**May 26, 2025**



Table of content

[1. Database fild, Data type and Define Relationships 1](#_Toc32060)

[1.1 Database fild and Data type 1](#_Toc10685)

[Roles 1](#_Toc14011)

[Users 1](#_Toc23923)

[Projects 2](#_Toc12454)

[Tasks 2](#_Toc4010)

[Files 3](#_Toc7352)

[Comments 4](#_Toc7020)

[UserProjects 5](#_Toc295)

[1.2 Define Relationships 5](#_Toc28707)

[2. ER DIAGRAM 9](#_Toc12692)

[3 Key Functional Dependencies 9](#_Toc13639)

Table of table

[Table 1 roles 1](#_Toc25936)

[Table 2 users 1](#_Toc13662)

[Table 3 projects 2](#_Toc8952)

[Table 4 tasks 3](#_Toc21938)

[Table 5 files 3](#_Toc8112)

[Table 6 Comments 4](#_Toc24888)

[Table 7 userprojects 5](#_Toc483)

# Database fild, Data type and Define Relationships

## 1.1 Database fild and Data type

### Roles

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| roleKey | INT | Primary Key, Auto-Increment | Unique identifier for each role. |
| role\_name | VARCHAR(50) | Unique, Not Null | Name of the role (e.g., Intern, Mentor). |
| description | TEXT | None | Optional description of the role. |

Table 1 roles

### Users

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| userKey | INT | Primary Key, Auto-Increment | Unique identifier for each user. |
| email | VARCHAR(255) | Unique, Not Null | User’s email for authentication. |
| password\_hash | VARCHAR(255) | Not Null | Hashed password (bcrypt, SRS 5.3). |
| first\_name | VARCHAR(50) | None | User’s first name (optional). |
| last\_name | VARCHAR(50) | None | User’s last name (optional). |
| roleKey | INT | Foreign Key (Roles.roleKey), Not Null | Links to user’s role. |
| created\_at | TIMESTAMP | Not Null | Timestamp of user creation. |
| updated\_at | TIMESTAMP | None | Timestamp of last update (optional). |
| is\_active | BOOLEAN | Default: true | Indicates if user is active. |

Table 2 users

### Projects

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| projectKey | INT | Primary Key, Auto-Increment | Unique identifier for each project. |
| name | VARCHAR(100) | Not Null | Project name. |
| description | TEXT | None | Optional project description. |
| created\_at | TIMESTAMP | Not Null | Timestamp of project creation. |
| updated\_at | TIMESTAMP | None | Timestamp of last update (optional). |
| creatorKey | INT | Foreign Key (Users.userKey), Not Null | Links to user who created the project. |

Table 3 projects

### Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| taskKey | INT | Primary Key, Auto-Increment | Unique identifier for each task. |
| projectKey | INT | Foreign Key (Projects.projectKey), Not Null | Links to the project. |
| title | VARCHAR(100) | Not Null | Task title. |
| description | TEXT | None | Optional task description. |
| status | VARCHAR(20) | Not Null, Enum: [To Do, In Progress, In Review, Done] | Task status (SRS 4.2, REQ-5). |
| priority | VARCHAR(20) | Not Null, Enum: [High, Medium, Low] | Task priority (SRS 4.2, REQ-5). |
| created\_at | TIMESTAMP | Not Null | Timestamp of task creation. |
| updated\_at | TIMESTAMP | None | Timestamp of last update (optional) |
| deadline | TIMESTAMP | None | Optional task deadline (SRS 4.2, REQ-7). |
| creatorKey | INT | Foreign Key (Users.userKey), Not Null | Links to user who created the task. |
|  | | | |

Table 4 tasks

### TaskAssignees

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| TaskKey | |  |  | | --- | --- | |  | INT | | Primary Key, Foreign Key (Tasks.TaskKey), Not Null | Links to task. |
| UserKey | INT | Primary Key, Foreign Key (Users.UserKey), Not Null | Links to assigned user. |

