Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

# Databases

**Malte Luttermann – Institute for Humanities-Centered Artificial Intelligence (CHAI)**

January 15, 2026

Version from January 13, 2026

# Topics for Today

- ACID properties
- Serializability
- Anomalies
- Locks

# Acknowledgement

- Some of the upcoming content is taken from the following lecture (and corresponding exercises) and has been translated and partially modified:
  - PD Dr. habil. Özgür L. Özçep: »Datenbanken«

# ACID Properties

What are the ACID properties?

# ACID Properties

What are the ACID properties?

- **Atomicity**: Either all or none of the value changes in a transaction are transferred to the database state.

# ACID Properties

What are the ACID properties?

- **Atomicity**: Either all or none of the value changes in a transaction are transferred to the database state.
- **Consistency**: A transaction transfers a consistent state (FDs, integrity conditions) to another consistent state.

# ACID Properties

What are the ACID properties?

- **Atomicity**: Either all or none of the value changes in a transaction are transferred to the database state.
- **Consistency**: A transaction transfers a consistent state (FDs, integrity conditions) to another consistent state.
- **Isolation**: A transaction does not take into account the effects of other parallel transactions in its calculations.

# ACID Properties

What are the ACID properties?

- **Atomicity**: Either all or none of the value changes in a transaction are transferred to the database state.
- **Consistency**: A transaction transfers a consistent state (FDs, integrity conditions) to another consistent state.
- **Isolation**: A transaction does not take into account the effects of other parallel transactions in its calculations.
- **Durability**: The effects of a successful transaction are made persistent.

# Comprehension Questions

Are the following statements correct?

- »The plan $\langle r_1(A), r_1(A), r_2(A), r_1(A) \rangle$ is serializable.«
- »Operations within a transaction never conflict with each other.«
- »The absence of cycles in the serialization graph is a sufficient condition for the serializability of a sequential plan.«

# Comprehension Questions

Are the following statements correct?

- »The plan $\langle r_1(A), r_1(A), r_2(A), r_1(A) \rangle$ is serializable.«
  - Correct. There are no write operations, so no conflicts.
- »Operations within a transaction never conflict with each other.«

- »The absence of cycles in the serialization graph is a sufficient condition for the serializability of a sequential plan.«

# Comprehension Questions

Are the following statements correct?

- »The plan $\langle r_1(A), r_1(A), r_2(A), r_1(A) \rangle$ is serializable.«
  - Correct. There are no write operations, so no conflicts.
- »Operations within a transaction never conflict with each other.«
  - Correct. Within a transaction, operations are executed sequentially.
- »The absence of cycles in the serialization graph is a sufficient condition for the serializability of a sequential plan.«

# Comprehension Questions

Are the following statements correct?

- »The plan $\langle r_1(A), r_1(A), r_2(A), r_1(A) \rangle$ is serializable.«
  - Correct. There are no write operations, so no conflicts.
- »Operations within a transaction never conflict with each other.«
  - Correct. Within a transaction, operations are executed sequentially.
- »The absence of cycles in the serialization graph is a sufficient condition for the serializability of a sequential plan.«
  - Correct. This is the definition of serializability.

# Serializability

Consider the following sequential plan $S$ for the transactions $T_1$, $T_2$, and $T_3$:

$$S = \langle w_3(A), r_1(A), w_1(A), r_3(A), w_1(A), r_2(A), w_2(A) \rangle.$$

Draw the serialization graph (conflict graph) for the sequential plan $S$.

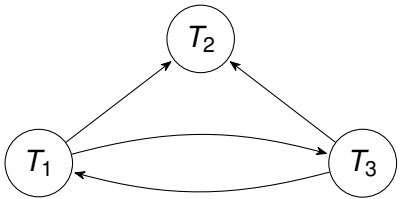
# Serializability

Consider the following sequential plan $S$ for the transactions $T_1$, $T_2$, and $T_3$:

$$S = \langle w_3(A), r_1(A), w_1(A), r_3(A), w_1(A), r_2(A), w_2(A) \rangle.$$

Draw the serialization graph (conflict graph) for the sequential plan $S$.

