University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___2_____

Date: __2021/Oct/24__________

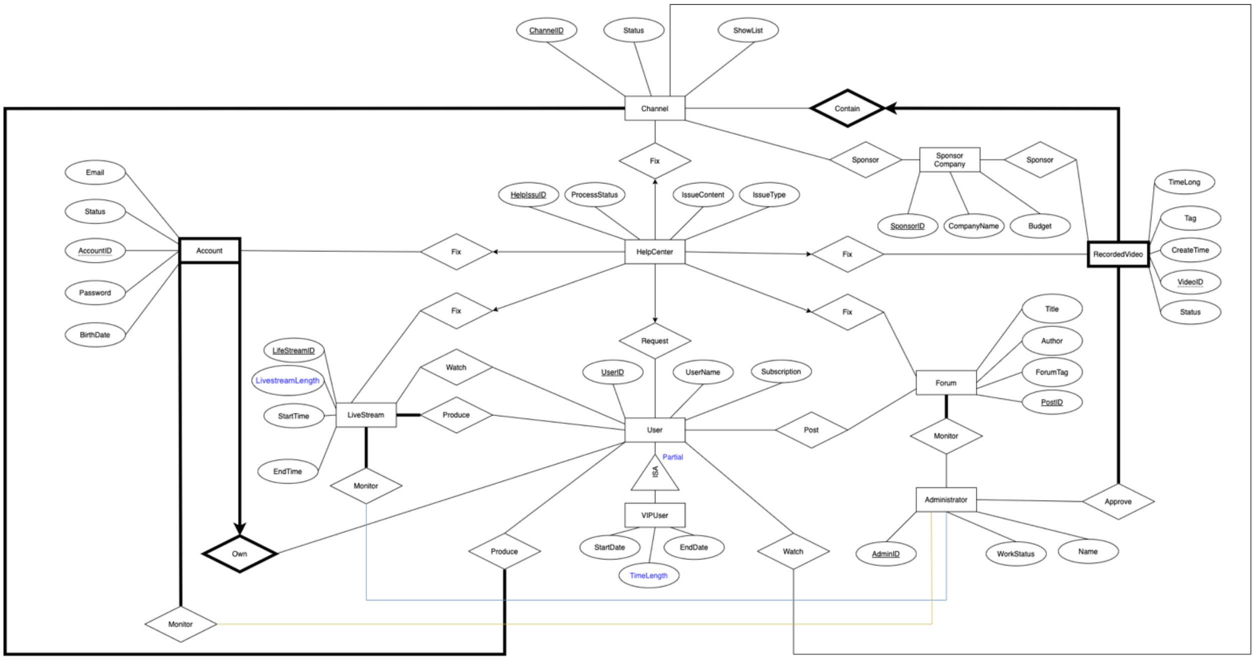
Group Number: ___52_______

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Yufei (Suki) Cai	25616533	x1k3b	yufeicaimail@gmail.com
Yixuan (Hanna) Li	58727413	e5v2m	xuanlee17@gmail.com
Qingxian (Andy) Liu	89451330	m3j3b	liu2001qingxian@163.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

ER Diagram



Update (Oct 24):

1. Add new attribute “LivestreamLength” to entity LiveStream to store the time length of livestreams.
2. Add new attribute “TimeLength” to entity LiveStream to store the time length of vip users.
3. Add “Partial” to ISA to denote that the ISA relationship between user and VIP user is partial.

Schema, FDs and Normalizations

channel(ChannelID: string, ChannelStatus: integer, ShowList: string)
Primary keys: ChannelID
Candidate keys: ChannelID
Foreign keys: N/A
FDs:
BCNF
ChannelID -> ChannelStatus, ShowList

forum(PostID: string, Title: string, Author: string, ForumTag: string)
Primary keys: PostID
Candidate keys: PostID
Foreign keys: N/A
FDs:
BCNF
PostID -> Title, Author, ForumTag

//explanation: the reason we change the user into ubcer is because the user has already been defined in sql.
ubcer(UserID: string, UserName: string, Subscription: string)
Primary keys: UserID
Candidate keys: UserID
Foreign keys: N/A
FDs:
BCNF
UserID -> UserName, Subscription

VIPUser(**UserID: string,** StartDate: date, EndDate: date, TimeLength: integer)
Primary keys: **UserID**
Candidate keys: **UserID**
Foreign keys: **UserID**
FDs:
2NF
UserID -> StartDate, EndDate, TimLength
StartDate, TimeLength -> EndDate
StartDate, EndDate -> TimeLength
// the valid time period of VIP user is one year

