

**Department of Computer Engineering**

**CSE5041 Database Design & Development  
Project Report**

SOCIAL NETWORK

**BUĞRA MERİÇ DEĞİRMENCİ**

**TABLE OF CONTENTS**

[**1 INTRODUCTION** 4](#_Toc352925434)

[1.1 PROJECT DESCRIPTION 4](#_Toc352925435)

[**2 ENTITY RELATIONAL MODEL** 5](#_Toc352925436)

[2.1 ENHANCED ER DIAGRAM 5](#_Toc352925437)

[2.2 RELATIONAL SCHEMA & MAPPING 6](#_Toc352925438)

[**3 NORMALIZATION** 7](#_Toc352925439)

[3.1 FUNCTIONAL DEPENDENCIES 7](#_Toc352925440)

[3.2 UNNORMALISED FORM 7](#_Toc352925441)

[3.3 FIRST NORMAL FORM 7](#_Toc352925442)

[3.4 SECOND NORMAL FORM 7](#_Toc352925443)

[3.5 THIRD NORMAL FORM 7](#_Toc352925444)

**LIST OF FIGURES**

[Figure 1: EER diagram of the Social Network Database ............................... 5](#_ENHANCED_ER_DIAGRAM)

[Figure 2: Relational schema of the Social Network Database with arrows indicating referential integrity....................................................................................................... 6](#_RELATIONAL_SCHEMA_&)

# INTRODUCTION

## PROJECT DESCRIPTION

**An Example:**

The Social Network Database (SocialNetwork) stores information about the users of a social network application. The following data have been identified and to be represented:

• Each user’s first name, last name, E-Mail, Password, sex,birthday, the date they have joined and a profile picture are stored in the User table in the database. Data like hobbies and user’s profession are set on different tables The ProfileDescription table stores the user’s description in their profile page. The visibility of the data(for example E-Mail, sex, profile picture) can be adjusted through the settings stored in the PrivacySettings table.

