Legend:
- 1 .. INF (bold line)
- 1 .. 1 (line with fork/loop end)
- 0 .. 1 (arrow)
- 0 .. INF (plain line)

Game — attributes: name, gameId, releaseDate, description

Company — attributes: companyId, name, companyType

News — attributes: newsId, text

User — attributes: points, isModerator, password, username

Media — attributes: mediaId, altText, link, isVideo

Submission — attributes: approved, submissionId, submitDate, submissionType, rawData

Relationships:
- Game — worked_on — Company ("worked on" / "worked on by")
- Game — news_on — News ("news about" / "has news")
- Game — submits — User
- Company — submits — User
- News — submits — User
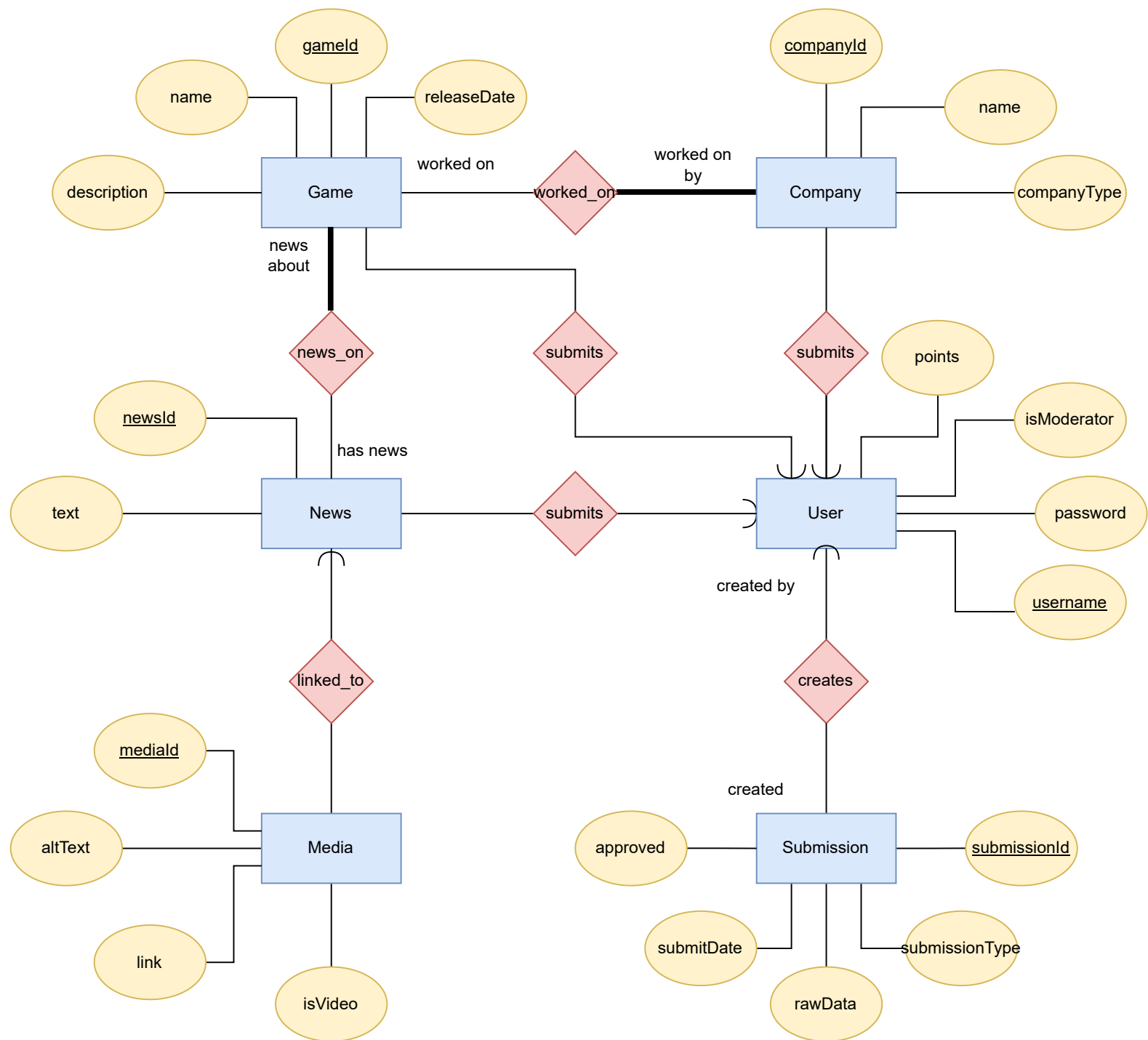- News — linked_to — Media
- User — creates — Submission ("created by" / "created")

