Databázové systémy 1
Projekt – YouTube-like platforma

Jméno: Phat Tran Dai

Osobní číslo: TRA0163

Datum: 07. 12. 2024

Obsah

Relační model

1.1. Grafický tvar relačního datového modelu
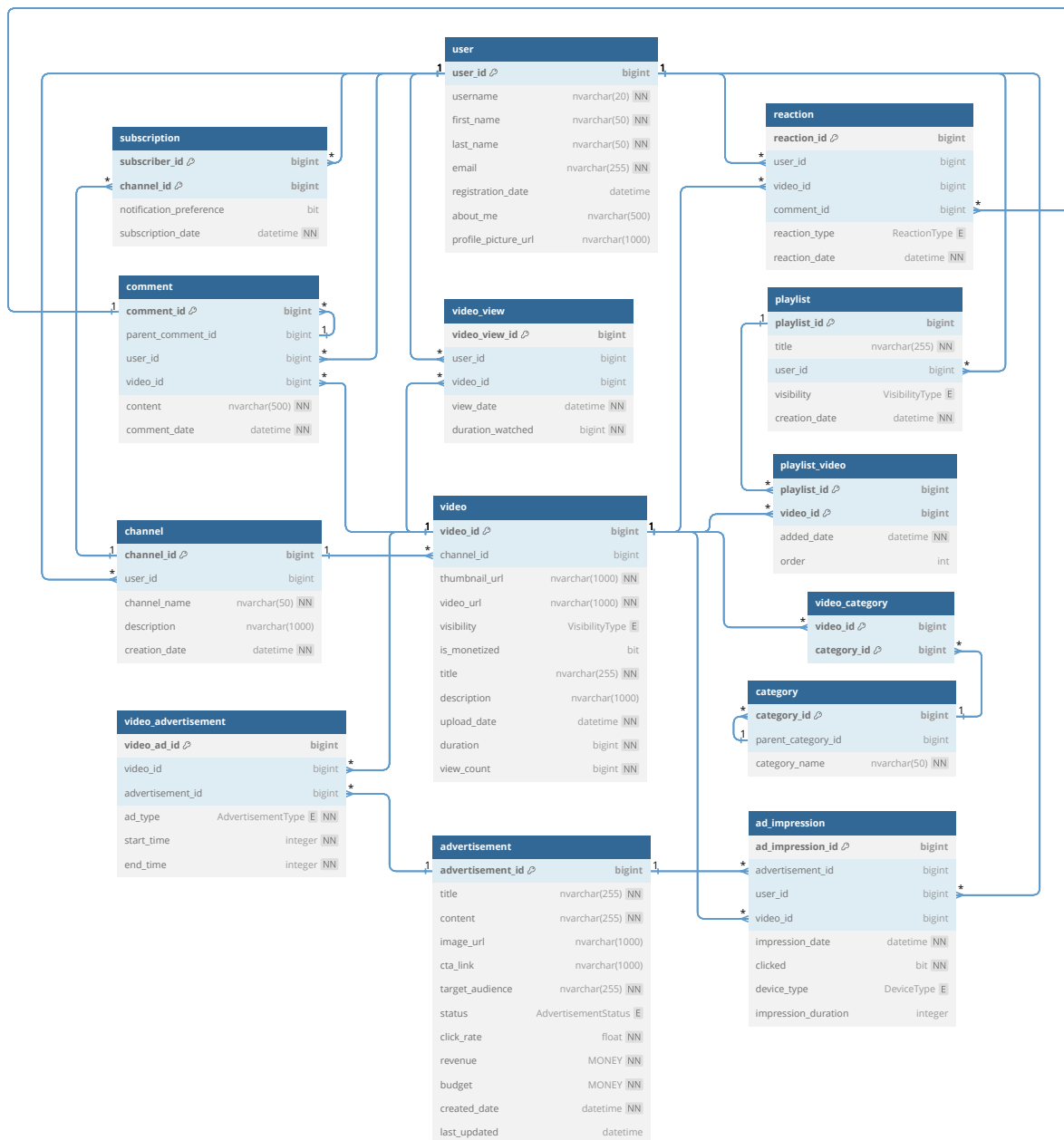
Obrázek 1: Relační model

1.2. Popis tabulek

- **_user_** - Záznam reprezentuje jednoho uživatele s osobními informacemi, jako je uživatelské jméno, e-mail, křestní a příjmení, datum registrace, URL profilového obrázku a další detaily, jako je sekce „o mně".

- **_channel_** - Záznam reprezentuje jeden kanál vlastněný uživatelem. Obsahuje název kanálu, popis a datum vytvoření, spolu s referencí na uživatele, který kanál vlastní.

- **_video_** - Záznam reprezentuje jedno video nahrané na kanál. Obsahuje detaily, jako je název videa, popis, URL, náhledový obrázek, nastavení viditelnosti, status monetizace, datum nahrání, délka a počet zhlédnutí, spolu s referencí na kanál, ke kterému video patří.

- **_playlist_** - Záznam reprezentuje jeden playlist vytvořený uživatelem. Obsahuje název playlistu, uživatele, který playlist vytvořil, a nastavení viditelnosti playlistu, spolu s datem vytvoření.

- **_playlist_video_** - Záznam reprezentuje video v playlistu. Spojuje konkrétní video s konkrétním playlistem a uchovává datum, kdy bylo video do playlistu přidáno.

