



Administración de bases de datos

D07

M-J 7-9 AM



## **Práctica 8: Consulta Distribuida**

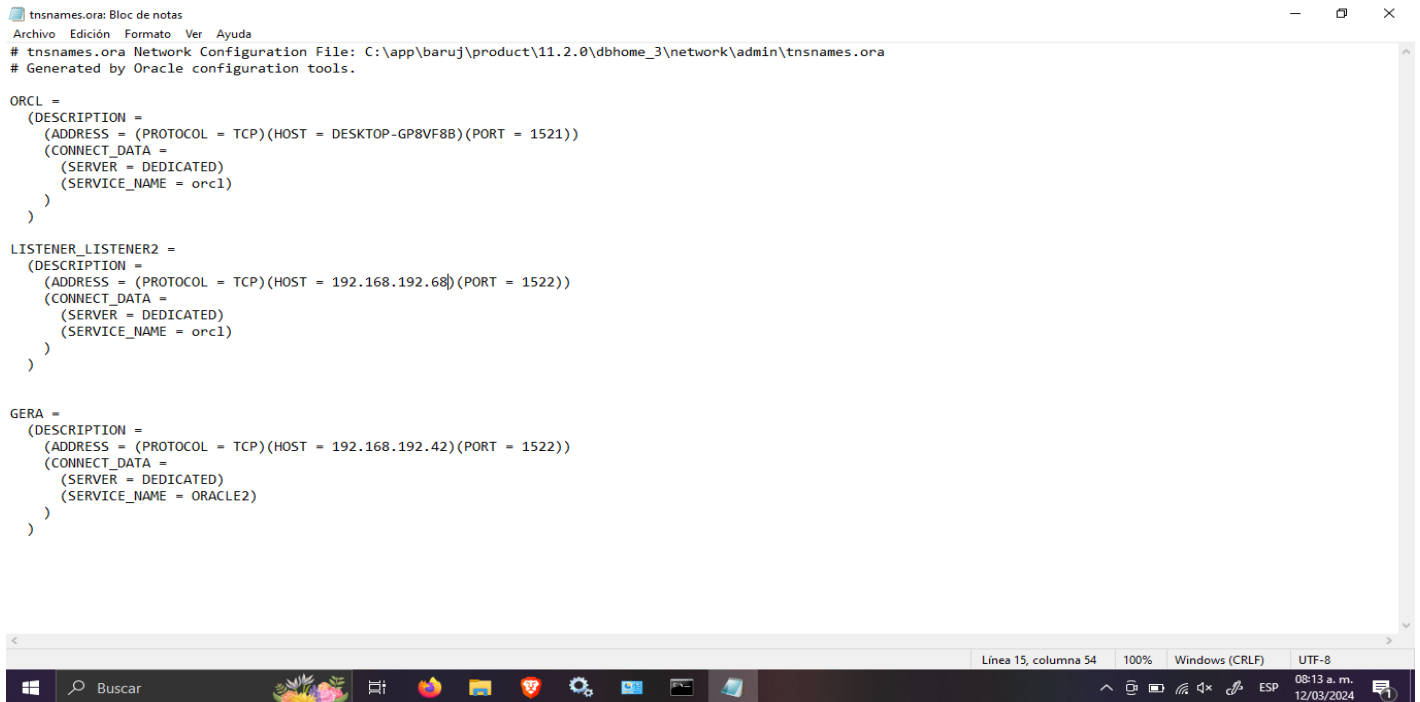
Centro Universitario de Ciencias Exactas e Ingenierías

Universidad de Guadalajara

2024<sup>a</sup>

La actividad se realizó con Gerardo Jasihel Rodriguez Cruz

Se modifica el archivo tnsnames.ora para realizar la conexión con la otra base de datos



