



Laboratorio de Bases de Datos Distribuidas

Práctica No.: 4

Grupo No.: 9

Integrantes:

- Luis Enrique Pérez Señalín.

Tema: Diseño de la base de datos distribuida

Objetivos:

1. Escenario (requerimientos)
2. Grafo relacional
3. Sedes y Roles
4. Campo y condición de fragmentación
5. Esquema de fragmentación
6. Esquema de replicación
7. Esquema de ubicación/asignación

Marco teórico:

El grafo relacional constituye un modelo de datos que utiliza nodos y aristas para representar las entidades y sus relaciones dentro de una base de datos. Cada nodo corresponde a una entidad, mientras que las aristas indican las conexiones entre estas entidades. Este enfoque visual facilita la comprensión de la estructura y las relaciones en un entorno de bases de datos distribuido.

La planificación del esquema de ubicación es crucial en el diseño distribuido, ya que determina cómo se distribuyen las tablas y los datos entre los diversos nodos del sistema. Cada nodo almacena un conjunto específico de tablas o datos, guiado por factores como la ubicación geográfica, la carga de trabajo y los requisitos de rendimiento.

Desarrollo de la práctica:

Describir el escenario.

La biblioteca de sistemas “FIS biblioteca”, contiene libros de materias básicas y de materias de carrera como “computación gráfica”, “Diseño de software”, “Redes”, etc. La biblioteca puede ser accedida por cualquier estudiante de la Universidad o profesores. Cuenta con 2 sedes, una en la planta baja de la facultad de sistemas y otra en el CEC.

- Cada biblioteca cuenta con un inventario propio y realiza los alquileres de los libros por separado, pero comparten información de los estudiantes y profesores.
- Los bibliotecarios acceden a la información del inventario de la biblioteca local y la otra biblioteca.



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

- Los estudiantes y profesores pueden ver toda la información de los libros para ambas bibliotecas, sin acceder al inventario disponible.
- Los estudiantes y profesores pueden alquilar un libro en cualquiera de las dos bibliotecas, pero deben devolverlo en la biblioteca donde lo alquilaron.
- Los estudiantes y profesores solo pueden alquilar 1 libro o documento para llevarse, y pueden alquilar máximo 3 libros para usarlos dentro de la biblioteca, un libro solo puede ser alquilado por una persona a la vez.
- Al alquilar un libro debe constar la fecha, el código del libro, el código único del estudiante o profesor, el tipo de alquiler (“Para llevar” o “Uso en biblioteca”), la sucursal y el tiempo de alquiler, las sucursales solo pueden ver los registros de alquiler de la otra sucursal, pero no añadir datos.
- Todos los bibliotecarios deben registrarse en de forma compartida, de modo que puedan trabajar en cualquier sucursal, debe contener su número de cédula, nombres completos y números de teléfono.
- Deben registrarse todos los proveedores de libros en general, no existe una diferenciación por sede, estos deben contar con su información como: “nombre del proveedor”, “dirección”, “teléfono”, “Ruc”, etc.

El diseño de la base de datos se enfoca en manejar los alquileres de los libros de forma individual, los estudiantes pueden acceder a la información de los libros en general sin importar la sede y solo el bibliotecario puede consultar en el inventario de ambas sedes, se debe garantizar que los libros existan en la base de datos de ambas sedes, así como los proveedores y los estudiantes.

Se identifican _ entidades fundamentales:

Biblioteca: Contiene libros de la facultad de sistemas.

Libro: Textos escritos para explicar temas y materias de distintas áreas.

Inventario: Datos que contienen la información y la cantidad de libros en cada biblioteca.

Usuario: Estudiante, profesor o personal de la Universidad.

Alquiler: Servicio de préstamos de libros para los estudiantes o profesores.

Proveedor: Vendedor autorizado se encarga de suministrar los libros a las bibliotecas.

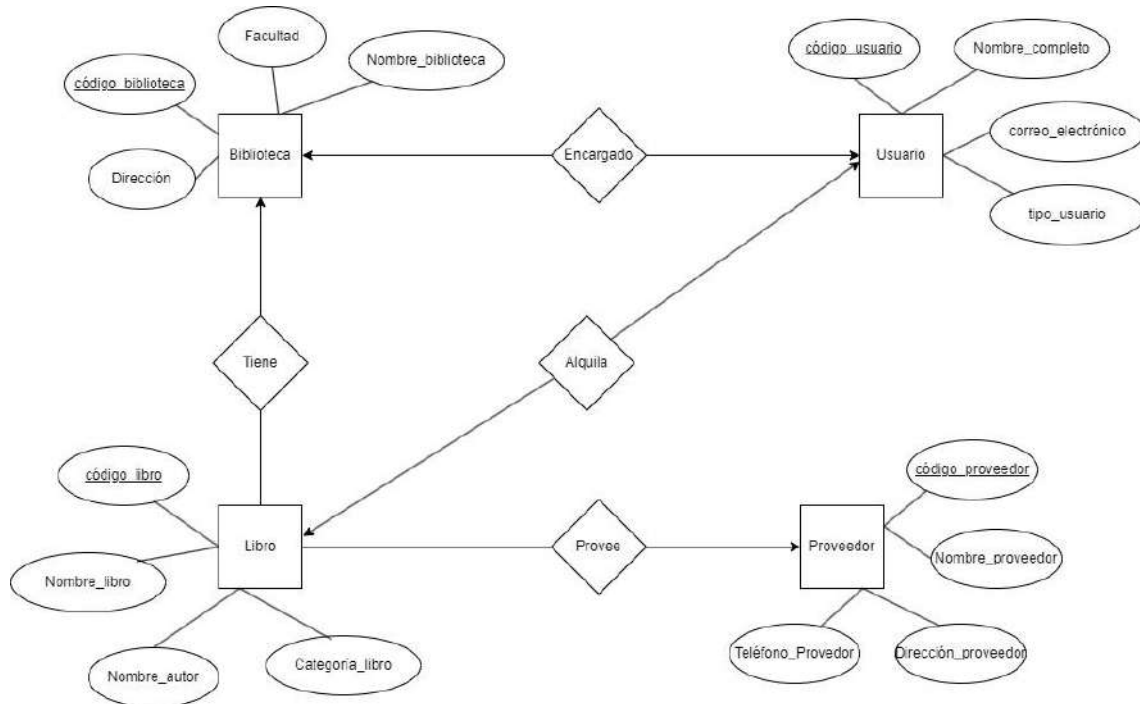
Las entidades se relacionan entre sí de la siguiente manera:

- Una biblioteca tiene un único usuario que es el encargado.
- Los libros solo pueden pertenecer a una biblioteca.
- Un usuario puede alquilar un único libro a la vez.
- Un libro solo puede ser alquilado por un solo usuario a la vez.
- Un proveedor provee libros.
- Un libro solo puede ser proveído por un solo proveedor.

Modelo entidad relación

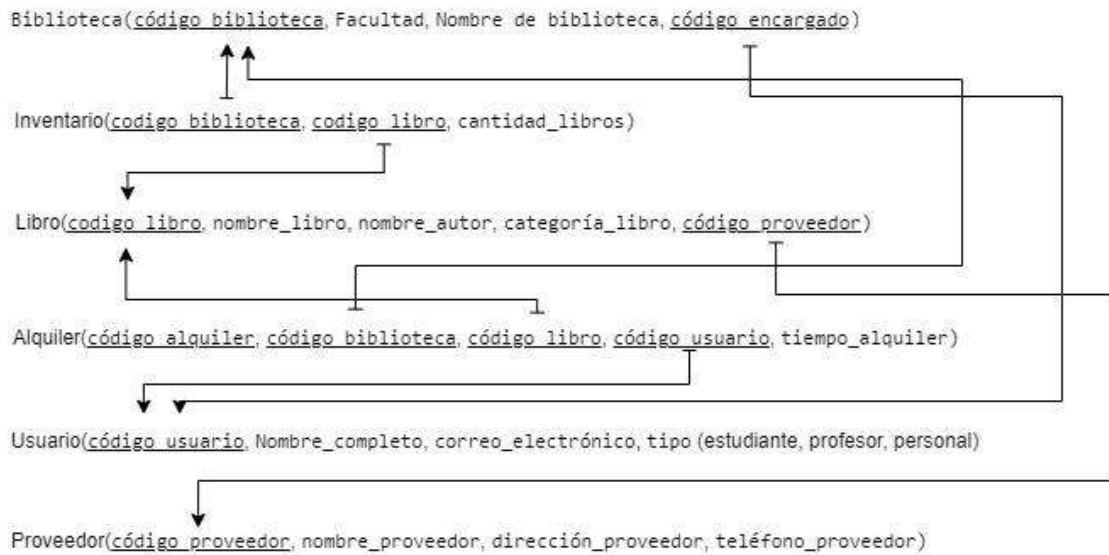


ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



Los atributos y las relaciones se obtuvieron directamente de la descripción del problema.

Grafo Relacional



Conversión del MER a un modelo relacional, donde se representan las entidades como tablas con atributos, además de las relaciones entre ellas mediante claves primarias y foráneas.

Sedes y Roles



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

Sede Facultad de Ingeniería en Sistemas (FIS): Registro de libros, gestión de estudiantes, gestión de encargados en la biblioteca en la FIS, alquileres en la biblioteca en la FIS y proveedores de libros.

Sede Centro de Educación Continua (CEC): Registro de libros, gestión de estudiantes, gestión de encargados en la biblioteca en el CEC, alquileres en la biblioteca en el CEC y proveedores de libros.