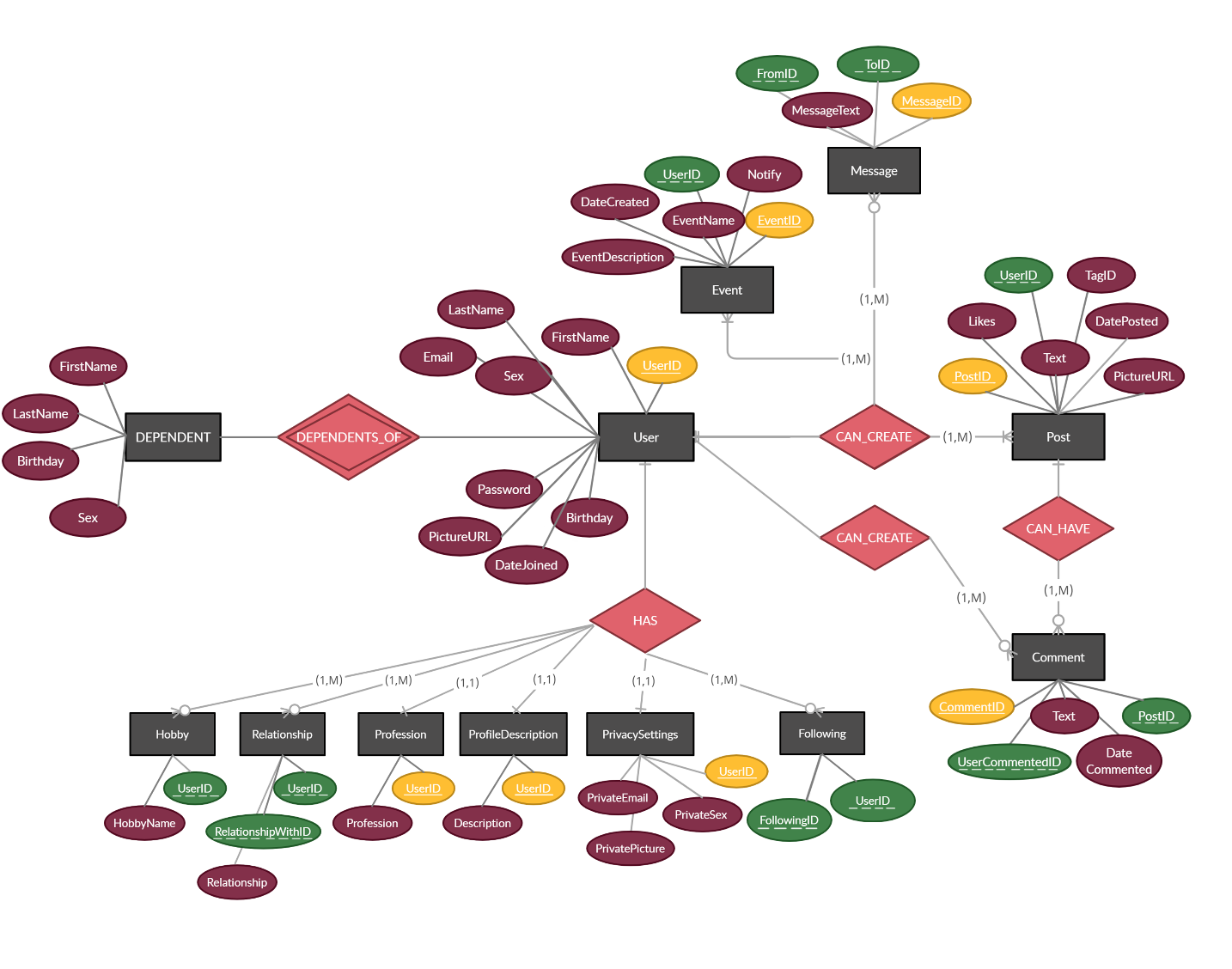
•The database also keeps track of the relationships, followers and private messages in their respective tables.

• Every user can create an event with a name, description and a choice to notify followers through messages when the event is created. These information is stored on the Event table.

•Every user can post a text, with or without a picture. Every post has a number of likes and a tag option. If the user tags another user while creating a post, the tagged person gets a notification through messages. The application also allows users to comment on their or other user’s posts. Every data is stored in their respective tables.