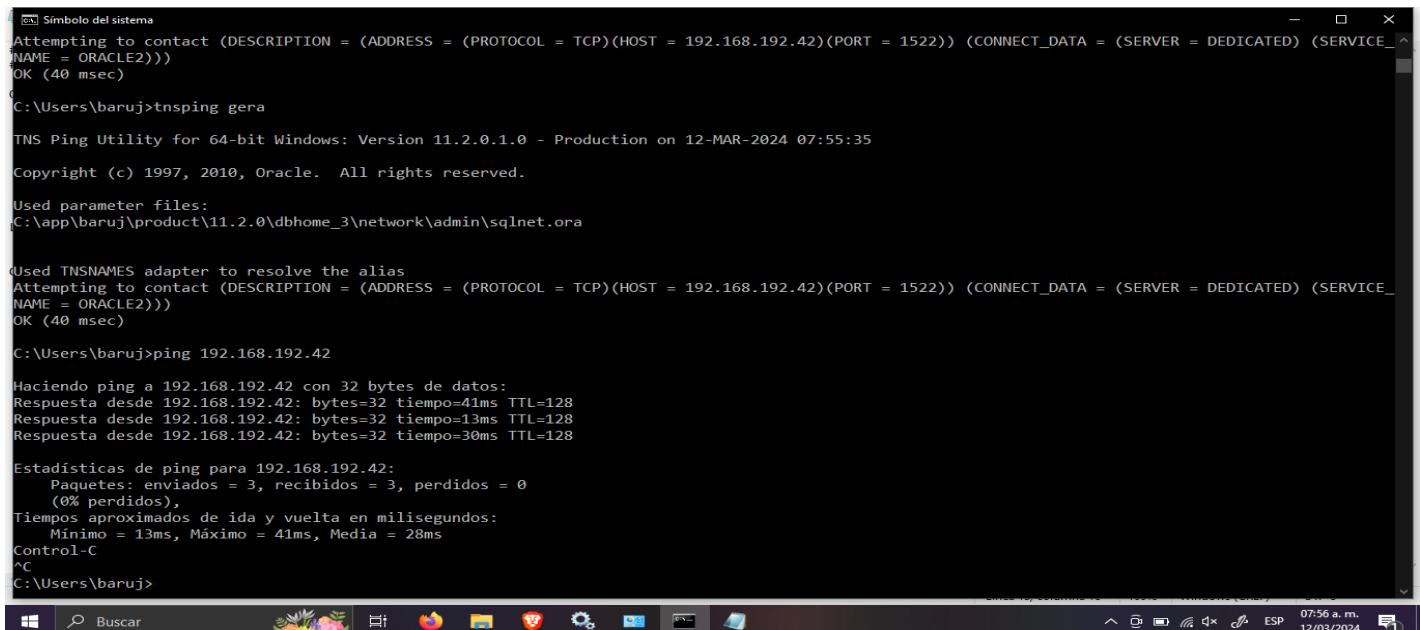
```
tnsnames.ora: Bloc de notas
Archivo Edición Formato Ver Ayuda
# tnsnames.ora Network Configuration File: C:\app\baruj\product\11.2.0\dbhome_3\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = DESKTOP-GP8VF8B)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )

LISTENER_LISTENER2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.192.68)(PORT = 1522))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )

GERA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.192.42)(PORT = 1522))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORACLE2)
    )
  )
```

Se revisa la conexión con tnsping más la configuración del anterior archivo.



```
Símbolo del sistema
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.192.42)(PORT = 1522)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_
NAME = ORACLE2)))
OK (40 msec)
C:\Users\baruj>tnsping gera

TNS Ping Utility for 64-bit Windows: Version 11.2.0.1.0 - Production on 12-MAR-2024 07:55:35

Copyright (c) 1997, 2010, Oracle. All rights reserved.

Used parameter files:
C:\app\baruj\product\11.2.0\dbhome_3\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.192.42)(PORT = 1522)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_
NAME = ORACLE2)))
OK (40 msec)
C:\Users\baruj>ping 192.168.192.42

Haciendo ping a 192.168.192.42 con 32 bytes de datos:
Respuesta desde 192.168.192.42: bytes=32 tiempo=41ms TTL=128
Respuesta desde 192.168.192.42: bytes=32 tiempo=13ms TTL=128
Respuesta desde 192.168.192.42: bytes=32 tiempo=30ms TTL=128

Estadísticas de ping para 192.168.192.42:
Paquetes: enviados = 3, recibidos = 3, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 13ms, Máximo = 41ms, Media = 28ms
Control-C
^C
C:\Users\baruj>
```

