Midterm Review Solutions 4/29/2019

SQL

CSE 344 Final March 18, 2019, 2:30-4:20

1 Relational Data Model

1. (35 points)

   A Web browser stores the local data in a relational database[1] with the following schema:

   ```
   History(url, ts)
   Cache(url, content, size)
   Bookmark(name, url)
   ```

   - Every time the user visits a page, the browsers inserts a record in `History`; `url` is the URL of the Webpage and has type `Text`; `ts` is the time stamp when the user accesses the page and has type `Int`.
   - Some Web pages are stored in the local cache of the browser; this is the table `Cache`; `content` is the HTML text of the page in the cache and has type `Text`, while `size` is its size in bytes and has type `Int`.
   - The user may create bookmarks and give them unique names; all bookmarks are stored in `Bookmark`.

   (a) (5 points) Sometimes users created duplicate bookmarks; they give two or more names to the same URL. Write a SQL query that returns all duplicate bookmarks. Your query should return a list of names and URLs, sorted by the URLs.

   Solution:

   ```
   -- drop/create tables for testing only
   drop table if exists history;
   drop table if exists bookmarks;
   drop table if exists cache;
   create table history(url text, ts int);
   create table cache(url text primary key, content text, size int);
   create table bookmark(name text primary key, url text);

   select distinct x.url, x.name
   from Bookmark x, Bookmark y
   where x.url = y.url and x.name != y.name
   order by x.url;
   ```

(b) (5 points) The browser wants to store a new page in the cache. There is no more space, and it decides to evict the oldest pages in order to make room. Write a SQL query that lists for each URL in the cache its size and the latest timestamp when that page was last accessed. Your query should return triples `url`, `size`, `ts`, ordered increasingly by `ts`.

Solution:

```
select y.url, max(x.ts) as tsmax, y.size
from History x, Cache y
where x.url = y.url
group by y.url, y.size
order by tsmax;
```

(c) (15 points) The new page to be added to the cache has 1000 bytes, and the browser decides to evict from the cache the oldest pages in order to make room from the new page. For example, if there are four pages in the cache last accessed at time stamps 1,2,3,4 respectively, and their sizes are 500, 300, 300, 600, then the browser wants to delete the oldest three pages, since $500 + 300 + 300 \geq 1000$. Write a SQL query to return the URLs of all pages in the cache that the browser needs to delete to make room for 1000 bytes; your query should return a list of URL's (no need to actually delete them from `Cache`).

Note: only attempt to answer this question if you have answered question b. If you answered it, then you may refer to the query in b (no need to write it again).

Solution:

```
with tmp as
     (select y.url, max(x.ts) as tsmax, y.size
      from History x, Cache y
      where x.url = y.url
      group by y.url, y.size)
select x.url
from tmp x, tmp y
where x.url = y.url and y.tsmax < x.tsmax
group by x.url
having sum(y.size) < 1000;
```

RA

CSE 344 Midterm Feb. 15, 2019, 1:30-2:20

2 Relational Algebra

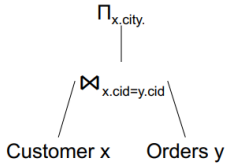
2. (25 points)

Consider the same relational schema as before:

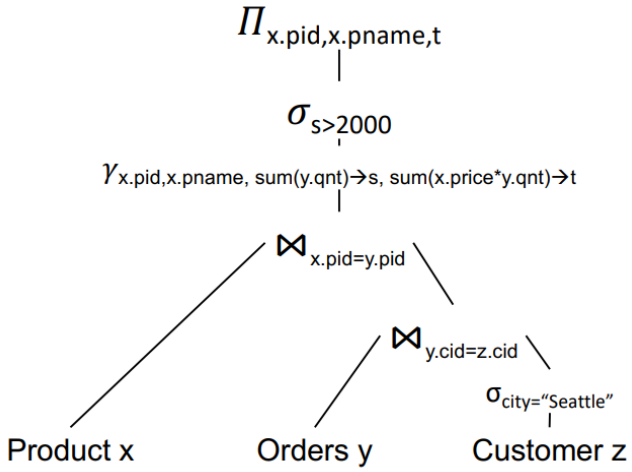
```
Product(pid,pname,price)
Orders(oid,pid,cid,qnt,date)
Customer(cid,cname,city)
```