Normalization:
BCNF
vip_user_main(**UserID: string, Since: date, TimePeriod: integer**)
Primary keys: **UserID**

Candidate keys: **UserID**
Foreign keys: **UserID, Since, TimePeriod**

Taking out: StartDate, TimeLength -> EndDate
BCNF
vip_user_data(StartDate: date, TimeLength: integer, EndDate: date)
Primary keys: StartDate, TimeLong
Candidate keys: (StartDate, TimeLong), (StartDate,EndDate)
Foreign keys: **N/A**

LiveStream(LiveStreamID: string, StartTime: string, EndTime: string, PeriodLength: string)
Primary keys: LiveStreamID
Candidate keys: LiveStreamID
Foreign keys: N/A
FDs:
2NF
LiveStreamID -> StartTime, EndTime, PeriodLength
StartTime, PeriodLength -> EndTime
StartTime, EndTime -> PeriodLength

Normalization:
BCNF
LiveStream_main(LiveStreamID: string, **BeginTime: date, PeriodLength: string**)
Primary keys: LiveStreamID
Candidate keys: LiveStreamID
Foreign keys: **BeginTime, PeriodLength**

Taking out: StartTime, PeriodLength -> EndTime
BCNF
LiveStream_data(StartTime: date, LiveStreamLength: date, EndTime: string)
Primary keys: (StartTime, LiveStreamLength)
Candidate keys: (StartTime, LiveStreamLength)
Foreign keys: **N/A**

recorded_video_contain(VideoID: string, **ChannelID: string**, TimeLong: string, Tag: string, CreateTime: string, VideoStatus: integer)
Primary keys: (VideoID, **ChannelID**)
Candidate keys: (VideoID, **ChannelID**)
Foreign keys: **ChannelID**

FDs:
BCNF
VideoID, ChannelID -> TimeLong, Tag, CreateTime, VideoStatus

account_own(AccountID: string, **UserID: string**, AccountStatus: integer, Email: string, AccountPassword: string, BirthDate: date)
Primary keys: (AccountID, **UserID**)
Candidate keys: (AccountID, **UserID**), (Email, AccountPassword)
Foreign keys: **UserID**
FDs:
BCNF
AccountID, UserID -> AccountStatus, Email, AccountPassword, BirthDate
Email, AccountPassword -> AccountStatus, BirthDate, AccountID, UserID

//Assertion needed: all the channels, videos, accounts, posts and livestreams are fixed by help center.

helpcenter(HelpIssueID: string, **UserID: string, ChannelID: string, VideoID: string, AccountID: string, PostID: string, LiveStreamID: string,** ProcessStatus: integer ,IssueContent: string, IssueType: string)
Primary keys: HelpIssueID
Candidate keys: HelpIssueID
Foreign keys: **ChannelID, VideoID, AccountID, UserID, PostID, LiveStreamID**

//explanation: as all the “request” and “fix” relationships between “HelpCenter " entity and the other entites are similar, we combine them together to deduce the redundancy.
FDs:
BCNF
HelpIssueID -> ChannelID, VideoID, AccountID, PostID, LiveStreamID, ProcessStatus, IssueContent, IssueType

Administrator(AdminID: string, WorkStatus: integer, Name: string)
Primary keys: AdminID
Candidate keys: AdminID, Name
Foreign keys: N/A
FDs:
2NF
AdminID -> WorkStatus, Name
Name -> WorkStatus

Normalization:
BCNF
admini_main(AdminID: string, **AdminName: string**)
Primary keys: AdminID
Candidate keys: AdminID
Foreign keys: **AdminName**

Taking out: Name -> WorkStatus
BCNF
admini_status(WorkStatus: integer, AdminName: string)
Primary keys: AdminName
Candidate keys: AdminName
Foreign keys: N/A

sponsor_company(SponsorID: string, CompanyName: string, Budget: integer)
Primary keys: SponsorID
Candidate keys: SponsorID
Foreign keys: N/A
FDs:
BCNF
SponsorID -> CompanyName, Budget

Assertion needed: all the accounts are monitored by administrators
admini_monitor_account(**AdminID: string, AccountID: string, UserID: string**)
Primary keys: (**AdminID, AccountID, UserID)**
Candidate keys: (**AdminID, AccountID, UserID)**
Foreign keys: (**AdminID, AccountID, UserID)**
BCNF

