

Concetto	Definizione	Condizioni	Test/Verifica
Conflict-Equivalente	Due schedule hanno lo stesso insieme di transazioni e lo stesso ordinamento di tutte le coppie di operazioni conflittuali	Per ogni coppia di operazioni conflittuali (O_i, O_j), se O_i precede O_j nello schedule S_1 , allora O_i precede O_j anche nello schedule S_2	Confronto diretto dell'ordinamento delle operazioni conflittuali
Conflict-Serializzabile (CSR)	Uno schedule è conflict-equivalente a qualche schedule seriale	Esiste almeno uno schedule seriale che è conflict-equivalente allo schedule dato	Costruzione del grafo dei conflitti: $CSR \Leftrightarrow$ grafo aciclico
View-Equivalente	Due schedule soddisfano le stesse relazioni di lettura e scrittura finale	1. Read-from : se T_i legge X da T_j in S_1 , lo stesso vale in S_2 2. Initial read : se T_i legge il valore iniziale di X in S_1 , lo stesso vale in S_2 3. Final write : se T_i scrive l'ultimo valore di X in S_1 , lo stesso vale in S_2	Verifica delle tre condizioni di equivalenza
View-Serializzabile (VSR)	Uno schedule è view-equivalente a qualche schedule seriale	Esiste almeno uno schedule seriale che è view-equivalente allo schedule dato	Test più complesso: ricerca esaustiva o algoritmi specifici

Il punto cruciale è che la view-serializzabilità **non fissa** un ordinamento specifico delle transazioni, ma garantisce solo che **esista almeno un ordinamento seriale** che produca gli stessi effetti.

View-serializzabile significa: "Anche se avete giocato tutti mescolati, alla fine è come se aveste giocato uno alla volta".

Il punto importante: Non importa se nel gioco "uno alla volta" l'ordine è Marco-Sofia-Luca oppure Sofia-Luca-Marco. L'importante è che almeno uno di questi ordini dia lo stesso risultato del gioco mescolato.

Relazioni Fondamentali

- **$CSR \subseteq VSR$:** Ogni schedule conflict-serializzabile è anche view-serializzabile
- **$\text{Conflict-equivalenza} \subseteq \text{View-equivalenza}$:** Ogni coppia di schedule conflict-equivalenti

è anche view-equivalente

- Esistono schedule VSR ma non CSR (il famoso controesempio: $r_1(X)w_2(X)w_1(X)$)

VSR e Ordinamento delle Transazioni

La view-serializzabilità NON implica un ordinamento totale fisso delle transazioni.

Spiegazione:

- Uno schedule è VSR se è view-equivalente a *qualche* schedule seriale
- Possono esistere **multipli schedule seriali** view-equivalenti allo stesso schedule
- L'importante è che *esista almeno uno* schedule seriale equivalente, non che ce ne sia uno unico

Esempio:

Schedule: $r_1(X)r_2(Y)w_1(X)w_2(Y)$

Questo schedule è view-equivalente a:

- $T_1; T_2$ (prima T_1 poi T_2)
- $T_2; T_1$ (prima T_2 poi T_1)

Entrambi gli ordinamenti seriali sono validi perché le transazioni non hanno dipendenze di lettura tra loro.

Implicazione Pratica:

La VSR garantisce **correttezza semantica** (stesso risultato di un'esecuzione seriale) ma non determina un ordinamento unico. Il DBMS può scegliere qualsiasi ordinamento seriale equivalente per l'esecuzione.