Table 5 TaskAssignees

### Files

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| fileKey | INT | Primary Key, Auto-Increment | Unique identifier for each file. |
| projectKey | INT | Foreign Key (Projects.projectKey), Not Null | Links to the project. |
| file\_name | VARCHAR(255) | Not Null | Name of the file. |
| file\_type | VARCHAR(50) | Not Null, Enum: [Report, Code, Presentation, Video, Quiz, PDF] | File type (SRS 4.3, BRD 5). |
| version | INTEGER | Not Null | File version number (SRS 4.3, REQ-9). |
| file\_path | VARCHAR(255) | Not Null | Storage path of the file. |
| size | INT | Not Null, Check (size <= 10485760) | File size in bytes (max 10MB). |
| uploaderKey | INT | Foreign Key (Users.userKey), Not Null | Links to user who uploaded the file. |
| uploaded\_at | TIMESTAMP | Not Null | Timestamp of file upload. |
| is\_current | BOOLEAN | Default: true | Indicates if this is the current version. |
|  | | | |

Table 6 files

Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| commentKey | INT | Primary Key, Auto-Increment | Unique identifier for each comment. |
| taskKey | INT | Foreign Key (Tasks.taskKey), Nullable | Links to task (optional, XOR with fileKey). |
| fileKey | INT | Foreign Key (Files.fileKey), Nullable | Links to file (optional, XOR with taskKey). |
| parent\_commentKey | INT | Foreign Key (Comments.commentKey), Nullable | Links to parent comment (threading). |
| content | TEXT | Not Null | Comment text (SRS 4.4, REQ-12) |
| creatorKey | INT | Not Null | Links to user who created the comment. |
| created\_at | TIMESTAMP | Not Null | Timestamp of comment creation. |
| mentions | TEXT | None | Mentions (@username) in comment (REQ-13) |

Table 7 Comments

UserProjects

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Constraints | Description |
| userKey | INT | Primary Key, Foreign Key (Users.userKey), Not Null | Links to user. |
| projectKey | INT | Primary Key, Foreign Key (Projects.projectKey), Not Null | Links to project. |

Table 8 userprojects

## 1.2 Define Relationships

**1. Users Roles**

* **Source Entity**: Users
* **Target Entity**: Roles
* **Cardinality**: Many-to-One
* **Connecting Attribute**: roleKey (Foreign Key in Users references roleKey in Roles)
* **Constraint**: roleKey in Users is not null, ensuring every user has a role (SRS 4.1, REQ-2).
* **Description**: Each user is assigned one role (e.g., Intern, Mentor, Admin), while a role can be assigned to multiple users. This supports role-based access control (SRS 4.1).

**2. Projects Users (creator)**

* **Source Entity**: Projects
* **Target Entity**: Users
* **Cardinality**: Many-to-One
* Connecting Attribute: creatorKey (Foreign Key in Projects references userKey in Users)
* **Constraint**: creatorKey is not null, indicating every project must have a creator (BRD 3.2).
* **Description**: Each project is created by one user, while a user can create multiple projects. This tracks project ownership (SRS 2.2).

**3. Tasks Projects**

* **Source Entity**: Tasks
* **Target Entity**: Projects
* **Cardinality**: Many-to-One
* **Connecting Attribute**: projectKey (Foreign Key in Tasks references projectKey in Projects)
* **Constraint**: projectKey is not null, ensuring every task belongs to a project (SRS 4.2).
* **Description**: Each task is associated with one project, while a project can contain multiple tasks. This supports Kanban task organization (SRS 4.2).
* **Performance Note**: The updated ERD suggests indexing projectKey to optimize Kanban board loading (<2-second load time, SRS 5.1).

**4. Tasks Users (assignees,Via TaskAssignees)**

* **Source Entity**: Tasks
* **Target Entity**: Users
* **Cardinality**: Many-to-Many
* **Connecting Table**:TaskAssignees
* **Connecting Attribute**:

TaskKey (Foreign Key in TaskAssignees references TaskKey in Tasks)

UserKey (Foreign Key in TaskAssignees references UserKey in Users)

* Constraint: Both TaskKey and UserKey are not null and form a composite primary key
* **Description**: A task can be assigned to multiple user, and a user can be assigned multiple tasks. This supports task assignment flexibility beyond the original requirement (SRS 4.2,REQ-7).

