USER(<u>username</u>, password, email, last_access_date, creation_date, first_name, last_name)
//None of the above values can be null

VIDEO_GAME(<u>vg_id</u>, title, esrb_rating)
//ersb_rating cannot be null, there is a set domain, title can not be null, title is unique

COLLECTION(<u>collection_id</u>, *<u>username</u>*, name)
//name cannot be null
//username foreign key refers to username in USER, not null

GENRE(<u>name</u>)

PLATFORM(<u>platform_id</u>, name, os)
//name cannot be null

CREATOR(<u>creator_id</u>, name)
//name cannot be null

FRIENDS_WITH(*<u>uid</u>*, *<u>fid</u>*)
//uid foreign key refers to username in USER, not null
//fid foreign key refers to username in USER, not null

PLAYS(<u>date_played</u>, *<u>username</u>*, *<u>vg_id</u>*, total_playtime)
//username foreign key refers to username in USER, not null
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null
//total_playtime cannot be null, defaults to zero

RATES(*<u>username</u>*, *<u>vg_id</u>*, rating)
//rating cannot be NULL
//username foreign key refers to username in USER, not null
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null

USER_HAS/OWNS_PLATFORM(*<u>username</u>*, *<u>platform_id</u>*)
//username foreign key refers to username in USER, not null
//platform_id foreign key refers to platform_id in PLATFORMS, not null

VIDEO_GAME_ON/HAS_PLATFORM(*<u>vg_id</u>*, *<u>platform_id</u>*, release_date, price)
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null
//platform_id foreign key refers to platform_id in PLATFORMS, not null

PUBLISHES(*creator_id*, *vg_id*)
//creator_id foreign key refers to creator_id in CREATOR, not null
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null

DEVELOPS(*creator_id*, *vg_id*)
//creator_id foreign key refers to creator_id in CREATOR, not null
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null

HAS_GENRE(*vg_id*, *genre_name*)
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null
//genere_name foreign key refers to name in GENRE, not null

COLLECTION_CONTAINS(*collection_id*, *vg_id*)
//collection_id foreign key refers to collection_id in COLLECTION, not null
//vg_id foreign key refers to vg_id in VIDEO_GAME, not null