Assertion needed: all the forums are monitored by administrators
admini_monitor_forum(**AdminID: string, PostID: string**)
Primary keys: (**AdminID, PostID)**
Candidate keys: (**AdminID, PostID)**
Foreign keys: (**AdminID, PostID)**
BCNF

Assertion needed: all the livestreams are monitored by administrators
admini_monitor_livestream(**AdminID: string, LiveStreamID: string**)
Primary keys: (**AdminID, LiveStreamID)**
Candidate keys: (**AdminID, LiveStreamID)**
Foreign keys: (**AdminID, LiveStreamID)**
BCNF

Assertion needed: all the videos are approved by administrators before they are published
admini_approve_recorded_video(**AdminID: string, VideoID:integer, ChannelID: string**)
Primary keys: (**AdminID, VideoID, ChannelID)**
Candidate keys: (**AdminID, VideoID, ChannelID)**
Foreign keys: (**AdminID, VideoID, ChannelID)**
BCNF

user_produce_livestream(**LiveStreamID: string, UserID: string**)
Primary keys: (**LiveStreamID, UserID)**
Candidate keys:(**LiveStreamID, UserID)**
Foreign keys: (**LiveStreamID, UserID)**
BCNF

user_produce_channel(**ChannelID: string, UserID: string**)
Primary keys: (**ChannelID, UserID)**
Candidate keys: (**ChannelID, UserID)**
Foreign keys: (**ChannelID, UserID)**
BCNF

user_watch_livestream(**LiveStreamID: string, UserID: string)**
Primary keys: (**LiveStreamID, UserID)**
Candidate keys: (**LiveStreamID, UserID)**
Foreign keys: (**LiveStreamID, UserID)**
BCNF

user_watch_channel(**ChannelID: string, UserID: string)**
Primary keys: (**ChannelID, UserID)**
Candidate keys: (**ChannelID, UserID)**
Foreign keys: (**ChannelID, UserID)**
BCNF

posted(**PostID: string, UserID: string)**
Primary keys: (**PostID, UserID)**
Candidate keys: (**PostID, UserID)**
Foreign keys: (**PostID, UserID)**
BCNF

sc_sponsor_recorded_video(**SponsorID: string, VideoID:integer, ChannelID: string**)
Primary keys: (**SponsorID, VideoID,ChannelID)**
Candidate keys: (**SponsorID, VideoID,ChannelID)**
Foreign keys: (**SponsorID, VideoID,ChannelID)**
BCNF

SCSponsorChannel(**SponsorID: string, ChannelID: string**)
Primary keys: (**SponsorID, ChannelID)**
Candidate keys: (**SponsorID, ChannelID)**
Foreign keys: (**SponsorID, ChannelID)**
BCNF

