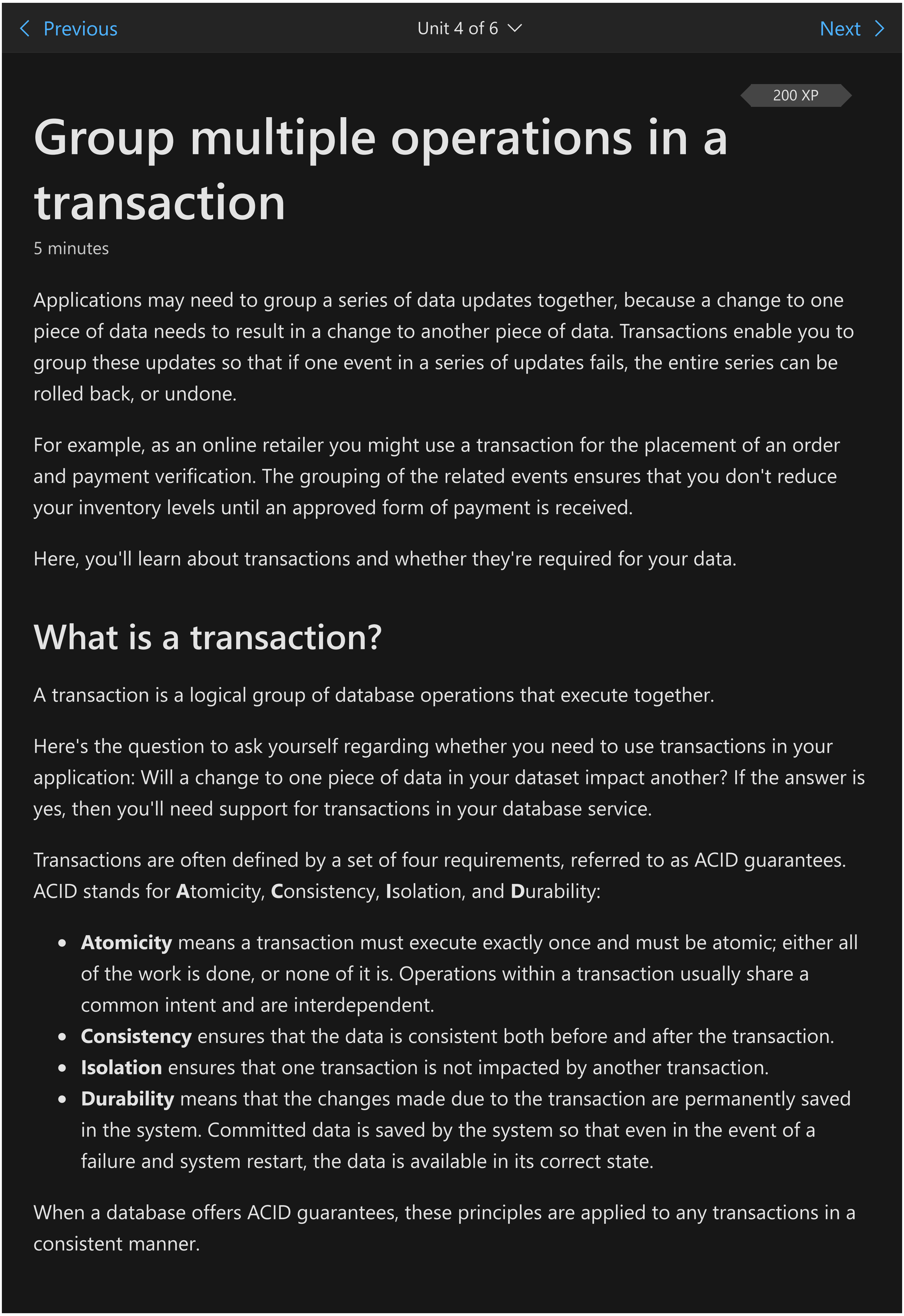

Previous

Unit 4 of 6

Next

200 XP

Group multiple operations in a transaction

5 minutes

Applications may need to group a series of data updates together, because a change to one piece of data needs to result in a change to another piece of data. Transactions enable you to group these updates so that if one event in a series of updates fails, the entire series can be rolled back, or undone.

For example, as an online retailer you might use a transaction for the placement of an order and payment verification. The grouping of the related events ensures that you don't reduce your inventory levels until an approved form of payment is received.

Here, you'll learn about transactions and whether they're required for your data.

What is a transaction?

A transaction is a logical group of database operations that execute together.

Here's the question to ask yourself regarding whether you need to use transactions in your application: Will a change to one piece of data in your dataset impact another? If the answer is yes, then you'll need support for transactions in your database service.

Transactions are often defined by a set of four requirements, referred to as ACID guarantees. ACID stands for **A**tomicity, **C**onsistency, **I**solation, and **D**urability:

- **Atomicity** means a transaction must execute exactly once and must be atomic; either all of the work is done, or none of it is. Operations within a transaction usually share a common intent and are interdependent.
- **Consistency** ensures that the data is consistent both before and after the transaction.
- **Isolation** ensures that one transaction is not impacted by another transaction.
- **Durability** means that the changes made due to the transaction are permanently saved in the system. Committed data is saved by the system so that even in the event of a failure and system restart, the data is available in its correct state.

When a database offers ACID guarantees, these principles are applied to any transactions in a consistent manner.

OLTP vs OLAP

Transactional databases are often called OLTP (Online Transaction Processing) systems. OLTP systems commonly support lots of users, have quick response times, and handle large volumes of data. They are also highly available (meaning they have very minimal downtime), and typically handle small or relatively simple transactions.

On the contrary, OLAP (Online Analytical Processing) systems commonly support fewer users, have longer response times, can be less available, and typically handle large and complex transactions.

The terms OLTP and OLAP aren't used as frequently as they used to be, but understanding them makes it easier to categorize the needs of your application.

Now that you're familiar with transactions, OLTP, and OLAP, let's walk through each of the data sets in the online retail scenario, and determine the need for transactions.

Product catalog data

Product catalog data should be stored in a transactional database. When users place an order and the payment is verified, the inventory for the item should be updated. Likewise, if the customer's credit card is declined, the order should be rolled back, and the inventory should not be updated. These relationships all require transactions.

Photos and videos

Photos and videos in a product catalog don't require transactional support. These files are changed only when an update is made or new files are added. Even though there is a relationship between the image and the actual product data, it's not transactional in nature.

Business data

For the business data, because all of the data is historical and unchanging, transactional support is not required. The business analysts working with the data also have unique needs in that they often require working with aggregates in their queries, so that they can work with the totals of other smaller data points.

Summary

Ensuring that your data is in the correct state is not always an easy task. Transactions can help by enforcing data integrity requirements on your data. If your data benefits from ACID principles, then choose a storage solution that supports transactions.

Check your knowledge

1. Which type of transactional database system would work best for product data?

○ OLAP

○ OLTP

2. Suppose the operations to update inventory and process payments are in the same transaction. A user is attempting to apply store credit for the full amount of an order, and submitted the exact same order (for the full amount) using their phone and laptop at the same time - so two identical orders are received. The database behind the scenes is an ACID-compliant database, what would happen?

○ Both orders would be processed and use the in-store credit.

○ One order would be processed and use the in-store credit, and the other order would update the remaining inventory for the items in the basket, but would not complete the order.

○ One order would be processed and use the in-store credit, and the other order would not be processed.

Check your answers