

Video WebSite Database

Database Term Project – Oracle SQL



Team Purple

Eve Yifang Wang

Lou Alexander Salvador

Thea Alvarez

Contents

[project Introduction 3](#_Toc480102120)

[Business Rules (Based Crow’s Foot Notation) 3](#_Toc480102121)

[Database Structure 3](#_Toc480102122)

[User Table: 3](#_Toc480102123)

[Video Table: 4](#_Toc480102124)

[Channel Table: 5](#_Toc480102125)

[Playlist Table: 5](#_Toc480102126)

[Playlist Item Table: 5](#_Toc480102127)

[Subscription Table: 5](#_Toc480102128)

[Chatroom Table: 6](#_Toc480102129)

[Chatroom Member List Table: 6](#_Toc480102130)

[Implementation Coding 7](#_Toc480102131)

[Create Tables 7](#_Toc480102132)

[--1. User Table 7](#_Toc480102133)

[--2. Video Table 7](#_Toc480102134)

[--3. Channel 7](#_Toc480102135)

[--4. Playlist 8](#_Toc480102136)

[--5. Playlist Item 8](#_Toc480102137)

[--6. Subscription 8](#_Toc480102138)

[--7. Chatroom 9](#_Toc480102139)

[--8.Chatroom Member 9](#_Toc480102140)

[Foreign Key Constraints 9](#_Toc480102141)

[--User 9](#_Toc480102142)

[--Video 9](#_Toc480102143)

[--Playlist Item 9](#_Toc480102144)

[--Subscription 10](#_Toc480102145)

[--Chatroom Menber List 10](#_Toc480102146)

[Testing the Database – SQL Query 11](#_Toc480102147)

[--Video Length: Less Than 15 Minutes, More Than 45 Minutes 11](#_Toc480102148)

[--User And Their Video With Screening Status 11](#_Toc480102149)

[--Channel, Channel Owner And Its Videos 11](#_Toc480102150)

[--Playlist And Its Items 11](#_Toc480102151)

[--Chatroom And Its Members 11](#_Toc480102152)

[--Subscription 11](#_Toc480102153)

[--Channel 11](#_Toc480102154)

[--User 12](#_Toc480102155)

[Appendix A: ER Diagram 13](#_Toc480102156)

[Appendix B: SQL files 13](#_Toc480102157)

# Project Introduction

Team purple’s project is a video website database similar to that of YouTube. Our database shows the core build of a video hosting website which normally consist of the following: user accounts, channels, videos, playlists, chatrooms and/or comment section(we are using chatrooms for this specific project).

# Business Rules (Based Crow’s Foot notation)

* Each ***channel*** is owned by one or more ***user*** ***accounts,*** each ***user account***may own zero or one ***channel***
* Each ***channel*** can be subscribed to by one or more ***user account***s, each ***user account*** may subscribe to one or more ***channel***s
* Each ***channel*** may store one or more channel ***video***s, each channel ***video*** is stored in one or more ***channel***s
* Each ***playlist*** contains one or more playlist ***video***s, each playlist ***video*** is contained into one or more ***playlist***s
* Each ***user account*** may upload one or more ***video***s, each ***video***is uploaded by one ***user account***
* Each chatroom ***user*** may join one or more ***chatroom***s, each ***chatroom*** can be joined by one or more chatroom ***user***s
* Each ***user account*** may subscribe to one or more (other)***user account***s, each (other)***user account*** can be subscribed to one or many account ***user account***s

# Database Structure

## User Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| User ID | Uniquely identifies the account | Primary Key | VarChar2 | 7 |
| User Name | User’s name will be on the website | Not Null | VarChar2 | 20 |
| License | Approval that user is allowed to post on website  **Default: Personal** | Check  (LICENSE in('personal', 'business')) | VarChar2 | 15 |
| Channel ID | Channel associated to the account | Foreign Key | VarChar2 | 7 |
| Email | Email used to make the account | Unique key  Not Null | VarChar2 | 50 |