SQL DDL & Insertions

```
drop table SCSponsorChannel;
drop table sc_sponsor_recorded_video;
drop table posted;
drop table user_watch_channel;
drop table user_watch_livestream;
drop table user_produce_channel;
drop table user_produce_livestream;
drop table admini_approve_recorded_video;
drop table admini_monitor_livestream;
drop table admini_monitor_forum;
drop table admini_monitor_account;
drop table sponsor_company CASCADE CONSTRAINT;
drop table admini_main;
drop table admini_status;
drop table helpcenter;
drop table account_own CASCADE CONSTRAINT;
drop table recorded_video_contain CASCADE CONSTRAINT;
drop table Livestream_main CASCADE CONSTRAINT;
drop table Livestream_data CASCADE CONSTRAINT;
drop table vip_user_main;
drop table vip_user_data;
drop table ubcer CASCADE CONSTRAINT;
drop table forum CASCADE CONSTRAINT;
drop table channel CASCADE CONSTRAINT;

CREATE TABLE channel (
    ChannelID Varchar(10) PRIMARY KEY,
    ChannelStatus Integer,
    Showlist Varchar(100));
grant select on channel to public;

CREATE TABLE forum (
    PostID Varchar(10) PRIMARY KEY,
    ForumTag Varchar(20),
   Author Varchar(20),
    Title Varchar(20));
grant select on forum to public;

CREATE TABLE ubcer (
    UserID Varchar(10) PRIMARY KEY,
    UserName Varchar(20),
    Subscription Varchar(1000));
grant select on ubcer to public;

CREATE TABLE vip_user_data (
    TimeLength Integer,
    StartDate Date,
    EndDate Date,
    PRIMARY KEY (TimeLength, StartDate));
grant select on vip_user_data to public;
```

```
CREATE TABLE vip_user_main (
   UserID Varchar(10) PRIMARY KEY,
   Timeperiod Integer,
   Since Date,
   FOREIGN KEY (UserID) REFERENCES ubcer
   ON DELETE CASCADE,
   --ON UPDATE CASCADE
   FOREIGN KEY (Timeperiod, Since) REFERENCES vip_user_data
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on vip_user_main to public;

CREATE TABLE Livestream_data (
   LivestreamLength Integer,
   StartTime ,
   EndTime ,
   PRIMARY KEY (LivestreamLength, StartTime));
grant select on Livestream_data to public;

CREATE TABLE Livestream_main (
   LiveStreamID Varchar(10) PRIMARY KEY,
   PeriodLength Integer,
   BeginTime ,
   FOREIGN KEY (PeriodLength, BeginTime) REFERENCES Livestream_data
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on Livestream_main to public;

CREATE TABLE recorded_video_contain (
   VideoID Varchar(10),
   ChannelID Varchar(10),
   VideoLength Varchar(30),
   Tag Varchar(20),
   CreateTime Varchar(40),
   VideoStatus Integer,
   PRIMARY KEY(VideoID, ChannelID),
   FOREIGN KEY (ChannelID) REFERENCES channel
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on recorded_video_contain to public;

CREATE TABLE account_own (
   AccountID Varchar(10),
   UserID Varchar(10),
   AccountStatus Integer,
   Email Varchar(50),
   AccountPassword Varchar(100),
   BirthDate Date,
   PRIMARY KEY (AccountID, UserID),
   FOREIGN KEY (UserID) REFERENCES ubcer
   ON DELETE CASCADE);
```

```
    --ON UPDATE CASCADE);
grant select on account_own to public;

CREATE TABLE helpcenter (
    HelpIssueID Varchar(10) PRIMARY KEY,
    ChannelID Varchar(10),
    UserID Varchar(10),
    VideoID Varchar(10),
    AccountID Varchar(10),
    PostID Varchar(10),
    LiveStreamID Varchar(10),
    ProcessStatus Integer,
    IssueContent Varchar(1000),
    IssueType Varchar(20),
    FOREIGN KEY (VideoID, ChannelID) REFERENCES recorded_video_contain,
    FOREIGN KEY (AccountID, UserID) REFERENCES account_own,
    FOREIGN KEY (PostID) REFERENCES forum,
    FOREIGN KEY (LiveStreamID ) REFERENCES Livestream_main);
grant select on helpcenter to public;

CREATE TABLE admini_status (
    AdminName Varchar(20) PRIMARY KEY,
    WorkStatus Integer);
grant select on admini_status to public;

CREATE TABLE admini_main (
    AdminID Varchar(10) PRIMARY KEY,
    AdminstratorName Varchar(20),
    FOREIGN KEY (AdminstratorName) REFERENCES admini_status
    ON DELETE CASCADE);
    --ON UPDATE CASCADE);
grant select on admini_main to public;

CREATE TABLE sponsor_company (
    SponsorID Varchar(10) PRIMARY KEY,
    CompanyName Char(20),
    Budget Integer);
grant select on sponsor_company to public;

CREATE TABLE admini_monitor_account (
    AdminID Varchar(10),
    AccountID Varchar(10),
    UserID Varchar(10),
    PRIMARY KEY (AdminID, AccountID, UserID),
    FOREIGN KEY (AdminID) REFERENCES admini_main
