CSCI585 - Database Systems
Spring 2008

Final Exam

Please Complete the Following:

Name	
Student ID #	
Location (if remote)	

Notes:

1. ***Duration of the exam is 1 hour and 30 minutes***
2. ***Total number of pages including this page: 23***

For grader use only:

	Maximum	Received
Problem 1	10	
Problem 2	24	
Problem 3	22	
Problem 4	12	
Problem 5	20	
Problem 6	12	
Total	100	

XML
Problem 1 (10 points)

Consider the following XML documents and the corresponding XML schemas. Are the schemas correct? If not, make minimal required changes to correct them.

Customer.xml:

```
<Customer>
    <Dob> 2000-01-12T12:13:14Z </Dob>
    <Address>
        <Line1>34 thingy street, someplace</Line1>
        <Line2>sometown, w1w8uu </Line2>
     </Address>
</Customer>
```

Supplier.xml:

```
<Supplier location="CA">
    <Phone>0123987654</Phone>
    <Address>
        <Line1>22 whatever place, someplace</Line1>
        <Line2>sometown, ss1 6gy </Line2>
       </Address>
</Supplier>
```

Customer.xsd:

```
<xs:complexType name="AddressType">
<xs:sequence>
        <xs:element name="Line1" type="xs:string">
        <xs:element name="Line2" type="xs:string">
</xs:sequence>
</xs:complexType>

<xs:element name="Customer" type=”xsd:string”>
    <xs:complexType>
     <xsd:SimpleContent>
        <xs:sequence>
             <xs:element name="Dob" type="xs:date">
              <xs:element name="Address" type="AddressType">
           </xs:sequence>
     </xsd:SimpleContent>
    </xs:complexType>
</xs:element>
```

Supplier.xsd:

```
<xs:element name="supplier" type=”xsd:string”>
        <xs:complexType>
                <xsd:SimpleContent>
                        <xsd:attribute ref="SupplierID">
                         <xsd:attribute name=”location” use="required">
                                 <xsd:SimpleType>
                                         <xsd:restriction base=”xsd:integer”>
                                                 <xsd:enumeration value=”CA”>
                                                 <xsd:enumeration value=”NY”>
                                                 <xsd:enumeration value=”OC”>
                                         </xsd:restriction>
                                 </xsd:SimpleType>
                        <xs:sequence>
                                 <xs:element name="address" type="AddressType">
                                 <xs:element name="phone" type="xs:integer">
                        </xs:sequence>
                </xsd:SimpleContent>
        </xs:complexType>
</xs:element>
<xsd:attribute name="SupplierID" type="xsd:string" use="optional">
```

Customer.xsd:

```
<xs:complexType name="AddressType">
<xs:sequence>
        <xs:element name="Line1" type="xs:string"/>
        <xs:element name="Line2" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Customer" remove:type=”xsd:string”>
    <xs:complexType>
      Remove:<xsd:SimpleContent>
         <xs:sequence>
            <xs:element name="Dob" type="xs:date"/>
               <xs:element name="Address" type="AddressType"/>
            </xs:sequence>
      Remove:</xsd:SimpleContent>
    </xs:complexType>
</xs:element>
```

Supplier.xsd:

```
<xs:element name="supplier"  remove:type=”xsd:string”>
   <xs:complexType>
      Remove:<xsd:SimpleContent>
       <xsd:attribute ref="SupplierID" use="optional >
       <xsd:attribute name=”location” use="required">
          <xsd:SimpleType>
            <xsd:restriction base=”xsd:string”>
                        <xsd:enumeration value=”CA” />
                              <xsd:enumeration value=”NY” />
                     <xsd:enumeration value=”OC” />
            </xsd:restriction>
          </xsd:SimpleType>
<xs:sequence>
             <xs:element name="address" type="AddressType"/>
             <xs:element name="phone" type="xs:integer"/>
        </xs:sequence>
      Remove:<</xsd:SimpleContent>
   </xs:complexType>
</xs:element>
<xsd:attribute name="SupplierID" type="xsd:string" remove:use="optional">
```

XML
Problem 2 (24 points)

Consider the following xml files storing the account numbers and transactions between accounts:

Accounts.xml:

```
<?xml version="1.0"?>
<Accounts>
<Account >
<OwnerName>Farnoush Banaei-Kashani </OwnerName>
<AccountNumber>123-456-789</AccountNumber>
</Account>
…
</Accounts>
```

