



# Data Interaction and Database Transaction

## Database Fundamentals

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Welcome to Data Interaction and Database Transaction.

# What you will learn

## At the core of the lesson

You will learn how to do the following:

- List different ways to interact with relational databases.
- Define the characteristics of a transaction.

Key terms:

- Database analyst
- Database administrator
- Transaction
- Atomicity, consistency, isolation, and durability (ACID)



You will learn how to do the following:

- List different ways to interact with relational databases.
- Define the characteristics of a transaction.

# Data sharing

## Making data available to multiple users

More and more, data needs to be made accessible to users who previously did not have access or the ability to update information in a timely manner.

From developers to end users, companies can now empower their employees to use data within databases and interact with these databases in various ways.



A database provides an efficient solution to share the same data between multiple users who are using different applications.

## Database interaction: Database roles

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You'll begin by learning about database interaction.

# Roles interacting with relational databases

People have different roles and different ways of interacting with a database.

- **Application developer**
  - Creates applications that populate and manipulate the data within a database according to the application's functional requirements
- **End user**
  - Uses reports that are created from the information within the database
- **Data analyst**
  - Collects, cleans, and interprets data within a database system
- **Database administrator**
  - Designs, implements, administers, and monitors data in database systems
  - Ensures consistency, quality, and security of the database



This slide lists the roles of individuals who commonly interact with a database. Each role has differing ways of interacting with the database. Direct interaction requires the user to know how to issue a structure query language (SQL) command in a specific database management system (DBMS). Therefore, roles that might have less knowledge of SQL would interact with the database in other ways.

## Role interaction types (1 of 2)



Application developer

### Develops and tests applications

- Creates different applications that use the data



Developer computer

The application developer develops and tests the application that uses the database.

The server returns the results to the application.



End user

### Occasionally interacts directly

- Occasionally accesses the database directly if they have knowledge of SQL



End user computer

The end user queries and updates the database.

The server returns the results to the end user computer.



A database is typically accessed through an application that an application developer creates. The developer embeds the SQL statements needed to interact with the database in the application's code. Therefore, end users can query and update the data in the context of a business task without having to know SQL.

If the end user has a little SQL knowledge, they might occasionally access the database directly, typically to perform read-only queries.

## Role interaction types (2 of 2)



Data analyst

### Enters SQL commands

- Views and manipulates data by directly entering SQL
- Mainly uses the **SELECT** command



Analyst computer

The data analyst sends a SQL query to the server.

The server returns the results.



Database administrator

### Enters SQL commands

- Manages all of the components in the database
- Uses all SQL commands



Administrator computer

The database administrator sends a SQL query to the server.

The server returns the results.

Data analysts and database administrators also interact with databases. Individuals in these roles have a greater level of expertise than a typical end user. Data analysts and database administrators would be able to create and run a SQL command.

## Data interaction models

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You'll now go into further detail on each model of database interaction.



## Interacting with relational databases

### Types of database interaction:

- Client-server
- Three-tier web application



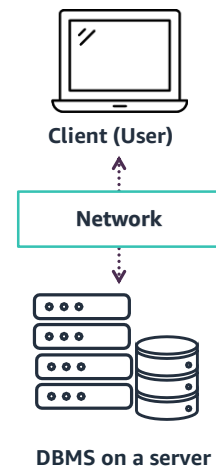
The roles mentioned previously in the section on database interaction and database roles list the ways that different job roles interact with databases. In this section, you will learn about two primary models for database interaction:

- Client-server
- Three-tier web application

# Client-server interaction

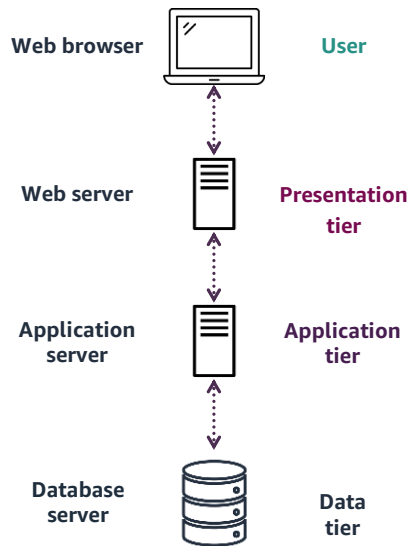
## Client-server interaction model:

1. Users use computers and devices that run client applications, which use SQL to request data.
2. The applications use SQL that is sent to the server over a network to communicate with the database.
3. The server runs a database management system, which receives the requests, processes the SQL, and returns the response.



In the client-server model, the user can use an application on the client machine to send a SQL command to the server over the network. The DBMS interface on the server receives the command, runs it, and returns the result back to the client.

## Three-tier web application interaction



1. The user uses a client computer or device that runs a web browser. A webpage that is running in the web browser captures the user's input and sends a request to the web server.
2. The web server gathers the information on the webpage and forwards the request to the application server for processing.
3. A web application component that is running on the application server receives the request. It contains the SQL commands to access the database to satisfy the request. The component sends the commands to the database server.
4. The DBMS that runs on the database server receives and processes the SQL commands. The DBMS returns the results to the application server.
5. The web application component on the application server processes the results and returns them to the web server.
6. The web server formats the results into a webpage.
7. The web browser on the client device displays the webpage that contains the SQL results to the user.

This slide illustrates and describes the typical steps that are performed when a user accesses a database by using a three-tier web application.

## Embedded SQL in application code

- In both interaction models, an application contains the SQL commands that the user requires.
- An application developer embeds SQL statements in the application code so that the application can perform database tasks.
- The application is installed on a user computer or an application server.