Campo y condición de fragmentación

Campo de fragmentación: código_biblioteca

Condición de fragmentación: código_biblioteca = {'FIS', 'CEC'}

Esquema de fragmentación

Fragmentación horizontal primaria

$BIBLIOTECA_FIS_i: \sigma_{\text{código_biblioteca}='FIS'}(Biblioteca)$
 $BIBLIOTECA_CEC_i: \sigma_{\text{código_biblioteca}='CEC'}(Biblioteca)$
 $ALQUILERES_FIS_i: \sigma_{\text{código_biblioteca}='FIS'}(Alquiler)$
 $ALQUILERES_CEC_i: \sigma_{\text{código_biblioteca}='CEC'}(Alquiler)$
 $INVENTARIO_FIS: \sigma_{\text{código_biblioteca}='FIS'}(Inventario)$
 $INVENTARIO_CEC: \sigma_{\text{código_biblioteca}='FIS'}(Inventario)$

Estas son las fragmentaciones horizontales primarias, donde “código_biblioteca” está presente.

Fragmentación horizontal derivada

$LIBROS_FIS = LIBRO_{\text{código_libro}} \bowtie_{\text{código_libro}} INVENTARIO_FIS_i$
 $LIBROS_CEC = LIBRO_{\text{código_libro}} \bowtie_{\text{código_libro}} INVENTARIO_CEC_i$
 $PROVEEDORES_FIS = PROVEEDOR_{\text{código_proveedor}} \bowtie_{\text{código_proveedor}} LIBROS_FIS_i$
 $PROVEEDORES_CEC = PROVEEDOR_{\text{código_proveedor}} \bowtie_{\text{código_proveedor}} LIBROS_CEC_i$
 $USUARIOS_FIS = USUARIO_{\text{código_usuario}} \bowtie_{\text{código_usuario}} ALQUILERES_FIS_i$
 $USUARIOS_CEC = USUARIO_{\text{código_usuario}} \bowtie_{\text{código_usuario}} ALQUILERES_CEC_i$

Las fragmentaciones horizontales derivadas no contienen “código_biblioteca” y necesitan de algo intermedio, en este caso “código_libro”.

Fragmentación vertical

$DATOS_PROVEEDORES: \pi_{\text{código_proveedor}, \text{nombre_proveedor}}(Proveedor)$



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Esquema de replicación

Se crea una replicación de tipo merge, desde el nodo FIS se publican las tablas Datos_Proveedores.

Esquema de ubicación

SEDES	FIS	CEC
BIBLIOTECA	BIBLIOTECA_FIS	BIBLIOTECA_CEC
LIBROS	LIBROS_FIS	LIBROS_CEC
ALQUILER	ALQUILERES_FIS	ALQUILERES_CEC
USUARIO	USUARIOS_FIS	USUARIOS_CEC
PROVEEDORES	PROVEEDORES_FIS DATOS_PROVEEDORES	PROVEEDORES_CEC

Evidencia de las tablas y su contenido de las dos sedes

Sede FIS

Base de Datos: BIBLIOTECA_FIS

Biblioteca_FIS

Se extraen los datos de la tabla Biblioteca de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada Biblioteca_FIS, utilizando el campo código_biblioteca como criterio de fragmentación.

```
CREATE TABLE Biblioteca_FIS (
    codigo_biblioteca VARCHAR(3) PRIMARY KEY, -- Usa VARCHAR en lugar de BINARY
    facultad NVARCHAR(100),
    nombre_biblioteca NVARCHAR(100),
    codigo_encargado INT,
    CONSTRAINT chk_codigo_biblioteca CHECK (codigo_biblioteca IN ('FIS', 'CEC')) -- Restricción de valores
)

INSERT INTO Biblioteca_FIS (codigo_biblioteca, facultad, nombre_biblioteca, codigo_encargado)
SELECT * FROM OPENQUERY(MySQL2, 'SELECT * FROM BibliotecaDistribuida.Biblioteca')
WHERE codigo_biblioteca = 'FIS';

Select * FROM Biblioteca_FIS;
```

codigo_biblioteca	facultad	nombre_biblioteca	codigo_encargado
FIS	Facultad de Ingeniería	Biblioteca FIS	15

Inventario_FIS



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Se extraen los datos de la tabla Inventario de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada Inventario_FIS.

```
--CREATE TABLE Inventario_FIS (  
    codigo_biblioteca VARCHAR(3) CHECK (codigo_biblioteca IN ('FIS', 'CEC')),  
    codigo_libro INT,  
    cantidad_libros INT,  
    PRIMARY KEY (codigo_biblioteca, codigo_libro)  
);  
  
--INSERT INTO Inventario_FIS (codigo_biblioteca, codigo_libro, cantidad_libros)  
SELECT codigo_biblioteca, codigo_libro, cantidad_libros  
FROM OPENQUERY(MySQL2, '  
    SELECT codigo_biblioteca, codigo_libro, cantidad_libros  
    FROM BibliotecaDistribuida.Inventario  
    WHERE codigo_biblioteca = "FIS"  
');  
  
--SELECT * FROM Inventario_FIS;
```

codigo_biblioteca	codigo_libro	cantidad_libros
FIS	2	4
FIS	3	2
FIS	10	3
FIS	12	1
FIS	16	3
FIS	20	5
FIS	21	4
FIS	22	4
FIS	24	5
FIS	27	2
FIS	29	1

LIBROS_FIS

Se extraen los datos de la tabla Libro de la base de datos central en MySQL y se crea una fragmentación horizontal derivada llamada Libros_FIS, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los libros que estén en la biblioteca FIS.

```
--CREATE TABLE Libro_FIS (  
    codigo_libro INT PRIMARY KEY,  
    nombre_libro NVARCHAR(255),  
    nombre_autor NVARCHAR(255),  
    categoria_libro NVARCHAR(100),  
    codigo_proveedor INT  
);  
  
--INSERT INTO Libro_FIS (codigo_libro, nombre_libro, nombre_autor, categoria_libro, codigo_proveedor)  
SELECT codigo_libro, nombre_libro, nombre_autor, categoria_libro, codigo_proveedor  
FROM OPENQUERY(MySQL2, '  
    SELECT l.codigo_libro, l.nombre_libro, l.nombre_autor, l.categoria_libro, l.codigo_proveedor  
    FROM BibliotecaDistribuida.Libro AS l  
    WHERE l.codigo_libro IN (  
        SELECT codigo_libro FROM BibliotecaDistribuida.Inventario WHERE codigo_biblioteca = "FIS"  
    )  
');  
  
--SELECT * FROM Libro_FIS;
```

codigo_libro	nombre_libro	nombre_autor	categoria_libro	codigo_proveedor
2	The Pragmatic Programmer	Andrew Hunt, David Thomas	Desarrollo de Software	2
3	Design Patterns: Elements of Reusable Object-Oriented Software	Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides	Arquitectura de Software	4
10	Software Engineering	Ian Sommerville	Ingeniería de Software	2
12	Code Complete	Steve McConnell	Desarrollo de Software	2
16	Deep Learning	Ian Goodfellow, Yoshua Bengio, Aaron Courville	Aprendizaje Automático	3
20	Compilers: Principles, Techniques and Tools	Afred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman	Compiladores	4
21	The Mythical Man-Month	Frederick P. Brooks Jr.	Ingeniería de Software	6
22	Modern Operating Systems	Andrew S. Tanenbaum, Herbert Bos	Sistemas Operativos	5
24	Cryptography and Network Security	William Stallings	Criptografía	4
27	Blockchain Basics	Daniel Drescher	Blockchain	1
29	Software Architecture in Practice	Len Bass, Paul Clements, Rick Kern	Arquitectura de Software	5

PROVEEDOR_FIS



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Se extraen los datos de la tabla Proveedor de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada Proveedor_FIS, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los proveedores que tengan libros en la biblioteca FIS.

```
-- CREATE TABLE Proveedor_FIS (
    codigo_proveedor INT PRIMARY KEY, -- Se usa IDENTITY en lugar de AUTO_INCREMENT
    nombre_proveedor NVARCHAR(255),
    direccion_proveedor NVARCHAR(255),
    telefono_proveedor NVARCHAR(20)
);

INSERT INTO Proveedor_FIS (codigo_proveedor, nombre_proveedor, direccion_proveedor, telefono_proveedor)
SELECT codigo_proveedor, nombre_proveedor, direccion_proveedor, telefono_proveedor
FROM OPENQUERY(MYSQL2, '
    SELECT codigo_proveedor, nombre_proveedor, direccion_proveedor, telefono_proveedor
    FROM BibliotecaDistribuida.Proveedor
') AS p
WHERE p.codigo_proveedor IN (
    Select codigo_proveedor FROM Libro_FIS
);

Select * FROM Proveedor_FIS;
```

codigo_proveedor	nombre_proveedor	direccion_proveedor	telefono_proveedor
1	Person Education	3333 Burr Ridge Pkwy, Burr Ridge, IL, USA	+1 800-822-6579
2	O'Reilly Media	1005 Gravenstein Hwy North, Sebastopol, CA, USA	+1 707-827-7000
3	Springer	Tiergartenstr. 17, Heidelberg, Germany	+49 6221 4870
4	MIT Press	One Rogers Street, Cambridge, MA, USA	+1 617-375-2885
5	Packt Publishing	Library Place, Birmingham, UK	+44 121-265-6404
6	McGraw-Hill Education	Two Penn Plaza, New York, NY, USA	+1 877-835-5524