## Video Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| Video ID | Uniquely identifies the video | Primary Key | VarChar2 | 7 |
| Video Name | Name of the video | Not Null | VarChar2 | 20 |
| User ID | How the user table can be connected to video table | Foreign Key | VarChar2 | 7 |
| Channel ID | How the channel table can be connected to video table. Channel ID owns a video when it is uploaded | Foreign Key | VarChar2 | 7 |
| Category | Category of the video | Check  IN ('MUSIC', 'SPORTS', 'GAMING', 'FILMS', 'TV SHOWS', 'NEWS') | VarChar2 | 15 |
| Length | Use to store how many seconds the video has | Not Null | Number | 6 |
| Screening | Make sure the videos follow the guidelines  **Default: “N”** | Not Null  Check  IN ('Y', 'N') | VarChar2 | 1 |

## Channel Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| Channel ID | Uniquely identifies the channel | Primary Key | VarChar2 | 7 |
| Channel Name | Name of the channel | Not Null | VarChar2 | 20 |
| Category | Category of videos a channel provides | Check | VarChar2 | 15 |
| Description | What the channel will be about |  | VarChar2 | 140 |

## Playlist Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| Playlist ID | Uniquely identifies the playlist | Primary Key | VarChar2 | 7 |
| Playlist Name | Name of playlist | Not Null | VarChar2 | 20 |
| Category | Category of the video | Check | VarChar2 | 15 |

\**disabled when no foreign key used*

## Playlist Item Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| Playlist ID | Uniquely identifies the playlist | Primary Key  Foreign Key | VarChar2 | 7 |
| Video ID | Uniquely identifies the playlist. | Primary Key | VarChar2 | 7 |

## Subscription Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| User ID | Uniquely identifies subscription. Users who follows the channel | Primary Key  Foreign Key | VarChar2 | 7 |
| Followed ID | Uniquely identifies subscription. Users the channel is following | Primary Key | VarChar2 | 7 |

## Chatroom Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| Chatroom ID | Uniquely identifies the account | Primary Key | VarChar2 | 7 |
| Chatroom Name | Name of chatroom | Not Null | VarChar2 | 20 |
| Category | Category of what the chatroom is about |  | VarChar2 | 15 |

## Chatroom Member List Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/Constraint** | **Data Type** | **Filed Length** |
| Chatroom ID | Uniquely identifies the account | Primary Key  Foreign Key | VarChar2 | 7 |
| User ID | How the chatroom table can be connected to user table | Primary Key  Foreign Key | VarChar2 | 7 |

# Implementation Coding

## Create Tables

### --1. USER TABLE

CREATE TABLE “USER”(USER\_ID VARCHAR2(7),

USER\_NAME VARCHAR2(50) NOT NULL,

LICENSE VARCHAR2(15) DEFAULT 'PERSONAL',

CHANNEL\_ID VARCHAR2(7),

EMAIL VARCHAR2(20) NOT NULL,

CONSTRAINT USER\_ID\_PK PRIMARY KEY(USER\_ID),

CONSTRAINT LICENSE\_CHK CHECK(LICENSE IN ('PERSONAL', 'BUSINESS '))ENABLE,

CONSTRAINT EMAIL\_UK UNIQUE(EMAIL));

### --2. VIDEO TABLE

CREATE TABLE VIDEO

(VIDEO\_ID VARCHAR2(7),

CONSTRAINT VIDEO\_ID\_PK PRIMARY KEY (VIDEO\_ID),

VIDEO\_NAME VARCHAR2(20) NOT NULL,

USER\_ID VARCHAR2(7), --FK

CHANNEL\_ID VARCHAR2(7), --FK

CATEGORY VARCHAR2(15), --CK

CONSTRAINT VIDEO\_CATEGORY\_CK CHECK

(CATEGORY IN ('MUSIC', 'SPORTS', 'GAMING', 'FILMS', 'TV SHOWS', 'NEWS')),

LENGTH NUMBER(6) NOT NULL, --HOW MANY SECONDS THIS VIDEO HAS

SCREENING VARCHAR2(1) DEFAULT 'N',

CONSTRAINT VIDEO\_SCREENING\_CK CHECK (SCREENING IN ('Y', 'N')));

