Google

Module 1 | **Lesson 8**

Digital Buildings Ontology (DBO)

Before you get started

This onboarding deck has interactive features and activities that enable a self-guided learning experience.
To help you get started, here are two tips for viewing and navigating through the deck.

1 **View this deck in presentation mode.**
- To enter presentation mode, you can either:
  - Click the **Present** or **Slideshow** button in the top-right corner of this page.
  - Press **Ctrl+F5** (Windows), **Cmd+Enter** (macOS), or **Ctrl+Search+5** (Chrome OS) on your keyboard.
- To exit presentation mode, press the **Esc** key on your keyboard.

2 **Navigate by clicking the buttons and links.**
- Click the **Back** or **Next** buttons to go backward or forward in the deck. Moving forward, you'll find them in the bottom corners of every slide.
- Click blue text to go to another slide in this deck or open a new page in your browser.
- For the best learning experience, using your keyboard or mouse wheel to navigate is discouraged.

**Ready to get started?**   Let's go!

Conceptual model revisited

Here's another look at the DBO conceptual model from Lesson 2.

In this lesson, you'll explore one modeling concept from the abstract model. Remember, the following modeling concepts are used to describe the relationships that can occur between entities:

- Mappings
  - Translations
  - Links
- Connections

**Do you see these concepts in the diagram?**
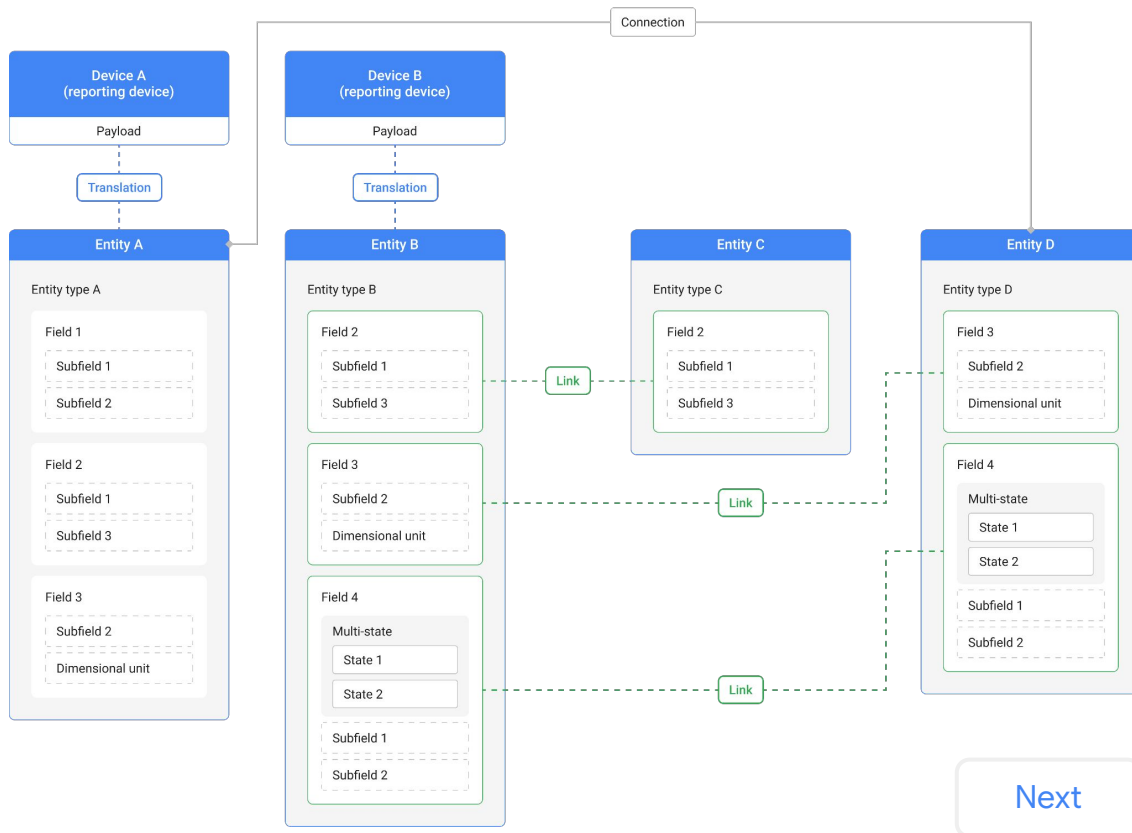
Back

Next

Lesson 8

Connections

What you'll learn about:

