

**DBMS ASSIGNMENT-1**

**ONLINE STREAMING  
OF VIDEOS DATABASE  
MAINTENANCE  
SYSTEM**

**1602-18-737-104**

**-M.SANNIHITHA**

## **ABSTRACT**

This project is an application software developed for monitoring the online video streaming which mainly focuses on basic operations like uploading a video, updating new information, searching videos and identifying the members who are genuinely watching the video. It is implemented with the help of 5 tables namely USERS, VIEWS, VIDEOS, UPLOADS, ADMIN.

The USERS and ADMIN tables consists the information regarding members and the admin respectively. Admin uploads a video which is described in another table namely UPLOADS while the relationship between USERS and VIDEOS tables is established by VIEWS.

Uid, Vid, Aid are the primary keys of the tables USERS, VIDEOS, ADMIN respectively and are foreign keys in the tables VIE and UPLOADS. A one to one mapping for ADMIN table and a total participation of USERS table is established.

This article aims to provide a structured approach that admin can use identify the members who watch the video genuinely and generate reports accordingly.

# REQUIREMENTS ANALYSIS

## List of Tables:

1. USERS
2. ADMIN
3. VIDEOS
4. UPLOADS
5. VIEWS

## List of attributes with their domain types:

### USERS:

User ID: uid-NUMBER()

User name: name-VARCHAR()

User age: age-NUMBER()

User password: password VARCHAR()

### ADMIN:

Admin ID: aid-NUMBER()

Admin name: name-VARCHAR()

Admin age: age-NUMBER()

## VIDEOS:

Video ID: vid-NUMBER()

Video duration: duration-NUMBER()

Number of views: numviews-NUMBER()

Video topic: topic-VARCHAR()

## UPLOADS:

Time of uploading: when-NUMBER()

## VIEWS:

Genuineness of video viewed: genuineness-VARCHAR()