    ON DELETE CASCADE,
    --ON UPDATE CASCADE
    FOREIGN KEY (UserID, AccountID) REFERENCES account_own
    ON DELETE CASCADE);
    --ON UPDATE CASCADE);
grant select on admini_monitor_account to public;
```

```
CREATE TABLE admini_monitor_forum (
   AdminID Varchar(10),
   PostID Varchar(10),
   PRIMARY KEY (AdminID, PostID),
   FOREIGN KEY (AdminID) REFERENCES admini_main
   ON DELETE CASCADE,
   --ON UPDATE CASCADE
   FOREIGN KEY (PostID ) REFERENCES forum
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on admini_monitor_forum to public;

CREATE TABLE admini_monitor_livestream (
   AdminID Varchar(10),
   LiveStreamID Varchar(10),
   PRIMARY KEY (AdminID, LiveStreamID),
   FOREIGN KEY (AdminID) REFERENCES admini_main
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (LiveStreamID) REFERENCES Livestream_main
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on admini_monitor_livestream to public;

CREATE TABLE admini_approve_recorded_video (
   AdminID Varchar(10),
   VideoID Varchar(10),
   ChannelID Varchar(10),
   PRIMARY KEY (AdminID, VideoID, ChannelID),
   FOREIGN KEY (AdminID) REFERENCES admini_main
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (VideoID, ChannelID) REFERENCES recorded_video_contain
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on admini_approve_recorded_video to public;

CREATE TABLE user_produce_livestream (
   LiveStreamID Varchar(10),
   UserID Varchar(10),
   PRIMARY KEY (LiveStreamID, UserID),
   FOREIGN KEY (UserID) REFERENCES ubcer
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (LiveStreamID) REFERENCES Livestream_main
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on user_produce_livestream to public;

CREATE TABLE user_produce_channel (
   ChannelID Varchar(10),
   UserID Varchar(10),
   PRIMARY KEY (ChannelID, UserID),
   FOREIGN KEY (UserID ) REFERENCES ubcer
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (ChannelID) REFERENCES channel
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on user_produce_channel to public;

CREATE TABLE user_watch_livestream (
   LiveStreamID Varchar(10),
   UserID Varchar(10),
   PRIMARY KEY (LiveStreamID, UserID),
   FOREIGN KEY (UserID) REFERENCES ubcer
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (LiveStreamID) REFERENCES Livestream_main
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on user_watch_livestream to public;

CREATE TABLE user_watch_channel (
   ChannelID Varchar(10),
   UserID Varchar(10),
   PRIMARY KEY (ChannelID, UserID),
   FOREIGN KEY (UserID) REFERENCES ubcer
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (ChannelID) REFERENCES channel
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on user_watch_channel to public;

CREATE TABLE posted (
   UserID Varchar(10),
   PostID Varchar(10),
   PRIMARY KEY (UserID, PostID),
   FOREIGN KEY (UserID) REFERENCES ubcer
   ON DELETE CASCADE,
   --ON UPDATE CASCADE,
   FOREIGN KEY (PostID) REFERENCES forum
   ON DELETE CASCADE);
   --ON UPDATE CASCADE);
grant select on posted to public;

CREATE TABLE sc_sponsor_recorded_video (
   VideoID Varchar(10),
   SponsorID Varchar(10),
   ChannelID Varchar(10),
   PRIMARY KEY (VideoID, SponsorID, ChannelID),
   FOREIGN KEY (SponsorID) REFERENCES sponsor_company
```

```
    ON DELETE CASCADE,
    --ON UPDATE CASCADE,
    FOREIGN KEY (VideoID, ChannelID) REFERENCES recorded_video_contain
    ON DELETE CASCADE);
    --ON UPDATE CASCADE);
grant select on sc_sponsor_recorded_video to public;

CREATE TABLE SCSponsorChannel  (
    ChannelID Varchar(10),
    SponsorID Varchar(10),
    PRIMARY KEY(ChannelID, SponsorID),
    FOREIGN KEY (ChannelID) REFERENCES channel
    ON DELETE CASCADE,
    --ON UPDATE CASCADE,
    FOREIGN KEY (SponsorID) REFERENCES sponsor_company
    ON DELETE CASCADE);
    --ON UPDATE CASCADE);
grant select on SCSponsorChannel to public;

insert into channel
values('C-00000001', 1, 'V-00000001, V-00000002, V-00000003, V-00000004, V-00000005');
insert into channel
values('C-00000002', 0, 'V-00000001, V-00000002, V-00000003, V-00000004');
insert into channel
values('C-00000003', 0, 'V-00000001, V-00000002, V-00000003');
insert into channel
values('C-00000004', 1, 'V-00000001, V-00000002');
insert into channel
values('C-00000005', 1, 'V-00000001, V-00000006');
insert into channel
values('C-00000006', 1, 'V-00000002, V-00000003');

