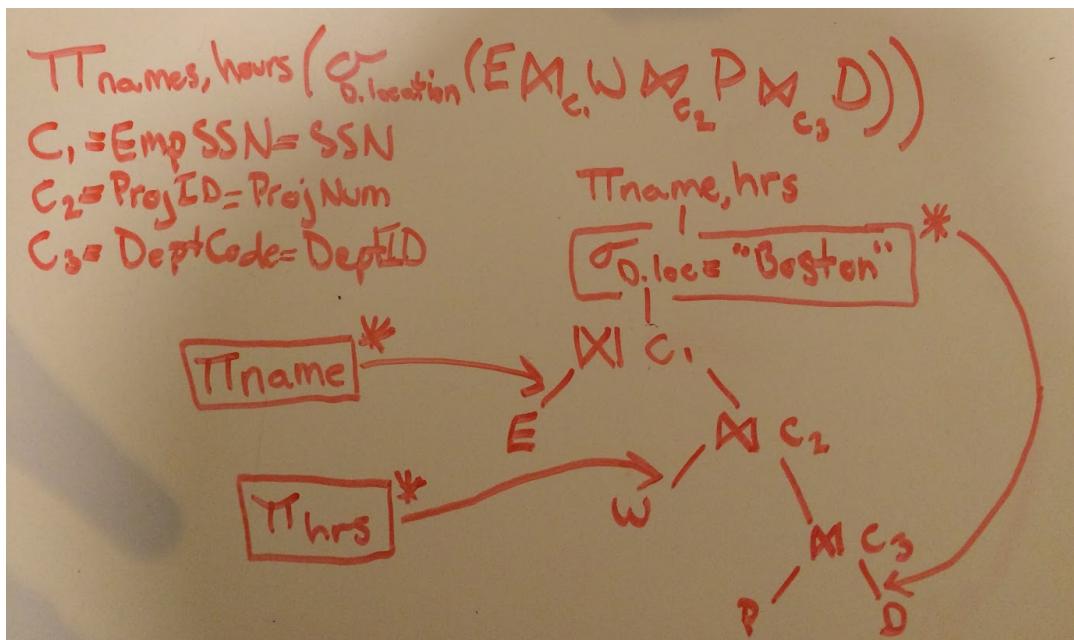


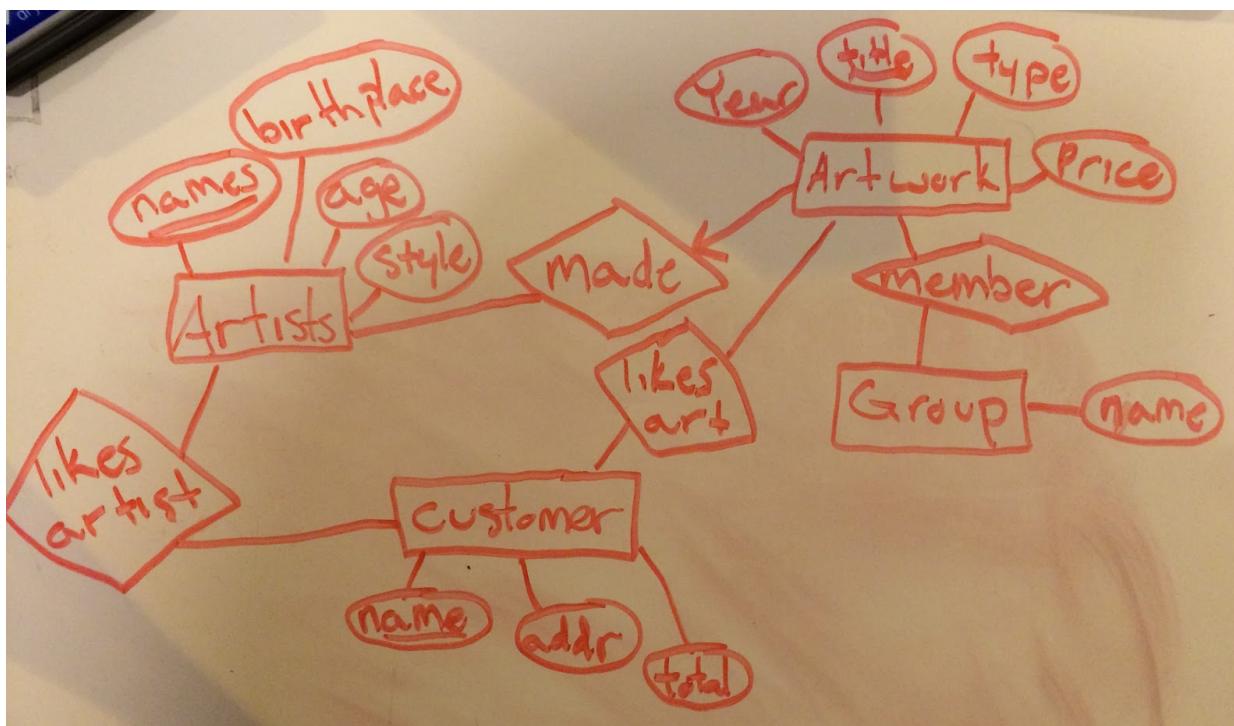
Tyler Waltze
 Professor Kolios
 CS 460
 April 30th, 2015
 Written Assignment #4