Transactions.xml:

```
<?xml version="1.0"?>
<Transactions>
<Transaction  id =0>
<PayerAccount>234-456-789</PayerAccount>
<Amount Unit=”$”>80</Amount>
<PayeeAccount>123-456-789</ PayeeAccount>
</Transaction>
…
</Transactions>
```

Write the corresponding XQueries for the following queries assuming the given XML files.

a) Display the owner names and the account numbers which have received more than $50 from other accounts in average. The query output should be in the following format. (8 points)

```
<Qa>
<Account>
<OwnerName> Farnoush Banaei-Kashani </OwnerName>
< AccountNumber >123-456-789</ AccountNumber >
<AvgMoneyReceived>56</AvgMoneyReceived>
</Account>
…
</Qa>
```

```
<Qa>
{
for $ac in document("accounts.xml")//Account
let $t := document("transactions.xml")//transaction[PayeeAccount =
$ac/AccountNumber]
where avg($t/Amount) > "50"
return
<Account>
{$ac/OwnerName}
{$ac/AccountNumber}
< AvgMoneyReceived >
{avg($t/ Amount)}
</ AvgMoneyReceived >
</Account>
}
</Qa>
```

b) Display all the people who sent money to "Farnoush Banaei-Kashani ". Sort the output alphabetically. The output should be in the following format. (8 points)

```
<Qb>
        <Name> Shahin Shayandeh </Name>
        <Name> Ugur Demiryurek</Name>
</Qb>
```

```
<Qb>
{
let $p2 := document("accounts.xml")//Account[OwnerName = " Farnoush
Banaei-Kashani "]
let $t := document("transactions.xml")//Transaction[PayeeAccount =
$p2/AccountNumber]
for $p1 in distinct-
values(document("accounts.xml")//Account[AccountNumber =
$t/PayerAccount])
return
<Name>
{$p1/OwnerName/text()}
</Name>
sortby (Name ascending)
}
</Qb>
```

c) Display all the account numbers owned by each person sorted by the account number in descending order. The output should be in the following format. (8 points)

```
<Qc>
<Owner>
<Name>Farnoush Banaei-Kashani </Name>
<Number>123-456-789</Number>
<Number>213-456-689</Number>
</Owner>
…
</Qc>
```

```
<Qc>
{
for $o in distinct-
values(document("accounts.xml")//Account/OwnerName)
let $p := document("accounts.xml")//Account[OwnerName = $o]
return
<Owner>
```

```
<Name>
{$o/text()}
</Name>
{$p/AccountNumber}
sortby (AccountNumber descending)
</Owner>
}
</Qc>
```

OLAP
Problem 3 (22 points)

Consider the following 2-dimensional cube. Each cell defines salary for the corresponding state and age ("salary" is measure attribute while "age" and "state" are dimension attributes).

	CA	NY	DC	OR	AZ	TX
20	20	20	40	20	30	30
25	50	40	50	40	40	50
30	50	50	40	50	60	50
35	70	50	70	70	50	60
40	60	60	80	80	70	70

a) Draw the corresponding Prefix-Sum cube. (3 points)

20	40	80	100	130	160
70	130	220	280	350	430
120	230	360	470	600	730
190	350	550	730	910	1100
250	470	750	1010	1260	1520
320	620	990	1340	1660	2000

b) Use the Prefix-Sum cube to answer the following query. (3 points)
(Note: Identify the Prefix-Sum cells that you use to answer the query.)

"What is the average salary in states AZ and TX and for ages between 30 and 40?"
(AZ<=state<=TX and 30<=age<=40)

20	40	80	100	130	160
70	130	220	280	350	430
120	230	360	470	600	730
190	350	550	730	910	1100
250	470	750	1010	1260	1520
320	620	990	1340	1660	2000

Sum= 1520-1010-430+280=360
AVG=360/6=60

c) How much does the above query cost? (1 points)
(Note: Count the number of I/O operations required to answer the query, assuming each cell is stored in a separate disk block.)

4

d) Draw the corresponding Space Efficient Relative Prefix Sum (SRPS) data cube. (6 points)

20	**20**	**60**	**100**	**30**	**60**
50	40	90	**180**	40	90
100	90	180	**370**	100	200
190	**160**	**360**	**730**	**180**	**370**
60	60	140	**280**	70	140
130	140	310	**610**	140	290

e) Answer the query mentioned in part (b) using the SRPS cube. (4 points)
(Note: Identify the cells that you use to answer the query.)

