



İSTANBUL TEKNİK ÜNİVERSİTESİ

Dijital Makale Satış Platformu Veritabanı Arayüzü

Bilgi Teknolojileri Tezsiz Yüksek Lisans Programı
2023-2024 Güz Dönemi
Veri Tabanı Sistemleri Dersi

Ad-Soyad : İsmail Emre UÇAR

No : 701231004

Dersin Kodu/CRN: BTE 513B/14967

Teslim Tarihi: 26.12.2023

Danışman: Doç. Dr. Ahmet Cüneyd TANTUĞ

Projenin Amacı:

Bilimsel araştırma makalelerini yayınlamakta ve erişimini sağlayan yayıncılar (Elsevier, Springer, Wiley, Taylor&Francis vb.) vardır. Burada çalışan birinin veritabanına erişimi bir arayüz aracılığıyla sağlanarak, veri girişi ve makale satışlarını manuel olarakta yapabilmesi amaçlanmıştır.

Projenin Kapsamı:

Arayüz kullanıcısının sahip olduğu yetenekler sırasıyla;

- **Kullanıcı hesap bilgilerinin;** isim, soyisim ve e-mail adresi aracılığıyla veritabanına girişi yapılabilmektedir.
- **Makale bilgilerini;** makale başlığı, yayın tarihi, yayınlandığı dergi ve yazarları yazar sırasına göre veritabanında saklayabilmektedir.
- **Yazar bilgilerini;** yazar isim/soyisimleri, çalıştıkları makaleler ve yazar sıralarına göre veritabanı bünyesinde tutabilmektedir.
- **Makale satışları;** satılacak makaleye ait bilgiler kullanıcı e-mail adresi üzerinden yapılmaktadır.

İşlevsel Bağımlılıklar:

Yapılan varsayımlar ve veritabanının işleyişi:

- Aynı başlığı kullanan makaleler aynı dergilerde varolamazlar.
- Aynı isim ve soyada sahip insanlar olabilir, ancak aynı mail adresini almalarına müsaade edilemez. E-mail adresleri benzersiz olmak zorunda bırakılmıştır.
- Benzersiz olan e-mail adresleri arayüz kullanıcısının erişimine açık olup hesapların tutulduğu veritabanındaki hesap id'leri güvenlik amacıyla kapalı tutulmuştur. Bu sayede kullanıcı satışları mail adresleri üzerinden eşleyebilir.
- Aynı isimde bir dergi(journal) defalarca veritabanına kaydedilemez.
- Bir derginin veritabanında kaydının oluşabilmesi için makale yayınlamış olma zorunluluğu vardır. Bu yüzden dergiler tek başına kayıt alamaz.
- Satış işlemleri dijital platformda gerçekleştiği için satış anı timestamp ile anlık olarak alınarak tarih bilgilerini veritabanında tutar.
- Müşteri hesapları isim, soyad veya e-mail adresleri üzerinden aranarak hesap bilgilerine ulaşılabilir.

- Makaleler başlık, yazar veya dergilere göre aranabilir.
- Bir dergide yayınlanan tüm makaleler gösterilebilir.

author_id → {author_name, author_surname}

user_id → {user_name, user_surname, user_email}

sale_id → {sale_id, user_id, article_id, sale_date}

article_id → {article_title, article_year, journal_id}

journal_id → {journal_name}

Normalizasyon Adımları:

1NF: Bir hücre birden fazla değer (atomiklik) içermiyordur. Tanımlama için birincil anahtar oluşturulmuştur. Tekrarlanan satır veya sütun yoktur. Herbir sütun tablodaki her satır için yalnızca bir değere sahiptir

id	author_name	author_surname	author_order	user_name	user_surname	user_email	journal_name	article_title	article_year	sale_date
.
.
.