### --3. CHANNEL

CREATE TABLE CHANNEL

(CHANNEL\_ID VARCHAR2(7),

CHANNEL\_NAME VARCHAR2(20) NOT NULL,

CATEGORY VARCHAR2(15),

CONSTRAINT CHANNEL\_CATEGORY\_CK CHECK

(CATEGORY IN ('MUSIC', 'SPORTS', 'GAMING', 'FILMS', 'TV SHOWS', 'NEWS')),

DESCRIPTION VARCHAR2(140),

CONSTRAINT CHANNEL\_ID\_PK PRIMARY KEY(CHANNEL\_ID));

### --4. PLAYLIST

CREATE TABLE PLAYLIST

(PLAYLIST\_ID VARCHAR2(7),

PLAYLIST\_NAME VARCHAR(20) NOT NULL,

CATEGORY VARCHAR2(15),

CONSTRAINT PLAYLIST\_CATEGORY\_CK CHECK

(CATEGORY IN ('MUSIC', 'SPORTS', 'GAMING', 'FILMS', 'TV SHOWS', 'NEWS')),

CONSTRAINT PLAYLIST\_ID\_PK PRIMARY KEY (PLAYLIST\_ID));

### --5. PLAYLIST ITEM

CREATE TABLE PLAYLISTITEM

(PLAYLIST\_ID VARCHAR2(7),

VIDEO\_ID VARCHAR2(7),

CONSTRAINT PLAYLISTITEM\_ID\_PK PRIMARY KEY (PLAYLIST\_ID, VIDEO\_ID));

### --6. SUBSCRIPTION

CREATE TABLE SUBSCRIPTION

(USER\_ID VARCHAR2(7),

FOLLOWED\_ID VARCHAR2(7),

CONSTRAINT SUBSCRIPTION\_ID\_PK PRIMARY KEY (USER\_ID, FOLLOWED\_ID));

### --7. CHATROOM

CREATE TABLE CHATROOM

(CHATROOM\_ID VARCHAR2(7),

CONSTRAINT CHATR\_ID PRIMARY KEY (CHATROOM\_ID),

CHATROOM\_NAME VARCHAR2(20) NOT NULL,

CATEGORY VARCHAR2(15),

CONSTRAINT CHATR\_CATEGORY\_CK CHECK

(CATEGORY IN ('MUSIC', 'SPORTS', 'GAMING', 'FILMS', 'TV SHOWS', 'NEWS')));

### --8.CHATROOM MEMBER

CREATE TABLE CHATROOM\_MEMBER

(CHATROOM\_ID VARCHAR2(7),

USER\_ID VARCHAR2(7),

CONSTRAINT CHATMBR\_ID\_PK PRIMARY KEY (CHATROOM\_ID, USER\_ID));

## Foreign Key Constraints

### --USER

ALTER TABLE "USER" ADD CONSTRAINT USER\_CHANNELID\_FK FOREIGN KEY (Channel\_ID) REFERENCES CHANNEL (CHANNEL\_ID);

### --VIDEO

ALTER TABLE VIDEO ADD CONSTRAINT VIDEO\_USERID\_FK FOREIGN KEY (USER\_ID) REFERENCES "USER" (USER\_ID);

ALTER TABLE VIDEO ADD CONSTRAINT VIDEO\_CHANNELID\_FK FOREIGN KEY (CHANNEL\_ID) REFERENCES CHANNEL (CHANNEL\_ID);

### --PLAYLIST ITEM

ALTER TABLE PLAYLISTITEM ADD CONSTRAINT PLITEM\_PLAYLISTID\_FK FOREIGN KEY (PLAYLIST\_ID) REFERENCES PLAYLIST (PLAYLIST\_ID);

