2017 Exam - Information Management
CS3041

INSTRUCTOR: Vincent Wade & Seamus Lawless
vincent.wade@scss.tcd.ie
seamus.lawless@scss.tcd.ie

Two Hour Exam

Question 1 Mandatory

Answer 2 out of 3 Questions out of Q2-4

25 marks per question

40 mins per question

1% per mark

Question 1

Question 2

a) What are the four essential properties of a transaction? Explain each of them.

Atomicity: Each transaction should be an atomic unit of processing. It should be performed in its entirety or not at all.

Consistency Preservation: A transaction should preserve the consistency of a DB. Each transaction should take the DB from one consistent state to another.

Isolation: Each transaction should appear as if acting in isolation. The execution of one transaction should not be interfered with by another transaction.

Durability: The changes applied by a transaction must persist in the database. The changes must not be lost due to a failure.

b) Concurrently executing transactions can cause a number of problems if they are not correctly scheduled. Discuss, with examples, the concurrency protocol problems of "Lost Update", "Temporary Update (Dirty Read)" and "Incorrect Summary".

Lost Update: This occurs when two or more transactions access the same data item and are executed concurrently. They would be interleaved in such a way that results in an incorrect value being written to the database.

Temporary Update: This occurs when two or more transactions access the same data item concurrently and one transaction fails and must rollback.

Incorrect Summary: This occurs when two or more transactions access the same data item concurrently and one transaction is calculating an aggregate summary on attributes while another transaction is updating those attributes.

c) What operations in a schedule are deemed to be in conflict? Illustrate with examples.

Two operations in a schedule are deemed to be in conflict if:

1. They belong to different transactions
2. They access the same item(X)
3. At least one of the operations is write(X)

e.g

$r_1(X)$	|
$w_1(X)$	| $r_2(X)$

d) Serializable schedules are those which are said to be equivalent to a serial schedule. How is equivalence measured?

Equivalence in schedules is measured using the following two:

Result Equivalence: When the operation of A and B produce the same result regardless of the order in which they are executed.

Conflict Equivalence: If the order of any two conflicting operations in a serial schedule and a non-serial schedule is the same then they can be considered conflict equivalent.

e) Explain how concurrency control algorithms ensure that concurrently executing transactions do not interfere with each others execution. Make reference to both binary and read-write locking. Two-phase locking is an additional locking protocol. Discuss two-phase locking and the benefits it offers.

Concurrency control algorithms are implemented by the DB to ensure concurrency exists and all schedules are correct. They make use of the following to ensure concurrency.

Locking Protocols: Data items within transactions are locked when they are accessed/read. There are two main types of locks; Binary locks and Read-Write locks. Binary locks maintain two states; locked or unlocked. When a transaction attempts to access a resource it must first attempt to lock it and when it is finished it must also unlock it. This can delay transactions seriously especially if a lock must be acquired in order to just read a resource. Whereas read-write locks are more flexible as they allow for data items to be concurrently read by multiple transactions however only one transaction can obtain a write-lock on a resource at any given time.

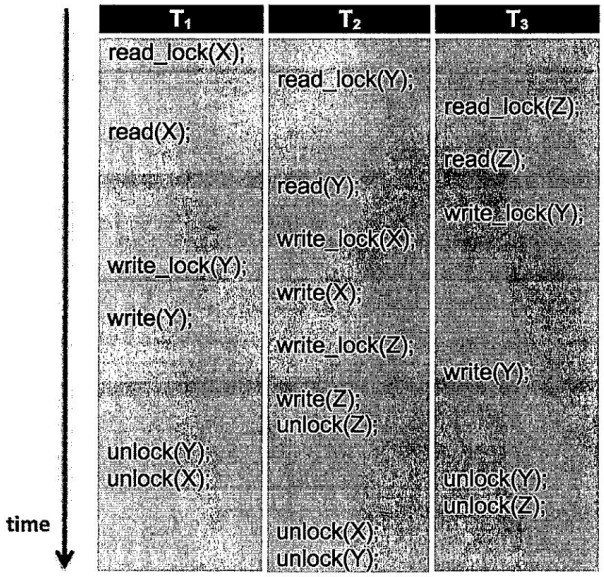
Timestamps: Timestamps are used to identify transactions based upon start times and are also used to verify who had access to a given lock first in order to determine priority.

Two-Phase Locking: This concerns the positioning of locking and unlocking in transactions. Within this method locking and unlocking can be performed in two phases/cycles known as the expanding and shrinking phase. In the expanding phase new locks can be acquired but no locks can be released. In the shrinking phase existing locks can be released but no new locks can be acquired. If transactions follow this locking protocol the schedule is said to be serializable however it comes at an extra cost. Transactions might not be able to release a resource when they are finished with it and must wait until the next shrinking phase. This reduces concurrency significantly and increases query execution latencies.

f) Outline the operation of the "Wound-Wait" algorithm. Indicate how "Wound-Wait" would execute on the following schedule. You may assume that T_1 is older than T_2 etc. State any assumptions you make in determining transaction operation ordering.

