

scheduler_api_test module

hook_install()

after creating the field definition

Drupal::entityDefinitionUpdateManager()

->installFieldStorageDefinition

which is in /core/lib/Drupal/Core/Entity/EntityDefinitionUpdateManager.php

function installFieldStorageDefinition

calls \$this->entityManager->onFieldStorageDefinitionCreate

which is in /core/lib/Drupal/Core/Entity/Sql/SqlContentEntityStorage.php

function onFieldStorageDefinitionCreate(FieldStorageDefinitionInterface
\$storage_definition)

not much here.

allowsSharedTableStorage is true

then:

```
$this->wrapSchemaException(function () use ($storage_definition) {  
    $this->getStorageSchema()-  
>onFieldStorageDefinitionCreate($storage_definition);  
});
```

for my site this is in /core/lib/Drupal/Core/Entity/Sql/
SqlContentEntityStorageSchema.php

```
public function onFieldStorageDefinitionCreate(FieldStorageDefinitionInterface  
$storage_definition) {  
    $this->performFieldSchemaOperation('create', $storage_definition);  
}
```

in the same file

```
protected function performFieldSchemaOperation($operation,  
FieldStorageDefinitionInterface $storage_definition, FieldStorageDefinitionInterface  
$original = NULL) {
```

```
    ...  
    $this->{$operation . 'SharedTableSchema'}($storage_definition, $original);  
}
```

in same file

```
protected function createSharedTableSchema(FieldStorageDefinitionInterface  
$storage_definition, $only_save = FALSE) {
```

for each of the four possible tables, `$table_mapping->getFieldNames($table_name)`
returns all the existing fields, but the new one is not given, hence the rest of the
function does nothing.

So something has to be done before this stage, to have the new field in that list.