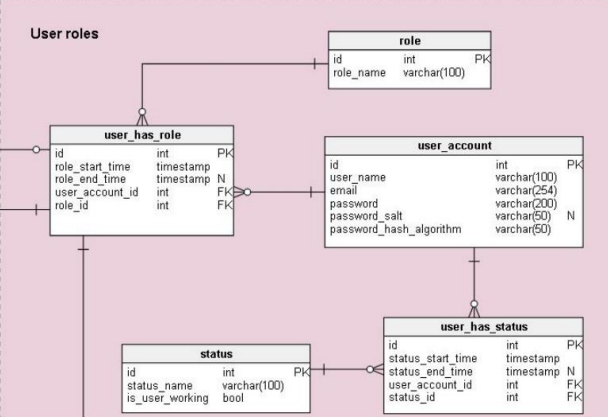
**Relational Database Design**

**Problem Statement:**

How to convert a relational design into tables in SQL Server Topics: In this project, you will work on converting a relational design that enlists various users, their roles, user accounts, and their statuses into different tables in SQL Server and insert data into them. Having at least two rows in each of the tables, you have to ensure that you have created respective foreign keys.

**Tasks to Be Performed:**

* Define relations/attributes.
* Define primary keys
* Create foreign keys



1. Insert data into each of the above tables. With at least two rows in each of the table, make sure, that you have created respective foreign keys.
2. Delete all the data from each of the tables.

-- Project 3

-- Problem Statement: How to convert a relational design into tables in SQL Server

-- Table 1 - User Role

CREATE TABLE role(

id INT PRIMARY KEY,

role\_name VARCHAR(100)

);

-- Table 2 - user\_has\_role

CREATE TABLE user\_has\_role(

id INT PRIMARY KEY,

role\_start\_time DATETIME,

role\_end\_time DATETIME,

user\_account\_id INT FOREIGN KEY(user\_account\_id) REFERENCES user\_has\_role(id),

role\_id INT FOREIGN KEY (role\_id) REFERENCES user\_has\_role(id)

);

-- Table 3 - user account

CREATE TABLE user\_account(

id INT PRIMARY KEY,

username VARCHAR(100),

email VARCHAR(254),

password VARCHAR(200),

password\_salt VARCHAR(50),

password\_hash\_algorithm VARCHAR(50));

-- Table 4 - status

CREATE TABLE status(

id INT PRIMARY KEY,

status\_name VARCHAR(100),

is\_user\_working BIT

);

-- Table 5 - user\_has\_status

CREATE TABLE user\_has\_status(

id INT PRIMARY KEY,

status\_start\_time DATETIME,

status\_end\_time DATETIME,

user\_account\_id INT FOREIGN KEY(user\_account\_id) REFERENCES user\_has\_role(id),

status\_id INT FOREIGN KEY (status\_id) REFERENCES user\_has\_role(id)

);

/\* 1. Insert data into each of the above tables. With at least two rows in each of the table, make sure, that you have created respective foreign keys. \*/

-- Insert data into 'role' table

INSERT INTO role (id, role\_name) VALUES

(1, 'Admin'),

(2, 'User');

select \* from role;

-- Insert data into 'user\_has\_role' table

INSERT INTO user\_has\_role (id, role\_start\_time, role\_end\_time, user\_account\_id, role\_id) VALUES

(1, '2023-01-01 00:00:00', '2023-01-31 23:59:59', 1, 1),

(2, '2023-02-01 00:00:00', '2023-02-28 23:59:59', 2, 2);

select \* from user\_has\_role;

-- Insert data into 'user\_account' table

INSERT INTO user\_account (id, username, email, password, password\_salt, password\_hash\_algorithm) VALUES

(1, 'admin\_user', 'admin@example.com', 'admin\_password\_hash', 'admin\_salt', 'SHA256'),

(2, 'regular\_user', 'user@example.com', 'user\_password\_hash', 'user\_salt', 'SHA256');

select \* from user\_account;

-- Insert data into 'status' table

INSERT INTO status (id, status\_name, is\_user\_working) VALUES

(1, 'Active', 1),

(2, 'Inactive', 0);

select \* from status;

-- Insert data into 'user\_has\_status' table

INSERT INTO user\_has\_status (id, status\_start\_time, status\_end\_time, user\_account\_id, status\_id) VALUES

(1, '2023-01-01 00:00:00', '2023-01-31 23:59:59', 1, 1),

(2, '2023-02-01 00:00:00', '2023-02-28 23:59:59', 2, 2);

select \* from user\_has\_status;

-- 2. Delete all the data from each of the tables.

-- Delete all data from 'role' table

DELETE FROM role;

-- Delete all data from 'user\_has\_role' table

DELETE FROM user\_has\_role;

-- Delete all data from 'user\_account' table

DELETE FROM user\_account;

-- Delete all data from 'status' table

DELETE FROM status;

-- Delete all data from 'user\_has\_status' table

DELETE FROM user\_has\_status;

-- Truncate all tables

TRUNCATE TABLE role;

TRUNCATE TABLE user\_has\_role;

TRUNCATE TABLE user\_account;

TRUNCATE TABLE status;

TRUNCATE TABLE user\_has\_status;