**Entities**

An entity has neither behavior nor data; instead, it identifies which pieces of data belong together. [Systems](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/ecs_systems.html) provide the behavior, and [components](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/ecs_components.html) store the data.

An entity is essentially an ID. Entity IDs are stable; you can use them to store a reference to another component or entity.

An [EntityManager](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html) manages all of the entities in a [World](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.World.html). An EntityManager maintains the list of entities and organizes the data associated with an entity for optimal performance.

As you create entities and add components to them, the EntityManager keeps track of the unique combinations of components on the existing entities. Such a unique combination is called an **Archetype**. The EntityManager creates an [EntityArchetype](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityArchetype.html) struct as you add components to an entity. You can use existing EntityArchetypes to create new entities that conform to that archetype. You can also create an EntityArchetype in advance and use that to create entities.

**Creating entities**

The easiest way to create an entity is in the Unity Editor. You can set ECS to convert both GameObjects placed in a Scene and Prefabs into entities at runtime. For more dynamic parts of your game or application, you can create spawning systems that create multiple entities in a job. Finally, you can use one of the [EntityManager.CreateEntity](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html" \l "Unity_Entities_EntityManager_CreateEntity) functions to create entities one at a time.

Creating entities with an EntityManager

Use one of the [EntityManager.CreateEntity](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html" \l "Unity_Entities_EntityManager_CreateEntity) functions to create an entity. ECS creates the entity in the same World as the EntityManager.

You can create entities one-by-one in the following ways:

* Create an entity with components that use an array of [ComponentType](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.ComponentType.html) objects.
* Create an entity with components that use an [EntityArchetype](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityArchetype.html).
* Copy an existing entity, including its current data, with [Instantiate](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html#Unity_Entities_EntityManager_Instantiate_Unity_Entities_Entity_)
* Create an entity with no components and then add components to it.

You can also create multiple entities at a time:

* Fill a NativeArray with new entities with the same archetype using [CreateEntity](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html" \l "Unity_Entities_EntityManager_CreateEntity).
* Fill a NativeArray with copies of an existing entity, including its current data, using [Instantiate](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html#Unity_Entities_EntityManager_Instantiate_Unity_Entities_Entity_).
* Explicitly create chunks populated with a specified number of entities with a given archetype with [CreateChunk](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityManager.html" \l "Unity_Entities_EntityManager_CreateChunk_).

**Adding and removing components**

After an entity has been created, you can add or remove components. When you do this, the archetype of the affected entities change and the EntityManager must move altered data to a new chunk of memory, as well as condense the component arrays in the original chunks.

Changes to an entity that cause structural changes — that is, adding or removing components that change the values of SharedComponentData, and destroying the entity — cannot be done inside a job because these could invalidate the data that the job is working on. Instead, you add the commands to make these types of changes to an [EntityCommandBuffer](https://docs.unity3d.com/Packages/com.unity.entities@0.9/api/Unity.Entities.EntityCommandBuffer.html) and execute this command buffer after the job is complete.

The EntityManager provides functions to remove a component from a single entity as well as all of the entities in a NativeArray. For more information, see the documentation on [Components](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/ecs_components.html).

**Iterating entities**

Iterating over all entities that have a matching set of components, is at the center of the ECS architecture. See [Accessing entity Data](https://docs.unity3d.com/Packages/com.unity.entities@0.9/manual/chunk_iteration.html).

记忆：

An entity is essentially an ID.

An entity has neither behavior nor data; instead, it identifies which pieces of data belong together.