insert into forum
values('P-00000001', 'music', 'Taylor Swift', 'Love Story');
insert into forum
values('P-00000002', 'TV', 'Chace Gonzalez', 'Talent Show');
insert into forum
values('P-00000003', 'meal', 'Gorden Ramsay', 'Wonderful Steak');
insert into forum
values('P-00000004', 'learnning', 'Joma Tech', 'Leetcode Solution');
insert into forum
values('P-00000005', 'sport', 'Joel Levy', 'Swimming technique');
insert into forum
values('P-00000006', 'sport', 'Missi Ivy', 'Soccer technique');

insert into ubcer
values('U-00000001', 'Taylor Swift', 'C-00000001, C-00000002, C-00000003, C-00000004, C-00000005');
insert into ubcer
values('U-00000002', 'Chace Gonzalez', 'C-00000001, C-00000002, C-00000003, C-00000004');
insert into ubcer
values('U-00000003', 'Gorden Ramsay', 'C-00000001, C-00000002, C-00000003');
insert into ubcer
```

```
values('U-00000004', 'Joma Tech', 'C-00000001, C-00000002');
insert into ubcer
values('U-00000005', 'Joel Levy', 'C-00000001');

insert into vip_user_data
values(3, TO_DATE('2021-10-23', 'yyyy-mm-dd'), TO_DATE('2021-10-26', 'yyyy-mm-dd'));
insert into vip_user_data
values(6, TO_DATE('2021-11-23', 'yyyy-mm-dd'), TO_DATE('2021-11-29', 'yyyy-mm-dd'));
insert into vip_user_data
values(9, TO_DATE('2021-12-23', 'yyyy-mm-dd'), TO_DATE('2022-01-02', 'yyyy-mm-dd'));
insert into vip_user_data
values(12, TO_DATE('2022-01-23', 'yyyy-mm-dd'), TO_DATE('2021-01-30', 'yyyy-mm-dd'));
insert into vip_user_data
values(1, TO_DATE('2022-02-23', 'yyyy-mm-dd'), TO_DATE('2022-02-24', 'yyyy-mm-dd'));

insert into vip_user_main
values('U-00000001', 3, TO_DATE('2021-10-23', 'yyyy-mm-dd'));
insert into vip_user_main
values('U-00000002', 6, TO_DATE('2021-11-23', 'yyyy-mm-dd'));
insert into vip_user_main
values('U-00000003', 9, TO_DATE('2021-12-23', 'yyyy-mm-dd'));
insert into vip_user_main
values('U-00000004', 12, TO_DATE('2022-01-23', 'yyyy-mm-dd'));
insert into vip_user_main
values('U-00000005', 1, TO_DATE('2022-02-23', 'yyyy-mm-dd'));

insert into Livestream_data
values (20, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'), TO_DATE('2021-10-23 10:00:20',
'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_data
values (30, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'), TO_DATE('2021-10-23 10:00:30',
'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_data
values (40, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'), TO_DATE('2021-10-23 10:00:40',
'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_data
values (50, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'), TO_DATE('2021-10-23 10:00:50',
'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_data
values (60, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'), TO_DATE('2021-10-23 10:00:00',
'yyyy-mm-dd HH:MI:SS'));

insert into Livestream_main
values ('L-00000001', 20, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_main
values ('L-00000002', 30, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_main
values ('L-00000003', 20, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_main
values ('L-00000004', 20, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'));
insert into Livestream_main
values ('L-00000005', 20, TO_DATE('2021-10-23 10:00:00', 'yyyy-mm-dd HH:MI:SS'));
```

```
insert into recorded_video_contain
values('V-00000001', 'C-00000001', '00:10:00','music', '2021-10-23 14:00:00', 1);
insert into recorded_video_contain
values('V-00000002', 'C-00000002', '00:10:00','TV', '2021-10-23 15:00:00', 0);
insert into recorded_video_contain
values('V-00000003', 'C-00000003', '00:10:00','meal', '2021-10-23 16:00:00', 1);
insert into recorded_video_contain
values('V-00000004', 'C-00000004', '00:10:00','learnning', '2021-10-23 17:00:00', 0);
insert into recorded_video_contain
values('V-00000005', 'C-00000005', '00:10:00','sport', '2021-10-23 18:00:00', 0);
insert into recorded_video_contain
values('V-00000006', 'C-00000005', '00:10:00','game', '2021-10-23 19:00:00', 1);

insert into account_own
values('AC-0000001', 'U-00000001', 0, 'yufeicaimail0@gmail.com', '12345678', TO_DATE('2001-08-31',
'yyyy-mm-dd'));
insert into account_own
values('AC-0000002', 'U-00000002', 1, 'yufeicaimail1@gmail.com', '1234567', TO_DATE('2001-09-30',
'yyyy-mm-dd'));
insert into account_own
values('AC-0000003', 'U-00000003', 1, 'yufeicaimail2@gmail.com', '123456', TO_DATE('2001-10-31',