ALQUILER_FIS

Se extraen los datos de la tabla Proveedor de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada ALQUILER_FIS, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los registro de alquileres que tengan el código de biblioteca FIS.

```
-- CREATE TABLE Alquiler_FIS (
    codigo_alquiler INT PRIMARY KEY,
    codigo_biblioteca VARCHAR(3) CHECK (codigo_biblioteca IN ('FIS', 'CEC')),
    codigo_libro INT,
    codigo_usuario INT,
    tiempo_alquiler INT, -- Días de alquiler
);

INSERT INTO Alquiler_FIS (codigo_alquiler, codigo_biblioteca, codigo_libro, codigo_usuario, tiempo_alquiler)
SELECT codigo_alquiler, codigo_biblioteca, codigo_libro, codigo_usuario, tiempo_alquiler
FROM OPENQUERY(MYSQL2, '
    SELECT codigo_alquiler, codigo_biblioteca, codigo_libro, codigo_usuario, tiempo_alquiler
    FROM BibliotecaDistribuida.Alquiler
    Where codigo_biblioteca = "FIS"
');

Select * FROM Alquiler_FIS;
```

codigo_alquiler	codigo_biblioteca	codigo_libro	codigo_usuario	tiempo_alquiler
1	FIS	20	27	3
2	FB	20	20	10
3	FB	23	14	9
4	FB	20	11	10
5	FB	5	10	10
6	FB	5	22	7
7	FB	5	1	6
8	FB	5	7	11
9	FB	5	30	15
10	FB	7	34	10
11	FB	4	33	13
12	FB	2	30	11

USUARIOS_FIS



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Se extraen los datos de la tabla Usuarios de la base de datos central en MySQL y se crea una fragmentación horizontal derivada llamada USUARIOS_FIS, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los registro de usuarios con registros de alquiler en la biblioteca FIS.

```
--CREATE TABLE Usuarios_FIS (
codigo_usuario INT PRIMARY KEY, -- Se usa IDENTITY en lugar de AUTO_INCREMENT
nombre_completo NVARCHAR(255),
correo_electronico NVARCHAR(255),
tipo NVARCHAR(20)
);

INSERT INTO Usuarios_FIS (codigo_usuario, nombre_completo, correo_electronico, tipo)
SELECT codigo_usuario, nombre_completo, correo_electronico, tipo
FROM OPENQUERY(MySQL2, '
SELECT codigo_usuario, nombre_completo, correo_electronico, tipo
FROM BibliotecaDistribuida.Usuarios
') AS u
WHERE u.codigo_usuario IN (
Select codigo_usuario FROM Alquiler_FIS
);

SELECT * FROM Usuarios_FIS;
```

codigo_usuario	nombre_completo	correo_electronico	tipo
1	AGUIRRE GUADALUPE ANTHONY SEBASTIAN	aguirre.guadalupe1@epn.edu.ec	estudiante
2	AGUIRRE GUADALUPE ANTHONY SEBASTIAN	aguirre.guadalupe1@epn.edu.ec	estudiante
3	AGUIRRE GUADALUPE ANTHONY SEBASTIAN	aguirre.guadalupe1@epn.edu.ec	estudiante
4	ALMEIDA PALACIOS JOHANN GABRIEL	almeida.palacios1@epn.edu.ec	estudiante
5	ALVARADO TIGRERO MIGUEL ANGEL	alvarado.tigrero1@epn.edu.ec	estudiante
6	ANIVANDIO ZAMBRANO JAMES MATTHEW	anivandio.zambrano1@epn.edu.ec	estudiante
7	ANZULES PALMA BRITANY DENNISE	anzules.palma1@epn.edu.ec	estudiante
8	ARCOIS CONLAGO SEBASTIAN ALEXANDER	arcois.conlagos1@epn.edu.ec	estudiante
9	ARIAS YUGO CHRISTIAN GEOVANNY	arias.yugo1@epn.edu.ec	estudiante
10	ENRIQUEZ BORRILLA KELLY DANIELA	enriquez.borrilla1@epn.edu.ec	estudiante
11	ESCOBAR CHAMBA GISEL ANACELY	escobar.chamba1@epn.edu.ec	estudiante
12	ESPINOZ URIARTE KEVIN ALEXANDER	espinoz.uriarte1@epn.edu.ec	estudiante

DATOS_PROVEEDORES

Se extraen los datos de la tabla Vehiculo de la base de datos central en MySQL mediante una fragmentación vertical.

```
--CREATE TABLE Datos_Proveedores (
codigo_proveedor INT PRIMARY KEY,
nombre_proveedor VARCHAR(255)
);

INSERT INTO Datos_Proveedores (codigo_proveedor, nombre_proveedor)
SELECT codigo_proveedor, nombre_proveedor
FROM OPENQUERY(MySQL2, '
SELECT codigo_proveedor, nombre_proveedor
FROM BibliotecaDistribuida.Proveedores
');

SELECT * FROM Datos_Proveedores;
```

codigo_proveedor	nombre_proveedor
1	Pearson Education
2	O'Reilly Media
3	Springer
4	MIT Press
5	Pack Publishing
6	McGraw-Hill Education
7	CRC Press
8	Cambridge University Press
9	Elsevier
10	John Wiley & Sons

SEDE CEC



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Base de Datos: BIBLIOTECA_CEC Biblioteca_CEC

Se extraen los datos de la tabla Biblioteca de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada Biblioteca_CEC, utilizando el campo código_biblioteca como criterio de fragmentación.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy, including the 'BIBLIOTECA_CEC' database. The right pane shows the SQL query editor with the following code:

```
--CREATE TABLE Biblioteca_CEC (  
    codigo_biblioteca VARCHAR(3) PRIMARY KEY, -- Usa VARCHAR en lugar de ENUM  
    facultad NVARCHAR(100),  
    nombre_biblioteca NVARCHAR(100),  
    codigo_encargado INT,  
    CONSTRAINT chk_codigo_biblioteca CHECK (codigo_biblioteca IN ('FIS', 'CEC')) -- Restricción de valores  
);  
  
--INSERT INTO Biblioteca_CEC (codigo_biblioteca, facultad, nombre_biblioteca, codigo_encargado)  
SELECT * FROM OPENQUERY([ENRIQUE-PS], 'SELECT * FROM OPENQUERY(MYSQL2, ''SELECT * FROM BibliotecaDistribuida.Biblioteca'')')  
WHERE codigo_biblioteca = 'CEC';  
  
SELECT * FROM Biblioteca_CEC
```

The bottom pane shows the results of the query execution, displaying a table with the following data:

codigo_biblioteca	facultad	nombre_biblioteca	codigo_encargado
CEC	Facultad de Ciencias	Reservorio CEC	21

Inventario_CEC

Se extraen los datos de la tabla Inventario de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada Inventario_CEC.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy, including the 'BIBLIOTECA_CEC' database. The right pane shows the SQL query editor with the following code:

```
--CREATE TABLE Inventario_CEC (  
    codigo_biblioteca VARCHAR(3) CHECK (codigo_biblioteca IN ('FIS', 'CEC')),  
    codigo_libro INT,  
    cantidad_libros INT,  
    PRIMARY KEY (codigo_biblioteca, codigo_libro)  
);  
  
--INSERT INTO Inventario_CEC (codigo_biblioteca, codigo_libro, cantidad_libros)  
SELECT codigo_biblioteca, codigo_libro, cantidad_libros  
FROM OPENQUERY([ENRIQUE-PS], 'SELECT * FROM OPENQUERY(MYSQL2, ''  
    SELECT codigo_biblioteca, codigo_libro, cantidad_libros  
    FROM BibliotecaDistribuida.Inventario  
    Where codigo_biblioteca = "CEC"  
    ''')');  
  
SELECT * FROM Inventario_CEC
```

The bottom pane shows the results of the query execution, displaying a table with the following data:

codigo_biblioteca	codigo_libro	cantidad_libros
CEC	6	2
CEC	7	4
CEC	8	4
CEC	9	4
CEC	11	2
CEC	13	1
CEC	14	5
CEC	15	5
CEC	17	3
CEC	18	1
CEC	19	1
CEC	23	5
CEC	26	3
CEC	28	2
CEC	29	3



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

LIBROS_CEC

Se extraen los datos de la tabla Libro de la base de datos central en MySQL y se crea una fragmentación horizontal derivada llamada Libros_CEC, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los libros que estén en la biblioteca CEC.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server structure, including the 'BIBLIOTECA_CEC' database. The right pane shows the SQL query editor with the following code:

```
CREATE TABLE Libro_CEC (
    codigo_libro INT PRIMARY KEY,
    nombre_libro NVARCHAR(255),
    nombre_autor NVARCHAR(255),
    categoria_libro NVARCHAR(100),
    codigo_proveedor INT
);

INSERT INTO Libro_CEC (codigo_libro, nombre_libro, nombre_autor, categoria_libro, codigo_proveedor)
SELECT codigo_libro, nombre_libro, nombre_autor, categoria_libro, codigo_proveedor
FROM OPENQUERY([ENRIQUE-PS], 'SELECT * FROM OPENQUERY(MySQL2, ''
SELECT l.codigo_libro, l.nombre_libro, l.nombre_autor, l.categoria_libro, l.codigo_proveedor
FROM BibliotecaDistribuida.Libro AS l
WHERE l.codigo_libro IN (
SELECT codigo_libro FROM BibliotecaDistribuida.Inventario WHERE codigo_bibliotecas = "CEC"
'')'');
SELECT * FROM Libro_CEC;
```

The bottom pane shows the results of the query, displaying a table with 12 columns: codigo_libro, nombre_libro, nombre_autor, categoria_libro, and codigo_proveedor. The data is as follows:

codigo_libro	nombre_libro	nombre_autor	categoria_libro	codigo_proveedor
1	Queen Code	Robert C. Martin	Diseño de Software	3
2	Artificial Intelligence: A Modern Approach	Stuart Russell, Peter Norvig	Inteligencia Artificial	3
3	Database Systems Concepts	Abraham Silberschatz, Henry Korth, S. Sudarshan	Bases de Datos	6
4	Introduction to the Theory of Computation	Michael Sipser	Técnicas de Computación	4
5	Computer Networks	Andrew S. Tanenbaum, David J. Wetherell	Redes de Computadoras	5
6	Operating System Concepts	Abraham Silberschatz, Peter B. Galvin, Greg Gagne	Sistemas Operativos	2
7	Machine Learning	Tom M. Mitchell	Aprendizaje Automático	9
8	Refactoring: Improving the Design of Existing Code	Martin Fowler	Diseño de Software	8
9	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald	Algoritmos	7
10	Computer Organization and Design	David A. Patterson, John L. Hennessy	Arquitectura de Computadoras	4
11	Artificial Intelligence: Foundations of Computational Intelligence	David L. Poole, Alan K. Mackworth	Inteligencia Artificial	1
12	The Art of Computer Programming	Donald Knuth	Algoritmos	6

