DBML Syntax Guide

Custom ERD Generator (Strict Chen's Notation)

Documentation

January 15, 2026

Introduction

This document outlines the specific DBML (Database Markup Language) syntax required to generate Strict Chen's Notation ER Diagrams using the custom converter tool. The parser supports standard DBML features along with specific "syntax sugar" and comment-based directives to handle advanced ER concepts like Weak Entities, Participation constraints, and Derived attributes.

Entity Definitions

Strong Entity

A standard entity is defined using the `Table` keyword.

```
Table User {
  user_id int [pk]
  name varchar
}
```

Weak Entity

To define a Weak Entity (represented by a double rectangle), include a `Note: 'weak'` line inside the table block.

```
Table Dependent {
  Note: 'weak'
  dep_name varchar
}
```

Attribute Types

Primary Key

Primary keys are underlined in the diagram. Use `[pk]` or `[primary key]`.

```
  id int [pk]
```

Discriminator (Partial Key)

For weak entities, the partial key is dashed-underlined. Use `[note: 'discriminator']`.

```
  sequence_no int [note: 'discriminator']
```

Multivalued Attribute

Multivalued attributes are enclosed in curly braces `{ }` in the diagram.

```
  {phone_numbers} varchar  // Option 1: Brackets
  tags varchar [note: 'multivalued'] // Option 2: Note
```

Derived Attribute

Derived attributes are enclosed in parentheses `( )` and dotted-underlined.

```
  age() int                // Option 1: Parentheses
  (total_cost) decimal     // Option 2: Parentheses wrapper
  score int [note: 'derived'] // Option 3: Note
```

Composite Attribute

Use dot notation to indicate hierarchy. Child attributes are indented.

```
  address           // Parent
  address.street    // Child
  address.city      // Child
```

Relationships

Relationships are defined using the `Ref` keyword. The parser uses the operators to determine the visual arrows (Cardinality).

Cardinality Operators

- `<` : Arrow points to the **Left** entity (Left is "One").
- `>` : Arrow points to the **Right** entity (Right is "One").
- `-` : No arrows (Many-to-Many).

```
Ref: User.id < Ride.id     // One User, Many Rides
Ref: Car.id - Driver.id    // Many Cars, Many Drivers
```

Participation Constraints (Total vs Partial)

Participation is controlled via keywords in the comment section.

- **Total Participation:** Double line.
- **Partial Participation:** Single line.

Syntax: `// leftParticipation-rightParticipation`

```
// User (Total) -- requests -- (Total) Ride
Ref: User.id < Ride.id // total-total

// Driver (Partial) -- drives -- (Total) Car
Ref: Driver.id - Car.id // partial-total
```

Identifying Relationships

Used for Weak Entities. Renders as a **Double Diamond**. Add `Note: 'identifying'` to the relationship.

```
Ref: Trip.id < Stop.seq
Note: 'identifying' // total-total consists_of
```

Cardinality Labels (Min..Max)

To display explicit text labels (e.g., `1..1`, `0..*`), put them in brackets `[L, R]` inside the comment.

```
Ref: User.id < Ride.id // [1..1, 0..*] total-total requests
```

Complete Syntax Cheatsheet

Feature	Syntax Example
Strong Entity	`Table Name { ... }`
Weak Entity	`Note: 'weak'` inside Table
Primary Key	`col type [pk]`
Discriminator	`col type [note: 'discriminator']`
Multivalued	`{col_name}`
Derived	`col_name()` or `(col_name)`
Composite	`parent.child`
One-to-Many	`Ref: A < B`
Many-to-One	`Ref: A > B`
Many-to-Many	`Ref: A - B`
Total Participation	`// total-total` (in comment)
Identifying Rel.	`Note: 'identifying'` (after Ref)
Cardinality Text	`// [1..1, 0..*]` (in comment)
Label	Text after `//` or brackets

Full Example

```
Table User {
  id int [pk]
  name.first varchar
  name.last varchar
  {phones} varchar
  age() int
}

Table Order {
  id int [pk]
  date datetime
}

// User (1) places (N) Orders
// User must place at least 1 order (Total)
// Order must belong to 1 User (Total)
Ref: User.id < Order.id // [1..1, 1..*] total-total places
```