5. Tasks Users (creatorKey)

* **Source Entity**: Tasks
* **Target Entity**: Users
* **Cardinality**: Many-to-One
* **Connecting Attribute**: creatorKey (Foreign Key in Tasks references userKey in Users)
* **Constraint**: creatorKey is not null, ensuring every task has a creator (SRS 4.2).
* **Description**: Each task is created by one user, while a user can create multiple tasks. This supports task auditing (SRS 5.3).

**6. Files Projects**

* **Source Entity**: Files
* **Target Entity**: Projects
* **Cardinality**: Many-to-One
* Connecting Attribute: projectKey (Foreign Key in Files references projectKey in Projects)
* **Constraint**: projectKey is not null, ensuring every file belongs to a project (SRS 4.3).
* **Description**: Each file is associated with one project, while a project can contain multiple files. This supports document management (SRS 4.3).

**7. Files Users (uploaded\_by)**

* **Source Entity**: Files
* **Target Entity**: Users
* **Cardinality**: Many-to-One
* Connecting Attribute: uploaderKey (Foreign Key in Files references userKey in Users)
* **Constraint**: uploaderKey is not null, ensuring every file has an uploader (SRS 4.3, REQ-9).
* **Description**: Each file is uploaded by one user, while a user can upload multiple files. This tracks document ownership (SRS 4.3).

**8. Comments Tasks**

* **Source Entity**: Comments
* **Target Entity**: Tasks
* **Cardinality**: Many-to-One
* **Connecting Attribute**: taskKey (Foreign Key in Comments references taskKey in Tasks)
* **Constraint**: taskKey is nullable, with an XOR constraint with fileKey (one must be null) (SRS 4.4).
* **Description**: A comment can be linked to one task, while a task can have multiple comments. This supports task-specific discussions (SRS 4.4).
* **Performance Note**: The updated ERD suggests indexing taskKey to optimize comment retrieval (SRS 5.1).

**9. Comments Files**

* **Source Entity**: Comments
* **Target Entity**: Files
* **Cardinality**: Many-to-One
* **Connecting Attribute**: fileKey (Foreign Key in Comments references fileKey in Files)
* **Constraint**: fileKey is nullable, with an XOR constraint with taskKey (one must be null) (SRS 4.4).
* **Description**: A comment can be linked to one file, while a file can have multiple comments. This supports file-specific discussions (SRS 4.4).

**10. Comments Comments (parent)**

* **Source Entity**: Comments (child)
* **Target Entity**: Comments (parent)
* **Cardinality**: Many-to-One
* **Connecting Attribute**: parent\_commentKey (Foreign Key in Comments references commentKey in Comments)
* **Constraint**: parent\_commentKey is nullable, allowing top-level comments (SRS 4.4, REQ-12).
* **Description**: A comment can be a reply to one parent comment, while a parent comment can have multiple replies. This enables threaded discussions (SRS 4.4).