Problem 2:



(*) Optimizations: By moving the selection down to the bottom, the number of rows used in the following 3 joins is reduced. Doing the projections on E and W limits the amount of data (row width) used in subsequent calls.

Problem 3:



```
CREATE TABLE `artist_likes` (
  `artist` varchar(64) NOT NULL,
  `customer` varchar(64) NOT NULL,
  PRIMARY KEY (`artist`,`customer`),
  KEY `customer` (`customer`),
  CONSTRAINT `artist_likes_ibfk_2` FOREIGN KEY (`customer`) REFERENCES `customers`(`name`),
  CONSTRAINT `artist_likes_ibfk_1` FOREIGN KEY (`artist`) REFERENCES `artists`(`name`)
);
```

```
CREATE TABLE `artists` (
  `name` varchar(64) NOT NULL,
  `birthplace` varchar(64) DEFAULT NULL,
  `age` int(3) DEFAULT NULL,
  `style` varchar(64) DEFAULT NULL,
  PRIMARY KEY (`name`)
);
```

```
CREATE TABLE `artwork` (
```

```

`title` varchar(64) NOT NULL,
`year` int(4) DEFAULT NULL,
`price` int(11) DEFAULT NULL,
`type` varchar(64) DEFAULT NULL,
`artist` varchar(64) NOT NULL,
PRIMARY KEY (`title`)
);

CREATE TABLE `artwork_likes` (
`artwork` varchar(64) NOT NULL,
`customer` varchar(64) NOT NULL,
PRIMARY KEY (`artwork`,`customer`),
KEY `customer` (`customer`),
CONSTRAINT `artwork_likes_ibfk_2` FOREIGN KEY (`customer`) REFERENCES `customers` (`name`),
CONSTRAINT `artwork_likes_ibfk_1` FOREIGN KEY (`artwork`) REFERENCES `artwork`(`title`)
);

CREATE TABLE `customers` (
`name` varchar(64) NOT NULL,
`address` varchar(128) DEFAULT NULL,
`total_spent` int(11) DEFAULT NULL,
PRIMARY KEY (`name`)
);

CREATE TABLE `groups` (
`name` varchar(64) NOT NULL,
PRIMARY KEY (`name`)
);

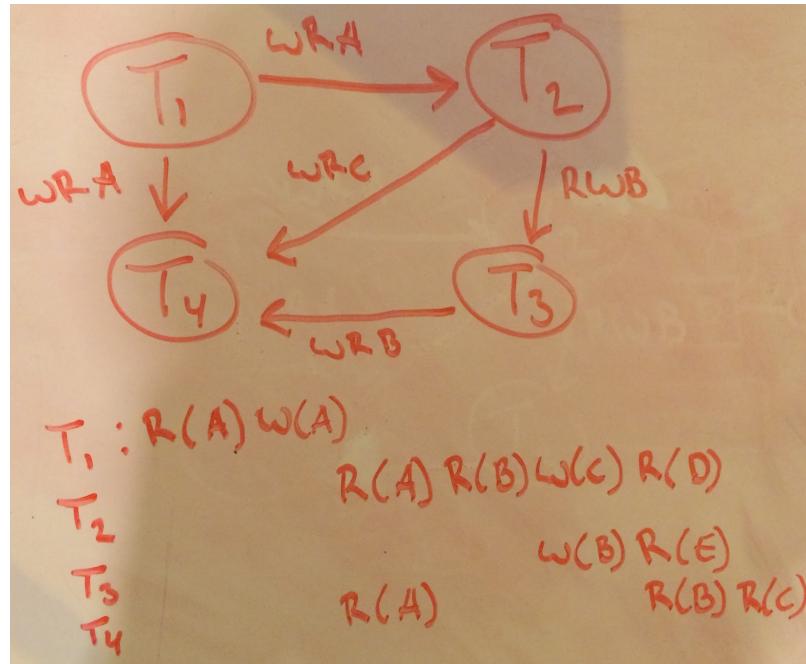
CREATE TABLE `members` (
`customer` varchar(64) NOT NULL,
`group` varchar(64) NOT NULL,
PRIMARY KEY (`customer`,`group`),
KEY `group` (`group`),
CONSTRAINT `members_ibfk_2` FOREIGN KEY (`group`) REFERENCES `groups`(`name`),
CONSTRAINT `members_ibfk_1` FOREIGN KEY (`customer`) REFERENCES `customers`(`name`)
);