- Connections
- Source and target entities
- Connection types
- Connection construction

By the end of this lesson, you'll be able to:

- Describe the concept of a connection.
- Identify a connection in source code.
- Understand the relationship between source and target entities.
- Recognize the different connection types.
- Construct a connection.
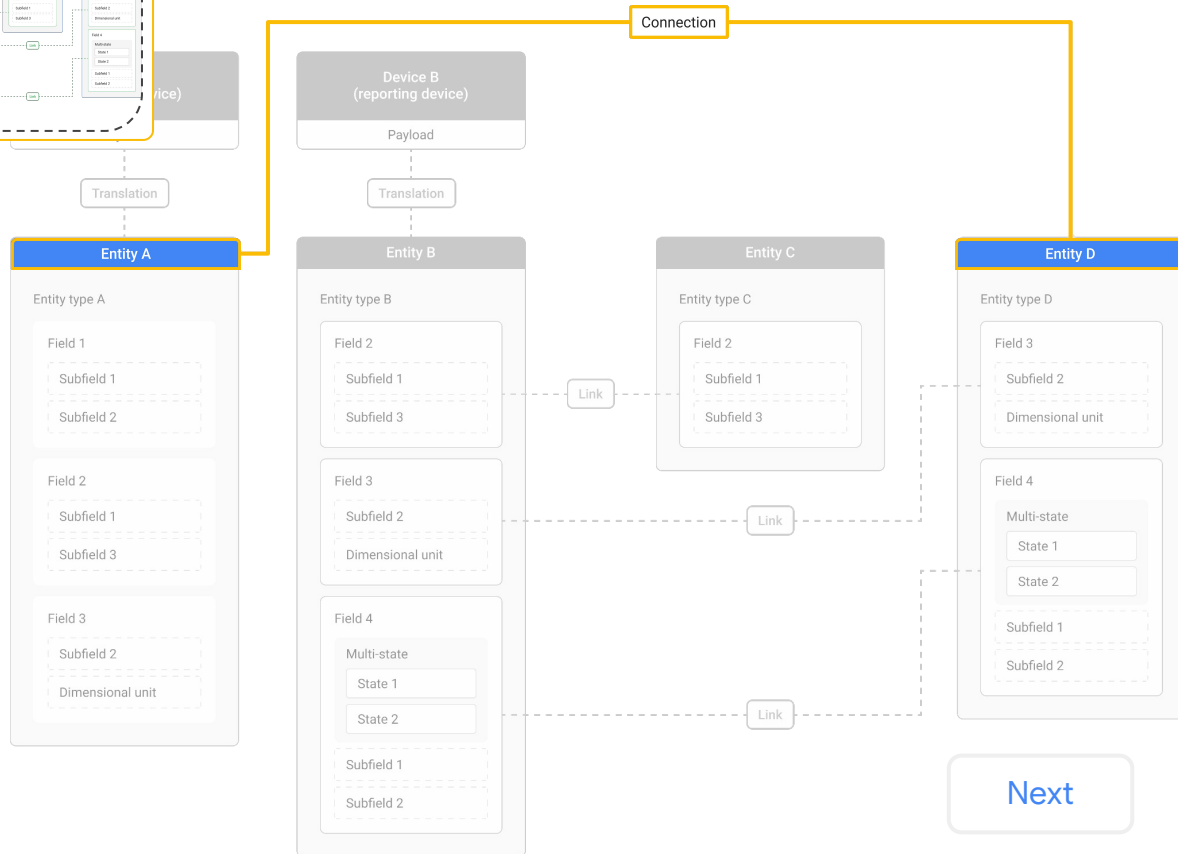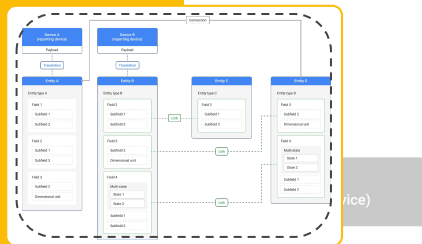
Back

Next

What's a connection?