'yyyy-mm-dd'));
insert into account_own
values('AC-0000004', 'U-00000004', 0, 'yufeicaimail3@gmail.com', '12345', TO_DATE('2001-11-30',
'yyyy-mm-dd'));
insert into account_own
values('AC-0000005', 'U-00000005', 0, 'yufeicaimail4@gmail.com', '1234', TO_DATE('2001-12-31',
'yyyy-mm-dd'));
insert into account_own
values('AC-0000006', 'U-00000005', 0, 'yufeicaimail5@gmail.com', '123', TO_DATE('2000-01-31',
'yyyy-mm-dd'));

insert into helpcenter
values ('H-00000001', 'C-00000001', 'U-00000001', 'V-00000001', 'AC-0000001', 'P-00000001',
'L-00000001', 0, 'channel content gone', 'channel-Content');
insert into helpcenter
values ('H-00000002', 'C-00000001', 'U-00000002', 'V-00000002', 'AC-0000002', 'P-00000002',
'L-00000002', 1, 'Can not create mutiple accounts', 'ubcer-Account');
insert into helpcenter
values ('H-00000003', 'C-00000003', 'U-00000003', 'V-00000003', 'AC-0000002', 'P-00000003',
'L-00000003', 0, 'Video are blocked', 'Video-Permission');
insert into helpcenter
values ('H-00000004', 'C-00000004', 'U-00000004', 'V-00000003', 'AC-0000004', 'P-00000004',
'L-00000004', 1, 'Log in failed', 'Account-LogIn');
insert into helpcenter
values ('H-00000005', 'C-00000005', 'U-00000005', 'V-00000005', 'AC-0000005', 'P-00000004',
'L-00000005', 1, 'Reporting racially discriminatory posted', 'posted-AgressiveContent');
insert into helpcenter
values ('H-00000006', 'C-00000006', 'U-00000005', 'V-00000006', 'AC-0000006', 'P-00000006',
'L-00000003', 0, 'Livestream cannot be recorded', 'Livestream-Record');
```

```
insert into admini_status
values('Valdi Gigges', 1);
insert into admini_status
values('Martin Potter', 0);
insert into admini_status
values('Angelina Watson', 1);
insert into admini_status
values('Chris Pang', 1);
insert into admini_status
values('David Hoffman', 0);
insert into admini_status
values('Eliza Gates', 1);
insert into admini_status
values('Sanai Banks', 1);

insert into admini_main
values('AD-0000001','Valdi Gigges');
insert into admini_main
values('AD-0000002','Martin Potter');
insert into admini_main
values('AD-0000003','Angelina Watson');
insert into admini_main
values('AD-0000004','Chris Pang');
insert into admini_main
values('AD-0000005','David Hoffman');
insert into admini_main
values('AD-0000006','Eliza Gates');
insert into admini_main
values('AD-0000007','Sanai Banks');

insert into sponsor_company
values('S-00000001','Addidas',10000);
insert into sponsor_company
values('S-00000002','Nike', 10000);
insert into sponsor_company
values('S-00000003','Blue Navy', null);
insert into sponsor_company
values('S-00000004','Puma', 2000);
insert into sponsor_company
values('S-00000005','Pop Mart', 1);

insert into admini_monitor_account
values('AD-0000001','AC-0000001','U-00000001');
insert into admini_monitor_account
values('AD-0000001','AC-0000006','U-00000005');
insert into admini_monitor_account
values('AD-0000002','AC-0000001','U-00000001');
insert into admini_monitor_account
values('AD-0000003','AC-0000002','U-00000002');
insert into admini_monitor_account
values('AD-0000004','AC-0000003','U-00000003');
insert into admini_monitor_account
```

```
values('AD-0000006','AC-0000004','U-00000004');
insert into admini_monitor_account
values('AD-0000007','AC-0000005','U-00000005');

insert into admini_monitor_forum
values('AD-0000001','P-00000001');
insert into admini_monitor_forum
values('AD-0000002','P-00000001');
insert into admini_monitor_forum
values('AD-0000002','P-00000002');
insert into admini_monitor_forum
values('AD-0000003','P-00000003');
insert into admini_monitor_forum
values('AD-0000004','P-00000004');
insert into admini_monitor_forum
values('AD-0000005','P-00000005');
insert into admini_monitor_forum
values('AD-0000003','P-00000006');

insert into admini_monitor_livestream
values('AD-0000001','L-00000001');
insert into admini_monitor_livestream
values('AD-0000002','L-00000001');
insert into admini_monitor_livestream
values('AD-0000002','L-00000002');
insert into admini_monitor_livestream
values('AD-0000003','L-00000003');
insert into admini_monitor_livestream
values('AD-0000004','L-00000004');
insert into admini_monitor_livestream
values('AD-0000005','L-00000005');

insert into admini_approve_recorded_video
values('AD-0000001','V-00000001','C-00000001');
insert into admini_approve_recorded_video
values('AD-0000007','V-00000001','C-00000001');