Se crea un link a la base de datos del otro nodo con “CREATE DATABASE LINK db\_gera CONNECT TO gerardom IDENTIFIED BY gerardom USING ‘GERA’;” y a partir de ahí vemos alguna de las tablas dentro de ese nodo.

```
SQL> CREATE DATABASE LINK db-gera CONNECT TO gerardom IDENTIFIED BY gerardom USING 'GERA';
CREATE DATABASE LINK db-gera CONNECT TO gerardom IDENTIFIED BY gerardom USING 'GERA'
*
ERROR at line 1:
ORA-00933: comando SQL no terminado correctamente

SQL> CREATE DATABASE LINK db_gera CONNECT TO gerardom IDENTIFIED BY gerardom USING 'GERA';
Database link created.

SQL> SELECT * FROM peliculas@db_gera;

CINE_ASIG  TITULO                DIRECTOR
-----
1 Titanic   James Cameron
2 El Padrino Francis Ford Coppola
3 La La Land Damien Chazelle
3 Forrest Gump Robert Zemeckis
3 Interestelar Christopher Nolan

SQL>
```

### FRAGMENTACIÓN HORIZONTAL (División en filas):

Para ello se crearon dos tablas a partir de una las cuales fueron renombradas

En mi nodo se creó películas\_part1 y en el otro nodo películas\_part2, se usó la misma estructura.

```
CREATE TABLE peliculas_part1 (
    id NUMBER PRIMARY KEY,
    titulo VARCHAR2(20),
    año_lanzamiento NUMBER
);
```

```
INSERT INTO peliculas_part1 VALUES (1, 'Titanic', 1997);
```

```
INSERT INTO peliculas_part1 VALUES (2, 'Matrix', 1999);
```

```
INSERT INTO peliculas_part1 VALUES (3, 'Toy Story', 1995);
```

Lo siguiente muestra la tabla creada en el nodo local, siendo esta las filas de la primera parte de la tabla películas, con sus correspondientes filas para ser completadas con el otro nodo de la base de datos distribuida.

```
Símbolo del sistema - sqlplus
Microsoft Windows [Versión 10.0.19045.4170]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\baruj>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Mar Mar 19 17:29:19 2024

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: mortal_baruj
Enter password:

Connected to:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production

SQL> select * from peliculas_part1;

   ID TITULO          ANO_LANZAMIENTO
-----
    1 Titanic          1997
    2 Matrix            1999
    3 Toy Story         1995

SQL>
```

Para ver la tabla del otro nodo de manera de fragmentación horizontal lo que se hace es utilizar el dblink creado con anterioridad e implementarlo en un comando para unirlos a la tabla se usó “SELECT \* FROM películas\_part1 UNION ALL SELECT \* FROM películas\_part2@db\_gera” para conectarnos y recuperar los datos del otro nodo.

```
Símbolo del sistema - sqlplus
datos remota

SQL> SELECT * FROM peliculas_part1 UNION ALL SELECT * FROM peliculas_part2@db_gera;

   ID TITULO          ANO_LANZAMIENTO
-----
    1 Titanic          1997
    2 Matrix            1999
    3 Toy Story         1995
    4 Avatar            2009
    5 Inception          2010
    6 The Dark Knight    2008

6 rows selected.

SQL>
```

### FRAGMENTACIÓN VERTICAL (Separación de columnas):

Se debe crear una tabla con columnas diferentes para realizar la fragmentación vertical.

CREATE TABLE actores\_part1 (

```
id NUMBER PRIMARY KEY,  
actor VARCHAR2(20)  
);
```

```
INSERT INTO actores_part1 VALUES (1, 'Leonardo Dicaprio');  
INSERT INTO actores_part1 VALUES (2, 'Keanu Reeves');  
INSERT INTO actores_part1 VALUES (3, 'Tom Hanks');  
INSERT INTO actores_part1 VALUES (4, 'Sam Worthington');  
INSERT INTO actores_part1 VALUES (5, 'Cillian Murphy');  
INSERT INTO actores_part1 VALUES (6, 'Cristian Bale');
```

