CSCI-2020 Database Fundamentals

Erin Lorelle

Final Exam Project, Fall 2017

December 2, 2017

Credential Manager Database
for

Renkcub Software Solutions

Table of Contents

Phase #1: Conceptual Design

Purpose

"Credential Manager" for Renkcub Software Solutions to store user credentials. The database will also track previously used passwords. The below ERD is based on the specific business needs requested. Note: All and any assumptions are noted on the diagram.

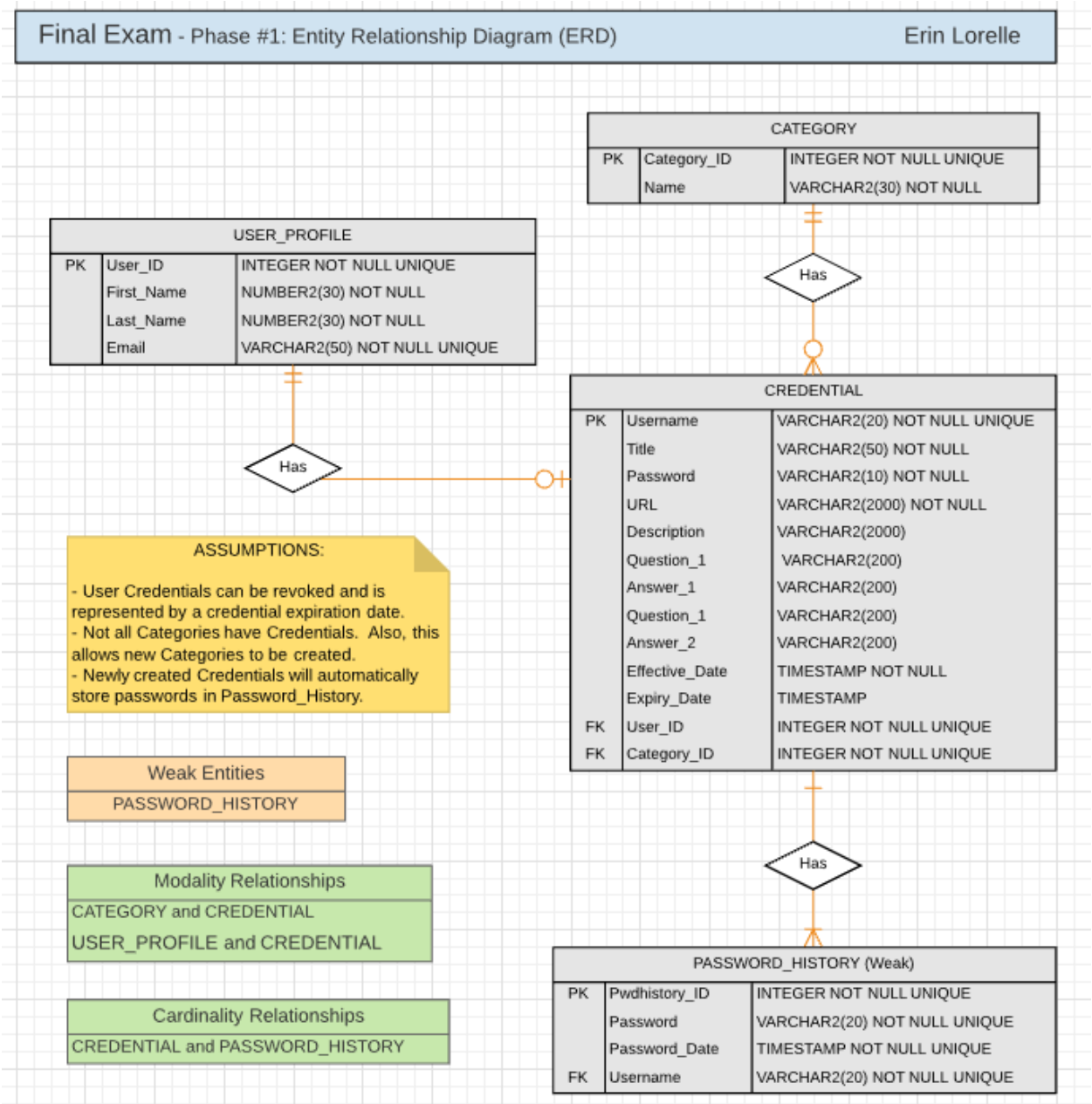

Table of Contents>>

Phase #2: Logical Design

Normalized through 3NF

ATTRIBUTES/FIELDS LIST

Category
Category ID
Category Name
Credential
Credential Title
Username
Password
Password Date
URL
Credential Description
Security Questions & Answers
Credential Effective Date
Credential Expiration Date
User ID
User Name
User Email

3NF

Note: Includes separate PASSWORD_HISTORY entity to accommodate multiple passwords tracked individually by date created.

CATEGORY (Category_ID, Name)

CREDENTIAL (Username, Title, Password, URL, Description, Question_1, Answer_1, Question_2, Answer_2, Effective_Date, Expiry_Date, User_ID, Category_ID)

PASSWORD_HISTORY (Pwdhistory_ID, Password, Password_Date, Username)

USER_PROFILE (User_ID, First_Name, Last_Name, Email)

Table of Contents>>

Phase #3: Physical Design

Data Dictionary

Column Name	Data Type	Constraints (PK, FK, NOT NULL, UNIQUE)
Category_ID	Integer	PK, FK, NOT NULL, UNIQUE
Name	Varchar2(30)	NOT NULL
Username	Varchar2(20)	PK, FK, NOT NULL, UNIQUE
Title	Varchar2(50)	NOT NULL
Password	Varchar2(20)	NOT NULL
