

## 1.3 EXCHANGING DATA · 1.3.2

# Transaction processing & ACID

Transactions, the **ACID** properties, and **record locking**. Spec 1.3.2(f).

## 01 The ACID properties

**Atomicity** All-or-nothing; rollback if any step fails.

**Consistency** Database stays in a valid state (rules kept).

**Isolation** Concurrent transactions don't interfere.

**Durability** Committed changes are permanent (non-volatile storage / log).

## 02 Transactions & locking

**Transaction** One logical operation, often several steps that all succeed or all fail.

**Record locking** Locks a record being changed → enforces **isolation**.

**Prevents** The **lost update** problem.

**Downsides** Delays; possible **deadlock**.

## 03 Worked example: last seat on a flight

**Without isolation** Two users both read "1 free" and both book it → overbooked.

**With locking** User 1 locks the seat; User 2 waits; after commit, User 2 sees 0 free.

**Atomicity** If payment fails after reserving, the whole booking rolls back and frees the seat.

## FINAL PASS BEFORE THE EXAM

## Rapid exam tips

Seven slips on transaction questions.

**01**

**Atomicity** = all-or-nothing (rollback). **Consistency** = stays in a valid state. Don't merge them.

**02**

**Durability**: committed changes saved to **non-volatile storage** (a transaction log) to survive a crash.

**03**

**Record locking** enforces **isolation** — name the property when asked.

**04**

Locking prevents the **lost update** problem.

**05**

Locking downsides: **delays** and **deadlock** — be ready to give one.

**06**

A transaction is often **several steps** that must all succeed together, not one action.

**07**

"State what CID refers to" comes up — learn all four letters and a one-line meaning for each.