(a) (5 points) Write a Relational Algebra expression in the form of a logical query plan (i.e., draw a tree) that is equivalent to the SQL query below. Your query plan does not have to be necessarily "optimal": however, points will be taken off for overly complex solutions.

Hint: to avoid renaming, use aliases in the query plan, like this




```
select x.pid, x.pname, sum(x.price*y.qnt) as revenue
from Product x, Orders y, Customer z
where x.pid = y.pid and y.cid = z.cid
  and z.city = 'Seattle'
group by x.pid, x.pname
having sum(y.qnt) > 2000;
```



(b) i. (2 points) Which of the following is the most accurate English interpretation of the SQL query below?

```
select distinct z.city
from Customer z
where not exists
       (select *
        from Orders y
        where y.cid = z.cid and y.qnt < 100);
```

Returns cities that ...

(A) had at least one order for < 100 units of a product.

(B) have a customer who placed at least one order for more < 100 units of a product.

(C) had only orders with ≥ 100 units.

(D) have a customer that placed only orders with ≥ 100 units.

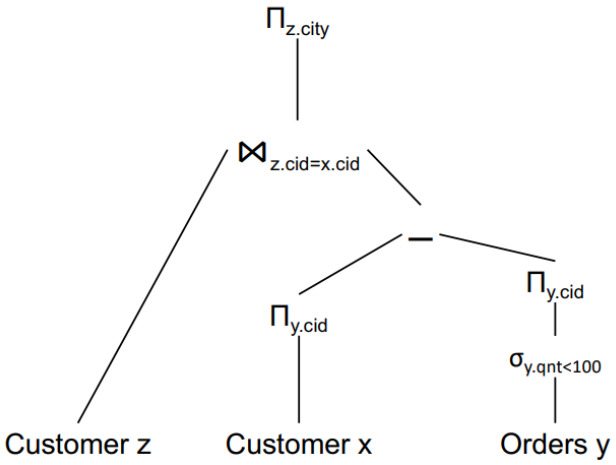
i. _____

A/B/C/D:

ii. (8 points) Write a Relational Algebra expression in the form of a logical query plan (i.e., draw a tree) that is equivalent to the SQL query above. Your query plan does not have to be necessarily "optimal": however, points will be taken off for overly complex solutions.

Answer to (i): D.

Answer to (ii):



Design Theory

CSE 344 Final Examination December 16, 2015

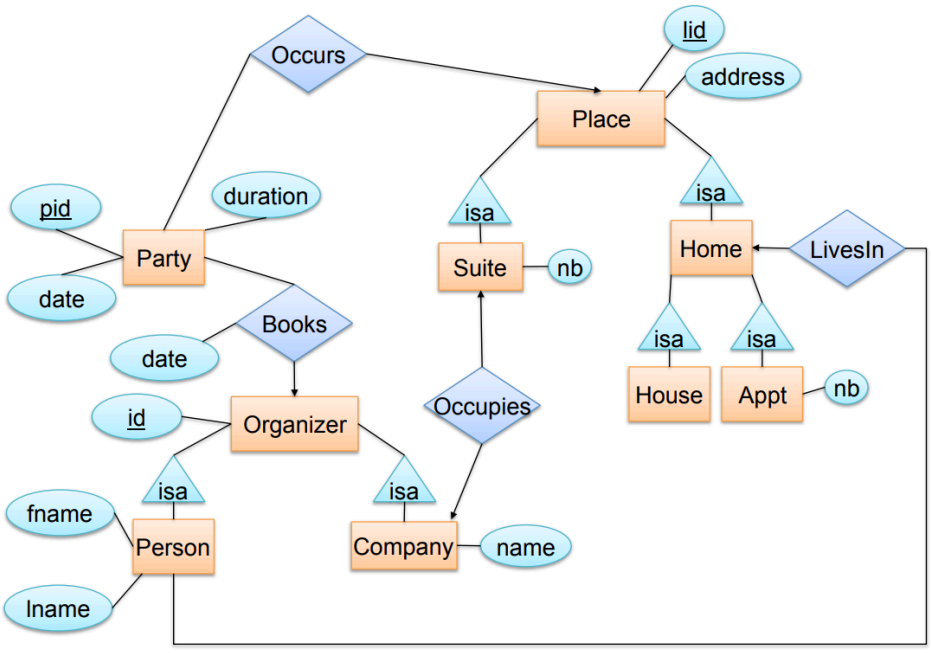
2 E/R Diagrams, Constraints, Conceptual Design