A connection is a named relationship between two entities.

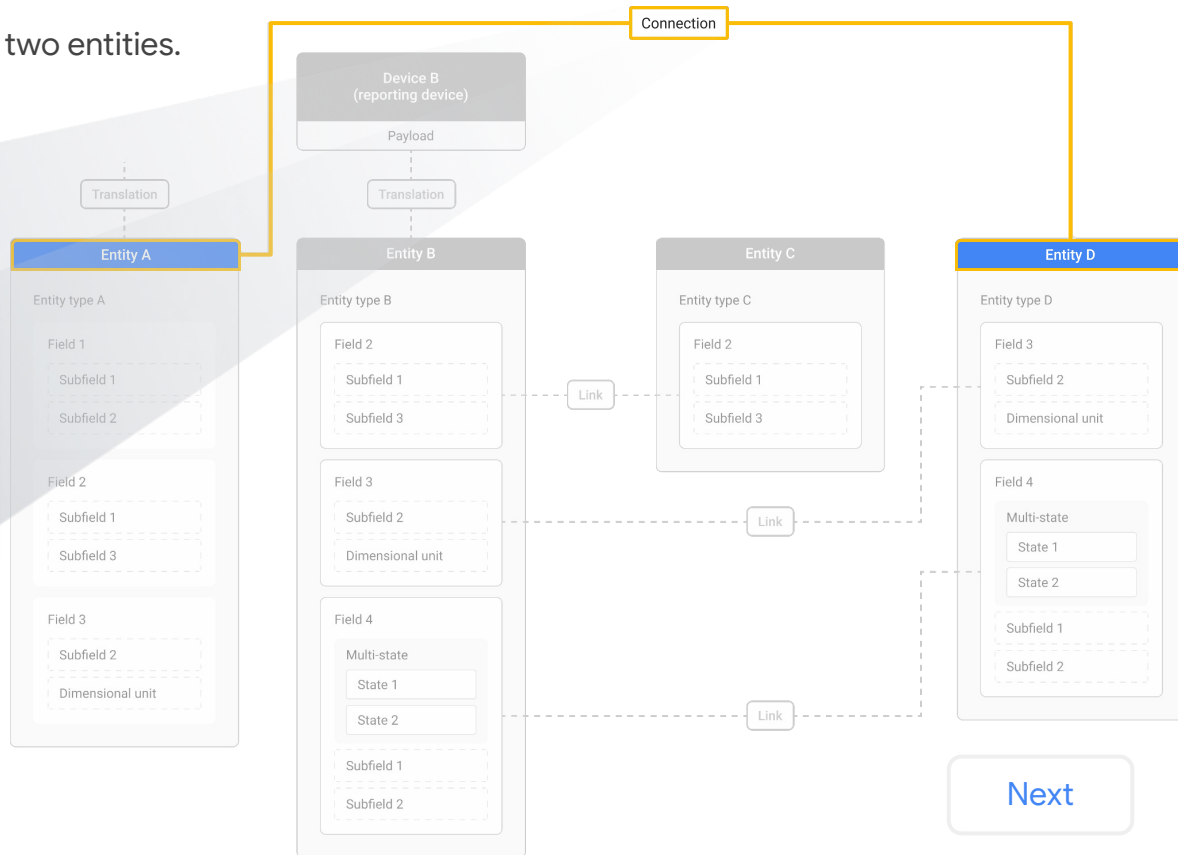Each connection has a specific definition that describes the relationship.

Example

```
VAV-32:
  type: HVAC/VAV
  connections:
    UK-LON-6PS-1: CONTAINS
    AHU-123: FEEDS
```

Connection

Device B
(reporting device)

Payload

Translation

Translation

Entity A

Entity type A

Field 1

Subfield 1

Subfield 2

Field 2

Subfield 1

Subfield 3

Field 3

Subfield 2

Dimensional unit

Entity B

Entity type B

Field 2

Subfield 1

Subfield 3

Field 3

Subfield 2

Dimensional unit

Field 4

Multi-state

State 1

State 2

Subfield 1

Subfield 2

Entity C

Entity type C

Field 2

Subfield 1

Subfield 3

Link

Entity D

Entity type D

Field 3

Subfield 2

Dimensional unit

Field 4

Multi-state

State 1

State 2

Subfield 1

Subfield 2

Link

Link

Link

Back

Next

Source and target entities

A connection is a directional relationship going from a source to a target.