20	**20**	**60**	**100**	**30**	**60**
50	40	90	**180**	40	90
100	90	180	**370**	100	200
190	**160**	**360**	**730**	**180**	**370**
60	60	140	**280**	70	140
130	140	310	**610**	140	290

SUM=370+140-60-90
AVG=360/6=60

f) How much does the query cost when answered using the SRPS cube? (1 points) (Note: Count the number of I/O operations required to answer the query, assuming each cell is stored in a separate disk block.)

4

g) Suppose we change the salary at cell <30,CA> to 60 in the original data cube. How many cells must be updated in the Prefix-Sum cube? How many cells must be updated in the SRPS cube? (4 points)

Prefix Sum

20	40	80	100	130	160
70	130	220	280	350	430
130	240	370	480	610	740
200	360	560	740	920	1110
260	480	760	1020	1270	1530
330	630	1000	1350	1670	2010

24 cells

SRPS

20	**20**	**60**	**100**	**30**	**60**
50	40	90	**180**	40	90
110	90	180	**380**	100	200
200	**160**	**360**	**740**	**180**	**370**
60	60	140	**280**	70	140
130	140	310	**610**	140	290

4 cells

OLAP
Problem 4 (12 points)

Consider the following one dimensional data cube:

A [1:8]=	16	8	4	12	32	16	4	8

a) Draw the corresponding Haar wavelet transformation assuming h=[1/√2 , 1/√2] and g=[1/√2 , -1/√2] . (4 points)

AW [1:8]=	25√2	-5√2	4	18	4√2	-4√2	8√2	-2√2

b) Answer the following query using the transformed array. (4 points)
(Note: Identify the transformed array cells you use.)

Sum(A[i]) where (2<=i<=6)

Q [1:8]=	0	1	1	1	1	1	0	0

QW [1:8]=	5/(2*√2)	1/(2*√2)	-0.5	1	-1/√2	0	0	0

AW*QW=125/2-5/2-2+18-4=72

c) How much is the cost for the above query? (1 point)

QW [1:8]=

5/(2*√2)	1/(2*√2)	-0.5	1	-1/√2	0	0	0

5 (number of non-zeros)

d) Suppose we update the cell A[4] from 12 to 20. How much does this update cost? (3 points)

log(8)+1=4

DDBMS
Problem 5 (20 points)

Consider the following database at BooksDotCom:

- CUSTOMER (Cid, Cname, Ccity, Cstate)
- BOOK (Book #, Author, Price, Topic, ...)
- ORDER (Order#,Cid,Book#, Order_date,Payment_type, ...)
- AUTHOR (Author, Affiliation,Author-type, ...)

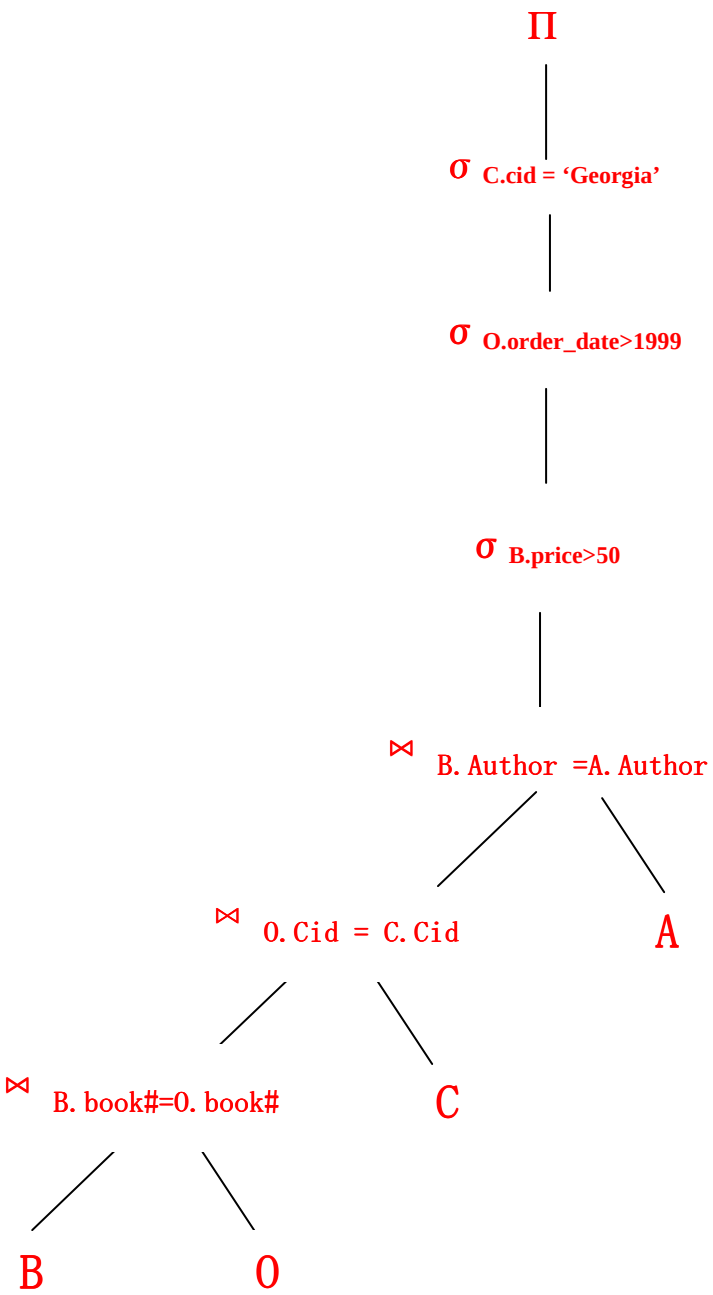
Each table has some additional fields that we are not interested in.