2. (30 points)

   (a) (10 points) Design an E/R diagram describing the following domain:

   - A **person** has attributes **id** (key), **fname**, and **lname**.
   - A **company** has attributes **id** (key) and **name**. Each company name is unique.
   - A **party** has attributes **pid**, **date**, and **duration**. The date attribute cannot be NULL. A party cannot last longer than 4 hours.
   - A party is **booked** by either a person or a company. We need to capture information about both **when the party takes place** and **when the party was booked**. The booking date must be earlier than the party date. A party is booked by zero or one organizers. A person or company can book many parties.
   - A party **occurs** in either a **house**, an **apartment**, or a **suite**. All these locations are uniquely identified with a **location id (lid)**. A house has an attribute **street address**. The address is an atomic string attribute. An apartment additionally has an **apartment number**. A suite has a street address and a **suite number**. A party occurs in zero or one place. A place can host multiple parties.
   - A company **occupies** zero or one suite and a suite hosts zero or one company (i.e., the application does not store historical data only current data).
   - A person **lives** in either a house or an apartment. A person lives in zero or one place. A given place can host many people.

Solution:



(c) (10 points) Consider the following relational schema and set of functional dependencies.

R(A,B,C,D,E,F,G) with functional dependencies:
E $\rightarrow$ C
G $\rightarrow$ AD
B $\rightarrow$ E
C $\rightarrow$ BF

- Give one example of non-trivial functional dependency implied by the ones above:

  Answer (Example FD):

  Solution: Many solutions are possible. One example is E $\rightarrow$ BF.

- Compute E^+, the closure of E.

  Answer (E^+):

  Solution: $\{E\}^+ = \{B, C, E, F\}$

- Why do we bother to decompose relations into BCNF? What does this normal form ensure?

  Answer (Explanation):

  Solution: BCNF ensures that, for each table, only "good" FDs exist: Whenever X $\rightarrow$ B is a non-trivial dependency, then X is a super key. Because only these good FDs exist, the relation will not suffer from data redundancy, update anomalies, nor deletion anomalies (see lecture 18 for details).

- Decompose R into BCNF. Show your work for partial credit. Your answer should consist of a list of table names and attributes and an indication of the keys in each table (underlined attributes).

  Answer (Decompose R into BCNF):

  Solution: $\{E\}^+ \rightarrow \{B, C, E, F\}$ violates BCNF because it is neither trivial nor contains all attributes (i.e., E is not a superkey). So we need to decompose. We decompose into R1(B, C, E, F) and R2(A, D, E, G).

  For R1(B, C, E, F), there are no more FD that violate BCNF, so we continue with R2(A, D, E, G). We find $\{G\}^+ = \{A, D\}$, so we decompose into R21(G, A, D) and R22(G, E). R22 has now two attributes, so it is necessarily in BCNF. For R21, there are no more FDs that violate BCNF.