Source entity   CONNECTION →   **Target entity**

A **source entity** is the entity that connects to another entity.

A **target entity** is the entity that is being connected.

Why does this matter?

It's important to properly identify the source entity and the target entity because valid connection construction follows a specific syntax. As shown in the example to the right, identifying source and target entities should be fairly intuitive in most cases.

Connections are always defined on the target entity. If a connection is mistakenly defined on the source entity, then it'll run into a validation deadlock. When a validation deadlock occurs, the source entity is invalidated and the target entity is deleted.

Example

A building contains a floor. The building is the source entity. The floor is the target entity. The defined connection is `contains`.

BUILDING — CONTAINS → FLOOR

An air handling unit (AHU) feeds a variable air volume (VAV). The AHU is the source entity. The VAV is the target entity. The defined connection is `feeds`.

AHU — FEEDS → VAV

A chilled water system (CHWS) has a pump (PMP). The CHWS is the source entity. The PMP is the target entity. The defined connection is `has_part`.

CHWS — HAS_PART → PMP
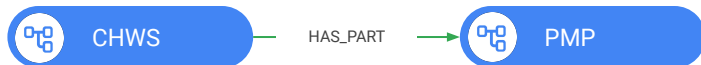
Back

Next

Connection definitions

There are five different types of connection definitions.

Each type of connection can be used to define the various system and spatial relationships that occur between two entities.

CONTAINS

This connection is used to show the source entity physically encapsulates at least part of the target entity.

CONTROLS

This connection is used to show the source entity determines or affects the internal state or behavior of the target entity.

FEEDS

This connection is used to show the source entity provides some media (e.g., water or air) to the target entity.

HAS_PART

This connection is used to show the source entity has a component or part that's defined by the target entity.

HAS_RANGE

This connection is used to show the source entity has a coverage or detection range defined by the target entity.

Back

Note: Connection definitions are always defined in the global namespace.

Next

Connection construction
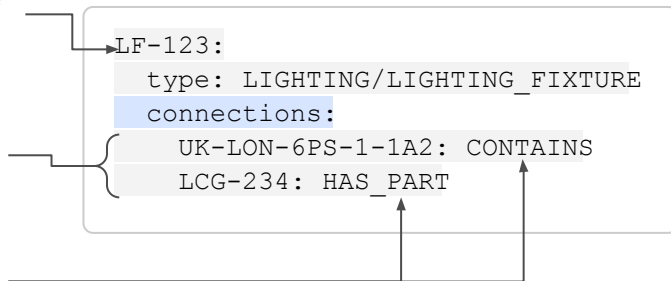
Connections are always defined on the target entity.

A connection is defined with the `connections` block.
A target entity will use the block to define its source entity. Then, the connection is defined with one of the five connection definitions.

First, you'll see the name of the target entity.

In the `connections` block, you'll see the names of the source entities...

...along with the connection definition.

```
LF-123:
  type: LIGHTING/LIGHTING_FIXTURE
  connections:
    UK-LON-6PS-1-1A2: CONTAINS
    LCG-234: HAS_PART
```

Example

The entity `VAV-32` has a defined connection:

```
VAV-32:
  type: HVAC/VAV
  connections:
    UK-LON-6PS-1: CONTAINS
    AHU-123: FEEDS
```

- The target entity is `VAV-32`, which is a variable air volume system.
- The source entity is `AHU-123`, which is an air handling unit.
- The defined connection is `FEEDS`, which means the source entity provides media to the target entity.

Note the syntax. `VAV-32`, the target entity, has defined its connection to `AHU-123`, the source entity. This means `AHU-123` feeds media to `VAV-32`. Put simply, an air handling unit feeds air to a variable air volume system.