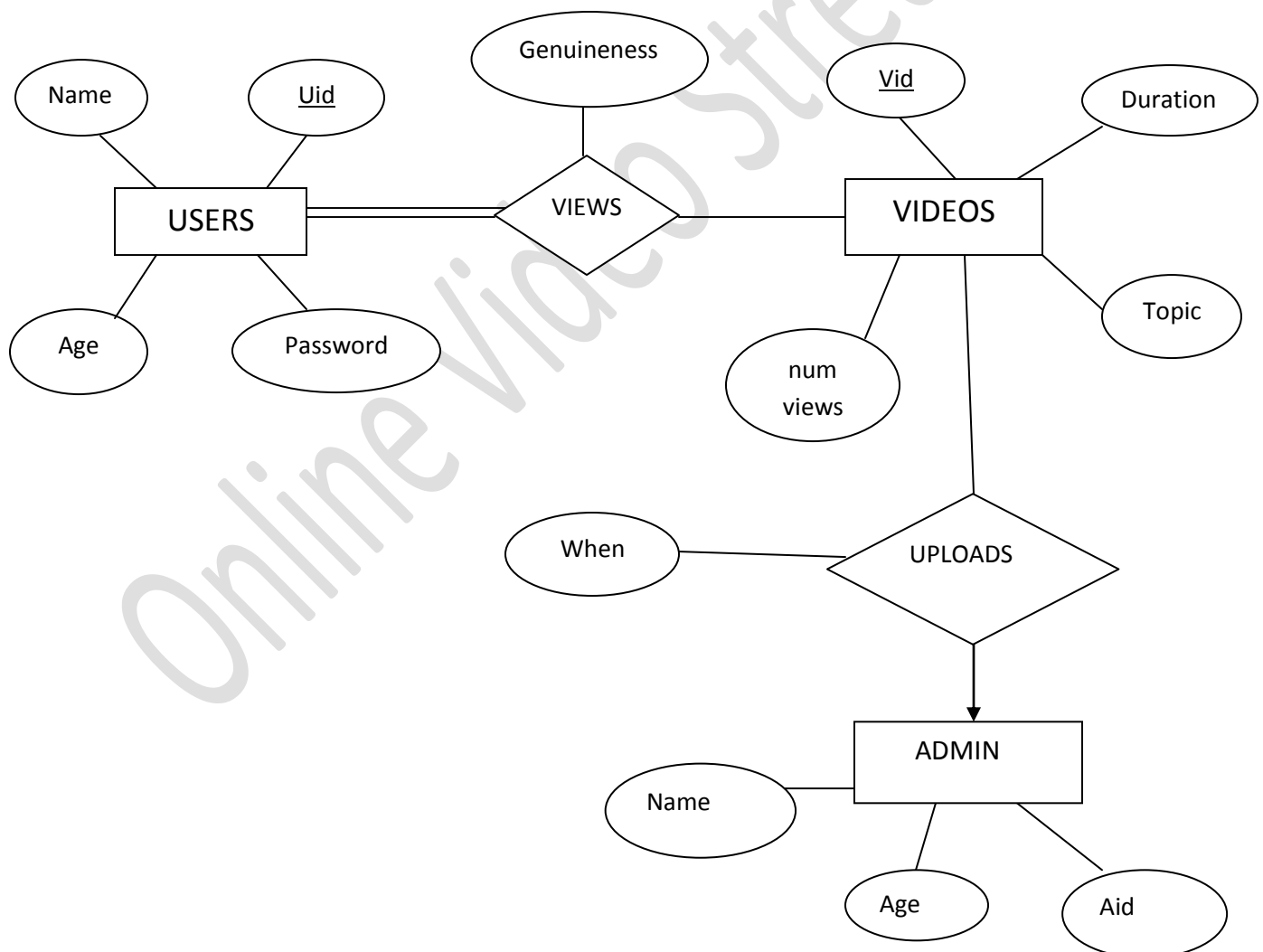
# DBMS

## ASSIGNMENT-1

1602-18-737-104

### ONLINE STREAMING OF VIDEOS DATABASE MAINTENANCE SYSTEM

#### ENTITY-RELATIONSHIP DIAGRAM



## **Mapping Cardinalities and Participation Constraints:**

A user can watch as many as videos as desired but all videos need not be watched by atleast one user. So, many to many participation between users and views .

As of now, there is only one admin so only one entity not a set.

Genuineness is the descriptive attribute of views table and it tells if a view is genuine or not.

When is the descriptive attribute of uploads tabe and it gives when the video was uploaded by the admin.

# OUTPUTS

1602-18-737-104

## DDL COMMANDS

### 1.USERS

```
SQL> create table users(usid NUMBER(5) Primary key,name VARCHAR(10),password VARCHAR(10),age NUMBER(5));
Table created.

SQL> desc users;
Name                               Null?    Type
-----
USID                               NOT NULL NUMBER(5)
NAME                               VARCHAR2(10)
PASSWORD                           VARCHAR2(10)
AGE                                NUMBER(5)

SQL>
```

### 2.VIDEOS

```
SQL> create table videos(vid NUMBER(5) Primary key,duration NUMBER(5),topic VARCHAR(10),numviews NUMBER(5));
Table created.

SQL> desc videos
Name                               Null?    Type
-----
VID                                NOT NULL NUMBER(5)
DURATION                           NUMBER(5)
TOPIC                              VARCHAR2(10)
NUMVIEWS                           NUMBER(5)

SQL>
```

### 3.ADMIN

```
SQL> create table admin(aid NUMBER(5) Primary key,name NUMBER(5),age NUMBER(5));
Table created.

SQL> desc admin;
Name                               Null?    Type
-----
AID                                NOT NULL NUMBER(5)
NAME                               NUMBER(5)
AGE                                NUMBER(5)

SQL>
```

#### 4.VIEWS

```
SQL> create table views(usid NUMBER(5),vid NUMBER(5),genuineness VARCHAR(5),foreign key(usid) references users,foreign key(vid) references videos);
```

Table created.

```
SQL> desc views;
```

Name	Null?	Type
USID		NUMBER(5)
VID		NUMBER(5)
GENUINENESS		VARCHAR2(5)

```
SQL>
```

#### 5.UPLOADS

```
SQL> create table uploads(vid NUMBER(5),aid NUMBER(5),since VARCHAR(5),foreign key(vid) references videos,foreign key(aid) references admin,primary key(vid,aid));
```