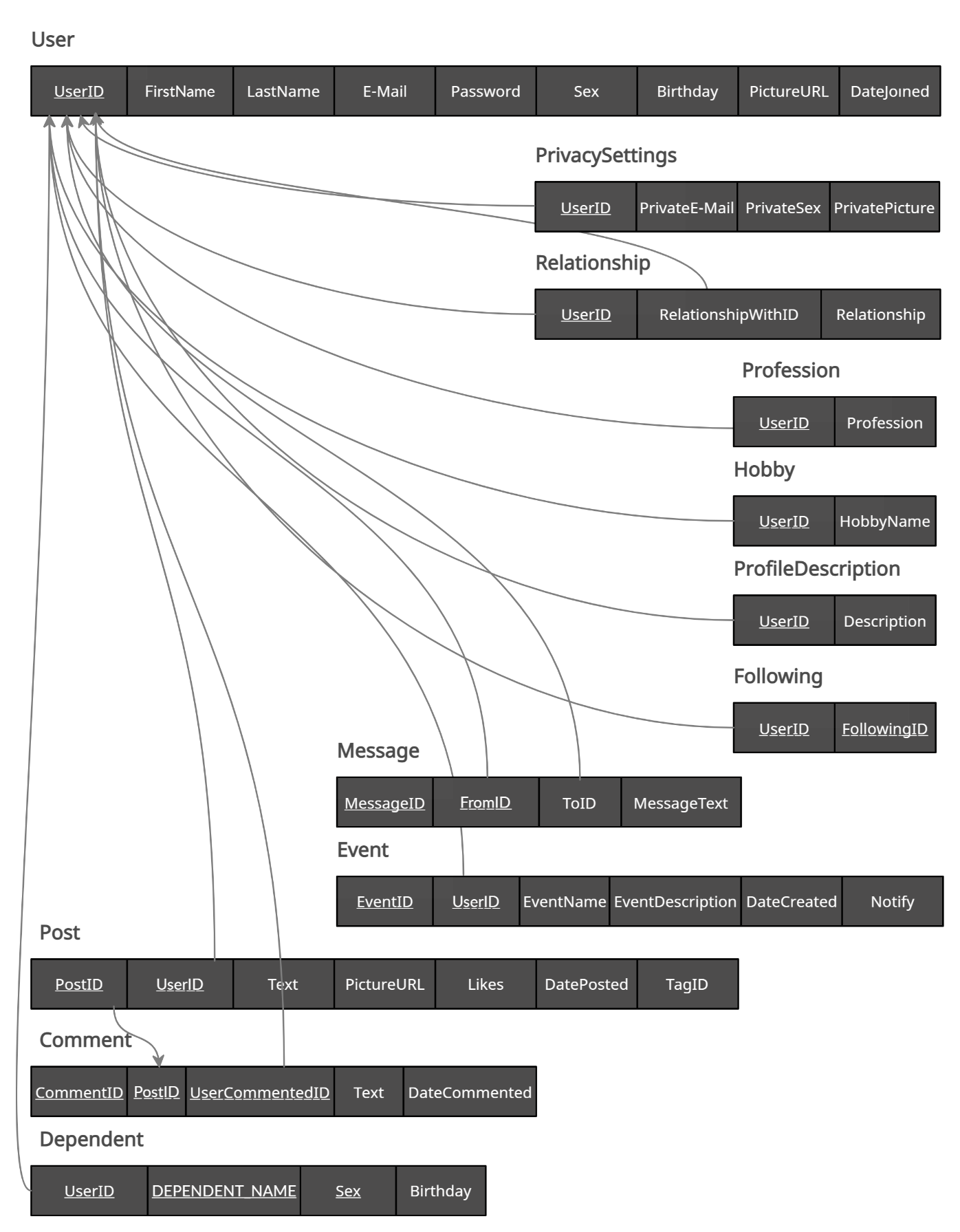
# ENTITY RELATIONAL MODEL

## ENHANCED ER DIAGRAM



**Figure 1:** EER diagram of the Social Network Database

## RELATIONAL SCHEMA & MAPPING



**Figure 2:** Relational schema of the Social Network Database with arrows indicating referential integrity

# NORMALIZATION

## FUNCTIONAL DEPENDENCIES

F ={FD1:UserID → FirstName, LastName, E-Mail, Password, Sex, Birthday, PictureURL,

DateJoined, PrivateE-Mail, PrivateSex, PrivatePicture, Description, Profession, HobbyName

FD2: PostID → UserID,Text, PictureURL, Likes, DatePosted, TagID

FD3: CommentID → PostID,UserCommentedID,Text,DateCommented

FD4: EventID → UserID, EventName, EventDescription, DateCreated, Notify

FD5: MessageID → FromID, ToID, MessageText

FD6: RelationshipWithID → UserID, Relationship

FD7:E-Mail → Password

}

## UNNORMALISED FORM

Users Table



Primary Key: UserID

The table is already at UF1 because we don’t have any attribute that has many values.