A su vez una tabla así, se creó en el otro nodo con la diferencia de que era con columnas distintas, aquí básicamente se crea la tabla y se empieza a insertar los datos y sus valores.

```
Simbolo del sistema - sqlplus  
C:\Users\baruj>sqlplus  
SQL*Plus: Release 11.2.0.1.0 Production on Mar Mar 19 21:25:49 2024  
Copyright (c) 1982, 2010, Oracle. All rights reserved.  
Enter user-name: mortal_baruj  
Enter password:  
Connected to:  
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production  
SQL> select * from peliculas_part1;  
  
ID TITULO ANO_LANZAMIENTO  
-----  
1 Titanic 1997  
2 Matrix 1999  
3 Toy Story 1995  
SQL> CREATE TABLE actores_part1 ( id NUMBER PRIMARY KEY, actor VARCHAR2(20));  
Table created.  
SQL> INSERT INTO actores_part1 VALUES (1, 'Leonardo Dicaprio');  
1 row created.  
SQL> INSERT INTO actores_part1 VALUES (2, 'Keanu Reeves');INSERT INTO actores_part1 VALUES (3, 'Tom Hanks');INSERT INTO actores_part1 VALUES (4, 'Sam  
Worthington');INSERT INTO actores_part1 VALUES (5, 'Cillian Murphy');INSERT INTO actores_part1 VALUES (6, 'Cristian Bale');  
INSERT INTO actores_part1 VALUES (2, 'Keanu Reeves');INSERT INTO actores_part1 VALUES (3, 'Tom Hanks');INSERT INTO actores_part1 VALUES (4, 'Sam Wort  
hington');INSERT INTO actores_part1 VALUES (5, 'Cillian Murphy');INSERT INTO actores_part1 VALUES (6, 'Cristian Bale')  
*  
ERROR at line 1:
```

Después de ingresar todos los registros para esta tabla fragmentada, esta fue la tabla creada con su columna

```
Simbolo del sistema - sqlplus
SQL> INSERT INTO actores_part1 VALUES (3, 'Tom Hanks');
1 row created.

SQL> INSERT INTO actores_part1 VALUES (4, 'Sam Worthington');
1 row created.

SQL> INSERT INTO actores_part1 VALUES (5, 'Cillian Murphy');
1 row created.

SQL> INSERT INTO actores_part1 VALUES (6, 'Cristian Bale');
1 row created.

SQL> Select * from actores_part1;

   ID ACTOR
-----
1 Leonardo Dicaprio
2 Keanu Reeves
3 Tom Hanks
4 Sam Worthington
5 Cillian Murphy
6 Cristian Bale

6 rows selected.

SQL>
```

Se utiliza la siguiente consulta para ver y juntar los datos de ambas tablas con sus respectivas columnas, lo que hace es unir los datos de las columnas de ambos nodos con sus fragmentos de la tabla “SELECT A.actor, B.nacionalidad FROM actores\_part1 AJOIN actores\_part2@db\_gera B ON A.id = B.id;” De manera que en este ocasión se utiliza el id como apoyo para unir de manera vertical las columnas de estas dos tablas.

```
Seleccionar Símbolo del sistema - sqlplus
SQL> SELECT A.actor, B.nacionalidad
2 FROM actores_part1 A
3 JOIN actores_part2@db_gera B ON A.id = B.id;

ACTOR              NACIONALIDAD
-----
Leonardo Dicaprio  Estado Unidense
Keanu Reeves        Canadiense
Tom Hanks           Estado Unidense
Sam Worthington     Britanico
Cillian Murphy      Irlandes
Cristian Bale       Britanico

6 rows selected.

SQL>
```