a) Write the following query in SQL. (3 points)

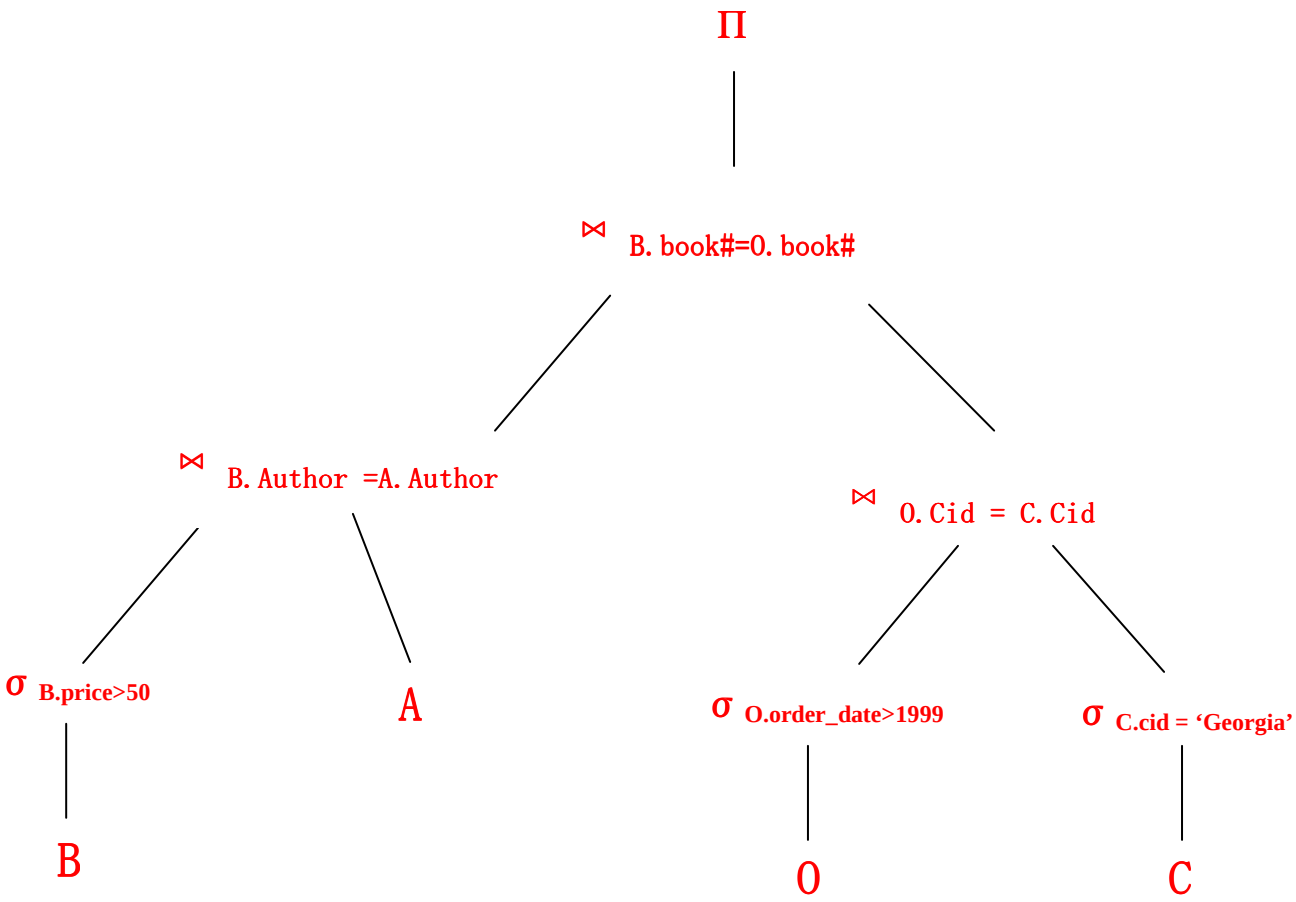
"List the Book #, Price, Author-type, Customer_city, Date of order for books ordered by customers in Georgia after 1998 (order_date > "01-JAN-1999"); only include books that cost more than $50.00"

```
Select B.Book#, B.Price, A.Author-type, C.Ccity, O.orade_date
From Customer C, Book B, Order  O, Author A
Where B.Book# = O.Book# AND
        O.Cid = C.Cid AND
        B.Author = A.Author AND
        B.Price > 50 AND
        O.order_data > "01-JAN-1999" AND
        C.Ccity = "Georgia"
```

