Solution:

-- Stores the User entity set information.
CREATE TABLE User (  
    userid INTEGER NOT NULL,
    account VARCHAR(30) NOT NULL, -- username
    password VARCHAR(60) NOT NULL,
    email VARCHAR(60) NOT NULL,
    jobtitle VARCHAR(60) NOT NULL,
    name_first VARCHAR(60) NOT NULL,
    name_middle VARCHAR(60),
    name_last VARCHAR(60) NOT NULL,
    user_since DATETIME NOT NULL,
) PRIMARY KEY (userid)
);

-- Stores the the degree attribute of the User entity set information.
CREATE TABLE Degree (  
    userid INTEGER NOT NULL,
    level VARCHAR(30) NOT NULL,
    year INTEGER NOT NULL,
    major VARCHAR(60) NOT NULL,
    school VARCHAR(100) NOT NULL,
) PRIMARY KEY (userid, level, year, major, school), -- may vary
FOREIGN KEY (userid) REFERENCES User (userid) ON DELETE CASCADE
);

-- Stores the Publisher entity set information.
CREATE TABLE Publisher (  
    publisherid INTEGER NOT NULL,
    name VARCHAR(60) NOT NULL,
    addr_street VARCHAR(100) NOT NULL,
    addr_city VARCHAR(60) NOT NULL,
    addr_state CHAR(2) NOT NULL,
    addr_zipcode CHAR(5) NOT NULL,
    website VARCHAR(100) NOT NULL,
    phone_area CHAR(3) NOT NULL,
    phone_number VARCHAR(20) NOT NULL,
    phone_extension VARCHAR(10),
) PRIMARY KEY (publisherid)
);
-- Stores the Poster entity set and the works_for relationship information.
-- Since works_for relationship is 1:N relationship between Publisher and Poster,
-- rather than having a separate relationship table,
-- we put the relationship information to the N side (Poster) of the relationship.
CREATE TABLE Poster (  
    userid INTEGER NOT NULL,  
    poster_since DATETIME NOT NULL,  
    works_since DATE,  
    publisherid INTEGER,  
    PRIMARY KEY (userid),  
    FOREIGN KEY (userid) REFERENCES User (userid) ON DELETE CASCADE,  
    FOREIGN KEY (publisherid) REFERENCES Publisher (publisherid) ON DELETE SET NULL
);  

-- Stores the Detector entity set information.
CREATE TABLE Detector (  
    userid INTEGER NOT NULL,  
    detector_since DATETIME NOT NULL,  
    PRIMARY KEY (userid),  
    FOREIGN KEY (userid) REFERENCES User (userid) ON DELETE CASCADE
);  

-- Stores the Article entity set information.
CREATE TABLE Article (  
    userid INTEGER NOT NULL,  
    articleid INTEGER NOT NULL,  
    title VARCHAR(100) NOT NULL,  
    content VARCHAR(1000) NOT NULL,  
    posting_datetime DATETIME NOT NULL,  
    popularity VARCHAR(20) NOT NULL,  
    quality VARCHAR(20) NOT NULL,  
    PRIMARY KEY (userid, articleid),  
    FOREIGN KEY (userid) REFERENCES Poster (userid) ON DELETE CASCADE
);  

-- Stores the topics attribute of Article entity set information.
CREATE TABLE Topic (  
    userid INTEGER NOT NULL,  
    articleid INTEGER NOT NULL
);
topic VARCHAR(60) NOT NULL,
    PRIMARY KEY (userid, articleid, topic),
    FOREIGN KEY (userid, articleid) REFERENCES Article (userid, articleid) ON
DEDELETE CASCADE
    );

-- Stores the reports relationship information.
-- Since this is an M:N relationship between Detector and Article,
-- we need to have a separate relation.
CREATE TABLE Reports (userid INTEGER NOT NULL,
    article_userid INTEGER NOT NULL,
    article_articleid INTEGER NOT NULL,
    reason VARCHAR(60) NOT NULL,
    PRIMARY KEY (userid, article_userid, article_articleid),
    FOREIGN KEY (userid) REFERENCES Detector (userid) ON DELETE
    CASCADE,
    FOREIGN KEY (article_userid, article_articleid) REFERENCES Article (userid,
    articleid) ON DELETE CASCADE
    );

-- Stores the recommendation relationship information.
-- Since this is a M:N:P relationship between User and Article,
-- we need to have a separate relation.
CREATE TABLE Recommendation (from_userid INTEGER NOT NULL,
    to_userid INTEGER NOT NULL,
    article_userid INTEGER NOT NULL,
    article_articleid INTEGER NOT NULL,
    PRIMARY KEY (from_userid, to_userid, article_userid, article_articleid),
    FOREIGN KEY (from_userid) REFERENCES User (userid) ON DELETE
    CASCADE,
    FOREIGN KEY (to_userid) REFERENCES User (userid) ON DELETE CASCADE,
    FOREIGN KEY (article_userid, article_articleid) REFERENCES Article (userid,
    articleid) ON DELETE CASCADE
    );

-- Stores the has_read relationship information.
-- Since this is M:N relationship between User and Article,
-- we need to have a separate relation.
CREATE TABLE Has_read (userid INTEGER NOT NULL,
    article_userid INTEGER NOT NULL,
article_articleid INTEGER NOT NULL,
like_score INTEGER NOT NULL,
PRIMARY KEY (userid, article_userid, article_articleid),
FOREIGN KEY (userid) REFERENCES User (userid) ON DELETE CASCADE,
FOREIGN KEY (article_userid, article_articleid) REFERENCES Article (userid, articleid) ON DELETE CASCADE
);

-- Stores the refers_to relationship information.
-- Since this is M:N relationship between Article and Article,
-- we need to have a separate relation. Citing is the referrer.
-- And source is the referee.
CREATE TABLE Refers_to (  
citing_userid INTEGER NOT NULL,
citing_articleid INTEGER NOT NULL,
source_userid INTEGER NOT NULL,
source_articleid INTEGER NOT NULL,
PRIMARY KEY (citing_userid, citing_articleid, source_userid, source_articleid),
FOREIGN KEY (citing_userid, citing_articleid) REFERENCES Article (userid, articleid) ON DELETE CASCADE,
FOREIGN KEY (source_userid, source_articleid) REFERENCES Article (userid, articleid) ON DELETE CASCADE
);