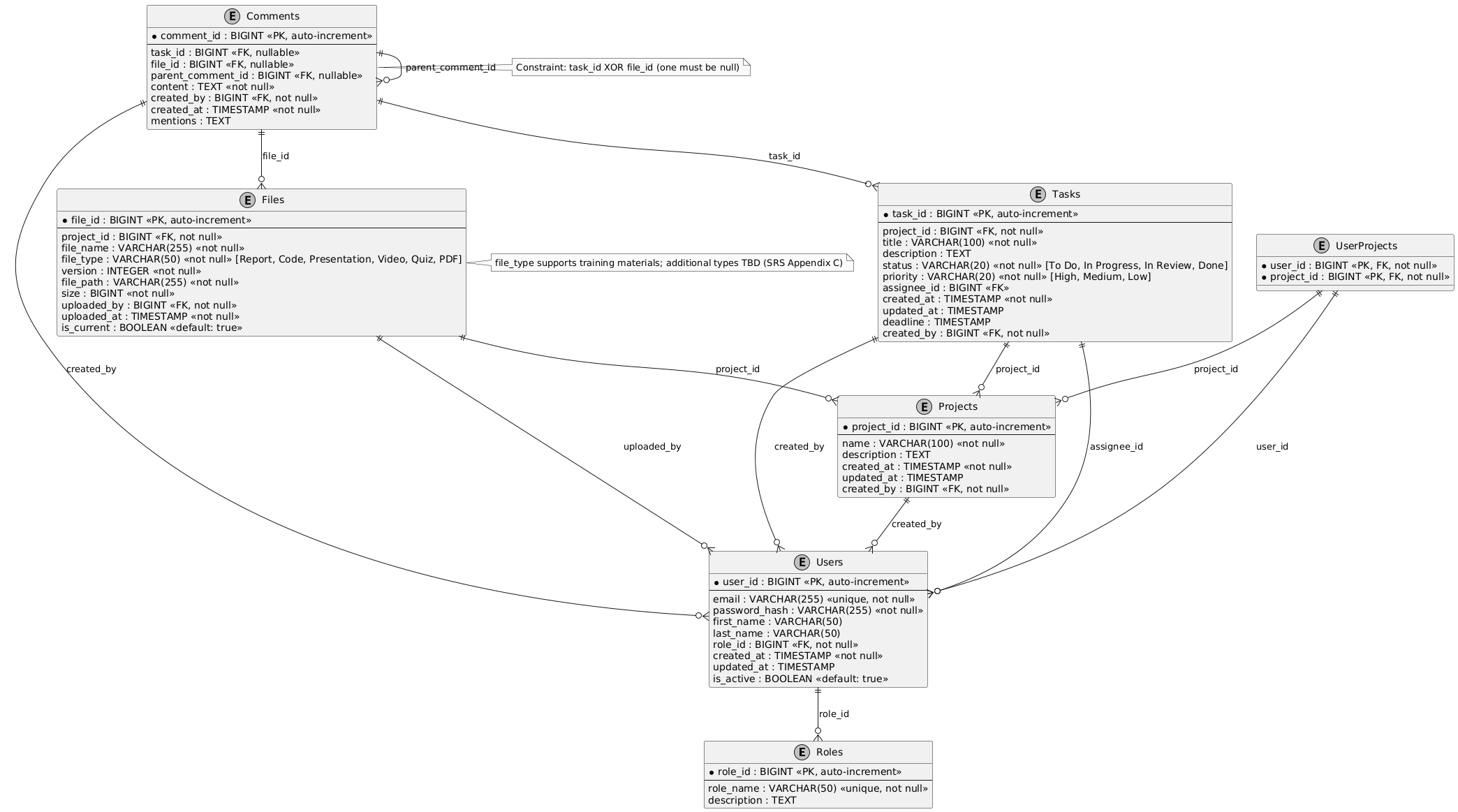
**11. Comments Users (creator)**

* **Source Entity**: Comments
* **Target Entity**: Users
* **Cardinality**: Many-to-One
* **Connecting Attribute**: creatorKey (Foreign Key in Comments references userKey in Users)
* **Constraint**: creatorKey is not null, ensuring every comment has an author (SRS 4.4).
* **Description**: Each comment is created by one user, while a user can create multiple comments. This tracks authorship (SRS 4.4).

**12. Users Projects (via UserProjects)**

* **Source Entity**: Users
* **Target Entity**: Projects
* **Cardinality**: Many-to-Many
* **Connecting Table**: UserProjects
* **Connecting Attributes**:
  + userKey (Foreign Key in UserProjects references userKey in Users)
  + projectKey (Foreign Key in UserProjects references projectKey in Projects)
* **Constraint**: Both userKey and projectKey are not null and form a composite primary key (BRD 3.2).
* **Description**: A user can be part of multiple projects, and a project can have multiple users. This supports team formation (BRD 3.2)

## ER DIAGRAM



## 3 Key Functional Dependencies

#### 1. ****Roles****

* **Primary Key**: roleKey
* **Functional Dependencies**:
  + roleKey → role\_name, description
    - Explanation: The roleKey (primary key) uniquely determines the role\_name (unique and not null) and description (optional). This reflects that each role has a distinct name (SRS 4.1, REQ-3).
* **Notes**: No additional dependencies due to the simplicity of the entity.