b) Draw the relational algebra query tree for the query. (3 points)



c) Use the heuristics for algebraic query optimization to transform/restructure the query-tree you generated at part (b) above into a more efficient query-tree.
(6 points)



d) Given the following database catalog information and the fragmentation of relations across sites, show how this query can be optimally executed in a distributed environment. Present your solution by generating a list of operations that must be performed in sequence to execute the query. Each operation is either a local query performed at a particular site or a data transfer between sites. In your list, number the operations in chronological order, and use the same number for the operations that can happen in parallel. (8 points)

Catalog information:

- Cardinality of BOOK is 10 million
- Cardinality of ORDER is 200 million
- Cardinality of CUSTOMER is 30 million
- Cardinality of AUTHOR is 3 million
- Orders are saved from Jan-01-1972 to Nov-27-2006
- Book prices range from $5 to $60.
- 5% of all customers are from Georgia
- Assume uniform distributions

Fragmentation:

- Site 1: Customers with Cid ≤ C5000000
- Site 2: Customers with Cid > C5000000
- Site 3: Orders corresponding to books with Cid ≤ C5000000
- Site 4: Orders corresponding to books with Cid > C5000000
- Site 5: Books with Book #'s ≤ B2000000
- Site 6: Books with Book #'s > B2000000
- Site 7: Authors of books with Book #'s ≤ B2000000
- Site 8: Authors of books with Book #'s > B2000000

a- Select customers with cid = 'georgia' in site 1, Select orders with order_date > '01-JAN-1999' in site 3 → join in site 1 or 3 (consider 1)

a- Select customers with cid = 'georgia' in site 2, Select orders with order_date > '01-JAN-1999' in site 4 → join in site 2 or 4 (consider 2)

a- Select Books with price > 50 in site 5, Select authors in site 7 → join in site 5 or 7 (consider 5)

a- Select Books with price > 50 in site 6, Select authors in site 8 → join in site 6 or 8 (consider 6)
b- send data from site 6 to site 5
b- send data from site 2 to site 1

c- send all data from site 5 to site 1

d- join over book# in site 1 and project the results

DDBMS
Problem 6 (12 points)

DotCom, an online retailer, carries three categories of products: books, music, and video. DotCom has three warehouses, one in New Jersey (NJ), one in California (CA), and one in Colorado (CO).

- The CA warehouse carries books, music and videos, and ships orders for the US-West and US-Central regions.
- The NJ warehouse carries music and videos, and ships orders for the US-East and US-Central regions.
- The CO warehouse carries books and music, and ships orders anywhere in the US.

The DotCom database has three tables:

- PROD(PID, CAT, DESC, ...)
- INV(PID, WID, UNITS, ...)
- ORD(OID, PID, SHIP, ...)

Each table has some additional fields we are not interested in. Some additional details:

- PID is the primary key for the products relation PROD.
- (PID, WID) is a key for the inventory relation INV.
- (OID, PID) is a key for the orders relation ORD.
- CAT is one of “book”, “music”, and “video”.
- WID is one of “CA”, “CO”, and “NJ”.
- SHIP is one of “US-West”, “US-Central”, and “US-East.”

