**Assignment 06**

**Distributed Databases**

1. Consider the following partial distributed database schema for a library system that shares resources (much like our University library system):

|  |  |  |
| --- | --- | --- |
| Tables | Fragments | Location |
| BOOKCOPY | BOOKCPY\_A BOOKCPY\_B | A B |
| BOOK | - | A |
| REQUEST | - | B |
| BRANCH | - | B |

For each of the following SQL transactions, identify the type of operation the distributed database must support (remote request, remote transaction, distributed transaction, distributed requests) and explain why you chose that type of operation.

* 1. BEGIN;

SELECT title

FROM book

WHERE isbn='0900596929';

COMMIT;

Ans: Remote Request, it consists SQL statement in which, the query accesses data at single location A.

* 1. BEGIN;

SELECT isbn, COUNT(isbn)

FROM bookcopy

GROUP BY isbn;

COMMIT;

Ans:

Distributed Request, the whole transaction refers multiple locations as well as the query is also referring multiple locations.

* 1. BEGIN;

SELECT title, isbn, COUNT(isbn)

FROM bookcopy, book

WHERE bookcopy.isbn = book.isbn

GROUP BY isbn,title;

COMMIT;

Ans: Distributed Request, the whole transaction refers multiple locations as well as the query is also referring multiple locations.

* 1. BEGIN;

SELECT isbn FROM request;

SELECT street\_addr FROM branch;

COMMIT;

Ans:

Remote Transaction, among the four statements, both the SELECT queries are accessing data at single site (both request and branch are at location B)

* 1. BEGIN;

INSERT INTO book VALUES (‘1491954461', 'MongoDB: The Definitive Guide: Powerful and Scalable Data Storage');

INSERT INTO request VALUES ('1491954461', 'branch1', '21-JUN-2020');

COMMIT;

Ans: Distributed Transaction, both the above insert statements are inserting values to tables at Location A and B (book and request).

1. Consider the following (partial) table of books:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| 1491954461 | OReilly | False | 2019 | 33.49 |
| B004G606K8 | OReilly | False | 2020 | 37.99 |
| 1848000693 | Springer | False | 2011 | 34.33 |
| 0394820371 | Bullseye Books | True | 1988 | 7.89 |
| B00HTJTX1C | Tor Fantasy | True | 1990 | 23.99 |
| B004G606K8 | Wizards of the Coast | True | 2010 | 7.99 |

* 1. Show how to fragment the above data horizontally by whether it is fiction or not. Call the fragments FICTION and NONFICTION

FICTION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| 0394820371 | Bullseye Books | True | 1988 | 7.89 |
| B00HTJTX1C | Tor Fantasy | True | 1990 | 23.99 |
| B004G606K8 | Wizards of the Coast | True | 2010 | 7.99 |

NON FICTION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| 1491954461 | OReilly | False | 2019 | 33.49 |
| B004G606K8 | OReilly | False | 2020 | 37.99 |
| 1848000693 | Springer | False | 2011 | 34.33 |

* 1. Fragment each of your fragments from part a by publisher. Name each of the new fragments using the existing fragment name, followed by an underscore, followed by the publisher’s name.

FICTION\_BULLSEYE\_BOOKS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| 0394820371 | Bullseye Books | True | 1988 | 7.89 |

FICTION\_TOR\_FANTASY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| B00HTJTX1C | Tor Fantasy | True | 1990 | 23.99 |

FICTION\_wIZARDS\_OF\_THE\_COAST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| B004G606K8 | Wizards of the Coast | True | 2010 | 7.99 |

NON\_FICTION\_OREILLY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| 1491954461 | OReilly | False | 2019 | 33.49 |
| B004G606K8 | OReilly | False | 2020 | 37.99 |

NON\_FICTION\_SPRINGER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASIN\_ISBN | PUBLISHER | IS\_FICTION | YEAR | PURCHASE\_PRICE |
| 1848000693 | Springer | False | 2011 | 34.33 |

1. Consider the following situation, adapted from Date. A simplified procurement (relational) database has the following three relations:

SUPPLIER (SUPPLIER\_NUMBER, CITY) 50,000 records stored in Detroit

PART (PART\_NUMBER, COLOR) 350,000 records stored in Chicago

SHIPMENT (SUPPLIER\_NUMBER, PART\_NUMBER) 2,500,000 records stored in Detroit

A query is made (in SQL) to list the supplier numbers for Cleveland suppliers of red parts:

SELECT SUPPLIER.SUPPLIER\_NUMBER

FROM SUPPLIER, SHIPMENT, PART

WHERE SUPPLIER.CITY = ‘Cleveland’

AND SHIPMENT.PART\_NUMBER = PART.PART\_NUMBER

AND SHIPMENT.SUPPLIER\_NUMBER = SUPPLIER.SUPPLIER\_NUMBER

AND PART.COLOR = ‘RED’;

Each record in each relation is 300 characters long. There are 30 red parts, a history of 100,000 shipments from Cleveland, and a negligible query computation time compared with communication time. Also, there is a very old communication system with a very slow data transmission time of 8,000 characters per second and two seconds access delay to send a message from one node to another. Consider operating time to execute the query as 4 seconds. Assume the time taken to transfer the query results as negligible.

Determine the time to process this remote query assuming the following strategy:

**Move SUPPLIER relation to Chicago; then move SHIPMENT relation to Chicago; process whole query at Chicago computer.**

Answer should be expressed in hours, with one decimal place.

Ans: 26.56 Hours.

2,500,000 records (SHIPMENT) + 50,000 records (SUPPLIER) = 2,550,000 records.

2,550,000 records \* 300 characters per record = 765,000,000 characters.

765,000,000 characters / 8,000 characters per second = 95,625seconds.

4 seconds access delay (2 sec for each query) and 8 seconds operating time (4 sec for each query).

Total Time = 95,625seconds + 12 seconds =95,637 seconds.