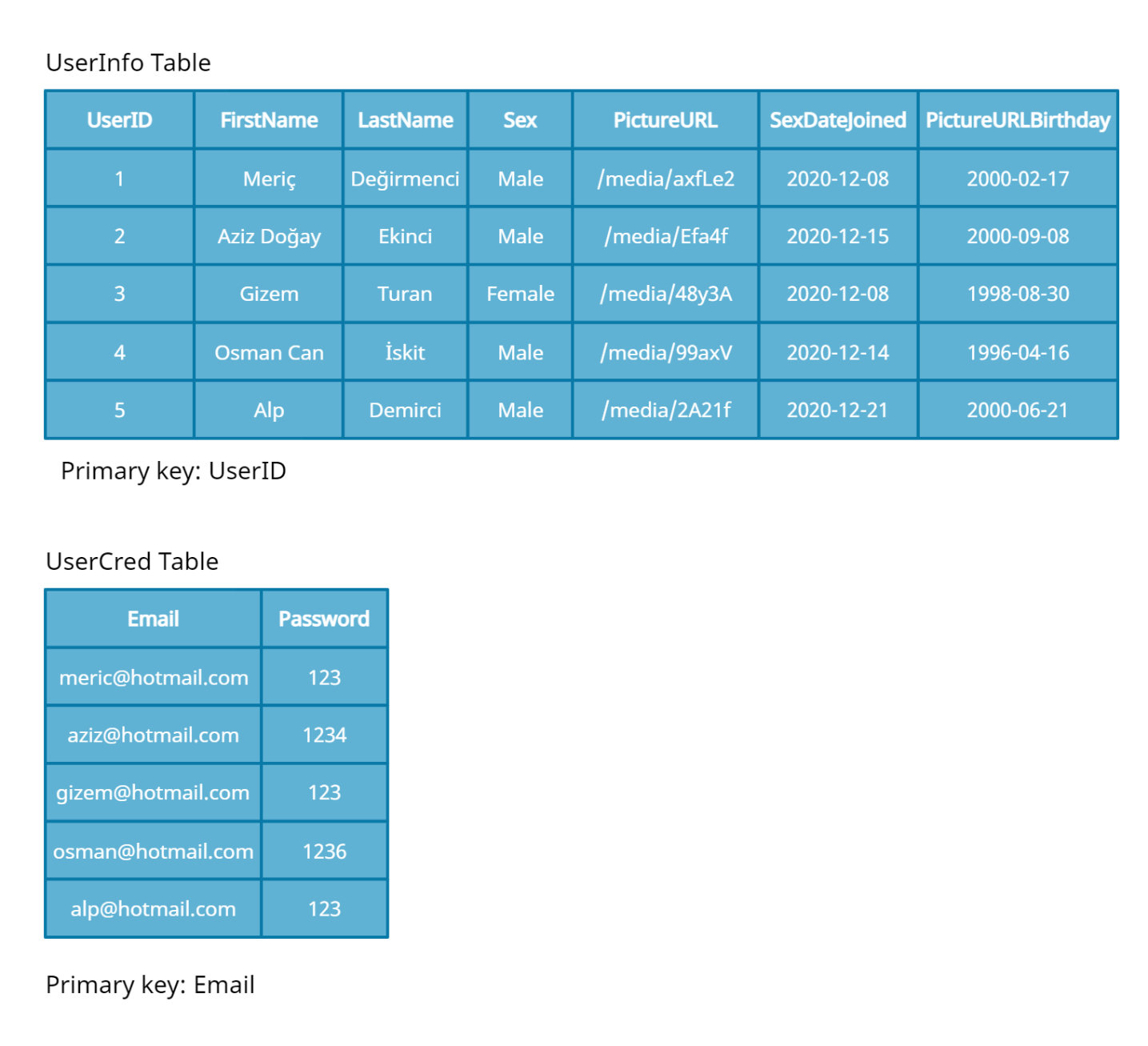
## FIRST NORMAL FORM

Users Table



Primary Key: UserID