Wound-Wait: Is a concurrency control algorithm. When T_x tries to lock an item X but can't because it is locked by another transaction T_y it performs the following:

- If T_x is older than T_y, abort T_y and restart it later with the same timestamp
- If T_x is younger than T_y, allow T_x to wait for T_y to finish and release



Conflicts

- $w_3(Y)$ and $r_2(Y)$ - T_3 is younger than T_2 and as a result is allowed to wait.
- $w_2(X)$ and $r_1(X)$ - T_2 is younger than T_1 and as a result is allowed to wait.
- $w_1(Y)$ and $r_2(Y)$ - T_1 is older than T_2 and as a result it wounds T_2 and causes it to restart later with the same timestamp.

Question 3

a) Distinguish between Security and Integrity in a relational database.

Data security is the protection of data against unauthorized access or corruption and is necessary to ensure data integrity. Data integrity is a desired result of data security, but the term data integrity refers only to the validity and accuracy of data rather than the act of protecting data.

Integrity is concerned with accidental corruption. Security is concerned with deliberate corruption. Integrity is imposed with integrity constraints. Security is imposed with security policies.

b) What is a database constraint? Distinguish between explicit constraints and semantic constraints. Define three basic types of integrity constraint that all relational databases must support.

A database constraint is a rule/policy that is imposed against the relational schema and data that ensures the integrity of the database and the validity of the data.

Explicit Constraints are constraints that are expressed directly within the relational schema.

Semantic Constraints are constraints that cannot be expressed in the relational schema. They are usually enforced by application programs.

The three types of integrity constraints that a DBMS must support are:

1. **Key:** Primary key, candidate keys. No duplicate values of primary key.
2. **Entity Integrity:** No NULL values in primary key.
3. **Referential Integrity:** Foreign keys must refer to valid primary key of another table.

c) What operations can violate referential integrity? What clauses and constraints can be used to avoid violating referential integrity? Use a CREATE TABLE statement for one of the tables to help illustrate your answer.

The SQL commands `DELETE, UPDATE` can all violate referential integrity. If we wanted to `DELETE` the primary key tuple from one table, this might have been referenced in multiple other tables and just deleting it in the primary table would violate referential integrity. As a solution to this the DBMS could propagate/cascade this deletion to all affected foreign key references. The same goes for UPDATE commands, these changes can be propagated, set to null or set to default.

```
[CONSTRAINT [symbol]] FOREIGN KEY
    [index_name] (col_name, ...)
    REFERENCES tbl_name (col_name,...)
    [ON DELETE reference_option]
    [ON UPDATE reference_option]

reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

d) Access Control is often used to secure a relational database. Discuss the various means by which a DBMS can manage access control. Make specific reference to privileges, propagation and the risks involved. Compare and contrast Discretionary Access Control and Mandatory Access Control.

Privileges are certain granted or restricted characteristics of a user functional ability. They are used in database systems to both allow and restrict DB users from performing certain operations/queries. Privileges can be defined on two levels:

1. Account Level: DBA can specify the privileges that each account holds independently of the relations of the database.

```
CREATE USER 'admin'@'%' IDENTIFIED BY 'password'

GRANT ALL PRIVILIGES ON *.* TO 'admin'@'%'
```

2. Relational Level: DBA can specify the privileges that each account holds independently of the relations of the database.

```
CREATE USER 'result_clerk'@'%' IDENTIFIED BY 'password'

GRANT SELECT, INSERT ON Result TO 'result_clerk'@'%'
```

Granting priviliges to a user is performed by:

```
GRANT ALL PRIVILIGES ON *.* TO 'admin'@'%'
```

Revoking priviliges from a user is performed by:

```
REVOKE ALL PRIVILIGES ON *.* FROM 'admin'@'%'
```

Propogation allows for a given user X to grant priviliges to another user Y and allow Y to pass such privileges on to another user Z.

```
GRANT ALL PRIVILIGES ON Result TO 'result_clerk'@'%' WITH GRANT OPTION
```

This means that result_clerk now also has the ability to grant these priviliges to any user he/she wishes. This can be quite dangerous as even after you have revoked result_clerk privileges, any privileges that he has granted cannot be revoked and still exist.

Discretionary Access Control is handled from a user level and is a security enforcement for events that you can expect to occur. Such an example is by granting priviliges and revoking priviliges. It is a form of access control fully at your discretion.

Mandatory Access Control defines security levels for both data and users. Certain users will have access to certain relations/features based on their security level/clearance. This is not commonly available in commercial DBMS' however is desirable for government and military intelligence.

e)

i) What is the SQL command to create a view?

```
CREATE VIEW managers_overview AS

SELECT m.manager_name, t.team_name

FROM Team t, Manager m

WHERE t.team_id=m.team_id
```

ii) How could we allow a user called "Elliot Anderson" to read and update the information contained within this view? How would you modify it to allow Elliot to pass these permissions to others?

1) `GRANT SELECT, UPDATE ON managers_overview TO 'elliot_anderson'@'%'`

2) `GRANT SELECT, UPDATE ON managers_overview TO 'elliot_anderson'@'%' WITH GRANT OPTION`

iii) How would we remove the update permission from Elliot?

```
REVOKE UPDATE ON managers_overview FROM 'elliot_anderson'@'%'
```