URL	Varchar2(200)	NOT NULL
Description	Varchar2(2000)	
Question_1	Varchar2(2000)	
Answer_1	Varchar2(2000)	
Question_2	Varchar2(2000)	
Answer_2	Varchar2(2000)	
Effective_Date	Timestamp	NOT NULL
Expiry_Date	Timestamp	
Pwdhistory_ID	Integer	PK, NOT NULL, UNIQUE
Password_Date	Timestamp	PK, NOT NULL, UNIQUE
Password	Varchar2(20)	NOT NULL
User_ID	Integer	PK, FK, NOT NULL, UNIQUE
First_Name	Varchar2(30)	NOT NULL
Last_Name	Varchar2(30)	NOT NULL
Email	Varchar2(200)	NOT NULL

Table of Contents>>

Phase #4: Database Implementation

PART I: SQL Data Definition Language (DDL) statements

```
-- Create USER_PROFILE table
CREATE TABLE user_profile(
    user_id INTEGER NOT NULL,
    first_name VARCHAR2(30) NOT NULL,
    last_name VARCHAR2(30) NOT NULL,
    email VARCHAR(50) NOT NULL UNIQUE,
    CONSTRAINT pk_user_profile PRIMARY KEY (user_id)
);

-- Create CATEGORY table
CREATE TABLE category(
    category_id INTEGER NOT NULL,
    name VARCHAR2(30) NOT NULL,
    CONSTRAINT pk_category PRIMARY KEY (category_id)
);

-- Create CREDENTIAL table
CREATE TABLE credential(
    username VARCHAR2(50) NOT NULL,
    title VARCHAR2(50) NOT NULL,
    password VARCHAR2(20) NOT NULL,
    url  VARCHAR2(200),
    description VARCHAR2(200),
    question_1 VARCHAR2(200),
    answer_1 VARCHAR2(200),
    question_2 VARCHAR2(200),
    answer_2 VARCHAR2(200),
    effective_date TIMESTAMP WITH TIME ZONE NOT NULL,
    expiry_date TIMESTAMP WITH TIME ZONE,
    user_id INTEGER NOT NULL,
    category_id INTEGER NOT NULL,
    CONSTRAINT pk_credential PRIMARY KEY (username)
);
```

