Compsci 557 Homework 6

Kayla Goetzke

Phil Sauvey

Chapter 10 # 1, 6, 8, 9

1A) There are 2 database requests required for an inventory update.

1B)

UPDATE PRODUCT

SET PROD\_QOH = PROD\_QOH +1

WHERE PROD\_CODE = "ABC";

UPDATE PART

SET PART\_QOH = PART\_QOH - 1

WHERE PART\_CODE = "A" OR

PART\_CODE = "B" OR

PART\_CODE = "C";

1C)

BEGIN TRANSACTION

UPDATE PRODUCT

SET PROD\_QOH = PROD\_QOH +1

WHERE PROD\_CODE = "ABC";

UPDATE PART

SET PART\_QOH = PART\_QOH - 1

WHERE PART\_CODE = "A" OR

PART\_CODE = "B" OR

PART\_CODE = "C";

COMMIT;

1D)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRL\_ID | TRX\_NUM | PREV\_PTR | NEXT\_PTR | OPERATION | TABLE | ROW\_ID | ATTRIBUTE | BEFORE VALUE | AFTER VALUE |
| 14 | 5 | NULL | 24 | START | \*\*START TRANSACTION |  |  |  |  |
| 24 | 5 | 14 | 27 | UPDATE | PRODUCT | ABD | PROD\_QOH | 77 | 78 |
| 27 | 5 | 24 | 28 | UPDATE | PART | A | PART\_QOH | 53 | 52 |
| 28 | 5 | 27 | 29 | UPDATE | PART | B | PART\_QOH | 42 | 41 |
| 29 | 5 | 28 | 33 | UPDATE | PART | C | PART\_QOH | 1 | 0 |
| 33 | 5 | 29 | NULL | COMMIT | \*\* END OF TRANSACTION |  |  |  |  |

1E)

1. Find the commit TRL\_ID 33
2. Using PREV\_PTR find the start of the transaction, TRL\_ID 14
3. Using NEXT\_PRT find each update operation and apply the changes to the disk using the AFTER VALUE, TRL\_IDs 24, 27, 28, and 29
4. Repeat for any other transaction.

6A)

BEGIN TRANSACTION

INSERT INTO INVOICE

VALUES(

10983,

10010,

'11-MAY-2016',

118.80,

"30",

"OPEN");

INSRT INTO LINE

VALUES(10983, 1, '11QER/31', 1, 110.00);

UPDATE PRODUCT

SET P\_QTYOH = P\_QTYOH - 1,

WHERE P\_CODE = "11QER/31";

UPDATE CUSTOMER

SET CUS\_BALANCE = CUS\_BALANCE + 118.80,

CUS\_DATELSTPUR = "11-MAY, 2016"

WHERE CUS\_CODE = 10010;

COMMIT;

6B)

BEGIN TRANSACTION

INSERT INTO PAYMENTS

VALUES(

3428,

"03-JUN-2016",

10010,

100.00,

"CASH",

"N/A");

UPDATE CUSTOMER

SET CUS\_BALANCE = CUS\_BALANCE - 100.00,

CUS\_DATELSTPAY = "03-JUN, 2016"

WHERE CUS\_CODE = 10010;

COMMIT;

8)

|  |  |
| --- | --- |
| Time | Step |
| 1 | Lock INVOICE |
| 2 | Insert row 10983 into INVOICE |
| 3 | Unlock INVOICE |
| 4 | Lock LINE |
| 5 | Insert into row 10983, 1 into LINE |
| 6 | Unlock LINE |
| 7 | Lock PRODUCT |
| 8 | Update PRODUCT 11QER/31, Subtract one from P\_QTYOH |
| 9 | Unlock PRODUCT |
| 10 | Lock CUSTOMER |
| 11 | Update CUSTOMER 10010, CUS\_BALANCE to current CUS\_BALANCE + 118.8 |
| 12 | Update CUSTOMER 10010, CUS\_DATELSTPUR to 11-May-2012 |
| 13 | Unlock CUSTOMER |

9)

|  |  |
| --- | --- |
| Time | Step |
| 1 | Lock INVOICE |
| 2 | Lock LINE |
| 3 | Lock PRODUCT |
| 4 | Lock CUSTOMER |
| 5 | Insert row 10983 into INVOICE |
| 6 | Insert into row 10983, 1 into LINE |
| 7 | Update PRODUCT 11QER/31, subtract one from P\_QTYOH |
| 8 | Update CUSTOMER 10010, update CUS\_BALANCE to current CUS\_BALANCE + 118.8 |
| 9 | Update CUSTOMER 10010, CUS\_DATELSTPUR to 11-May-2012 |
| 10 | Unlock INVOICE |
| 11 | Unlock LINE |
| 12 | Unlock PRODUCT |
| 13 | Unlock CUSTOMER |

Chapter 11 # 1, 2, 4, 5, 6

1) There are only two possible outcomes for the EMP\_SEX column, M or F. Therefore, EMP\_SEX has low sparsity.

2) Based on the code, we should create indexes in EMP\_AREACODE and a combination of EMP\_LNAME and EMP\_FNAME.

Index 1:

CREATE INDEX EMP\_NDX1 ON EMPLOYEE(EMP\_LNAME, EMP\_FNAME);

Index 2:

CREATE INDEX EMP\_NDX2 ON EMPLOYEE(EMP\_AREACODE);

4)The EMP\_DOB will have a high data sparsity because there are many different birthdates out there.

5)If you create an index in EMP\_DOB, it will not help this query because it uses the YEAR function.

6) It will probably use the Table Scan (Full) to read in all the rows of the EMPLOYEE table. The first operation in the table 11.3.