## SECOND NORMAL FORM

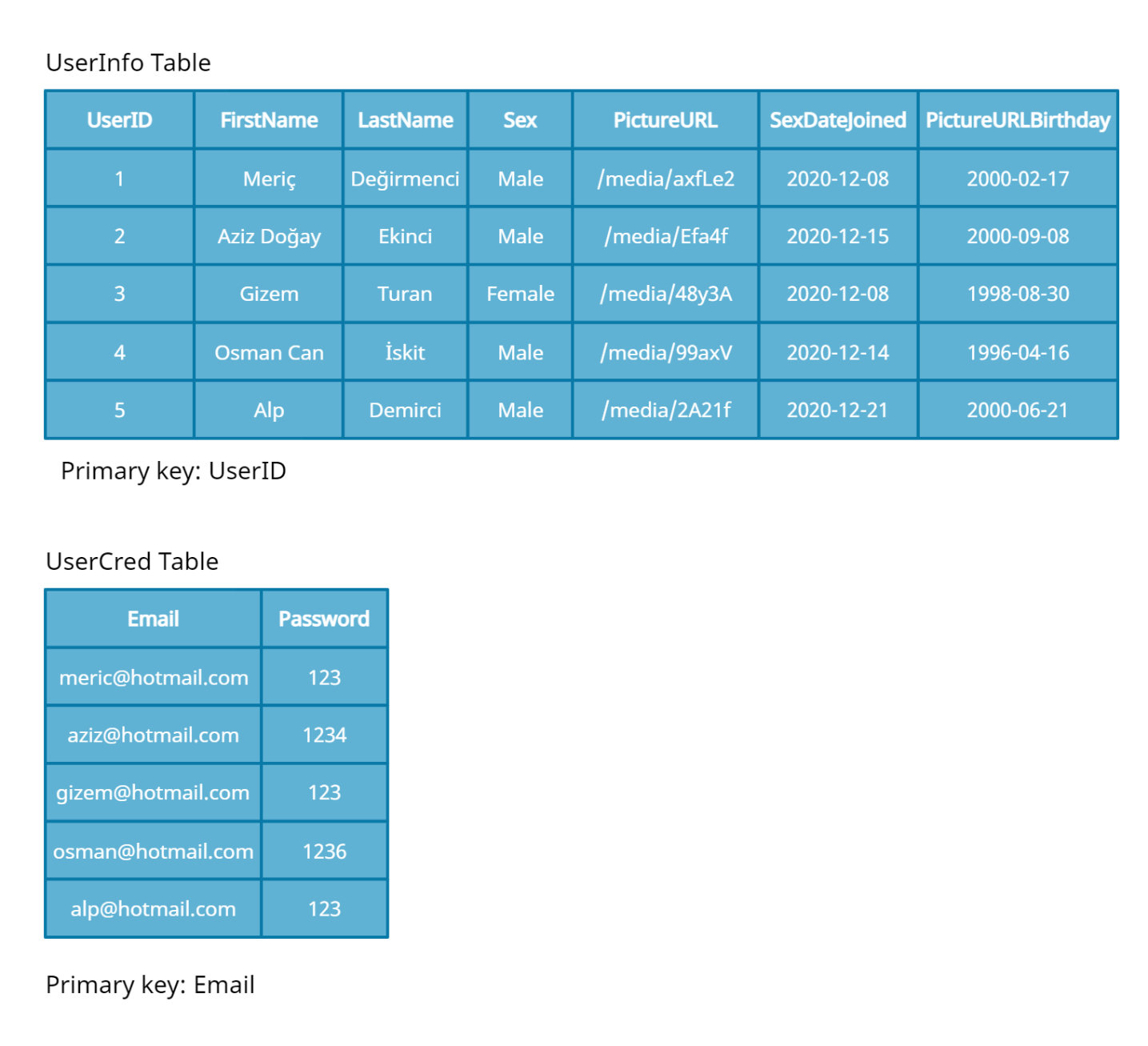


We split the table into UserInfo and UserCred table.