It's counterintuitive to switch `AHU-123` and `VAV-32` in the model. The reason the syntax is defined this way is to allow for the deletion of an entity (along with its connections) without breaking the source. This can be useful in editing of building config files.

Back

Next

Lesson 8

Knowledge check

Let's take a moment to reflect on what you've learned so far.

- The next slides will have questions about the concepts that were introduced in this lesson.

- Review each question and select the correct response.

If there are more than two answer options, you won't be able to move forward until the correct answer is selected.

*Click **Next** when you're ready to begin.*

Back

Next

Google

Knowledge check 1

A connection is a named relationship between two _____.

Fill in the blank.

Select the best answer from the options listed below.

entities

mappings

namespaces

entity types

Back

Next

Knowledge check 1

A connection is a named relationship between two _____.

Fill in the blank.

Select the best answer from the options listed below.

entities	mappings
namespaces	entity types

That's right! 🎉

Connections occur between two **entities**.

To define a connection, you'll use the `connections` block and apply one of the following definitions:

- `CONTAINS`
- `CONTROLS`
- `FEEDS`
- `HAS_PART`
- `HAS_RANGE`

Back

Next

Knowledge check 1

A connection is a named relationship between two _____.

Fill in the blank.

Select the best answer from the options listed below.

- entities
- **mappings**
- namespaces
- entity types

Hmm, that's not right! 🤔

Try again

Back

Next

Knowledge check 1

A connection is a named relationship between two _____.

Fill in the blank.

Select the best answer from the options listed below.

entities

mappings

namespaces

entity types

Hmm, that's not right! 🤔

Try again

Back

Next

Knowledge check 1

A connection is a named relationship between two _____.

Fill in the blank.

Select the best answer from the options listed below.

entities	mappings
namespaces	entity types

Hmm, that's not right! 🤔

Try again

Back

Next

Knowledge check 2

Connections must be defined on a specific entity to avoid a validation deadlock.

On which entity should a connection be defined?
Select the best answer from the options listed below.

On the source entity

On the target entity

Back

Next

Knowledge check 2

Connections must be defined on a specific entity to avoid a validation deadlock.

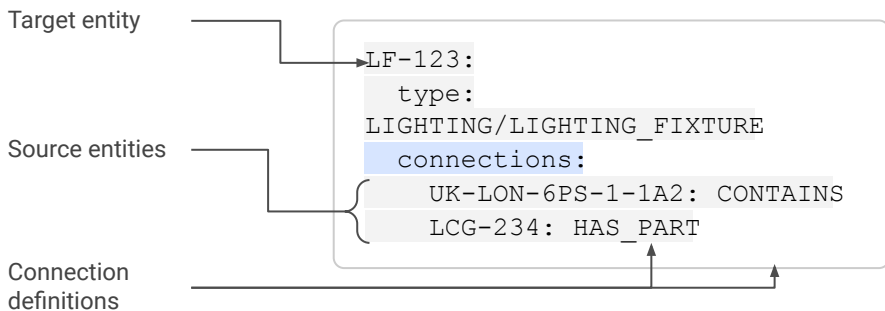## On which entity should a connection be defined?

Select the best answer from the options listed below.

On the source entity

On the target entity

Hmm, that's not right! 🤔

Connections are always defined **on the target entity**.

Target entity ──────┐
```
                    └──→ LF-123:
                           type:
                         LIGHTING/LIGHTING_FIXTURE
Source entities ─────────  connections:
                              UK-LON-6PS-1-1A2: CONTAINS
                              LCG-234: HAS_PART
Connection
definitions
```

If a connection is defined on the source entity, then a validation deadlock occurs. This is problematic because the source entity will be invalidated and the target entity will be deleted.

Back

Next

Knowledge check 2

Connections must be defined on a specific entity to avoid a validation deadlock.