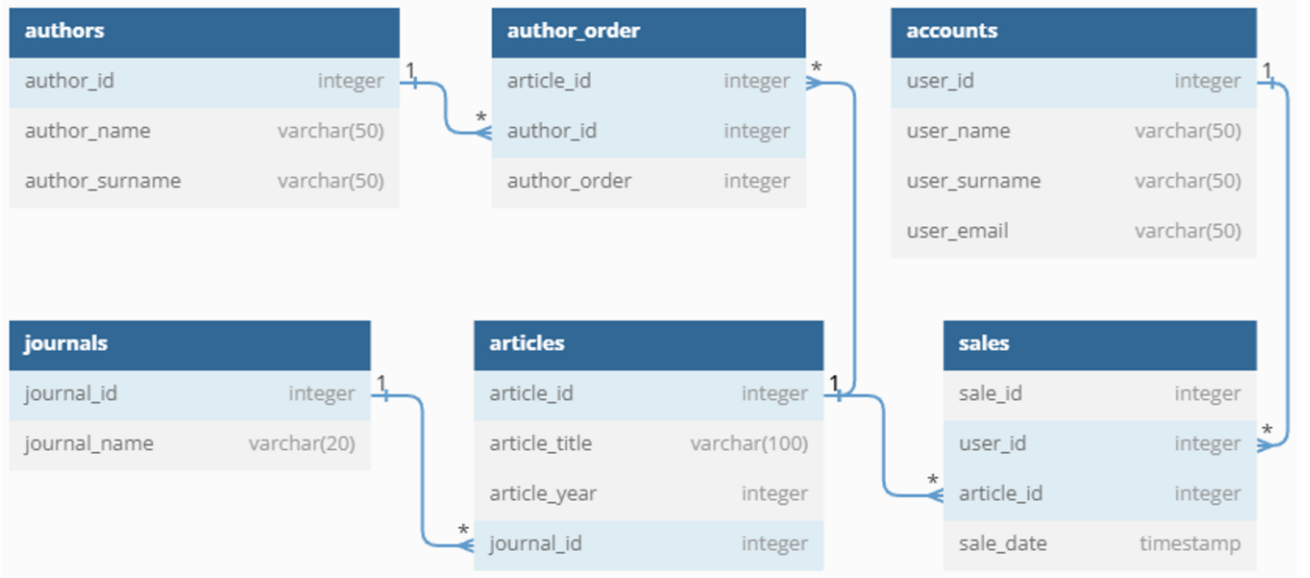
2NF: Her bir kolon kısmi bağımlılığı artık yoktur. Tüm anahtar olmayan nitelikler, birincil anahtara tamamen bağımlıdır.

id	author_name	author_surname	author_order
.	.	.	.
.	.	.	.
.	.	.	.

user_id	user_name	user_surname	user_email
.	.	.	.
.	.	.	.
.	.	.	.

article_id	journal_name	article_title	article_year	sale_date
.
.
.

3NF: Yapılan kabullerden dolayı anahtar olmayan nitelikler (journal_name ve author_order) kendisi gibi key olmayan (article_name ve author name) değerlere bağımlı olduğu için 3NF'e uygun tablo E-R diyagramı üzerinde gösterilmiştir.



Program ile ilgili,

WAMP veritabanı yönetim sistemine yardımcı olarak yüklenip SQL kodlar ve veritabanı oradan çalıştırılmıştır. Python programlama dili kullanılarak arka planda çalışan yazılım tamamlanmıştır. Kullanıcı arayüzü için main_menu.py dosyası çalıştırılmalıdır.

accounts.py: Hesaplar üzerindeki fonksiyonları, girdileri ve aramaları içerir.

articles_authors.py: Makaleler ve yazarlar ile ilgili fonksiyonları, girdileri ve aramaları içerir.

sales.py: Satışlar ile ilgili fonksiyonları, girdileri ve aramaları içerir.

sql_query.py: Uzun ve karışık olan tüm sorguların fonksiyonlarını içerir.

main_menu.py: Ana menüyü ve

```

import re
from os import system, name
import mysql.connector
from mysql.connector import Error

def mysql_connect():
    conn = None
    try:
        conn = mysql.connector.connect(host='localhost', database='bte513b_article_sale_website', user='root',password='')
        if conn.is_connected():
            return conn
    except Error as e:
        print(e)

def mysql_select(sql, val):
    mydb = mysql_connect()
    mycursor = mydb.cursor(dictionary=True)
    try:
        mycursor.execute(sql, val)
        myresult = mycursor.fetchall()
        for x in myresult:
            yield x
    except Error as e:
        mydb.rollback()
        print(e)
        return e

def mysql_insert(sql, val):
    mydb = mysql_connect()
    mycursor = mydb.cursor()
    try:
        mycursor.execute(sql, val)
        mydb.commit()
    except Error as e:
        mydb.rollback()
        print(e)
        return e

```

fonksiyonlarını içerir. Bu sayede veritabanı ile arasında gerekli olan kütüphaneleri ve veri girdilerini yapabilmektedir.

NOT: (host='localhost', database='bte513b_article_sale_website', user='root',password='')

video link: <https://vimeo.com/897958372?share=copy>