### FRAGMENTACIÓN MIXTA:

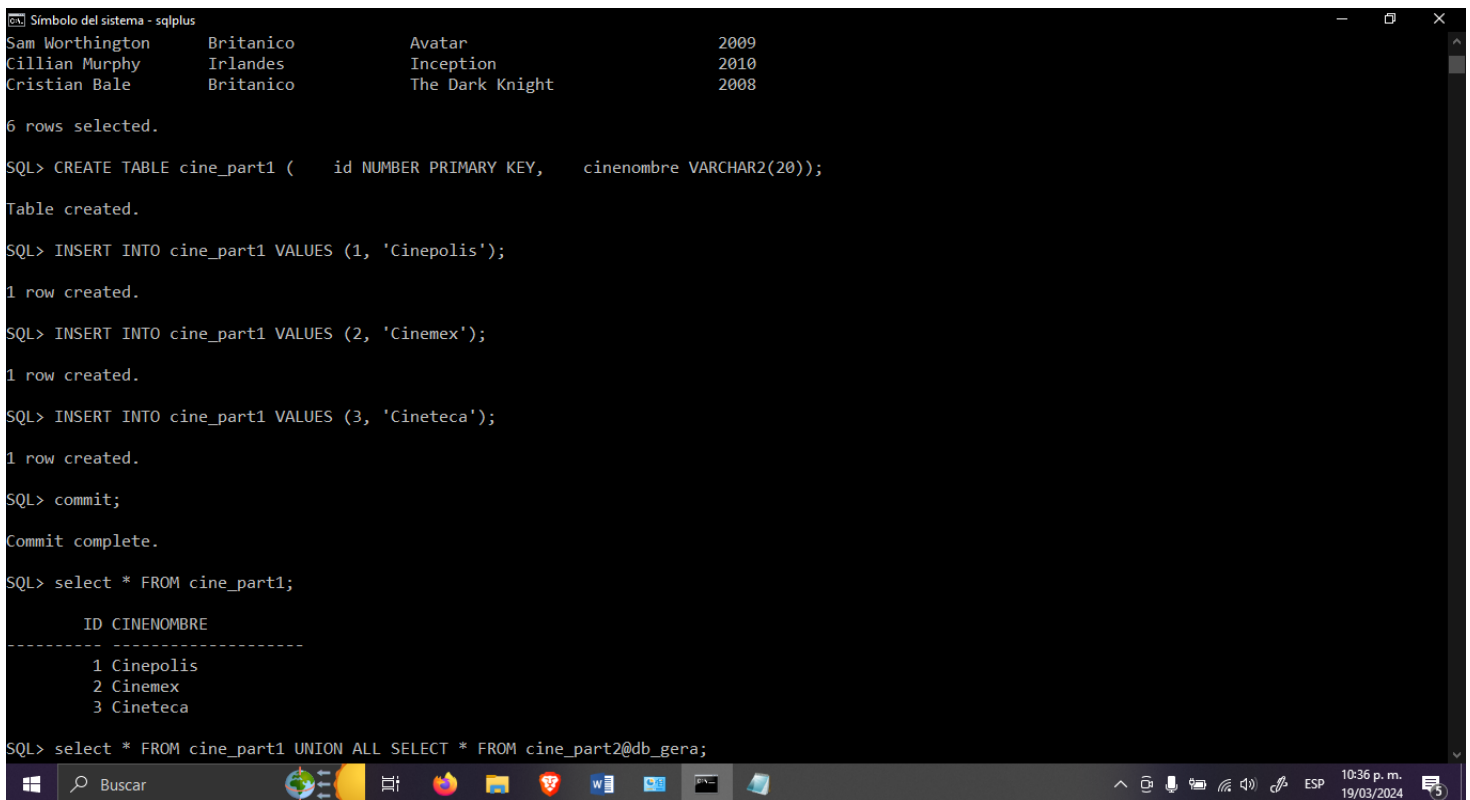
Se crea otra tabla para los registros de cine conocida como tabla cine, la cual se divide en cine\_part1 y cine\_part2, también se divide de manera horizontal, yo me quedé con cine\_part1, y se ingresan los siguientes datos:

```
CREATE TABLE cine_part1 (  
    id NUMBER PRIMARY KEY,  
    cinenombre VARCHAR2(20)  
);
```

```
INSERT INTO cine_part1 VALUES (1, 'Cinepolis');
```

```
INSERT INTO cine_part1 VALUES (2, 'Cinemex');
```

```
INSERT INTO cine_part1 VALUES (3, 'Cineteca');
```



```
Simbolo del sistema - sqlplus  
Sam Worthington      Britanico      Avatar      2009  
Cillian Murphy      Irlandes      Inception      2010  
Cristian Bale      Britanico      The Dark Knight      2008  
  
6 rows selected.  
  
SQL> CREATE TABLE cine_part1 (    id NUMBER PRIMARY KEY,    cinenombre VARCHAR2(20));  
Table created.  
  
SQL> INSERT INTO cine_part1 VALUES (1, 'Cinepolis');  
1 row created.  
  
SQL> INSERT INTO cine_part1 VALUES (2, 'Cinemex');  
1 row created.  
  
SQL> INSERT INTO cine_part1 VALUES (3, 'Cineteca');  
1 row created.  
  
SQL> commit;  
Commit complete.  
  
SQL> select * FROM cine_part1;  
  
    ID CINENOMBRE  
-----  
    1 Cinepolis  
    2 Cinemex  
    3 Cineteca  
  
SQL> select * FROM cine_part1 UNION ALL SELECT * FROM cine_part2@db_gera;
```