Table of Contents>>

```
-- Create PASSWORD_HISTORY table
CREATE TABLE password_history(
    pwdhistory_id INTEGER NOT NULL,
    password_date TIMESTAMP WITH TIME ZONE NOT NULL,
    username VARCHAR2(50) NOT NULL,
    CONSTRAINT pk_password_history
    PRIMARY KEY (pwdhistory_id)
);

-- Add foreign key constraints to tables
ALTER TABLE credential ADD CONSTRAINT fk_credential_user_id
    FOREIGN KEY (user_id) REFERENCES user_profile(user_id);

ALTER TABLE credential ADD CONSTRAINT fk_credential_category_id
    FOREIGN KEY (category_id) REFERENCES category(category_id);

ALTER TABLE password_history ADD CONSTRAINT
fk_password_history_username
    FOREIGN KEY (username) REFERENCES credential(username);
```

PART II: SQL Data Manipulation Language (DDL) statements

```
--Add six rows to CATEGORY table
INSERT INTO category (category_id, name) VALUES (1001, 'Web');
INSERT INTO category (category_id, name) VALUES (2001,
'Software');
INSERT INTO category (category_id, name) VALUES (3001,
'Hardware');
INSERT INTO category (category_id, name) VALUES (4001, 'Sales');
INSERT INTO category (category_id, name) VALUES (5001,
'Operations');
INSERT INTO category (category_id, name) VALUES (6001,
'Marketing');

--Add five rows to USER_PROFILE table
INSERT INTO user_profile (user_id, first_name, last_name, email)
    VALUES (12345, 'Bugs', 'Bunny', 'bugs@myfuzzysite.com');
```

Table of Contents>>

```
INSERT INTO user_profile (user_id, first_name, last_name, email)
    VALUES (678910, 'Weird Al', 'Yankovic',
'al@rekcubsoftware.com');

INSERT INTO user_profile (user_id, first_name, last_name, email)
    VALUES (246810, 'Bilbo', 'Baggins',
    'bilbo@rekcubsoftware.com');

INSERT INTO user_profile (user_id, first_name, last_name, email)
    VALUES (36972, 'Harry', 'Dresden',
    'harry@rekcubsoftware.com');

INSERT INTO user_profile (user_id, first_name, last_name, email)
    VALUES (48121, 'Marty', 'McFly','marty@rekcubsoftware.com');

--Add five rows to CREDENTIAL table
INSERT INTO credential (username, title, password, url,
     description, question_1, answer_1, question_2, answer_2,
     effective_date, expiry_date, user_id, category_id)
    VALUES ('duckseason', 'Manager', 'carr0ts_42',
    'www.renkcubsoftware.com', 'Sales Manager', 'What is your
    favorite color', 'blue', 'What is the name of your first
    grade teacher?', 'Ms Broomstick', CURRENT_TIMESTAMP, null,
    12345, 4001);

INSERT INTO credential (username, title, password, url,
     description, question_1, answer_1, question_2, answer_2,
     effective_date, expiry_date, user_id, category_id)
    VALUES ('pancreas', 'Owner', 'Franks2000inchTV',
    'www.renkcubsoftware.com', 'Owner and CEO', 'What is your
    favorite color', 'hawaiian red', 'What is the name of your
    first grade teacher?', 'Mr Nye the Science Guy',
    CURRENT_TIMESTAMP, null, 678910, 6001);

INSERT INTO credential (username, title, password, url,
     description, question_1, answer_1, question_2, answer_2,
     effective_date, expiry_date, user_id, category_id)
    VALUES ('myprecious', 'Dept Manager', 'Sting111',
    'www.renkcubsoftware.com', 'Department Manager and Head Web
    Designer', 'What is your favorite color', 'earthy brown',
    'What is the name of your first grade teacher?', 'Mr.
    Gandalf the Grey', CURRENT_TIMESTAMP, null, 246810, 1001);
```