- Conflicts:
  - $T_3 \rightarrow T_1$ due to $w_3(A)$ and $r_1(A)$
  - $T_1 \rightarrow T_3$ due to $w_1(A)$ and $r_3(A)$
  - $T_1 \rightarrow T_2$ due to $w_1(A)$ and $w_2(A)$
  - $T_3 \rightarrow T_2$ due to $w_3(A)$ and $w_2(A)$

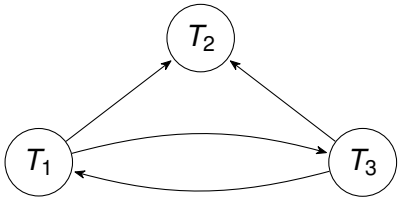


# Serializability

Consider the following sequential plan $S$ for the transactions $T_1$, $T_2$, and $T_3$:

$$S = \langle w_3(A), r_1(A), w_1(A), r_3(A), w_1(A), r_2(A), w_2(A) \rangle.$$

Can the serialization graph be used to make a statement about the serializability of $S$?



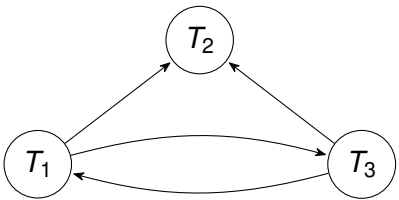
# Serializability

Consider the following sequential plan $S$ for the transactions $T_1$, $T_2$, and $T_3$:

$$S = \langle w_3(A), r_1(A), w_1(A), r_3(A), w_1(A), r_2(A), w_2(A) \rangle.$$

Can the serialization graph be used to make a statement about the serializability of $S$?

- The serialization graph contains a cycle
- No clear statement about serializability can be made based on the graph alone



# Serializability

Consider the following sequential plan $S$ for the transactions $T_1, \ldots, T_5$:

$$S = \langle r_1(A), r_2(A), r_1(B), w_2(A), r_3(C), w_3(D), r_4(D), w_1(B), r_5(E), w_5(D) \rangle.$$

Draw the serialization graph (conflict graph) for the sequential plan $S$.

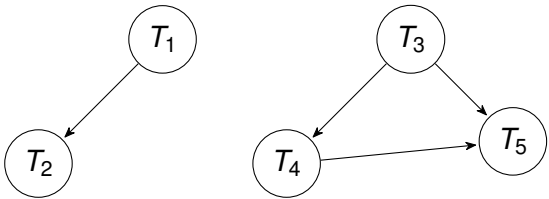
# Serializability

Consider the following sequential plan $S$ for the transactions $T_1, \ldots, T_5$:

$$S = \langle r_1(A), r_2(A), r_1(B), w_2(A), r_3(C), w_3(D), r_4(D), w_1(B), r_5(E), w_5(D) \rangle.$$

Draw the serialization graph (conflict graph) for the sequential plan $S$.

- Conflicts:
  - $T_1 \rightarrow T_2$ due to $r_1(A)$ and $w_2(A)$
  - $T_3 \rightarrow T_4$ due to $w_3(D)$ and $r_4(D)$
  - $T_3 \rightarrow T_5$ due to $w_3(D)$ and $w_5(D)$
  - $T_4 \rightarrow T_5$ due to $r_4(D)$ and $w_5(D)$

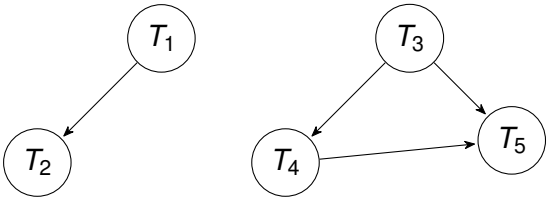


# Serializability

Consider the following sequential plan $S$ for the transactions $T_1, \ldots, T_5$:

$$S = \langle r_1(A), r_2(A), r_1(B), w_2(A), r_3(C), w_3(D), r_4(D), w_1(B), r_5(E), w_5(D) \rangle.$$

Is the plan $S$ serializable? If yes, provide a serial plan that is equivalent to $S$.



