Explain with an example the use of **ganache-time-traveller**.

Ganache-time-traveller is a tool that helps simulate time passing in smart contract testing.

Suppose we have a smart contract that locks tokens for seven days before they can be withdrawn. We can use a ganache-time-traveller to instantly simulate the passage of these seven days to test if the tokens are locked and become withdrawable after this period.

**Example Use Case:** Testing a lending contract where loans are supposed to be repaid within a specific time frame.

const { advanceTime } = require('ganache-time-traveller');

it('tests loan expiry', async () => {

// Advance time by 30 days

await advanceTime(30 \* 24 \* 60 \* 60); const isExpired = await

lendingContract.isLoanExpired(loanId);

assert.equal(isExpired, true, 'Loan should be expired after 30 days');

});

Explanation:

In blockchain testing, time is simulated using Ganache-Time-Traveller. In this instance, it's used to advance the blockchain's current state by 30 days in order to check whether a loan in a lending contract really has expired as expected. During tests, time can be adjusted using the advanceTime function, which is imported from the ganache-time-traveller package. This makes it easier to effectively validate time-dependent logic in smart contracts.