Table created.

```
SQL> desc uploads
```

Name	Null?	Type
VID	NOT NULL	NUMBER(5)
AID	NOT NULL	NUMBER(5)
SINCE		VARCHAR2(5)

```
SQL>
```



## DML COMMANDS

### 1.USERS

```
new 1: insert into users values(45,'sri','xyz',19)

1 row created.

SQL> /
Enter value for usid: 77
Enter value for name: vamsi
Enter value for password: secret
Enter value for age: 22
old 1: insert into users values(&usid,&name,&password,&age)
new 1: insert into users values(77,'vamsi','secret',22)

1 row created.

SQL> /
Enter value for usid: 100
Enter value for name: ria
Enter value for password: efg
Enter value for age: 16
old 1: insert into users values(&usid,&name,&password,&age)
new 1: insert into users values(100,'ria','efg',16)

1 row created.

SQL> /
Enter value for usid: 104
Enter value for name: manu
Enter value for password: paga
Enter value for age: 19
old 1: insert into users values(&usid,&name,&password,&age)
new 1: insert into users values(104,'manu','paga',19)

1 row created.

SQL> select * from users;

  USID NAME      PASSWORD      AGE
-----
    203 jai       abc          17
    45 sri       xyz          19
    77 vamsi     secret        22
   100 ria       efg          16
   104 manu     paga          19
```

## 2.VIDEOS

```
SQL> insert into videos values(&vid,&duration,&'&topic',&numviews);
Enter value for vid: 473
Enter value for duration: 29
Enter value for topic: sds
Enter value for numviews: 444
old 1: insert into videos values(&vid,&duration,&'&topic',&numviews)
new 1: insert into videos values(473,29,'sds',444)
```

1 row created.

```
SQL> /
Enter value for vid: 1111
Enter value for duration: 35
Enter value for topic: python
Enter value for numviews: 144
old 1: insert into videos values(&vid,&duration,&'&topic',&numviews)
new 1: insert into videos values(1111,35,'python',144)
```

1 row created.

```
SQL> /
Enter value for vid: 7538
Enter value for duration: 15
Enter value for topic: matlab
Enter value for numviews: 88
old 1: insert into videos values(&vid,&duration,&'&topic',&numviews)
new 1: insert into videos values(7538,15,'matlab',88)
```

1 row created.

```
SQL> select * from videos;
```

VID	DURATION	TOPIC	NUMVIEWS
2637	20	stats	100
7438	25	algebra	28
473	29	sds	444
1111	35	python	144
7538	15	matlab	88

```
SQL>
```

### 3.ADMIN

```
SQL>
SQL> insert into admin values(&aid,&'&name',&age);
Enter value for aid: 299
Enter value for name: sannihitha
Enter value for age: 18
old 1: insert into admin values(&aid,&'&name',&age)
new 1: insert into admin values(299,'sannihitha',18)

1 row created.

SQL> select * from admin;

      AID NAME                AGE
-----
      299 sannihitha          18

SQL> _
```

#### 4.VIEWS

SQL> Run SQL Command Line

1 row created.

SQL> /

Enter value for usid: 166

Enter value for vid: 473

Enter value for genuineness: yes

old 1: insert into views values(&usid,&vid,&'&genuineness')

new 1: insert into views values(166,473,'yes')

1 row created.

SQL> /

Enter value for usid: 104

Enter value for vid: 1111

Enter value for genuineness: no

old 1: insert into views values(&usid,&vid,&'&genuineness')

new 1: insert into views values(104,1111,'no')

1 row created.

SQL> /

Enter value for usid: 203

Enter value for vid: 2637

Enter value for genuineness: no

old 1: insert into views values(&usid,&vid,&'&genuineness')

new 1: insert into views values(203,2637,'no')

1 row created.


SQL> select \* from views;

USID	VID	GENUI
77	7438	yes
100	1111	no
77	7538	no
166	473	yes
104	1111	no
203	2637	no

6 rows selected.

SQL>

## 5.UPLOADS

 Run SQL Command Line

```
Enter value for since: 5 days
old  1: insert into uploads values(&vid,&aid,&since')
new  1: insert into uploads values(473,299,'5 days')

1 row created.

SQL> /
Enter value for vid: 7538
Enter value for aid: 299
Enter value for since: 1 hour
old  1: insert into uploads values(&vid,&aid,&since')
new  1: insert into uploads values(7538,299,'1 hour')

1 row created.

SQL> /
Enter value for vid: 2637
Enter value for aid: 299
Enter value for since: 1 week
old  1: insert into uploads values(&vid,&aid,&since')
new  1: insert into uploads values(2637,299,'1 week')

1 row created.

SQL> /
Enter value for vid: 7438
Enter value for aid: 299
Enter value for since: 5 hours
old  1: insert into uploads values(&vid,&aid,&since')
new  1: insert into uploads values(7438,299,'5 hours')

1 row created.

SQL> select * from uploads;

      VID      AID SINCE
-----
      1111      299 20 min
       473      299 5 days
      7538      299 1 hour
      2637      299 1 week
      7438      299 5 hours

SQL> 
```