# Serializability

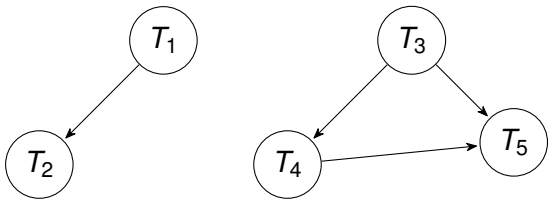
Consider the following sequential plan $S$ for the transactions $T_1, \ldots, T_5$:

$$S = \langle r_1(A), r_2(A), r_1(B), w_2(A), r_3(C), w_3(D), r_4(D), w_1(B), r_5(E), w_5(D) \rangle.$$

Is the plan $S$ serializable? If yes, provide a serial plan that is equivalent to $S$.

- The serialization graph is acyclic
- Thus, the plan is serializable
- One equivalent serial plan is:

$$S' = \langle r_1(A), r_1(B), w_1(B), r_2(A), w_2(A), r_3(C), w_3(D), r_4(D), r_5(E), w_5(D) \rangle.$$



# Anomalies

Consider the following transactions $T_1$ and $T_2$. What anomalies might occur when executing the transactions concurrently?

| $T_1$ | $T_2$ |
|---|---|
| $\alpha \leftarrow \mathit{read}(A)$; | |
| | $\beta \leftarrow \mathit{read}(A)$; |
| $\alpha \leftarrow \alpha - 100$; | |
| | $\beta \leftarrow \beta + 50$; |
| $\mathit{write}(A, \alpha)$; | |
| | $\mathit{write}(A, \beta)$; |

# Anomalies

Consider the following transactions $T_1$ and $T_2$. What anomalies might occur when executing the transactions concurrently?

| $T_1$ | $T_2$ |
|---|---|
| $\alpha \leftarrow \mathit{read}(A)$; | |
| | $\beta \leftarrow \mathit{read}(A)$; |
| $\alpha \leftarrow \alpha - 100$; | |
| | $\beta \leftarrow \beta + 50$; |
| $\mathit{write}(A, \alpha)$; | |
| | $\mathit{write}(A, \beta)$; |

- Lost update anomaly
- The update of one transaction is overwritten by the update of another transaction
- E.g., if the initial value of $A$ is 1000, the final value could be 900 or 1050 instead of 950

# Anomalies

Consider the following transactions $T_1$ and $T_2$. What anomalies might occur when executing the transactions concurrently?

| $T_1$ | $T_2$ |
|---|---|
| $\alpha \leftarrow \mathit{read}(A)$; | |
| $\alpha \leftarrow \alpha - 100$; | |
| $\mathit{write}(A, \alpha)$; | |
| | $\beta \leftarrow \mathit{read}(A)$; |
| | $\beta \leftarrow \beta + 50$; |
| $\mathit{abort}$; | |
| | $\mathit{write}(A, \beta)$; |

# Anomalies

Consider the following transactions $T_1$ and $T_2$. What anomalies might occur when executing the transactions concurrently?

| $T_1$ | $T_2$ |
|---|---|
| $\alpha \leftarrow read(A)$; | |
| $\alpha \leftarrow \alpha - 100$; | |
| $write(A, \alpha)$; | |
| | $\beta \leftarrow read(A)$; |
| | $\beta \leftarrow \beta + 50$; |
| $abort$; | |
| | $write(A, \beta)$; |

- Dirty read anomaly
- One transaction reads uncommitted data written by another transaction
- E.g., if the initial value of $A$ is 1000, $T_2$ might read 900 instead of 1000

# Granularity of Locking

There are locks with different granularity levels. What is the trade-off between choosing a finer or coarser granularity?

# Granularity of Locking

There are locks with different granularity levels. What is the trade-off between choosing a finer or coarser granularity?