Data Management

Name	2015	2016	2017	2018
Angola	100	110		115
Luxembourg	50		55	65

Suppose the above relation is stored as the table “Country” in a SQL database. The values are GDP amounts, corresponding to its row’s country name and its column’s year. **Write a query to unpivot GDP by year for each country.** Your query should result in the following relation, though the tuples may be returned in a different order:

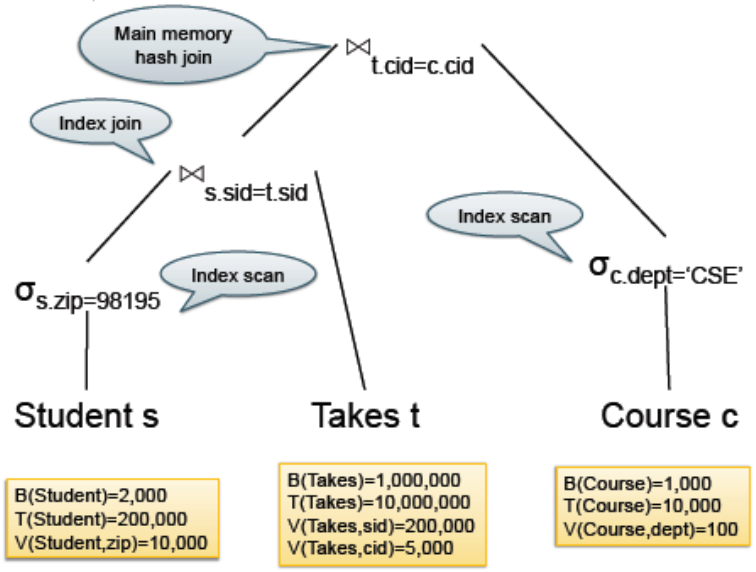
Name	Year	GDP
Angola	2015	100
Luxembourg	2015	50
Angola	2016	110
Angola	2018	115
Luxembourg	2017	55
Luxembourg	2018	65

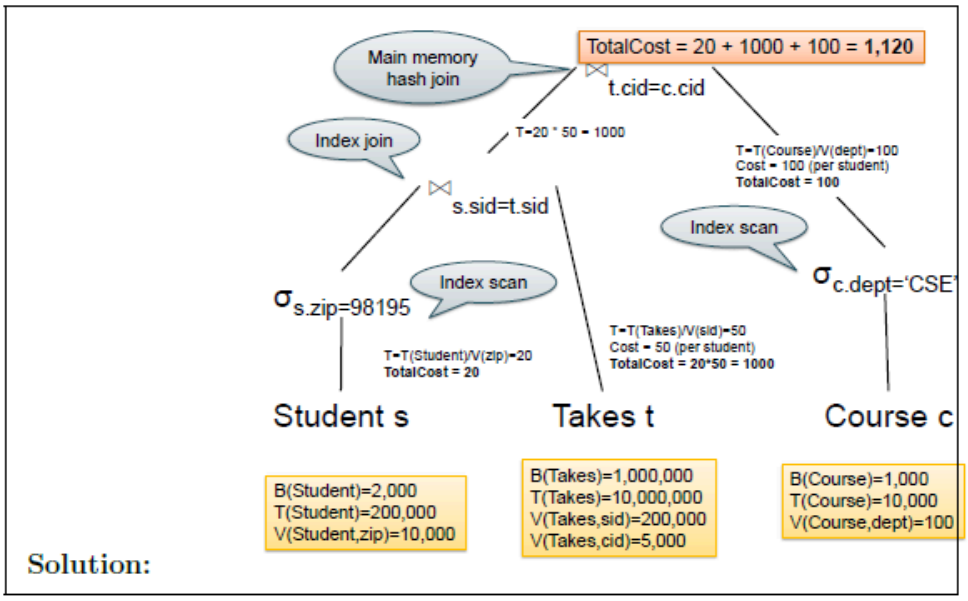
```
-- Table creation code for testing in SQLite:
CREATE TABLE Country(Name string, "2015" int, "2016" int, "2017" int, "2018" int);
INSERT INTO Country VALUES ("Angola", 100, 110, null, 115);
INSERT INTO Country VALUES ("Luxembourg", 50, null, 55, 65);

-- Solution:
SELECT C.Name, 2015 AS Year, C."2015" AS GDP FROM Country C WHERE C."2015" IS NOT NULL
UNION
SELECT C.Name, 2016 AS Year, C."2016" AS GDP FROM Country C WHERE C."2016" IS NOT NULL
UNION
SELECT C.Name, 2017 AS Year, C."2017" AS GDP FROM Country C WHERE C."2017" IS NOT NULL
UNION
SELECT C.Name, 2018 AS Year, C."2018" AS GDP FROM Country C WHERE C."2018" IS NOT NULL;
```