The CA warehouse issues queries of the form:

```
SELECT * FROM PROD, ORD, INV
WHERE PROD.PID = ORD.PID AND
PROD.PID = INV.PID AND
WID = "CA" AND
(SHIP = "US-West" OR SHIP = "US-Central")
```

The CO warehouse issues queries of the form:

*SELECT * FROM PROD, ORD, INV*
WHERE PROD.PID = ORD.PID AND
PROD.PID = INV.PID AND
WID = "CO" AND
(CAT= "book" OR CAT= "music")

The NJ warehouse issues queries of the form:

*SELECT * FROM PROD, ORD, INV*
WHERE PROD.PID = ORD.PID AND
PROD.PID = INV.PID AND
WID = "NJ" AND
(CAT="music" OR CAT="video") AND
(SHIP = "US-East" OR SHIP = "US-Central")

a) What are the sets of relevant simple predicates to (horizontally) fragment each of the tables PROD, ORD, and INV? (3 points)

- Predicates for PROD:
  PROD: CAT = "book" , CAT = "music" , CAT = "video"

- Predicates for ORD:
  ORD: SHIP = "US-West" , SHIP = "US-Central" , SHIP = "US-East"

- Predicates for INV:
  INV: WID = "CA" , WID = "CO" , WID = "NJ"

b) What is the corresponding primary fragment for each predicate derived in part (a) above? (3 points)

- Fragments for PROD:
  PROD: $\sigma_{CAT="book"}(PROD)$, $\sigma_{CAT="music"}(PROD)$, $\sigma_{CAT="video"}(PROD)$

- Fragments for ORD:
  ORD: $\sigma_{SHIP="US-West"}(ORD)$, $\sigma_{SHIP="US-Central"}(ORD)$, $\sigma_{SHIP="US-East"}(ORD)$

- Fragments for INV:
  INV: $\sigma_{WID="CA"}(INV)$, $\sigma_{WID="CO"}(INV)$, $\sigma_{WID="NJ"}(INV)$

c) We now wish to further fragment each of the INV fragments, using derived horizontal fragmentation with PROD as the owner relation (i.e., we want to partition each INV fragment from part (b) into "smaller" sub-fragments.). List all the derived sub-fragments which are valid/meaningful. (3 points)

Sub-fragments for all primary INV fragments are:

$\sigma_{WID="CA"}(INV) \ltimes \sigma_{CAT="book"}(PROD)$
$\sigma_{WID="CA"}(INV) \ltimes \sigma_{CAT="music"}(PROD)$
$\sigma_{WID="CA"}(INV) \ltimes \sigma_{CAT="video"}(PROD)$
$\sigma_{WID="CO"}(INV) \ltimes \sigma_{CAT="book"}(PROD)$
$\sigma_{WID="CO"}(INV) \ltimes \sigma_{CAT="music"}(PROD)$
$\sigma_{WID="NJ"}(INV) \ltimes \sigma_{CAT="music"}(PROD)$
$\sigma_{WID="NJ"}(INV) \ltimes \sigma_{CAT="video"}(PROD)$

d) In order to apply derived horizontal fragmentation to the primary fragments of ORD obtained in part (b), what relation should be used as the owner relation? Pick a primary fragment of ORD (i.e., any fragment of your choice) and list the corresponding derived sub-fragments. (3 points)

The owner relation is:
Owner Relation is PROD, because ORD $\ltimes$ INV fragmentation is not guaranteed to be disjoint, since PID is not a primary key for the INV table.

Sub-fragments for one primary ORD fragment are:

Either:
$\sigma_{SHIP="US-West"}(ORD) \ltimes \sigma_{CAT="book"}(PROD)$
$\sigma_{SHIP="US-West"}(ORD) \ltimes \sigma_{CAT="music"}(PROD)$
$\sigma_{SHIP="US-West"}(ORD) \ltimes \sigma_{CAT="video"}(PROD)$
Or
$\sigma_{SHIP="US-Central"}(ORD) \ltimes \sigma_{CAT="book"}(PROD)$
$\sigma_{SHIP="US-Central"}(ORD) \ltimes \sigma_{CAT="music"}(PROD)$
$\sigma_{SHIP="US-Central"}(ORD) \ltimes \sigma_{CAT="video"}(PROD)$
Or
$\sigma_{SHIP="US-East"}(ORD) \ltimes \sigma_{CAT="book"}(PROD)$
$\sigma_{SHIP="US-East"}(ORD) \ltimes \sigma_{CAT="music"}(PROD)$

$\sigma_{SHIP=\text{“US-East”}}(ORD) \square \sigma_{CAT=\text{“video”}}(PROD)$