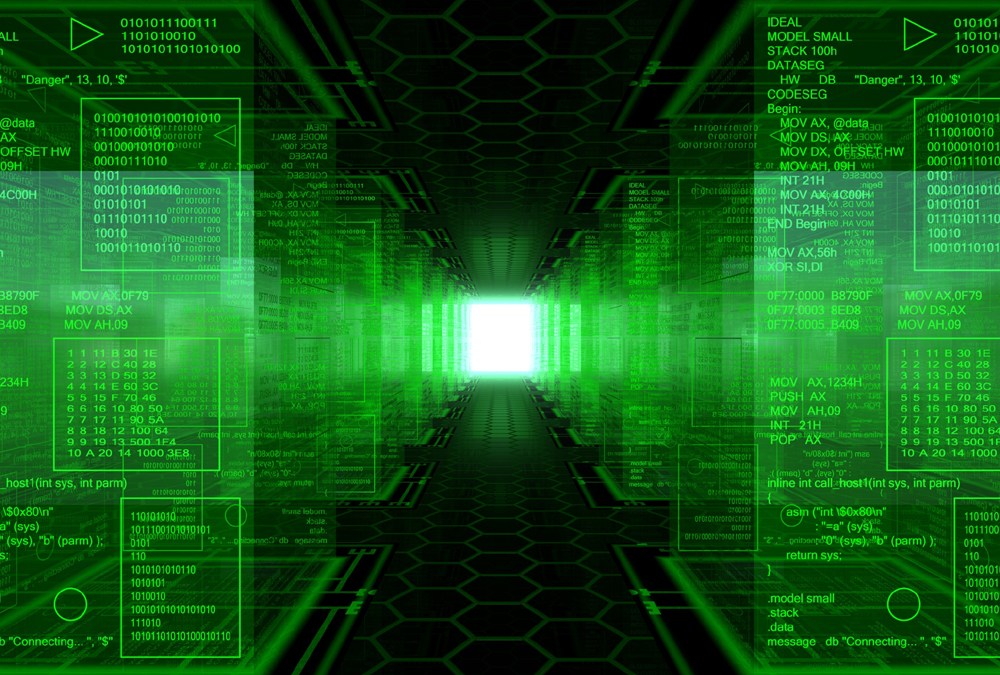
CSCI-2020 Database Fundamentals

Erin Lorelle

Final Exam Project, Fall 2017

December 2, 2017



**“PassBase”**

## Credential Manager Database

## for

## Renkcub Software Solutions

# **Table of Contents**

Phase #1: CONCEPTUAL DESIGN 3

1. [Entity relationship diagram (ERD)](#_Purpose) 3

Phase #2: Logical Design Section 4

1. [Table relations](#_Normalized_through_3NF) 4

Phase #3: Physical Design Section 5

1. [Data dictionary](#_Data_Dictionary) 5

Phase #4: Database Implementation Section 6

1. [Part I: SQL DDL statements](#_PART_I:_SQL) 6
2. [Part II: SQL DML statements](#_PART_II:_SQL) 7

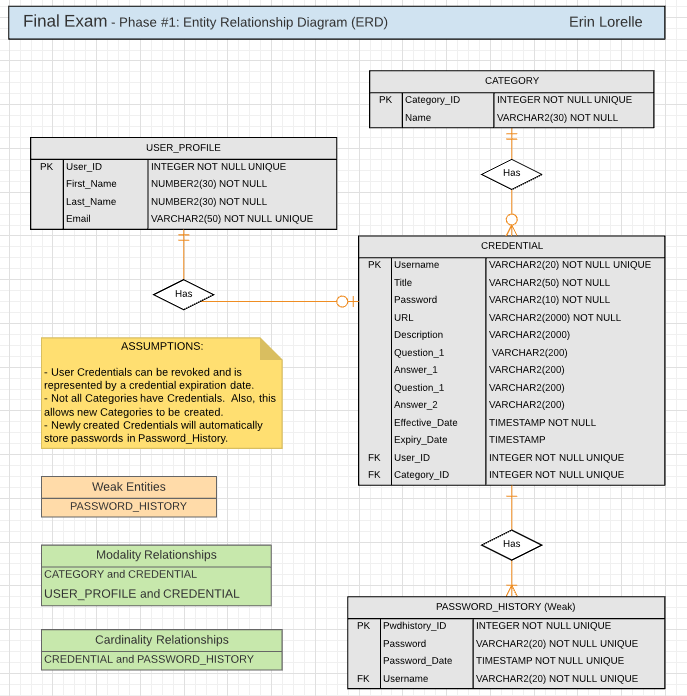
Phase #5: Database Maintenance Section 10

1. [Part I: Create specified view](#_PART_I:_Create) 10
2. [Part II: Create specified indexes](#_PART_II:_Create) 11

# 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>>](#_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>>](#_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>>](#_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>>](#_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>>](#_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>>](#_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>>](#_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>>](#_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>>](#_Table_of_Contents)