	1
6	INGAHURCO	Administración	1
7	QUEROCHACA	Agroindustria	1
8	QUEROCHACA	Sistemas	1

- Demostración de **transparencia**: una consulta global que no “sepa” de los fragmentos y devuelva datos unificados.

```

195
196 -- b) Conteo GLOBAL
197 SELECT COUNT(*) AS TotalGlobal FROM dbo.Alumnos;
198

```

70 % 3 0 Línea: 197 Carácter

Resultados Mensajes

	TotalGlobal
1	8

4. Conclusiones:

- La fragmentación de bases de datos, tal como se hizo en este ejemplo, tiene puntos a favor importantes, aunque también implica dificultades. Uno de los mayores beneficios es que el desempeño mejora, pues las búsquedas se hacen en porciones de datos más chicas y enfocadas. Esto ayuda mucho en sistemas que están repartidos. Además, puede hacer que sea más fácil crecer y que los datos estén más seguros, porque la información puede quedar separada en partes. Pero lo más complicado es volver a armar los datos. Es clave crear formas de ver todo

el panorama que puedan juntar los datos sin que se pierda nada, usando cosas como JOIN (para las partes verticales) y UNION ALL (para las partes horizontales).

- Para replicar este diseño con el objetivo de lograr alta disponibilidad, se podría implementar una arquitectura de replicación maestro-esclavo o multimaestro. Cada una de las "sedes" (HUACHI, INGAHURCO, QUEROCHACA) podría funcionar como una instancia física o virtual independiente, con sus fragmentos de datos replicados a un sitio de respaldo. De esta manera, si un sitio principal falla, la réplica puede tomar el control rápidamente, asegurando el acceso continuo a los datos. Por ejemplo, cada sede podría tener su propio servidor con su fragmento de la tabla Alumnos, y un servidor en espera podría reflejar esos datos en tiempo real para garantizar la redundancia y minimizar el tiempo de inactividad.