Table of Contents>>

```
INSERT INTO credential (username, title, password, url,
     description, question_1, answer_1, question_2, answer_2,
     effective_date, expiry_date, user_id, category_id)
    VALUES ('mrmister', 'Asst Manager', 'blueb3@tle',
    'www.renkcubsoftware.com', 'Assistant Manager and Director
    of Hardware and Such', 'What is your favorite color',
    'black', 'What is the name of your first grade
    teacher?', 'Mr. Bob', CURRENT_TIMESTAMP, null, 36972, 3001);

INSERT INTO credential (username, title, password, url,
     description, question_1, answer_1, question_2, answer_2,
     effective_date, expiry_date, user_id, category_id)
    VALUES ('gigawatt', 'Associate', '88mph',
    'www.renkcubsoftware.com', 'Administrative Assistant', 'What
    is your favorite color', 'delorean gray', 'What is the name
    of your first grade teacher?', 'Doc Brown',
    CURRENT_TIMESTAMP, null, 48121, 5001);

--Add five rows to PASSWORD_HISTORY table
INSERT INTO password_history VALUES (32, CURRENT_TIMESTAMP,
'duckseason');
INSERT INTO password_history VALUES (64, CURRENT_TIMESTAMP,
'pancreas');
INSERT INTO password_history VALUES (19, CURRENT_TIMESTAMP,
'myprecious');
INSERT INTO password_history VALUES (28, CURRENT_TIMESTAMP,
'mrmister');
INSERT INTO password_history VALUES (95, CURRENT_TIMESTAMP,
'gigawatt');

--Update first USER_PROFILE record to my first and last name
UPDATE user_profile SET first_name = 'Erin', last_name =
'Lorelle'
    WHERE (user_id = 12345);
```

Table of Contents>>

Phase #5: Database Maintenance

PART I: Create a view

```
CREATE VIEW All_User_Credentials(User_ID, Last_Name, First_Name,
Email,
    Username, Title, Password, URL, Job_Credential_Description,
Security_Question_1,
    Answer_1, Security_Question_2, Answer_2,
Credential_Effective_Date,
    Expiry_Date, Category_id, Category_Name, PasswordHistory_ID,
    Old_Password, OldPassword_Date) AS
SELECT user_id, last_name, first_name, email, username, title,
password, url,
    description, question_1, answer_1, question_2, answer_2,
effective_date,
    expiry_date, category_id, name, pwdhistory_id, password,
password_date
    FROM credential
    JOIN user_profile USING (user_id)
    JOIN category USING (category_ID)
    JOIN password_history USING (username)
    ORDER BY user_id, last_name;
SELECT*
    FROM All_User_Credentials;
```

Table of Contents>>

PART II: Create specified indexes

```
-- Test index
SELECT last_name, first_name
FROM user_profile
ORDER BY last_name;
--Cost: 4

CREATE INDEX idx_last_name ON user_profile(last_name,
first_name);
--Cost: 2
```

MISC: Notes

I attempted to create the below trigger based on research I found online. I wasn't able to get it to function properly but thought I would include it in my submission. I am curious if this is close to something that would actually be used.

```
--Attempt to add trigger to PASSWORD_HISTORY table
CREATE OR REPLACE TRIGGER "PASSWORD_CHANGE"
BEFORE UPDATE ON credential
FOR EACH ROW
BEGIN
    IF :new.password <> :old.password THEN

      DELETE password_history h WHERE h.pwdhistory_id = :old.pwdhistory_id
AND h.TIMESTAMP =
        (select min(TIMESTAMP) FROM password_history u WHERE u.pwdhistory_id
= :old.username
         HAVING COUNT(*) >= 20);

      INSERT INTO password_history (pwdhistory_id, TIMESTAMP, PASSWORD)
VALUES
                            (:old.pwdhistory_id, password_date,
:old.PASSWORD);
    END IF;
END;

-- Added UNIQUE constraint to PASSWORD_HISTORY table to insure password
  wasn't previously used.
ALTER TABLE password_history ADD CONSTRAINT pwd_is_not_reused UNIQUE
(pwdhistory_id, password);
```

Table of Contents>>