PROVEEDOR_CEC

Se extraen los datos de la tabla Proveedor de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada Proveedor_CEC, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los proveedores que tengan libros en la biblioteca CEC.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server structure, including the 'BIBLIOTECA_CEC' database. The right pane shows the SQL query editor with the following code:

```
CREATE TABLE Proveedor_CEC (
    codigo_proveedor INT PRIMARY KEY,
    nombre_proveedor NVARCHAR(255),
    direccion_proveedor NVARCHAR(255),
    telefono_proveedor NVARCHAR(20)
);

INSERT INTO Proveedor_CEC (codigo_proveedor, nombre_proveedor, direccion_proveedor, telefono_proveedor)
SELECT codigo_proveedor, nombre_proveedor, direccion_proveedor, telefono_proveedor
FROM OPENQUERY([ENRIQUE-PS], 'SELECT * FROM OPENQUERY(MySQL2, ''
SELECT codigo_proveedor, nombre_proveedor, direccion_proveedor, telefono_proveedor
FROM BibliotecaDistribuida.Proveedor
'') AS p
WHERE p.codigo_proveedor IN (
SELECT codigo_proveedor FROM Libro_CEC
);
SELECT * FROM Proveedor_CEC;
```

The bottom pane shows the results of the query, displaying a table with 4 columns: codigo_proveedor, nombre_proveedor, direccion_proveedor, and telefono_proveedor. The data is as follows:

codigo_proveedor	nombre_proveedor	direccion_proveedor	telefono_proveedor
1	Pearson Education	1303 Bur Ridge Pkwy, Burr Ridge, IL, USA	+1 800 822 0578
2	Citibank	1005 Greenway Way North, Suite 100, CA, USA	+1 707 421 1928
3	Springer	Tiergartenstr. 17, Heidelberg, Germany	+49 6221 4850
4	MIT Press	One Rogers Street, Cambridge, MA, USA	+1 617 253 2900
5	Pack Publishing	Livery Place, Birmingham, UK	+44 121 269 9684
6	McGraw-Hill Education	Two Penn Plaza, New York, NY, USA	+1 877 839 3264
7	CRC Press	800 Brookside Parkway NW, Boca Raton, FL, USA	+1 800 272 7757
8	Cambridge University Press	University Printing House, Cambridge, UK	+44 1223 326 000
9	Elsevier	Radwagweg 29, 1043 NX Amsterdam, Netherlands	+31 20 485 3911

ALQUILER_CEC



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Se extraen los datos de la tabla Proveedor de la base de datos central en MySQL y se crea una fragmentación horizontal primaria llamada ALQUILER_CEC, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los registro de alquileres que tengan el código de biblioteca CEC.

```
--CREATE TABLE Alquiler_CEC (  
    codigo_alquiler INT PRIMARY KEY,  
    codigo_biblioteca VARCHAR(3) CHECK (codigo_biblioteca IN ('FIS', 'CEC')),  
    codigo_libro INT,  
    codigo_usuario INT,  
    tiempo_alquiler INT, -- Días de alquiler  
);  
  
--INSERT INTO Alquiler_CEC (codigo_alquiler, codigo_biblioteca, codigo_libro, codigo_usuario, tiempo_alquiler)  
SELECT codigo_alquiler, codigo_biblioteca, codigo_libro, codigo_usuario, tiempo_alquiler  
FROM OPENQUERY([ENRIQUE-PS], 'SELECT * FROM OPENQUERY(MySQL2, ''  
    SELECT codigo_alquiler, codigo_biblioteca, codigo_libro, codigo_usuario, tiempo_alquiler  
    FROM BibliotecaDistribuida.Alquiler  
    Where codigo_biblioteca = "CEC"  
    ''')  
');  
SELECT * FROM Alquiler_CEC;
```

codigo_alquiler	codigo_biblioteca	codigo_libro	codigo_usuario	tiempo_alquiler
1	CEC	0	0	0
2	CEC	36	20	13
3	CEC	14	30	12
4	CEC	0	33	0
5	CEC	1	35	10
6	CEC	11	13	11
7	CEC	7	28	5
8	CEC	5	18	0
9	CEC	0	6	4
10	CEC	18	34	6
11	CEC	30	30	0
12	CEC	1	24	0

USUARIOS_CEC

Se extraen los datos de la tabla Usuarios de la base de datos central en MySQL y se crea una fragmentación horizontal derivada llamada USUARIOS_CEC, extraemos los datos desde MySQL y aplicamos el filtro de solo añadir los registro de usuarios con registros de alquiler en la biblioteca CEC.

```
--CREATE TABLE Usuario_CEC (  
    codigo_usuario INT PRIMARY KEY, -- Se usa IDENTITY en lugar de AUTO_INCREMENT  
    nombre_completo NVARCHAR(255),  
    correo_electronico NVARCHAR(255),  
    tipo NVARCHAR(20)  
);  
  
--INSERT INTO Usuario_CEC (codigo_usuario, nombre_completo, correo_electronico, tipo)  
SELECT codigo_usuario, nombre_completo, correo_electronico, tipo  
FROM OPENQUERY([ENRIQUE-PS], 'SELECT * FROM OPENQUERY(MySQL2, ''  
    SELECT codigo_usuario, nombre_completo, correo_electronico, tipo  
    FROM BibliotecaDistribuida.Usuario  
    WHERE u.codigo_usuario IN (  
        Select codigo_usuario FROM Alquiler_CEC  
    )  
    ''') AS u  
);  
SELECT * FROM Usuario_CEC;
```

codigo_usuario	nombre_completo	correo_electronico	tipo
1	AGUIRRE PALADRES MANUEL ALEJANDRO	aguirre.paladres1@epn.edu.ec	estudiante
2	ANRANGO ZAMBRANO JAMES MATTHEWS	anrango.zambrano1@epn.edu.ec	estudiante
3	ARCOS CONLAGE SEBASTIAN ALEXANDER	arcos.conlage1@epn.edu.ec	estudiante
4	ESPINOZA RASTICAS EVELYN ESTEFANIA	espinoza.rasticas1@epn.edu.ec	estudiante
5	ESPINOZA UVELO PADRA ELIZABETH	espinoza.uvelo1@epn.edu.ec	estudiante
6	MOYTA SALCABANCO ALVARO GABRIEL	moyma.salcabanco1@epn.edu.ec	estudiante
7	MULLO OÑA ANINI DAYANA	mullo.oña1@epn.edu.ec	estudiante
8	TORRES TORRES ESTEBAN EVERALDO	torres.torres1@epn.edu.ec	estudiante
9	YANCHAGUANO CHOGOLLO DAYRA ESTEFANIA	yanchaguano.chogollo1@epn.edu.ec	estudiante
10	YASLAGA HURTADO ANGIE DIMBELA	yaslagahurtado1@epn.edu.ec	estudiante
11	GARCIA ANCHAPANTA KEVIN RUBEN	garcia.anchapanta1@epn.edu.ec	estudiante
12	GONZALEZ DE SAPIEL SCHEME DANIELA	gonzalez.de.sapiel1@epn.edu.ec	estudiante

Añadir restricciones:



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

Para un fácil ingreso de datos no se aplicaron las restricciones hasta el final, tanto para el nodo FIS como para el nodo CEC.

Restricciones_FIS:

```
ALTER TABLE Libro_FIS ADD CONSTRAINT FK_LibroFIS_Proveedor FOREIGN KEY  
(codigo_proveedor) REFERENCES Proveedor_FIS(codigo_proveedor);
```

```
ALTER TABLE Inventario_FIS  
ADD CONSTRAINT fk_inventario_biblioteca  
FOREIGN KEY (codigo_biblioteca) REFERENCES Biblioteca_FIS(codigo_biblioteca);
```

```
ALTER TABLE Inventario_FIS  
ADD CONSTRAINT fk_inventario_libro  
FOREIGN KEY (codigo_libro) REFERENCES Libro_FIS(codigo_libro);
```

```
ALTER TABLE Alquiler_FIS  
ADD CONSTRAINT fk_alquiler_biblioteca  
FOREIGN KEY (codigo_biblioteca) REFERENCES Biblioteca_FIS(codigo_biblioteca);
```

```
ALTER TABLE Alquiler_FIS  
ADD CONSTRAINT fk_alquiler_libro  
FOREIGN KEY (codigo_libro) REFERENCES Libro_FIS(codigo_libro);
```

```
ALTER TABLE Alquiler_FIS  
ADD CONSTRAINT fk_alquiler_usuario  
FOREIGN KEY (codigo_usuario) REFERENCES Usuario_FIS(codigo_usuario);
```

Restricciones_CEC:

```
ALTER TABLE Libro_CEC ADD CONSTRAINT FK_LibroCEC_Proveedor FOREIGN KEY  
(codigo_proveedor) REFERENCES Proveedor_CEC(codigo_proveedor);
```

```
ALTER TABLE Inventario_CEC  
ADD CONSTRAINT fk_inventario_biblioteca  
FOREIGN KEY (codigo_biblioteca) REFERENCES Biblioteca_CEC(codigo_biblioteca);
```

```
ALTER TABLE Inventario_CEC  
ADD CONSTRAINT fk_inventario_libro  
FOREIGN KEY (codigo_libro) REFERENCES Libro_CEC(codigo_libro);
```