An application developer can create an application that users can use to perform database tasks. This way, users do not need to know the details of the database or learn SQL. This approach is used in both the client-server and three-tier web application models.



## Transactions in databases

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You will now learn more information about transactions in databases.

## What are transactions?

- A transaction is a collection of changes made to a database that must be performed as a unit.
- For example, suppose that you have an account table, which keeps track of bank account balances for customers. Suppose a customer wants to transfer \$100 from their checking account to their savings account. Each account type has a row in the table. You must do the following:
  - Reduce the balance of the checking account row by \$100.
  - Increase the balance of the savings account row by \$100.
  - The two change operations must either both succeed or both fail in order to preserve the integrity of the database. You cannot have one change succeed and the other fail.
- A transaction is not successful unless each of its operations is successful.

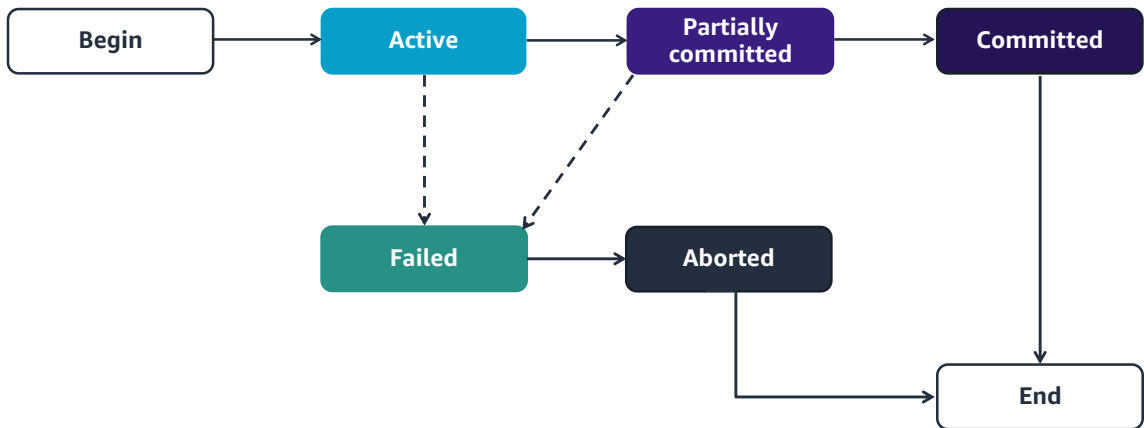


A transaction is also called a *logical unit of work*. In other words, either all of its operations succeed, and the transaction succeeds, or if one or more operations fail, then the entire transaction fails.

At the database level, either all the database changes related to the transaction are performed, or no change is made to the database at all. It's an all or nothing modification.

## Status of a transaction

From beginning to end, a transaction's state changes:



The status of a transaction can change as it moves from beginning to end. The statuses are as follows:

- **Active state:** In the initial state of every transaction and when the transaction is being run, the status is active.
- **Partially committed:** A transaction is in a partially committed state when it is completing its final operation.
- **Failed state:** A transaction is in a failed state when any checks made by the database recovery system fail.
- **Aborted state:** An aborted transaction occurs if the transaction is in a failed state, and the database rolls back to its original state before running the transaction.
- **Committed state:** When all of the operations within a transaction have been successfully performed, the transaction is considered committed.

## Transactions use cases

### You can use transactions to do the following:

- Run a set of operations so that the database never contains the result of partial operations.
  - If one operation fails, the database is restored to its original state.
  - If no errors occur, the full set of statements changes the database.
- Provide isolation between programs that access a database simultaneously.
  - If this isolation does not happen, the outcomes could prove to be incorrect.



Without transactions, if two or more requests attempt to change the same data in a database simultaneously, the order and effect of the changes are unpredictable. As a result, the database might end up in a corrupted state. Transactions provide a mechanism, called isolation, which ensures that the simultaneous change requests are processed one at a time and do not interfere with each other.



## Properties of transactions

**Transactions follow four standard properties—atomicity, consistency, isolation, and durability—which are known as ACID.**

- **A**tomicity ensures that changes are successfully completed all at once or not at all.
- **C**onsistency ensures that any changes will not violate the integrity of the database, including any constraints.
- **I**solation keeps all transactions in isolation. Transactions are isolated so that they do not interfere with the other transactions.
- **D**urability ensures that as soon as a transaction is committed, the change is permanent.

This slide lists the four important benefits of transactions, which are commonly referred to as the ACID properties.

## Checkpoint questions



**What are two different roles mentioned in this module that interact with databases?**



**What are the different transaction states?**



**What are two primary models for database interaction?**

1. What are two different roles mentioned in this module that interact with databases?

End users, data analysts, database administrators, and application developers

2. What are the different transaction states?

Active state, partially committed, failed state, aborted state, and committed state

3. What are two primary models for database interaction?

Client-server model and three-tier web application model

## Key takeaways



- The roles that interact with a database include the database administrator, the application developer, the data analyst, and the end user.
- An end user typically accesses a database through a client-server application or a three-tier web application.
- A database transaction is a collection of database changes that must be performed as a unit.
- Transactions follow four standard properties, which are known as ACID.

This module includes the following key takeaways:

- The roles that interact with a database include the database administrator, the application developer, the data analyst, and the end user.
- An end user typically accesses a database through a client-server application or a three-tier web application.
- A database transaction is a collection of database changes that must be performed as a unit.
- Transactions follow four standard properties, which are known as ACID.



# Thank you



© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.

Thank you for completing this module.