En la siguiente imagen podemos observar la parte de la tabla que me corresponde, además de visualizar los registros del mismo modo que se hizo con la fragmentación horizontal

```
Símbolo del sistema - sqlplus

1 row created.

SQL> INSERT INTO cine_part1 VALUES (3, 'Cineteca');

1 row created.

SQL> commit;

Commit complete.

SQL> select * FROM cine_part1;

   ID CINEOMBRE
-----
    1 Cinopolis
    2 Cinemex
    3 Cineteca

SQL> select * FROM cine_part1 UNION ALL SELECT * FROM cine_part2@db_gera;

   ID CINEOMBRE
-----
    1 Cinopolis
    2 Cinemex
    3 Cineteca
    4 Cinebox
    5 Cine bahia
    6 Cine toro

6 rows selected.

SQL> SELECT A.actor, B.nacionalidad, P.titulo, P.ano_lanzamiento, C.cinenombre
2 FROM (
3   SELECT A.actor, B.nacionalidad, P.titulo, P.ano_lanzamiento
```

Con la siguiente consulta, nos permite tomar los datos de las tres tablas y juntarlos para visualizar la consulta de la fragmentación mixta, donde se unen los datos por medio del id y con ello juntar la tabla películas, actores y cines:

```
SELECT AP.actor, AP.nacionalidad, AP.titulo, AP.ano_lanzamiento, C.cinenombre
FROM (
SELECT A.actor, B.nacionalidad, P.titulo, P.ano_lanzamiento, A.id
FROM actores_part1 A
JOIN actores_part2@db_gera B ON A.id = B.id
JOIN peliculas_part1 P ON A.id = P.id
UNION ALL
SELECT A.actor, B.nacionalidad, P.titulo, P.ano_lanzamiento, A.id
FROM actores_part1 A
JOIN actores_part2@db_gera B ON A.id = B.id
JOIN peliculas_part2@db_gera P ON A.id = P.id
) AP
JOIN (
SELECT * FROM cine_part1
UNION ALL
SELECT * FROM cine_part2@db_gera
```



) C ON AP.id = C.id;

```
Simbolo del sistema - sqlplus

6 rows selected.

SQL> SET LINESIZE 200
SQL> SELECT AP.actor, AP.nacionalidad, AP.titulo, AP.ano_lanzamiento, C.cinenombre
  2 FROM (
  3     SELECT A.actor, B.nacionalidad, P.titulo, P.ano_lanzamiento, A.id
  4     FROM actores_part1 A
  5     JOIN actores_part2@db_gera B ON A.id = B.id
  6     JOIN peliculas_part1 P ON A.id = P.id
  7     UNION ALL
  8     SELECT A.actor, B.nacionalidad, P.titulo, P.ano_lanzamiento, A.id
  9     FROM actores_part1 A
 10     JOIN actores_part2@db_gera B ON A.id = B.id
 11     JOIN peliculas_part2@db_gera P ON A.id = P.id
 12 ) AP
 13 JOIN (
 14     SELECT * FROM cine_part1
 15     UNION ALL
 16     SELECT * FROM cine_part2@db_gera
 17 ) C ON AP.id = C.id;
```

ACTOR	NACIONALIDAD	TITULO	ANO_LANZAMIENTO	CINENOMBRE
Leonardo Dicaprio	Estado Unidense	Titanic	1997	Cinepolis
Keanu Reeves	Canadiense	Matrix	1999	Cinemex
Tom Hanks	Estado Unidense	Toy Story	1995	Cineteca
Sam Worthington	Britanico	Avatar	2009	Cinebox
Cillian Murphy	Irlandes	Inception	2010	Cine bahia
Cristian Bale	Britanico	The Dark Knight	2008	Cine toro

```
6 rows selected.

SQL>
```

En conclusión, se pudo realizar la consulta mixta entre dos nodos con tres tablas, dos fragmentadas de manera horizontal y una en vertical, por lo que la actividad fue realizada satisfactoriamente.