Schema Visualization

Schema Visualization

Anonymous · 09/09/2026

Schema Visualization

Derafu ETL includes powerful tools for visualizing database schemas through various output formats. These visualizations are useful for documentation, planning, and understanding the structure of your data.

Available Visualization Formats

Derafu ETL supports multiple visualization formats through schema targets:

- **Markdown**: Human-readable documentation.
- **D2**: Interactive entity-relationship diagrams.
- **Text**: Simple text representation.
- **SQL**: Database creation scripts.
- **Doctrine Schema**: Programmatic schema representation.

Using Schema Targets

Schema targets implement the `SchemaTargetInterface` and convert a schema to a specific format.

Basic Usage

All schema targets follow a similar pattern:

```
use Derafu\ETL\Database\DatabaseManager;
use Derafu\ETL\Schema\Target\CustomSchemaTarget; // Just an example.

// Connect to the database.
$manager = new DatabaseManager();
$database = $manager->connect('database.sqlite');

// Get the schema.
$schema = $database->schema();

// Create the target.
$target = new CustomSchemaTarget();
```

```
// Apply the schema to the target.
$output = $target->applySchema($schema);

// Use the output.
file_put_contents('output-file.custom', $output);
```

Markdown Schema

The `MarkdownSchemaTarget` generates comprehensive Markdown documentation for your database schema.

```
use Derafu\ETL\Schema\Target\MarkdownSchemaTarget;

$target = new MarkdownSchemaTarget();
$markdown = $target->applySchema($schema);

file_put_contents('schema.md', $markdown);
```

The generated Markdown includes:

- Table of contents.
- Detailed table definitions.
- Column information with types and constraints.
- Primary keys, indexes, and foreign key relationships.

Example output:

```
# Database Schema

## Table of Contents

- [Table: users](#table-users)
- [Table: posts](#table-posts)

## Table: users {#table-users}

### Columns

| Column     | Type        | Attributes              | Description |
|------------|-------------|-------------------------|-------------|
| id         | integer     | PRIMARY KEY / NOT NULL  |             |
| username   | string(100) | NOT NULL                |             |
| email      | string(255) | NOT NULL                |             |
| created_at | datetime    | NOT NULL                |             |

### Primary Key
```

```
- Columns: `id`

### Indexes

| Name         | Columns  | Type   | Flags |
|--------------|----------|--------|-------|
| idx_username | username | INDEX  |       |
| idx_email    | email    | UNIQUE |       |
```

D2 Diagrams

The `D2SchemaTarget` generates diagrams in D2 format, a modern diagram scripting language.

```
use Derafu\ETL\Schema\Target\D2SchemaTarget;

$target = new D2SchemaTarget(
    detailLevel: D2SchemaTarget::DETAIL_FULL,
    direction: D2SchemaTarget::DIRECTION_RIGHT,
    layout: D2SchemaTarget::LAYOUT_DEFAULT,
    includeIndexes: true
);
$d2 = $target->applySchema($schema);

file_put_contents('schema.d2', $d2);
```

Options include:

- **Detail Level**: `DETAIL_FULL`, `DETAIL_KEYS_ONLY`, `DETAIL_MINIMAL`.
- **Direction**: `DIRECTION_UP`, `DIRECTION_DOWN`, `DIRECTION_LEFT`, `DIRECTION_RIGHT`.
- **Layout**: `LAYOUT_DEFAULT`, `LAYOUT_CLUSTERED`, `LAYOUT_HIERARCHICAL`.
- **Include Indexes**: Whether to include indexes in the diagram.

Example D2 output:

```
# Database Schema

direction: right

# Tables
users: {
  shape: sql_table
  id: integer NOT NULL pk
  username: string(100) NOT NULL column
  email: string(255) NOT NULL column
  created_at: datetime NOT NULL column
}
```

```
posts: {
  shape: sql_table
  id: integer NOT NULL pk
  user_id: integer NOT NULL fk
  title: string(255) NOT NULL column
  content: text column
  created_at: datetime NOT NULL column
}

# Relationships
posts -> users
```

Practical Applications

Documentation

Generate comprehensive documentation for your database:

```
use Derafu\ETL\Database\DatabaseManager;
use Derafu\ETL\Schema\Target\MarkdownSchemaTarget;

$manager = new DatabaseManager();
$database = $manager->connect('production.sqlite');

$target = new MarkdownSchemaTarget();
$docs = $target->applySchema($database->schema());

file_put_contents('database-schema.md', $docs);
```

Visual Modeling

Create visual diagrams for presentations or analysis:

```
use Derafu\ETL\Schema\Target\D2SchemaTarget;

// Filter to only show specific tables.
$target = new D2SchemaTarget(
    tableFilter: ['users', 'posts', 'comments']
);

$diagram = $target->applySchema($schema);
file_put_contents('entity-relationship.d2', $diagram);
```

Schema Migration

Export a schema definition to create a new database:

```
use Derafu\ETL\Schema\Target\SqliteSchemaTarget;

$target = new SqliteSchemaTarget();
$sql = $target->applySchema($schema);

file_put_contents('schema.sql', $sql);
```

Integration with ETL Pipeline

Schema visualization can be integrated with the ETL pipeline for comprehensive documentation:

```
use Derafu\ETL\Pipeline\Pipeline;
use Derafu\ETL\Schema\Target\MarkdownSchemaTarget;

// Run ETL pipeline.
$pipeline = new Pipeline();
$result = $pipeline
    ->extract('data.xlsx')
    ->transform()
    ->load('database.sqlite')
    ->execute();

// Document the resulting database.
$database = $result->target()->database();
$target = new MarkdownSchemaTarget();
$docs = $target->applySchema($database->schema());

file_put_contents('database-schema.md', $docs);
```

Last updated on 09/09/2026