```

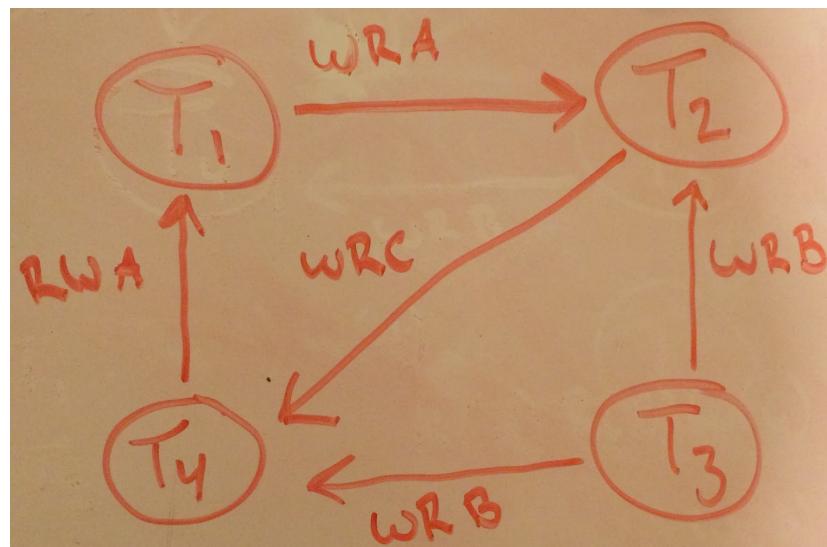
Problem 4:

1.

a. Conflict serializable



b. Not conflict serializable



2.

a. Completes

T_1	T_2	T_3
$LS(A)$	$LS(B)$	$LS(C)$
$R(A)$	$R(B)$	$R(C)$
$LX(B)$	$LX(C)$	$LX(D)$
		$W(D)$
	$UL(C)$	$UL(D)$
	$UL(B)$	$UL(C)$
$W(B)$		
$UL(A)$		
$UL(B)$		

b. Deadlocks

T_1	T_2	T_3
$LS(A)$	$LS(B)$	$LS(C)$
$R(A)$	$R(B)$	$R(C)$
$LS(B)$	$LS(C)$	$LS(A)$
$R(B)$	$R(C)$	$R(A)$
$LX(A)$	$LX(B)$	$LX(C)$

Problem 5:

1. It goes to the last recorded checkpoint and then creates a transaction table and dirty page table.
2. It will first go to LSN 20, then run each step in the log (Update P1 (T1), Update P2 (T1), etc).
3. It will write END T2 because it failed to do so prior to the crash, then it will undo the transaction done by T3.
4. END T2;
CLR; UNDO T3 LSN 70; UndoNxtLsn=50;
CLR; UNDO T3 LSN 50; UndoNxtLsn=null;
DONE;