ALTER TABLE PLAYLISTITEM ADD CONSTRAINT PLITEM\_VIDEOID\_FK FOREIGN KEY (VIDEO\_ID) REFERENCES VIDEO (VIDEO\_ID);

### --SUBSCRIPTION

ALTER TABLE SUBSCRIPTION ADD CONSTRAINT SUBSCRIP\_USERID\_FK FOREIGN KEY (USER\_ID) REFERENCES "USER" (USER\_ID);

ALTER TABLE SUBSCRIPTION ADD CONSTRAINT SUBSCRIP\_FOLLOWED\_USER\_FK FOREIGN KEY (FOLLOWED\_ID) REFERENCES "USER" (USER\_ID);

ALTER TABLE SUBSCRIPTION ADD CONSTRAINT SUBSCRIP\_FOLLOWED\_CHANNEL\_FK FOREIGN KEY (FOLLOWED\_ID) REFERENCES CHANNEL (CHANNEL\_ID);

### --CHATROOM MENBER LIST

ALTER TABLE CHATROOM\_MEMBER ADD CONSTRAINT CHATMBR\_CHATRID\_FK FOREIGN KEY (CHATROOM\_ID) REFERENCES CHATROOM (CHATROOM\_ID);

ALTER TABLE CHATROOM\_MEMBER ADD CONSTRAINT CHATMBR\_USERID\_FK FOREIGN KEY (USER\_ID) REFERENCES "USER" (USER\_ID);

# Testing the Database – SQL Query

## --Video Length: Less Than 15 Minutes, More Than 45 Minutes

SELECT VIDEO\_NAME, TRUNC(LENGTH/60,0) || ':'|| MOD(LENGTH, 60) AS LENGTH FROM VIDEO WHERE LENGTH < (15\*60);

SELECT VIDEO\_NAME, TRUNC(LENGTH/60,0) || ':'|| MOD(LENGTH, 60) AS LENGTH FROM VIDEO WHERE LENGTH > (45\*60);

## --User And Their Video With Screening Status

SELECT USER\_NAME AS Contributor, VIDEO\_NAME, CATEGORY, TRUNC(LENGTH/60,0) || ':'|| MOD(LENGTH, 60) AS LENGTH, SCREENING

FROM "USER" NATURAL JOIN VIDEO ORDER BY USER\_NAME;

## --Channel, Channel Owner And Its Videos

SELECT C.CHANNEL\_NAME AS CHANNEL, U.USER\_NAME AS OWNER, V.VIDEO\_NAME AS VIDEOS, TRUNC(V.LENGTH/60,0) || ':'|| MOD(V.LENGTH, 60) AS LENGTH

FROM CHANNEL C, "USER" U, VIDEO V

WHERE C.CHANNEL\_ID = U.CHANNEL\_ID

AND C.CHANNEL\_ID = V.CHANNEL\_ID;

## --Playlist And Its Items

SELECT PLAYLIST\_NAME, VIDEO\_NAME, TRUNC(LENGTH/60,0) || ':'|| MOD(LENGTH, 60) AS LENGTH

FROM PLAYLIST NATURAL JOIN PLAYLISTITEM NATURAL JOIN VIDEOS;

## --Chatroom And Its Members

SELECT CHATROOM\_NAME, USER\_NAME FROM CHATROOM NATURAL JOIN CHATROOM\_MEMBER NATURAL JOIN "USER";

## --Subscription

### --Channel

SELECT C.CHANNEL\_NAME AS CHANNEL, U.USER\_NAME AS FOLLOWER

FROM SUBSCRIPTION S, CHANNEL C, "USER" U

WHERE S.FOLLOWED\_ID = C.CHANNEL\_ID

AND S.USER\_ID = U.USER\_ID;

### --User

SELECT U.USER\_NAME AS "USER", F.USER\_NAME AS FOLLOWER

FROM SUBSCRIPTION S, "USER" F, "USER" U

WHERE S.FOLLOWED\_ID = U.USER\_ID

AND S.USER\_ID = F.USER\_ID;

# Appendix A: ER Diagram

# Appendix B: SQL files