insert into admini_approve_recorded_video
values('AD-0000002','V-00000002','C-00000002');
insert into admini_approve_recorded_video
values('AD-0000003','V-00000003','C-00000003');
insert into admini_approve_recorded_video
values('AD-0000004','V-00000004','C-00000004');
insert into admini_approve_recorded_video
values('AD-0000005','V-00000005','C-00000005');
insert into admini_approve_recorded_video
values('AD-0000001','V-00000006','C-00000005');

insert into user_produce_livestream
values('L-00000001','U-00000001');
insert into user_produce_livestream
values('L-00000002','U-00000001');
insert into user_produce_livestream
```

```
values('L-00000003','U-00000003');
insert into user_produce_livestream
values('L-00000004','U-00000005');
insert into user_produce_livestream
values('L-00000004','U-00000001');
insert into user_produce_livestream
values('L-00000004','U-00000002');
insert into user_produce_livestream
values('L-00000005','U-00000005');

insert into user_produce_channel
values('C-00000001','U-00000001');
insert into user_produce_channel
values('C-00000002','U-00000001');
insert into user_produce_channel
values('C-00000003','U-00000003');
insert into user_produce_channel
values('C-00000004','U-00000005');
insert into user_produce_channel
values('C-00000004','U-00000001');
insert into user_produce_channel
values('C-00000004','U-00000002');
insert into user_produce_channel
values('C-00000005','U-00000005');

insert into user_watch_livestream
values('L-00000002','U-00000001');
insert into user_watch_livestream
values('L-00000003','U-00000001');
insert into user_watch_livestream
values('L-00000001','U-00000002');
insert into user_watch_livestream
values('L-00000001','U-00000001');
insert into user_watch_livestream
values('L-00000001','U-00000005');
insert into user_watch_livestream
values('L-00000005','U-00000005');

insert into user_watch_channel
values('C-00000001','U-00000001');
insert into user_watch_channel
values('C-00000005','U-00000001');
insert into user_watch_channel
values('C-00000003','U-00000002');
insert into user_watch_channel
values('C-00000003','U-00000001');
insert into user_watch_channel
values('C-00000003','U-00000005');
insert into user_watch_channel
values('C-00000002','U-00000005');

insert into posted
```

```
values('U-00000001','P-00000001');
insert into posted
values('U-00000005','P-00000001');
insert into posted
values('U-00000001','P-00000002');
insert into posted
values('U-00000003','P-00000003');
insert into posted
values('U-00000002','P-00000004');
insert into posted
values('U-00000004','P-00000005');
insert into posted
values('U-00000001','P-00000006');

insert into sc_sponsor_recorded_video
values('V-00000001','S-00000005', 'C-00000001');
insert into sc_sponsor_recorded_video
values('V-00000003','S-00000005', 'C-00000003');
insert into sc_sponsor_recorded_video
values('V-00000004','S-00000005', 'C-00000004');
insert into sc_sponsor_recorded_video
values('V-00000004','S-00000001', 'C-00000004');
insert into sc_sponsor_recorded_video
values('V-00000004','S-00000002', 'C-00000004');


insert into SCSponsorChannel
values('C-00000001','S-00000005');
insert into SCSponsorChannel
values('C-00000003','S-00000005');
insert into SCSponsorChannel
values('C-00000004','S-00000005');
insert into SCSponsorChannel
values('C-00000004','S-00000001');
insert into SCSponsorChannel
values('C-00000004','S-00000003');
```

List of all the variety of queries

Insertion:

1. Add more than one accounts for a user.
2. Add more than one videos into a channel.
3. Add subscribed channels IDs into user’s subscription list.
4. Add more than one posts into a forum.

Delete Operation:

1. User can inactive an account.
2. Help center can delete help issues after they are fully processed.

Update Operation:

1. Administrators update work status when they start working.
2. Help center updates process status of each help issue when a help issue is received or fully processed.

Selection:

1. Select all “Video-Permission” problem from “HelpCenterFix” table to gather all similar problems.
2. Select all “game” type of forum to better find user’s interest later.

Projection:

Select “HelpIssueID”, “UserID” and “ProcessStatus” to rank the highest priority among help requests (in an ascending order of HelpIssueID) to deal with all the problems efficiently.