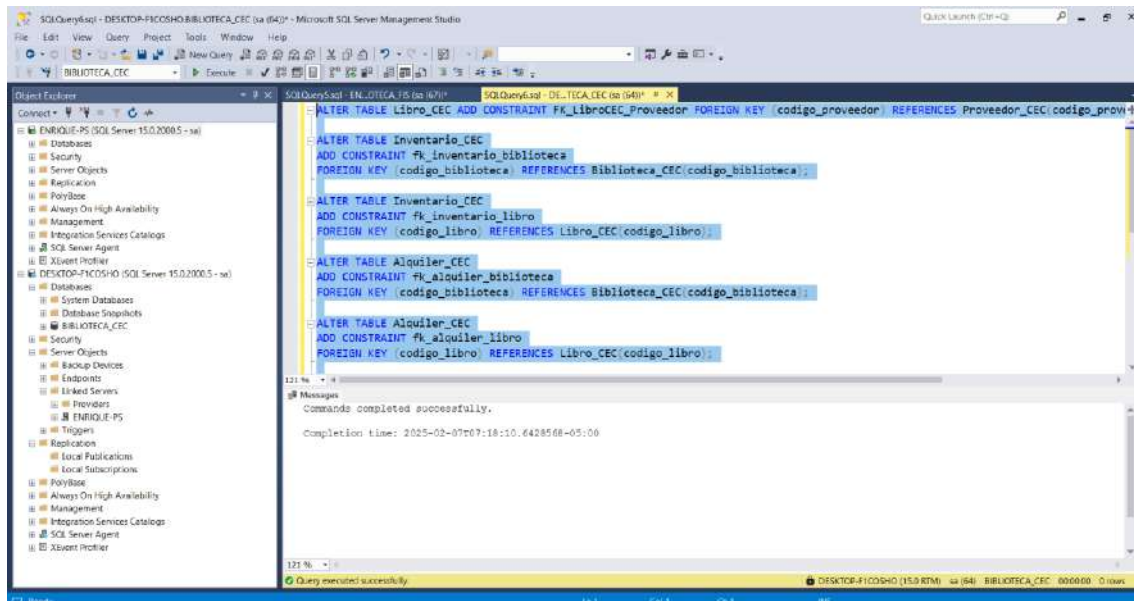
```
ALTER TABLE Alquiler_CEC  
ADD CONSTRAINT fk_alquiler_biblioteca  
FOREIGN KEY (codigo_biblioteca) REFERENCES Biblioteca_CEC(codigo_biblioteca);
```

```
ALTER TABLE Alquiler_CEC  
ADD CONSTRAINT fk_alquiler_libro  
FOREIGN KEY (codigo_libro) REFERENCES Libro_CEC(codigo_libro);
```



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

ALTER TABLE Alquiler_CEC
ADD CONSTRAINT fk_alquiler_usuario
FOREIGN KEY (codigo_usuario) REFERENCES Usuario_CEC(codigo_usuario);



Proceso de replicación (explique el modelo implementado y su demostración)

Proceso de replicación

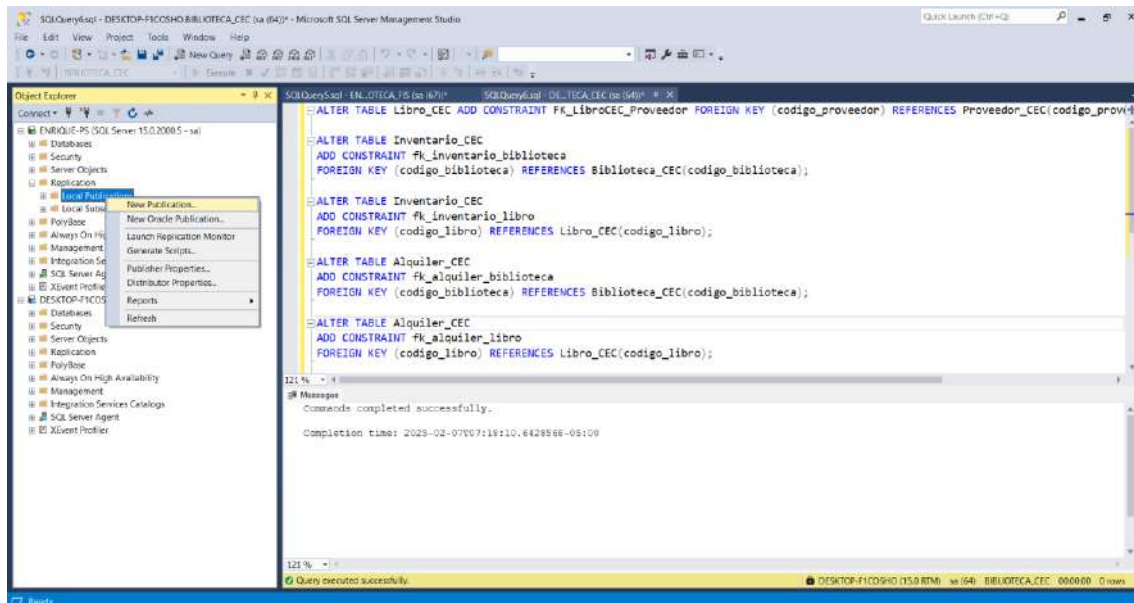
Publicación desde el nodo FIS

En este caso se replicará la tabla “Datos_Proveedores” tener los datos del nombre del proveedor y el código del proveedor en ambos nodos”.

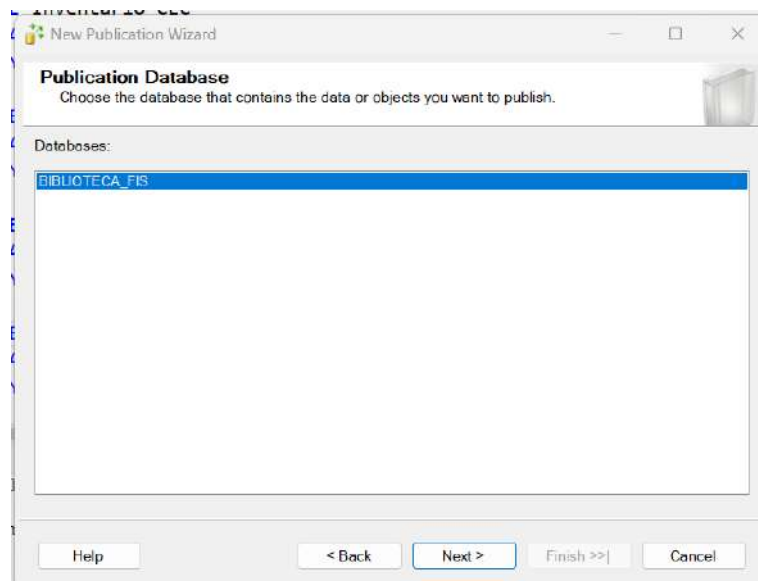
Nos dirigimos a la parte izquierda de SQL Server en Object Explorer, Replication , hacemos clic derecho sobre Local Publication en New Publication.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



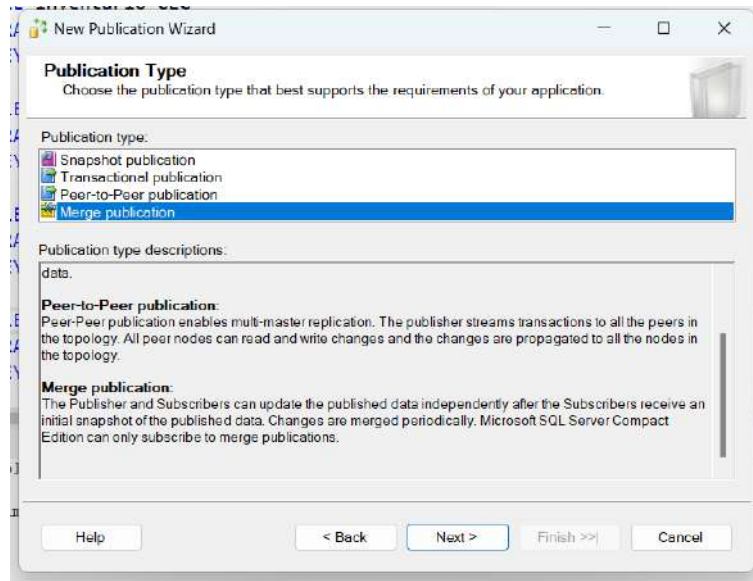
Seleccionamos la base de datos de la que queremos replicar las tablas, en este caso BIBLIOTECA_FIS.