Assumptions Entities:

- Game: Identified by a unique gameId, it holds all of the information about the game. Note that dates are represented in integers because of how our initial dataset is formatted.
- Company: Identified by a unique companyId, it holds all of the information about the company, including its "Type", like a publishing company or the developers.
- News: Identified by a unique newsId, it holds the text of the "news" (more similar in length and substance of a social media post).
- Media: Identified by a unique mediaId, it holds links to large types of media, such as images and videos, and alt text that may be attached to news posts.
- User: Identified by a unique username, it stores the password, "points" (to use as incentive to post news), and moderator status of the user. Moderators have greater access to the website, specifically allowing them to regulate submissions.
- Submissions: Identified by a unique submissionsId, it holds all of the data (submitted by the user on the website) to create a Game, News (including media), or Company entity. All proposed submissions will be stored here until a moderator (see above) approves or rejects it. If approved, the submission will be parsed and turned into the respective object. Approved or not, the submission will then be deleted to save space.

Assumptions Relations:

- Media <Linked to> News - many to one: We plan to have news posts hold larger forms of media (like videos and images) in a separate entity, with said Media entity being liked to that exact news post. News posts can have multiple videos, images, etc. linked to it, but each piece of media is "bound" to that post. Uploading the exact same image in different news posts should not matter since the media object encapsulating it would be different for different posts. **Deleting posts should delete the media it is linked to.**
- News <news on> Game - many to many: A game can have multiple news posts that relate to it, or none at all if it is obscure or new. A news post can be talking about one or more games, but never none (if a news article has no relation to any game, it should not be included on a game news website). We assume games with news will not be deleted, for archiving purposes.
- Company <worked on> Game - many to many: A Game must be associated with/ worked on by at least one Company (someone had to create it), and it can be worked on by multiple companies, such as having separate development teams and publishers. Companies can be associated with multiple games, but can also have no games it is associated with (such as new companies).
- User <submits> News/ Game/ Company - one to many: A User can submit new entries into each table through submissions, and if that entity is approved the User who created it keeps their name on it. A User can submit many entities or none, and an entity must be created by some User, which would include moderators and us. We assume entities cannot be submitted by multiple Users and that User profiles will not be deleted for archival purposes (even so, having a "deleted profile" User could be useful).
- User <creates> Submission - one to many: As stated above, users can create submissions. We assume Users can create many or no Submissions and that Submissions must be created by exactly one user.

Relational Schemas

- Game(gameId:INT [PK], name:VARCHAR(100), description:VARCHAR(2000), releaseDate:INT, submittedBy:VARCHAR(30) [FK to User.username])
- Company(companyId:INT [PK], name:VARCHAR(100), companyType:VARCHAR(30), submittedBy:VARCHAR(30) [FK to User.username])
- News(newsId:INT [PK], text:VARCHAR(2000), submittedBy:VARCHAR(30) [FK to User.username])
- Media(mediaId:INT [PK], altText:VARCHAR(1000), link:VARCHAR(1000), isVideo:BOOLEAN, linkedTo:INT [FK to News.newsId])
- User(username:VARCHAR(30) [PK], password:VARCHAR(30), isModerator:BOOLEAN, points:INT)
- Submission(submissionId:INT [PK], submissionType:VARCHAR(10), rawData:VARCHAR(4000), aproved:BOOLEAN, submitDate:INT, createdBy:VARCHAR(30) [FK to User.username])
- WorkedOn(game:INT [PK, FK to Game.gameId], company:INT [PK, FK to Company.companyId])
- NewsOn(game:INT [PK, FK to Game.gameId], news:INT [PK, FK to news.newsId])