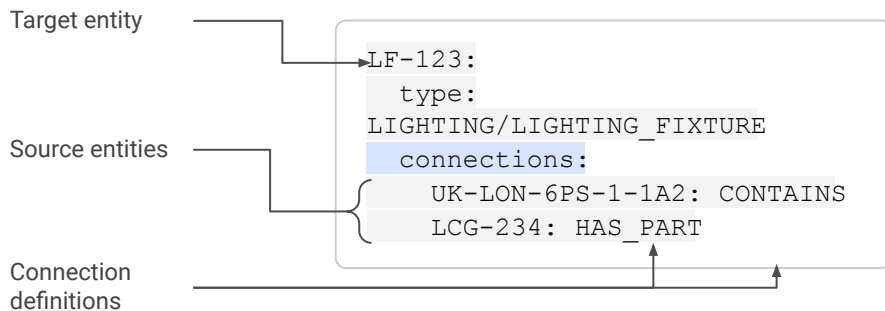## On which entity should a connection be defined?
Select the best answer from the options listed below.

On the source entity

On the target entity

That's right! 🎉

Connections are always defined **on the target entity**.

Target entity

```
LF-123:
  type:
LIGHTING/LIGHTING_FIXTURE
  connections:
    UK-LON-6PS-1-1A2: CONTAINS
    LCG-234: HAS_PART
```

Source entities

Connection definitions

If a connection is defined on the source entity, then a validation deadlock occurs. This is problematic because the source entity will be invalidated and the target entity will be deleted.

Back

Next

Knowledge check 3

Let's say you have a fan that delivers air to a single zone.
You'd like to model the `FAN` and the `ZONE`, and then define a
connection between them.
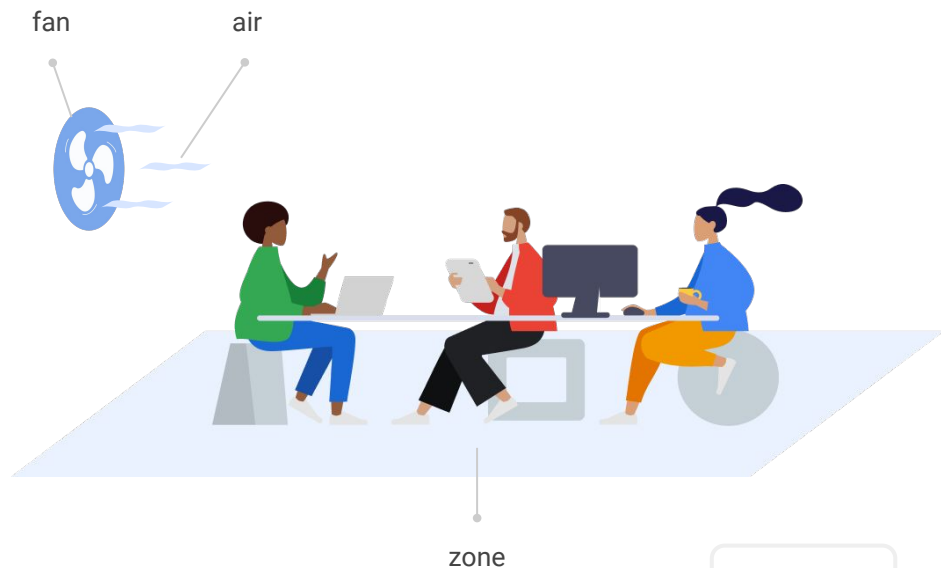
Which is the most appropriate connection definition?
Select the best answer from the options listed below.

`FAN` is the target, `ZONE` is the source, `FEEDS` is the connection

`FAN` is the source, `ZONE` is the target, `FEEDS` is the connection

`FAN` is the target, `ZONE` is the source, `CONTAINS` is the connection

`FAN` is the source, `ZONE` is the target, `CONTAINS` is the connection

fan   air

zone

Back

Next

Knowledge check 3

Let's say you have a fan that delivers air to a single zone. You'd like to model the `FAN` and the `ZONE`, and then define a connection between them.

Which is the most appropriate connection definition?

Select the best answer from the options listed below.

`FAN` is the target, `ZONE` is the source, `FEEDS` is the connection

`FAN` is the source, `ZONE` is the target, `FEEDS` is the connection

`FAN` is the target, `ZONE` is the source, `CONTAINS` is the connection

`FAN` is the source, `ZONE` is the target, `CONTAINS` is the connection

Close... but not quite right! 🤔

In this scenario, it doesn't make sense for the `fan` to be the target entity since it's delivering air to the `zone`.