#### 2. ****Users****

* **Primary Key**: userKey
* **Functional Dependencies**:
  + userKey → email, password\_hash, first\_name, last\_name, roleKey, created\_at, updated\_at, is\_active
    - Explanation: The userKey (primary key) uniquely determines all other attributes. email is unique and not null (SRS 4.1, REQ-1), password\_hash is not null (SRS 5.3), and roleKey links to Roles (SRS 4.1, REQ-2). created\_at, updated\_at, and is\_active are user-specific audit and status fields.
  + email → userKey (partial dependency)
    - Explanation: email is unique, implying it can determine userKey, though it’s not the primary key. This supports authentication (SRS 4.1, REQ-1).
* **Notes**: The uniqueness of email suggests a candidate key, but userKey is the chosen primary key for scalability (BRD 6).

#### 3. ****Projects****

* **Primary Key**: projectKey
* **Functional Dependencies**:
  + projectKey → name, description, created\_at, updated\_at, creatorKey
    - Explanation: The projectKey (primary key) uniquely determines all attributes. name is not null (BRD 3.2), creatorKey links to Users (SRS 2.2), and created\_at, updated\_at support auditing (SRS 5.3).
  + creatorKey, name → projectKey (partial dependency, theoretical)
    - Explanation: A user (creatorKey) creating a project with a specific name could theoretically determine projectKey if names are unique per creator, but this is not enforced (no unique constraint on name).
* **Notes**: No unique constraint on name alone, so the partial dependency is not strict.

#### 4. ****Tasks****

* **Primary Key**: taskKey
* **Functional Dependencies**:
  + taskKey → projectKey, title, description, status, priority, UserKey, created\_at, updated\_at, deadline, creatorKey
    - Explanation: The taskKey (primary key) uniquely determines all attributes. projectKey links to Projects (SRS 4.2), UserKey and creatorKey link to Users (SRS 4.2, REQ-7), and status, priority, deadline define task state (SRS 4.2, REQ-5).
  + projectKey, title → taskKey (partial dependency, theoretical)
    - Explanation: A task’s projectKey and title could theoretically determine taskKey if unique per project, but no unique constraint exists on title alone.
* **Notes**: The ERD suggests indexing projectKey for performance (SRS 5.1), but this does not affect FDs.

#### 5. ****Files****

* **Primary Key**: fileKey
* **Functional Dependencies**:
  + fileKey → projectKey, file\_name, file\_type, version, file\_path, size, uploaded\_by, uploaded\_at, is\_current
    - Explanation: The fileKey (primary key) uniquely determines all attributes. projectKey links to Projects (SRS 4.3), uploaded\_by links to Users (SRS 4.3, REQ-9), size has a max limit of 10MB (updated ERD), and version, is\_current support versioning (SRS 4.3, REQ-10).
  + projectKey, file\_name, version → fileKey (partial dependency)
    - Explanation: A file’s projectKey, file\_name, and version combination could determine fileKey, as versions are unique per file name within a project (implied by SRS 4.3, REQ-9).
* **Notes**: The check (size <= 10485760) constraint is a business rule, not a functional dependency.

#### 6. ****Comments****

* **Primary Key**: commentKey
* **Functional Dependencies**:
  + commentKey → taskKey, fileKey, parent\_commentKey, content, creatorKey, created\_at, mentions
    - Explanation: The commentKey (primary key) uniquely determines all attributes. taskKey or fileKey (XOR constraint) links to Tasks or Files (SRS 4.4), parent\_commentKey supports threading (SRS 4.4, REQ-12), and creatorKey links to Users (SRS 4.4).
  + taskKey, content → commentKey (partial dependency, if taskKey is not null)
    - Explanation: For comments on tasks, taskKey and content could theoretically determine commentKey if unique per task, but no unique constraint exists.
  + fileKey, content → commentKey (partial dependency, if fileKey is not null)
    - Explanation: Similarly, for comments on files, fileKey and content could determine commentKey if unique per file.
* **Notes**: The XOR constraint (taskKey or fileKey) is a business rule, not a functional dependency. Indexing taskKey is suggested for performance (SRS 5.1).

#### 7. ****UserProjects****

* **Composite Primary Key**: userKey, projectKey
* **Functional Dependencies**:
  + userKey, projectKey → (no other attributes)
    - Explanation: The composite primary key (userKey, projectKey) uniquely identifies each record in the junction table. There are no additional attributes, as it solely models the Many-to-Many relationship (BRD 3.2).
* **Notes**: This is a pure associative entity with no non-key attributes.