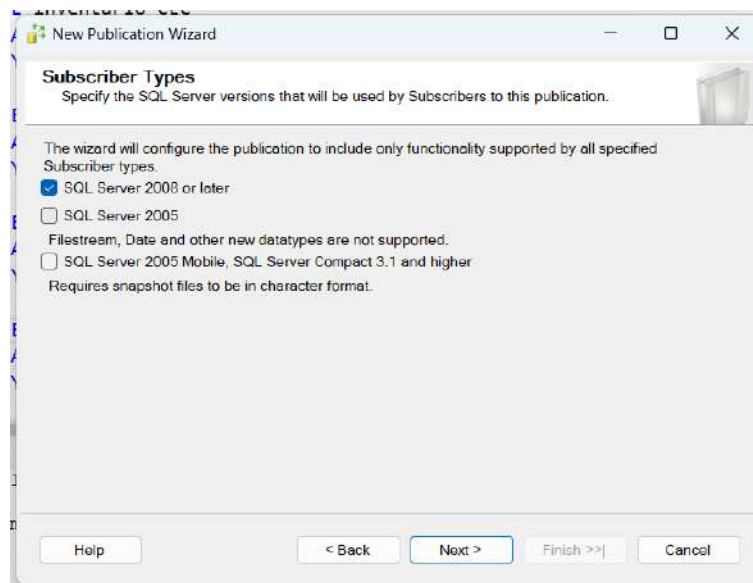
Seleccionamos el tipo de publicación, en este caso Merge Publication porque la replicación es de tipo bidireccional como se mencionó en el esquema de replicación



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



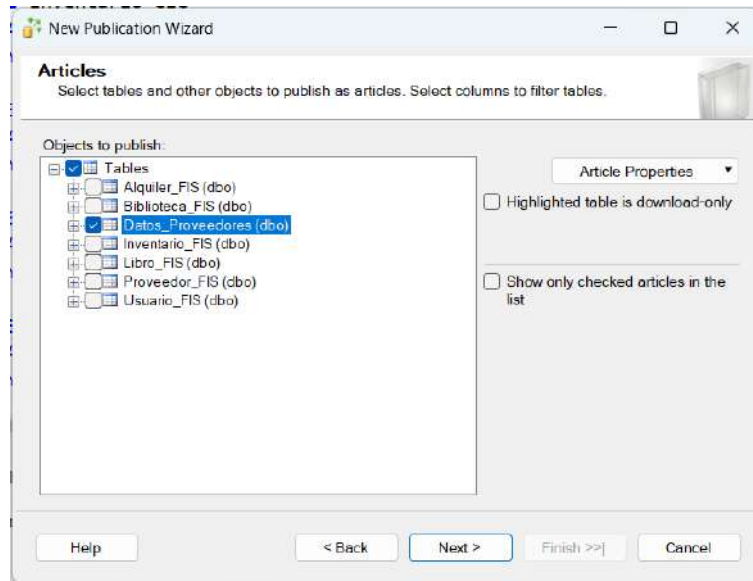
Seleccionar el tipo de subscriptor por defecto



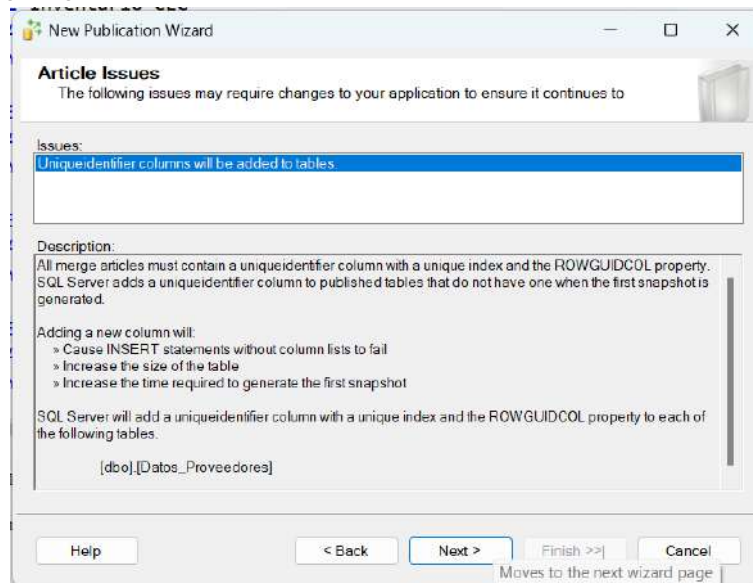
Seleccionamos las tablas que se van a publicar en este caso son Datos_proveedores.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



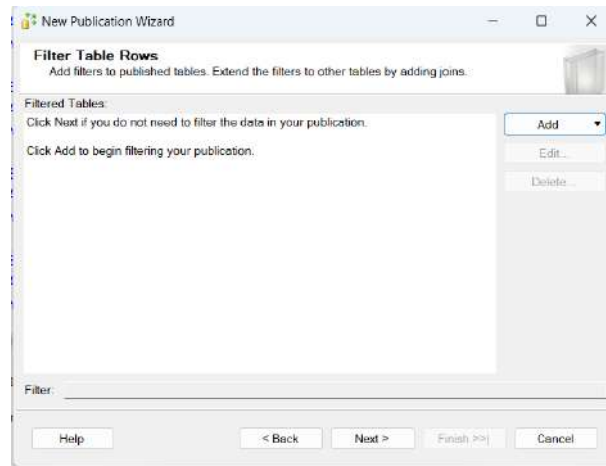
Hacemos clic en Next



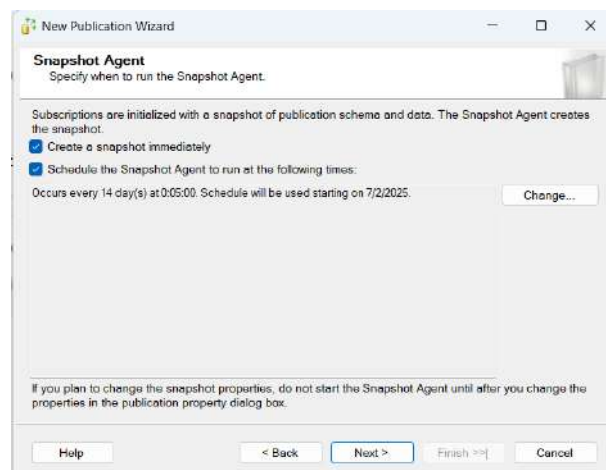
Hacemos click en next otra vez



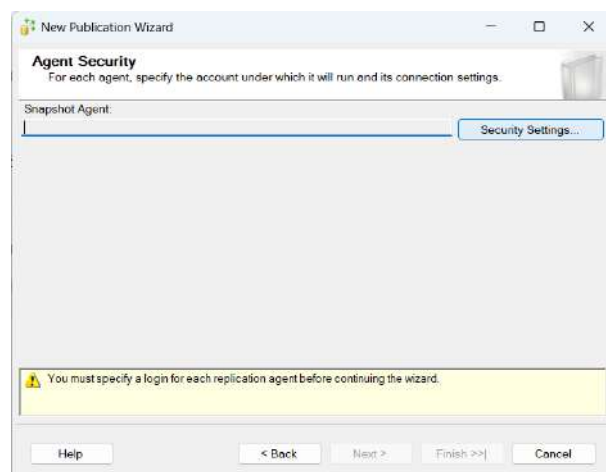
**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**



Seleccionamos el Snapshot Agent



Hacemos clic en Security Settings para configurar la seguridad del agente





ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

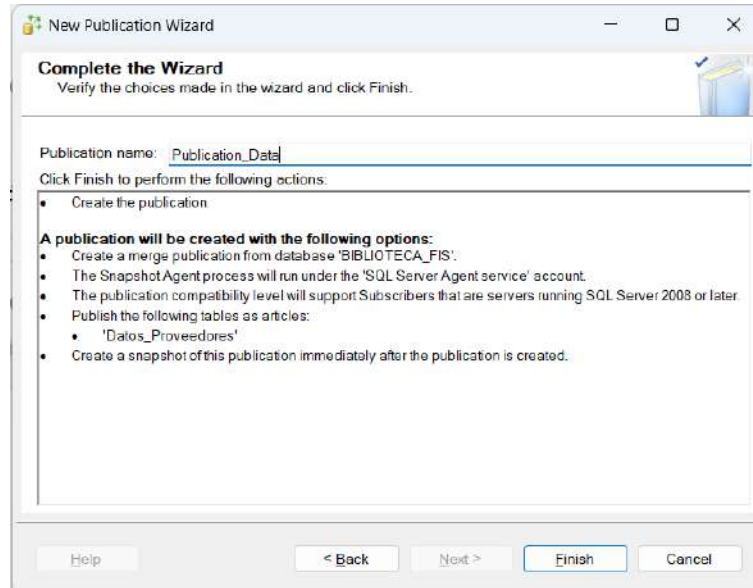
En la ventana que se abre, configuramos para que se puede acceder a la publicación se ingresen las credenciales de SQL del nodo Quito, en este caso es el usuario sa con la contraseña P@ssw0rd

Habiendo ya configurado la seguridad hacemos clic en Next

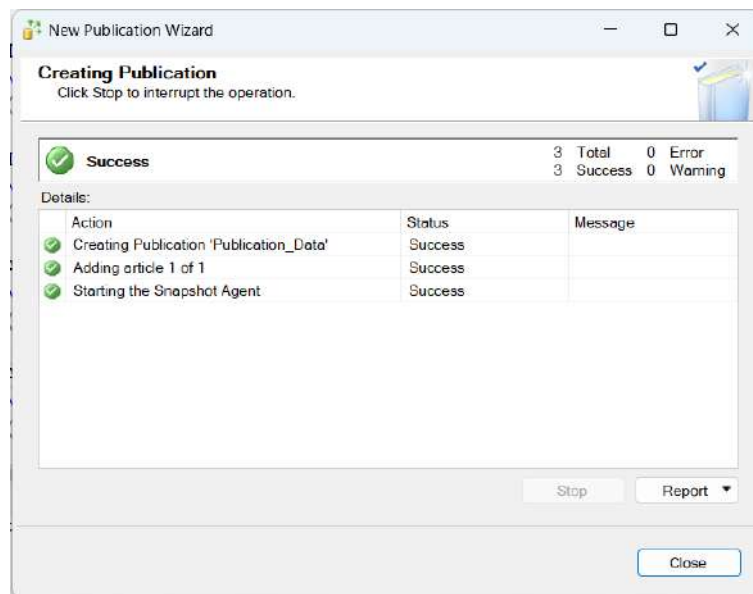


ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Nombrar la publicación para que pueda ser identificada, en este caso el nombre es Publicacion_Data y hacemos clic en Finish



