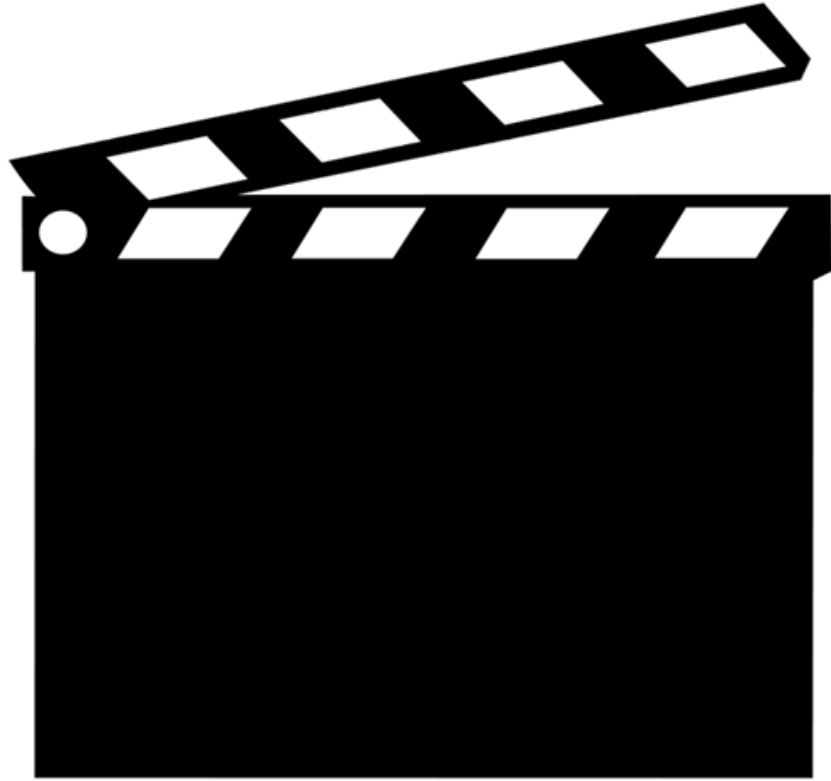


**C018SA-W1-S4**



# SEMAINE 1 : Transactions et concurrence

1. Introduction : les transactions
2. Les problèmes
3. Sérialisabilité
- 4. Estampillage**
5. Verrouillage à 2 phases
6. Degrés d'isolation dans les SGBD
7. Verrouillage hiérarchique

# Estampillage : principe

On associe à chaque transaction  $T_i$  un numéro distinct, appelé **estampille**, qu'on note  $E(T_i)$

Ce numéro est donné de manière croissante aux transactions selon leur date de début:

$$E(T_i) < E(T_j) \Leftrightarrow T_i \text{ a débuté avant } T_j$$

# Estampillage : principe

Chaque nuplet  $x_i$  va mémoriser :

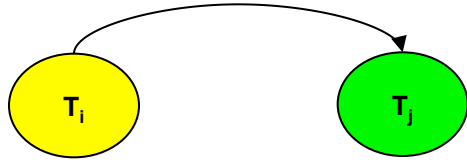
- L'estampille de la lectrice la plus récente, notée  $E_R(x_i)$
- L'estampille de l'écrivaine la plus récente, notée  $E_W(x_i)$

**Règle (estampillage partiel) :** Seule une transaction  $T_j$  t.q.

- $E(T_j) > E_W(x_i)$  pourra accéder en lecture ( $\rightarrow$  MAJ de  $E_R(X_i)$ )
- $E(T_j) > E_W(x_i)$  et  $E(T_j) > E_R(x_i)$  en écriture ( $\rightarrow$  MAJ de  $E_W(X_i)$ )

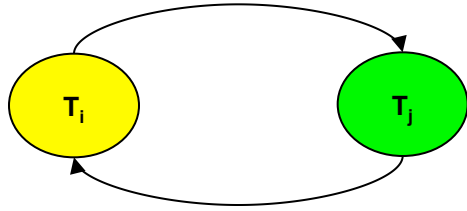
**Théorème :** Tout ordonnancement respectant la règle d'estampillage partiel est sérialisable.

# Preuve de la propriété de sérialisabilité



$T_i$  a accédé à  $x_k$  puis  $T_j$  a accédé à  $x_k$   
 $\rightarrow j > i$

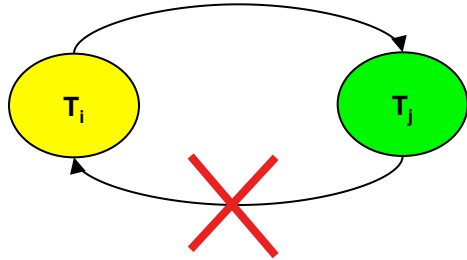
# Preuve de la propriété de sérialisabilité



$T_i$  a accédé à  $x_k$  puis  $T_j$  a accédé à  $x_k$   
 $\rightarrow j > i$

$T_j$  a accédé à  $x_k$  puis  $T_i$  a accédé à  $x_k$   
 $\rightarrow i < j$

# Preuve de la propriété de sérialisabilité



$T_i$  a accédé à  $x_k$  puis  $T_j$  a accédé à  $x_k$   
 $\rightarrow j > i$

$T_j$  a accédé à  $x_k$  puis  $T_i$  a accédé à  $x_k$   
 $\rightarrow i < j$

**IMPOSSIBLE PAR CONSTRUCTION**

Le graphe de précédence est un DAG par construction !  
D'après la définition 3 cela équivaut au fait que  
l'ordonnancement soit sérialisable.

# Une transaction viole la règle ?

Si une transaction  $T_j$  telle que  $E(T_j) < E_w(x_i)$  tente d'écrire une nouvelle valeur dans  $x_i \rightarrow T_j$  est annulée !

- On doit défaire toutes les opérations d'écriture de  $T_j$
- Toute transaction ayant lu  $x_i$  est annulée
- $T_j$  (et les autres) peut être relancée immédiatement

$\rightarrow$  on ne souhaite pas perdre les effets de la transaction



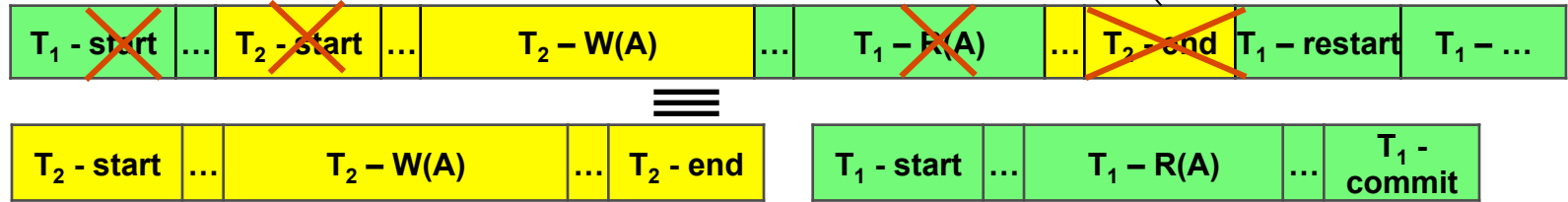
# Bilan de l'estampillage

- Algorithme simple : méthode préventive
- Méthode « équitable »

→ on prend les transactions dans l'ordre où elles arrivent, en cas de problème, on annule

**Vraiment équitable ?**

# Problèmes



- Abandon en cascade
- Abandon inutile (e.g. dans le cas de la lecture)
- Risque de privation pour les transactions longues