COMPSCI 3DB3

Assignment 3 Report

Mingzhe Wang

McMaster University

October 27, 2021

Part one

Subscriber

subscriber(<u>username</u>: VARCHAR(255), <u>number</u>: INT, email: VARCHAR(255), hash: BINARY(512), salt: BINARY(512)).

- The username and number should be the primary key.
- VARCHAR(255) is selected as the type for email, because an email address consists of variable length character.
- BINARY(512) is the type of hash and salt value, because they are stored in fixed-length binary strings.
- Note: because db2 seems not to support BINARY(512) type, in the code, we use VARCHAR(512) instead.

Friend_Of

friend_of(<u>fname</u>: VARCHAR(255), <u>fnumber</u>: INT, <u>tname</u>: VARCHAR(255), <u>tnumber</u>: INT).

- fname, fnumber, tname, tnumber should all be not null, primary key, and reference from subscriber(username) and subscriber(number).

Review

review(<u>uname</u>: VARCHAR(255), <u>unumber</u>: INT, <u>revision</u>: INT, <u>[Forfilm's primary key(s)]</u>: [corresponding type(s)], score: INT, timestamp: TIMESTAMP).

- If we assume the score is in the range $[0, 10]$, the we should mark this as a constraint and check it.
- We may need to update the ForFilm part later, as this information haven't been constructed.

VideoReview

video_review(uname: VARCHAR(255), unumber: INT, revision: INT, [Forfilm's primary key(s)]: [corresponding type(s)], video: BLOB).

- We use ER method to construct ISA, because a review could have both kinds of reviews (video and text).
- The uname, unumber, revision, and [Forfilm's primary key(s)] are foreign keys pointing to subscriber's.

TextReview

text_review(uname: VARCHAR(255), unumber: INT, revision: INT, [Forfilm's primary key(s)]: [corresponding type(s)], description: CLOB).

- Similar to videoreview.

Reaction

reaction(id: INT, byuname: VARCHAR(255), byunumber: INT, title: VARCHAR(255), content: CLOB).

- We use ER method here, because the ThreadR entity could have a relation with Reaction.
- We store the by subscriber's username and number here as foreign keys referencing subscriber(username, number), as reaction only participates exactly once in the"by" relation. It should also has a NOT NULL constraint.

ThreadR

reaction(id: INT, onid: INT)

- We need id for ISA relation and onid for On_Ration relation. Both id and onid should reference to the reaction(id).

ReviewR

review_r(id: INT, [Forfilm's primary key(s)]: [corresponding type(s)])

- We store the On_Review relation as foreign keys referencing to Review's primary keys, because the ReviewR entity participates exactly once in this relation. It should also has a NOT NULL constraint.

Part two

person

person(id: INT, name: VARCHAR(255), $\text{birthdate}_{\text{optional}}$: DATE)

film

film(<u>title</u>: VARCHAR(255), <u>year</u>: INT, <u>creator</u>: INT, duration: INTERVAL, budget: DECIMAL(50,2))

- We treat creator as a strict one-to-many relationship between film and person, so we store it in film.
- foreign key constrain: creator references to person(id).

film_info (view)

(view) film_info(<u>title</u>: VARCHAR(255), <u>year</u>: INT, <u>creator</u>: INT, duration: INTERVAL, budget: DECIMAL(50,2))

- this is a view, so we should use "select" sytax later.

role_as

role_as(<u>pid</u>: INT, <u>ftitle</u>: VARCHAR(255), <u>fyear</u>: INT, <u>fcreator</u>: INT, role: VARCHAR(20))

- we assume play_role_as as a many-to-many relation between person and film. The relationship should has an attribute role to denote the person's current role in the file. For cases where a person can perform many roles in a film, just add multiple instances with different role attributes.

constraint

this can be implemented using multi-table constraints by:
CREATE TABLE film(
title VARCHAR(255) ...,
year INT ...,
creater INT ...,
CHECK (creater IN (SELECT r.id FROM roleas r WHERE (r.ftitle = title AND r.fyear = year AND r.fcreator = creator) AND r.role = "director"));

Note

By now, we know what are the primary keys in film, so we need to replace every occurrence of "[Forfilm's primary key(s)]: [corresponding type(s)]", that is:

film

<u>title</u>: VARCHAR(255), <u>year</u>: INT, <u>creator</u>: INT,

Review

review(<u>uname</u>: VARCHAR(255), <u>unumber</u>: INT, <u>revision</u>: INT, <u>ftitle</u>: VARCHAR(255), <u>fyear</u>: INT, <u>fcreator</u>: INT, score: INT, timestamp: TIMESTAMP).

VideoReview

video_review(uname: VARCHAR(255), unumber: INT, revision: INT, ftitle: VARCHAR(255), fyear: INT, fcreator: INT, video: BLOB).

TextReview

text_review(uname: VARCHAR(255), unumber: INT, revision: INT, ftitle: VARCHAR(255), fyear: INT, fcreator: INT, description: CLOB).

ReviewR

review_r(id: INT, uname: VARCHAR(255), unumber: INT, revision: INT, ftitle: VARCHAR(255), fyear: INT, fcreator: INT)