La publicación fue creada con éxito

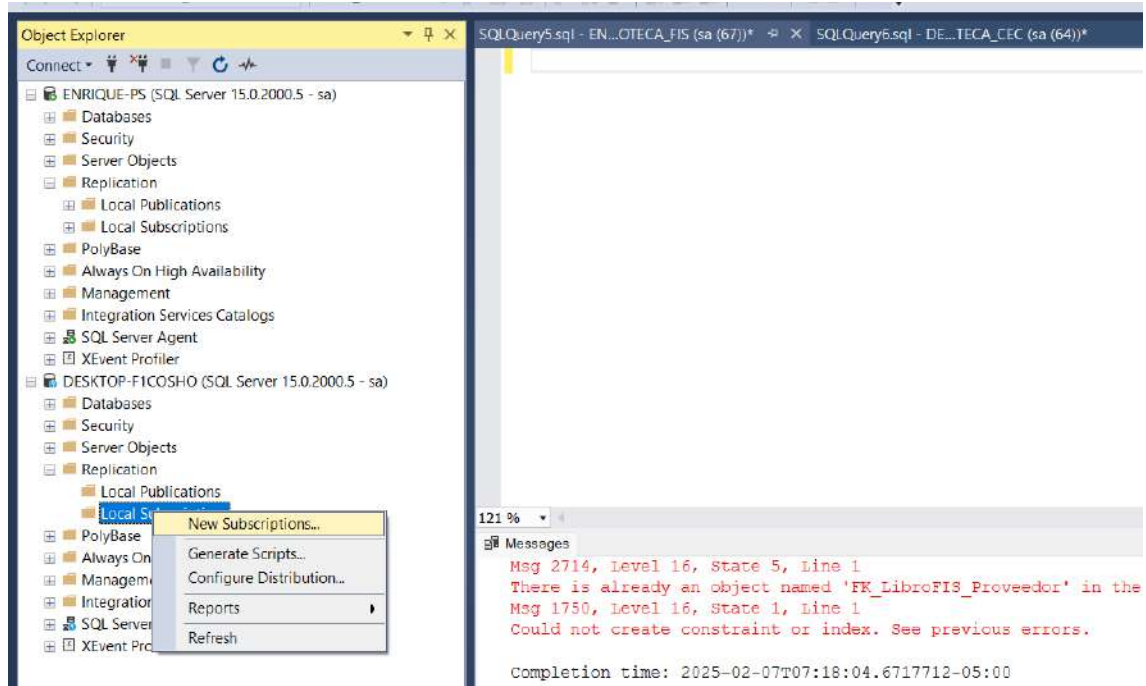


Proceso de suscripción

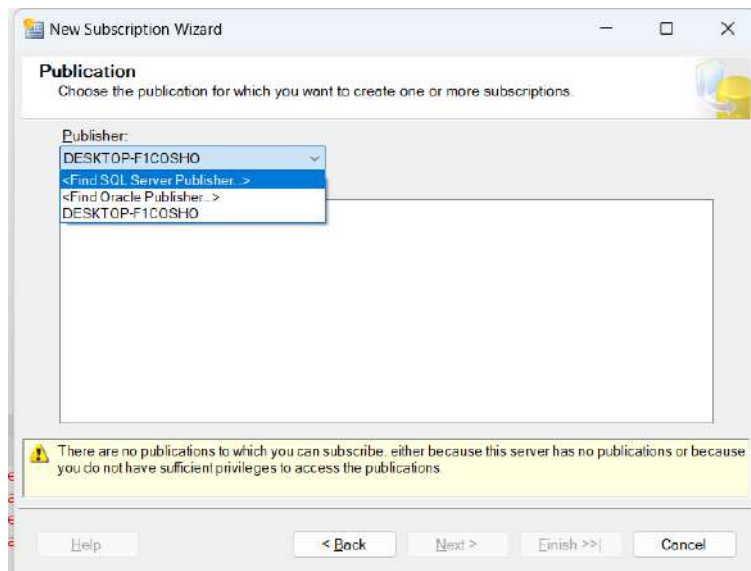
Nos dirigimos a la parte izquierda de SQL Server en Object Explorer, Replication , hacemos clic derecho sobre Local en New Subscriptions



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



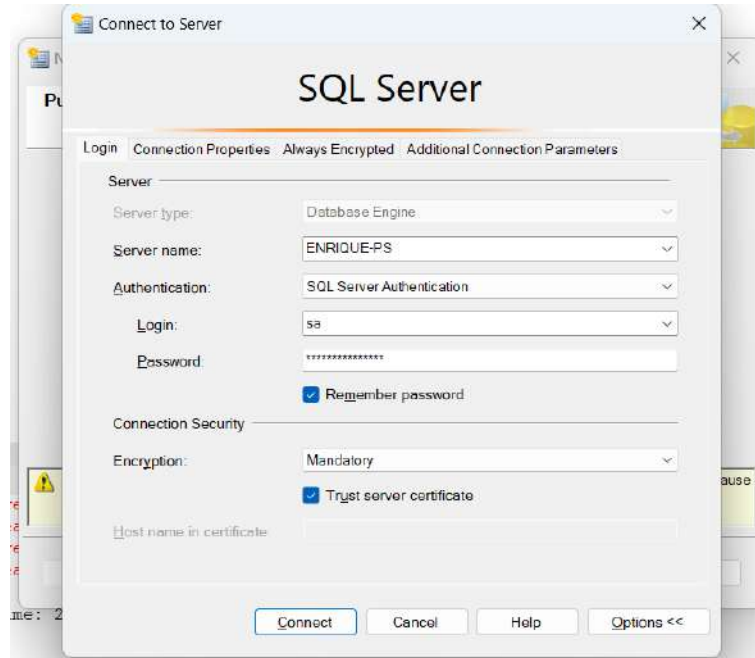
Elegimos el tipo de publicación a la que nos vamos a subscribir, en este caso es una publicación SQL.



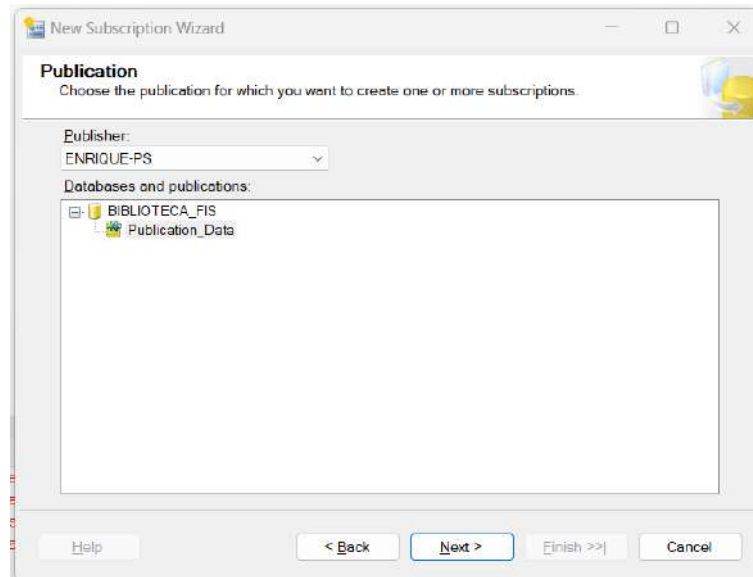


ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Nos conectamos al Nodo Central Quito mediante su IP y sus credenciales SQL



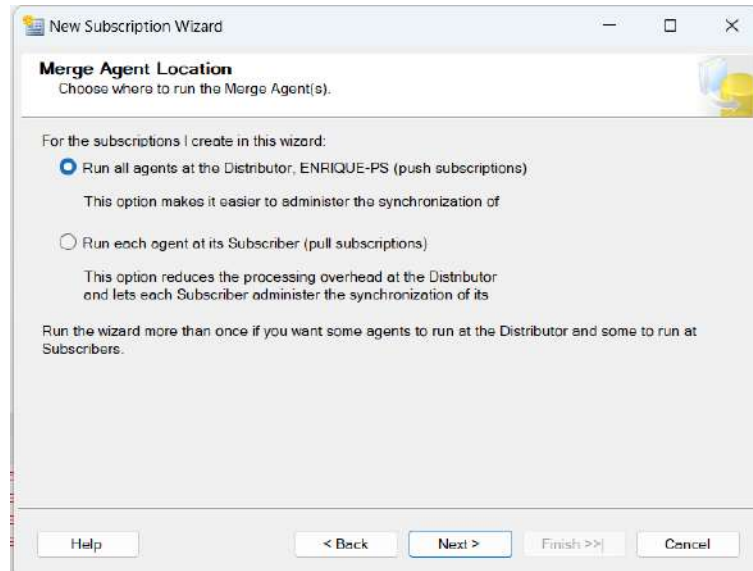
Al conectarnos ya podemos ver la publicación configurada en el nodo Quito llama Publicacion_Data



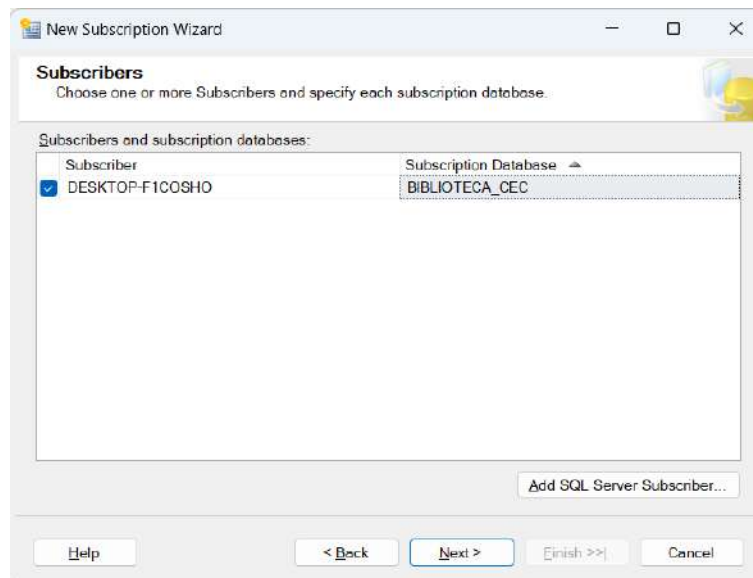


ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Configuramos el Merge Agent Location con la primera opción



Elegimos la base de datos local BIBLIOTECA_CEC, donde queremos que se repliquen las tablas del Nodo FIS, en este caso seleccionamos



Configurar la seguridad, para poder conectarnos al Nodo FIS, con la cuenta del Nodo FIS, su contraseña, junto con las credenciales del usuario SQL.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

☒ Run under the following Windows or Microsoft Entra account:

Process account:
Example: domain\account or account@domain.com

Password:
Confirm Password:

☐ Run under the SQL Server Agent service account (This is not a recommended security best practice.)

