ActiveRecord::ConnectionAdapters::TableDefinition

Instantiates a new column for the table. See connection.add_column for available options.

Additional options are:

- `:index` - Create an index for the column. Can be either `true` or an options hash.

This method returns `self`.

Examples

```
# Assuming +td+ is an instance of TableDefinition
td.column(:granted, :boolean, index: true)
```

Short-hand examples

Instead of calling column directly, you can also work with the short-hand definitions for the default types. They use the type as the method name instead of as a parameter and allow for multiple columns to be defined in a single statement.

What can be written like this with the regular calls to column:

```
create_table :products do |t|
  t.column :shop_id,     :integer
  t.column :creator_id,  :integer
  t.column :item_number, :string
  t.column :name,        :string, default: "Untitled"
  t.column :value,       :string, default: "Untitled"
  t.column :created_at,  :datetime
  t.column :updated_at,  :datetime
end
add_index :products, :item_number
```

can also be written as follows using the short-hand:

```
create_table :products do |t|
  t.integer :shop_id, :creator_id
  t.string  :item_number, index: true
  t.string  :name, :value, default: "Untitled"
  t.timestamps null: false
end
```

There's a short-hand method for each of the type values declared at the top. And then there's #timestamps that'll add `created_at` and `updated_at` as datetimes.

#references will add an appropriately-named _id column, plus a corresponding _type column if the `:polymorphic` option is supplied. If `:polymorphic` is a hash of options, these will be used when creating the `_type` column. The `:index` option will also create an index, similar to calling add_index. So what can be written like this:

```
create_table :taggings do |t|
  t.integer :tag_id, :tagger_id, :taggable_id
  t.string  :tagger_type
  t.string  :taggable_type, default: 'Photo'
end
add_index :taggings, :tag_id, name: 'index_taggings_on_tag_id'
add_index :taggings, [:tagger_id, :tagger_type]
```

Can also be written as follows using references:

```
create_table :taggings do |t|
  t.references :tag, index: { name: 'index_taggings_on_tag_id' }
  t.references :tagger, polymorphic: true, index: true
  t.references :taggable, polymorphic: { default: 'Photo' }
end
```