Each non key field is directly correlated with the primary key. So this is NF2

There aren’t any transit functional dependencies, so this is also NF3

## THIRD NORMAL FORM



**THE NORMALIZATIONS OF THE OTHER TABLES**

**Event Table**

**UNF:**

Event{EventID, EventName, EventDescription, DateCreated,UserID,Notify} Primary Key: EventID

**1NF:**

Event{EventID, EventName, EventDescription, DateCreated,UserID,Notify} Primary Key: EventID

\*Already at 1NF

**2NF:**

EventInfo{EventID, EventName, EventDescription, DateCreated, Notify} Primary Key: EventID

EventUser{UserID,EventID} Primary Key:UserID

\*We split the table for a separate EventInfo and EventUser

**3NF:**

EventInfo{EventID, EventName, EventDescription, DateCreated, Notify} Primary Key: EventID

EventUser{UserID,EventID} Primary Key:UserID

**Post Table**

**UNF:**

Post{PostID, Text, PictureURL, DatePosted, Likes, UserID, TagID} Primary Key: PostID

**1NF:**

Post{PostID, Text, PictureURL, DatePosted, Likes, UserID, TagID} Primary Key: PostID

\*Already at NF1

**2NF:**

PostInfo{PostID, Text, PictureURL, DatePosted, Likes, UserID} Primary Key: PostID

PostTag{PostID, TagID} Primary Key: PostID

\*We split the table. Because in some cases the TagID attribute is NULL and it is reserving unnecessary space .

**3NF:**

PostInfo{PostID, Text, PictureURL, DatePosted, Likes, UserID} Primary Key: PostID

PostTag{PostID, TagID} Primary Key: PostID

\*Alread at NF3 because there aren’t any transit functional dependencies

**Comment Table**

**UNF:**

Comment{CommentID, Text, DateCommented, UserCommentedID, PostID}

Primary Key: CommentID

**1NF:**

Comment{CommentID, Text, DateCommented, UserCommentedID, PostID}

Primary Key: CommentID

\*Already at 1NF

**2NF:**

CommentInfo{CommentID, Text, DateCommented, UserCommentedID}

Primary Key: CommentID

CommentPost{CommentID,UserID}

**3NF**

CommentInfo{CommentID, Text, DateCommented, UserCommentedID}

Primary Key: CommentID

CommentPost{CommentID,UserID}

\*No transit function dependencies

**Message Table**

**UNF:**

Message{MessageID, MessageText, FromID, ToID} Primary Key: MessageID

**1NF, 2NF and 3NF:**

Message{MessageID, MessageText, FromID, ToID} Primary Key: MessageID

**Hobby Table**

**UNF:**

Hobby{UserID, HobbyName} Primary Key: UserID

**1NF, 2NF and 3NF:**

Hobby{UserID, HobbyName} Primary Key: UserID

**Profession Table**

**UNF:**

Profession{UserID, ProfessionName} Primary Key: UserID

**1NF, 2NF and 3NF:**

Profession{UserID, ProfessionName} Primary Key: UserID

**ProfileDescription Table**

**UNF:**

ProfileDescription{UserID, Description} Primary Key: UserID

**1NF, 2NF and 3NF:**

ProfileDescription{UserID, Description} Primary Key: UserID

**Privacy Settings Table**

**UNF:**

PrivacySettings{UserID, PrivateEmail, PrivateSex, PrivatePicture} Primary Key: UserID

**1NF, 2NF and 3NF:**

PrivacySettings{UserID, PrivateEmail, PrivateSex, PrivatePicture} Primary Key: UserID

**Relationship Table**

**UNF:**

Relationship{UserID, RelationshipWithID, Relationship} Primary Key: UserID

**1NF, 2NF and 3NF:**

Relationship{UserID, RelationshipWithID, Relationship} Primary Key: UserID

**Following Table**

**UNF:**

Following{UserID, FollowingID} Primary Key: UserID

**1NF, 2NF and 3NF:**

Following{UserID, FollowingID} Primary Key: UserID