Physical Design

17au final, 3c. For each of the physical plans below, estimate its cost. Assume the size of the main memory is M = 2000 pages and that all indexes are unclustered, and are stored in main memory (hence accessing them requires zero disk I/O's).

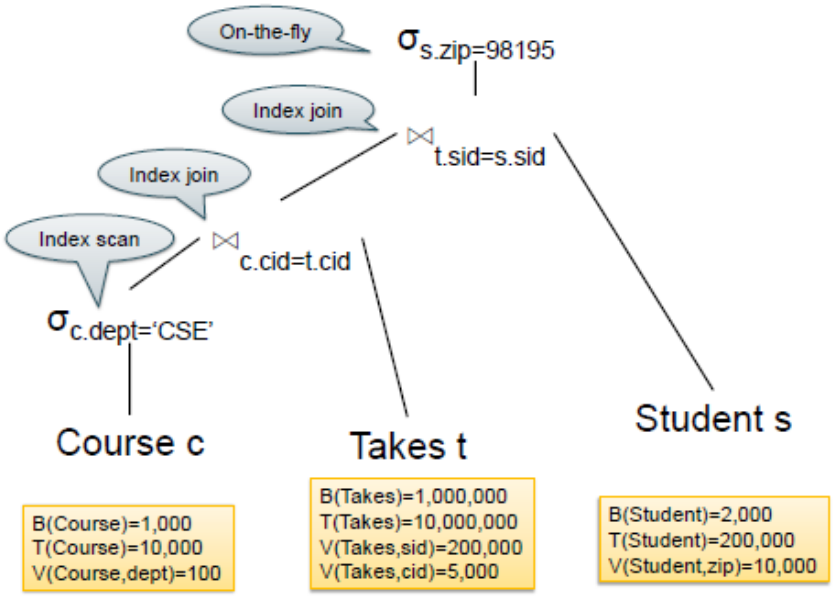
i. (5 points)