Connect to the Publisher and Distributor

☒ By impersonating the process account
☐ Using a SQL Server login

The connection to the server on which the agent runs must impersonate the process account. The process account must be a member of the Publication Access List.

Connect to the Subscriber

☐ By impersonating the process account
☒ Using the following SQL Server login:
☐ Using the following Microsoft Entra login:
☐ Using the following Microsoft Entra principal:

Login:
Password:
Confirm password:

The login used to connect to the Subscriber must be a database owner of the subscription database.

OK Cancel Help

Después de hacer click en OK y posteriormente en Next nos aparecerá la siguiente ventana, donde seleccionamos la opción de Run continuously

New Subscription Wizard

Synchronization Schedule
Specify the synchronization schedule for each agent:

Agent schedule:

Subscriber	Agent Location	Agent Schedule
DESKTOP-FICOSH0	Distributor	Run continuously

Help < Back Next > Finish >> Cancel



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Hacemos click en Next

Initialize Subscriptions
Specify whether to initialize each subscription with a snapshot of the publication data and

Subscription properties:

Subscriber	Memory Optimized	Initialize	Initialize When
DESKTOP-F1COS...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Immediately

A subscription database needs to be initialized with a snapshot of the publication data and schema unless it has already been specially prepared for the subscription.

The Snapshot Agent must run and generate a snapshot of the publication before the subscriptions can be initialized.

Help < Back Next > Finish >>| Cancel

Hacemos click en Next

Subscription Type
Specify the type of each subscription and assign a priority for conflict resolution.

Subscription properties:

Subscriber	Subscription Type	Priority for Conflict Resolution
DESKTOP-F...	Server	75,00

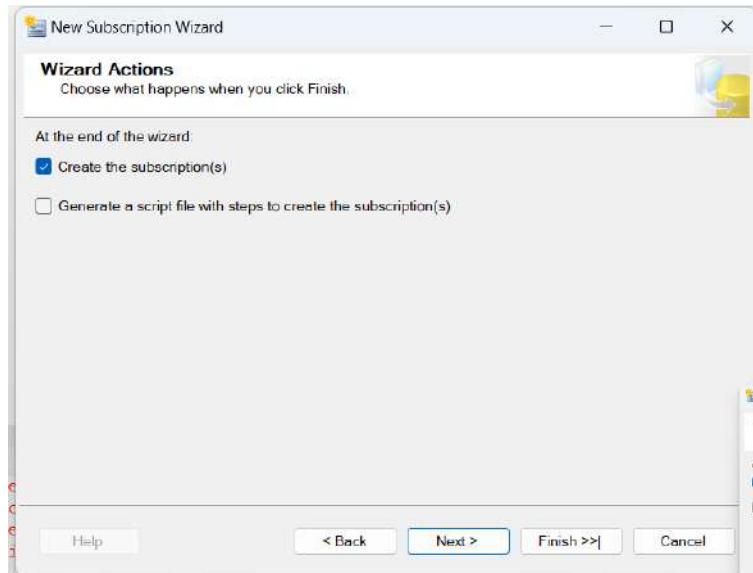
A server subscription can republish the data to, and be a synchronization partner with, other Subscribers. It has its own priority, a number between 0 (lowest priority) and 99.99 (highest priority), for resolving data conflicts. In addition, changes made to download-only articles at the Subscriber are replicated back to the Publisher.

Help < Back Next > Finish >>| Cancel

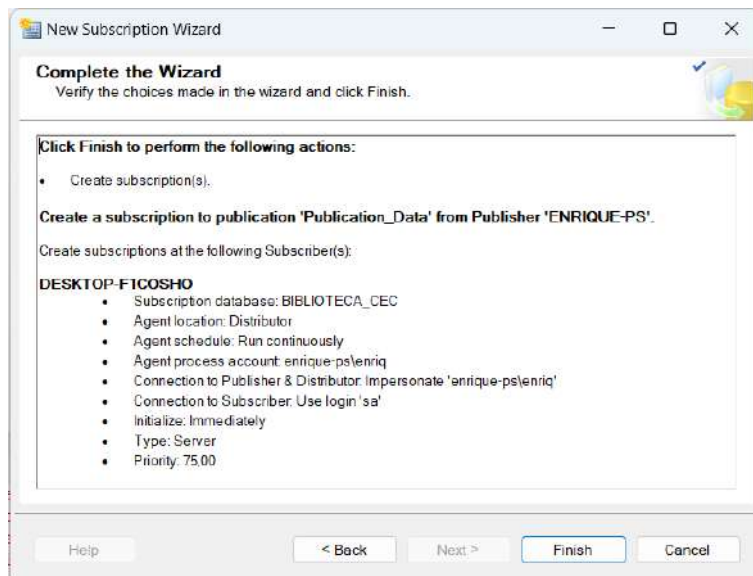


ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Hacemos click en Next



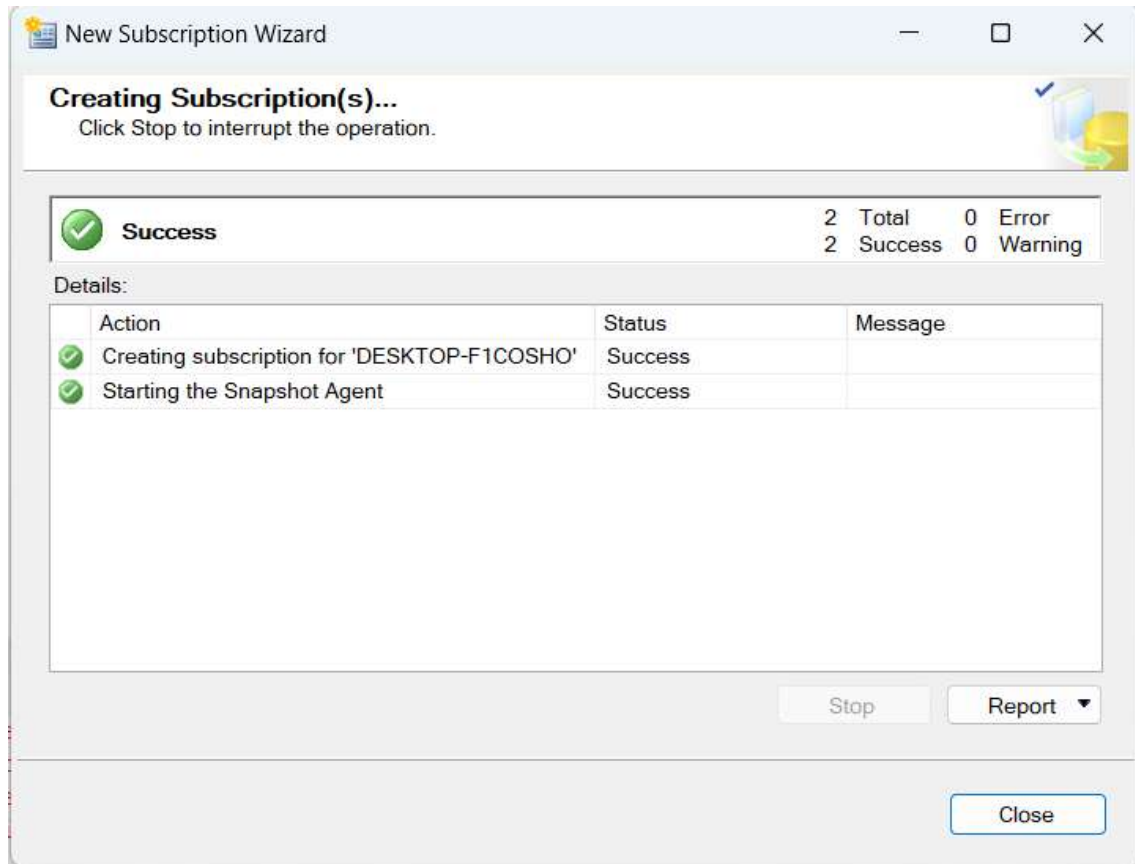
Hacemos click en Finish



La subscripción es creada con éxito como se ve en el Launch monitor



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN



Finalmente podemos verificar que las tablas DATOS_PROVEEDORES fueron replicadas en la base de datos BIBLIOTECA_CEC en el nodo CEC.

Conclusiones y recomendaciones:

- Se ha creado un diseño de un sistema distribuido de bases de datos, con dos nodos: Facultad de Ingeniería en Sistemas (FIS) y Centro de Educación Continua (CEC), que permite mejorar la disponibilidad y el rendimiento de la información con operaciones en ambas bibliotecas. Esto se traduce en una mayor eficiencia operativa y una mejor experiencia para los estudiantes.

Contribución:

Una sola persona realizó el trabajo.

Bibliografía



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

- [1] I. C. Clavijero, «Base de datos distribuidas,» 18 Enero 2023. [En línea]. Available:
https://cursos.clavijero.edu.mx/cursos/080_bdd/modulo3/contenidos/tema3.1.html?opc=0.

 - [2] U. a. d. e. d. Hidalgo, «Apuntes de Administración de Bases de Datos,» 18 Enero 2023.
[En línea]. Available:
<http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro21/index.html>.

 - [3] U. E. d. Milagro, «Diseño de Bases de Datos Distribuidas,» 18 Enero 2023. [En línea].
Available:
https://sga.unemi.edu.ec/media/archivologo/2021/10/26/archivologocompendio_20211026123418.pdf.
-