Try again

Back

Next

Knowledge check 3

Let's say you have a fan that delivers air to a single zone. You'd like to model the `FAN` and the `ZONE`, and then define a connection between them.

Which is the most appropriate connection definition?

Select the best answer from the options listed below.

`FAN` is the target, `ZONE` is the source, `FEEDS` is the connection

`FAN` is the source, `ZONE` is the target, `FEEDS` is the connection

`FAN` is the target, `ZONE` is the source, `CONTAINS` is the connection

`FAN` is the source, `ZONE` is the target, `CONTAINS` is the connection

That's right! 🎉

The `FAN` is the source entity. The `ZONE` is the target entity. The connection type is `FEEDS`.

FAN — FEEDS → ZONE

```
ZONE-123:
  type: HVAC/ZONE
  connections:
    UK-LON-6PS-1: CONTAINS
    FAN-123: FEEDS
```

Back

Next

Knowledge check 3

Let's say you have a fan that delivers air to a single zone. You'd like to model the **FAN** and the **ZONE**, and then define a connection between them.

Which is the most appropriate connection definition?

Select the best answer from the options listed below.

FAN is the target, ZONE is the source, FEEDS is the connection

FAN is the source, ZONE is the target, FEEDS is the connection

FAN is the target, **ZONE** is the source, **CONTAINS** is the connection

FAN is the source, ZONE is the target, CONTAINS is the connection

Close… but not quite right! 🤔

In this scenario, it doesn't make sense for the **fan** to be the target entity since it's delivering air to the **zone**. The **zone** might not contain the **fan** either.

Try again

Back

Next

Knowledge check 3

Let's say you have a fan that delivers air to a single zone. You'd like to model the `FAN` and the `ZONE`, and then define a connection between them.

Which is the most appropriate connection definition?
Select the best answer from the options listed below.

`FAN` is the target, `ZONE` is the source, `FEEDS` is the connection

`FAN` is the source, `ZONE` is the target, `FEEDS` is the connection

`FAN` is the target, `ZONE` is the source, `CONTAINS` is the connection

`FAN` is the source, `ZONE` is the target, `CONTAINS` is the connection

Close... but not quite right! 🤔

In this scenario, the `zone` might not contain the `fan`.

Try again

Back

Next

Lesson 8 summary

Let's review what you learned about:

- Connections
- Source and target entities
- Connection types
- Connection construction

Now you should be able to:

- Describe the concept of a connection.
- Identify a connection in source code.
- Understand the relationship between source and target entities.
- Recognize the different connection types.
- Construct a connection.
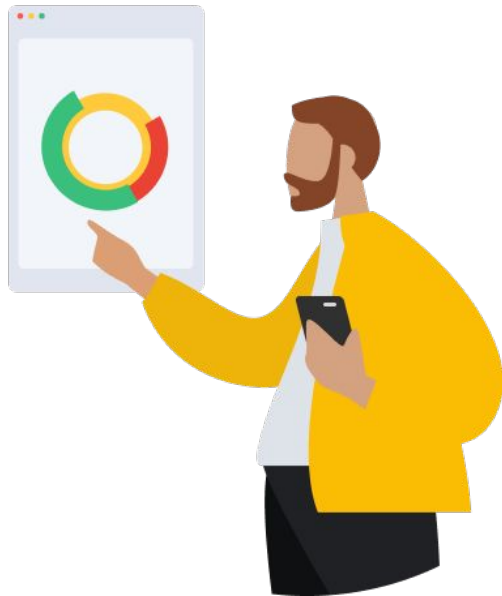
Back

Next

You completed Lesson 8!

Now's a great time to take a quick break before starting Lesson 9.

Ready for Lesson 9?

Let's go!

Back

Google

Have questions?

For future reference, keep these contacts and resources easily accessible for technical and procedural questions.

Key contacts

- For DBO questions: Trevor (tsodorff@) or Charbel (charbelk@)
- For UDMI questions: udmi-discuss@

Helpful resources

Bookmark these resources for future reference.

- digitalbuildings / ... / resources / connections
  Contains all of the available connections.
- digitalbuildings / ontology
  Contains the documentation and configuration files for the DBO.
- Building Configuration
  Describes the configuration format for mapping concrete assets to a model.