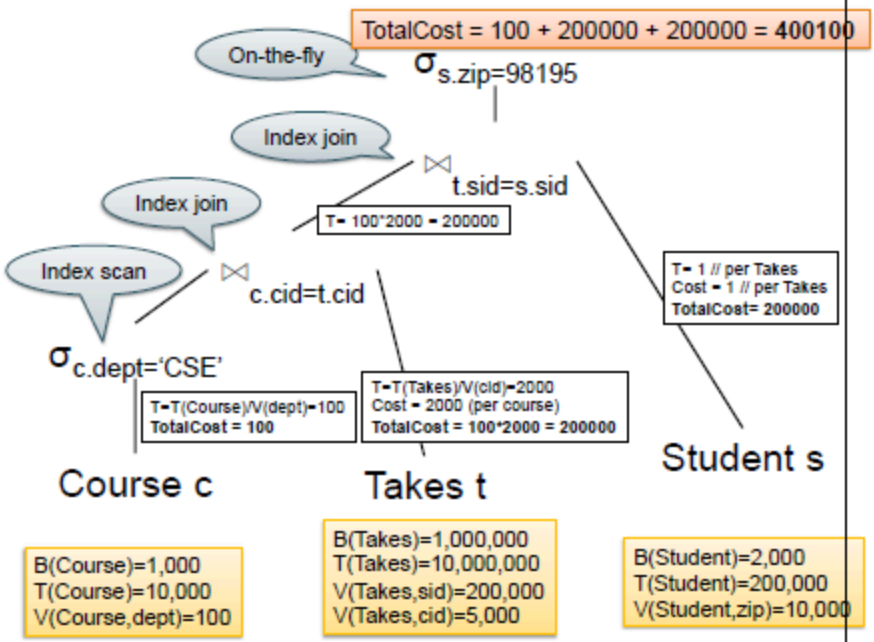


Solution:

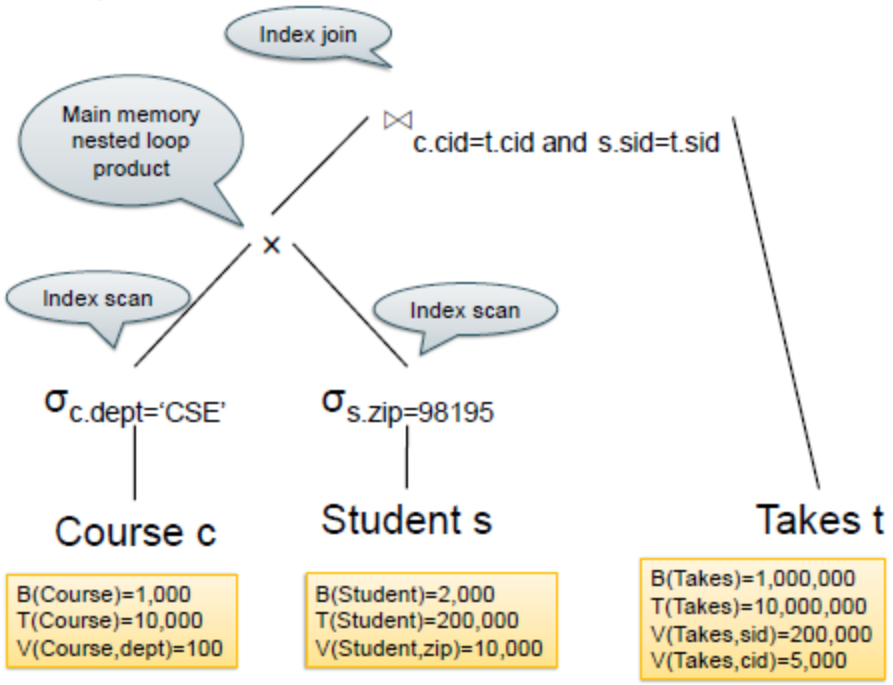
ii. (5 points)



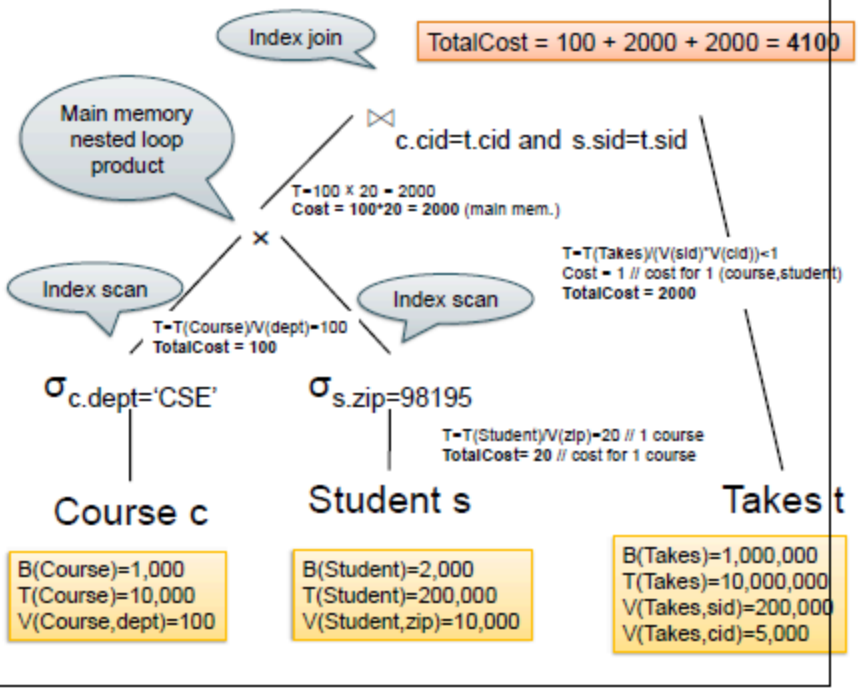
Solution:



iii. (5 points)



Solution:



Transactions

CSE 344 Final Examination March 15, 2016

4 Transactions

4. (54 points)

(a) Consider a concurrency control manager that uses strict two phase locking that schedules three transactions:

- $T1 : R_1(A), R_1(B), W_1(A), W_1(B), Co_1$
- $T2 : R_2(B), W_2(B), R_2(C), W_2(C), Co_2$
- $T3 : R_3(C), W_3(C), R_3(A), W_3(A), Co_3$

Each transaction begins with its first read operation, and commits with the Co statement. Asnwer the following questions for each of the schedules below:

- Is the schedule conflict-serializable? If yes, indicate a serialization order.
- Is this schedule possible under a strict 2PL protocol?
- If strict 2PL does not allow this schedule beause it denies a read or a write request, is the system in a deadlock at the time when the request is denied?

i. Schedule 1:

$$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), Co_3, R_2(C), W_2(C), Co_2, R_1(A), R_1(B), W_1(A), W_1(B), Co_1$$

Yes conflict serializable: a valid serialization order is 3, 2, 1.
Yes, possible under Strict 2PL.
No, no deadlock possible.

ii. Schedule 2:

$$R_2(B), W_2(B), R_3(C), W_3(C), R_1(A), R_1(B), W_1(A), W_1(B), Co_1, R_2(C), W_2(C), Co_2, R_3(A), W_3(A), Co_3$$

No, not conflict serializable.
No, not possible under Strict 2PL.
Yes, deadlock is possible: T_1 locks A, T_2 locks B, T_3 locks C

iii. Schedule 3:

$$R_1(A), R_1(B), R_2(B), W_2(B), R_2(C), W_2(C), Co_2, R_3(C), W_3(C), R_3(A), W_3(A), Co_3, W_1(A), W_1(B), Co_1$$

No, not conflict serializable.
No, not possible under Strict 2PL.
No, no deadlock possible.

iv. Schedule 4:

$$R_1(A), R_1(B), W_1(A), R_3(C), W_3(C), R_3(A), W_3(A), Co_3, W_1(B), R_2(B), W_2(B), Co_1, R_2(C), W_2(C), Co_2$$

Yes, conflict serializable: a valid serialization order is 1, 3, 2.
No, not possible under Strict 2PL. (All Strict 2PL schedules are conflict serializable, but some conflict serializable schedules are not Strict 2PL.) Look at item B; transaction 2 wants to lock B but transaction 1 must hold its lock on B until it commits.
No, no deadlock possible.

(b) (10 points) Consider the following three transactions:

- $T1 : R_1(A), W_1(B), Co_1$
- $T2 : R_2(B), W_2(C), Co_2$
- $T3 : R_3(C), W_3(D), Co_3$

Given an example of a conflict-serializable schedule that has the following properties: transaction $T1$ commits before transaction $T3$ starts, and the equivalent serial order is $T3, T2, T1$.

Solution: $R_1(A), R_2(B), W_1(B), Co_1, R_3(C), W_2(C), Co_2, W_3(D), Co_3$
Variations include: swap the first two reads (of A and B), and the last two writes (of C and D, together with the commit order)