- **_comment_** - Záznam reprezentuje jediný komentář napsaný uživatelem k videu. Obsahuje text komentáře, datum, a referenci na uživatele, který komentář napsal, a video, ke kterému byl komentář přidán. Může mít také referenci na nadřazený komentář, pokud se jedná o odpověď.

- **_reaction_** - Záznam reprezentuje reakci uživatele na cílový prvek (video, komentář nebo příspěvek). Obsahuje typ reakce (like, dislike, love), datum reakce, a referenci na cíl (video `video_id` nebo komentář `comment_id`) a uživatele, který reagoval.

- **_subscription_** - Záznam reprezentuje odběr uživatele na kanál. Obsahuje referenci na odběratele (uživatele) a kanál, na který se uživatel přihlásil, spolu s preferencemi pro oznámení a datem přihlášení k odběru.

- **_video_view_** - Záznam reprezentuje zhlédnutí videa uživatelem. Uchovává délku zhlédnutého času, datum zhlédnutí a referenci na video a uživatele, který video zhlédl.

- **_video_category_** - Záznam reprezentuje přiřazení kategorie k videu.

- **_category_** - Záznam reprezentuje jednu kategorii videí. Může mít nadřazenou kategorii (vytváří hierarchickou strukturu) a obsahuje název kategorie.

- **_advertisement_** - Záznam reprezentuje jednu reklamu s jejím obsahem, cílovou skupinou, URL obrázku, odkazem na CTA (Call to Action), stavem (aktivní, neaktivní, vypršelo), mírou prokliku, příjmem a rozpočtem.

- **_video_advertisement_** - Záznam reprezentuje umístění reklamy ve videu. Spojuje video s reklamou, specifikuje typ reklamy (pre-roll, mid-roll, post-roll nebo banner) a čas začátku a konce reklamy.

- **_ad_impression_** - Záznam reprezentuje zobrazení reklamy uživatelem ve videu. Obsahuje reklamu, uživatele, který reklamu viděl, video, ve kterém se reklama objevila, datum a čas zobrazení, zařízení, které uživatel použil, a informaci o tom, zda byla reklama zakliknuta.

Skripty

2.1. DDL Skript

```sql
1   CREATE TABLE [y_user] (
2       [user_id] bigint PRIMARY KEY IDENTITY(1, 1),
3       [username] nvarchar(20) UNIQUE NOT NULL,
4       [first_name] nvarchar(50) NOT NULL,
5       [last_name] nvarchar(50) NOT NULL,
6       [email] nvarchar(255) UNIQUE NOT NULL,
7       [registration_date] datetime DEFAULT (GETDATE()),
8       [about_me] nvarchar(500),
9       [profile_picture_url] nvarchar(1000)
10  )
11  GO
12
13  CREATE TABLE [y_channel] (
14      [channel_id] bigint PRIMARY KEY IDENTITY(1, 1),
15      [user_id] bigint,
16      [channel_name] nvarchar(50) UNIQUE NOT NULL,
17      [description] nvarchar(1000),
18      [creation_date] datetime NOT NULL DEFAULT (GETDATE())
19  )
20  GO
21
22  CREATE TABLE [y_video] (
23      [video_id] bigint PRIMARY KEY IDENTITY(1, 1),
24      [channel_id] bigint,
25      [thumbnail_url] nvarchar(1000) UNIQUE NOT NULL,
26      [video_url] nvarchar(1000) UNIQUE NOT NULL,
27      [visibility] nvarchar(255) NOT NULL
28          CHECK ([visibility] IN ('private', 'public', 'hidden')) DEFAULT 'public',
29      [is_monetized] bit DEFAULT (1),
30      [title] nvarchar(255) NOT NULL,
31      [description] nvarchar(1000),
32      [upload_date] datetime NOT NULL DEFAULT (GETDATE()),
33      [duration] bigint NOT NULL,
34      [view_count] bigint NOT NULL
35  )
36  GO
37
38  CREATE TABLE [y_playlist] (
39      [playlist_id] bigint PRIMARY KEY IDENTITY(1, 1),
40      [title] nvarchar(255) NOT NULL,
41      [user_id] bigint,
42      [visibility] nvarchar(255) NOT NULL
43          CHECK ([visibility] IN ('private', 'public', 'hidden')) DEFAULT 'public',
44      [creation_date] datetime NOT NULL DEFAULT (GETDATE())
45  )
46  GO
47
48  CREATE TABLE [y_playlist_video] (
49      [playlist_id] bigint,
50      [video_id] bigint,
51      [added_date] datetime NOT NULL DEFAULT (GETDATE()),
52      [order] int,
53      PRIMARY KEY ([playlist_id], [video_id])
54  )
55  GO
56
57  CREATE TABLE [y_comment] (
58      [comment_id] bigint PRIMARY KEY IDENTITY(1, 1),
59      [parent_comment_id] bigint,
60      [user_id] bigint,
61      [video_id] bigint,
62      [content] nvarchar(500) NOT NULL,
```

```sql
        [comment_date] datetime NOT NULL DEFAULT (GETDATE())
    )
    GO

    CREATE TABLE [y_reaction] (
        [reaction_id] bigint PRIMARY KEY IDENTITY(1, 1),
        [user_id] bigint,
        [video_id] bigint,
        [comment_id] bigint,
        [reaction_type] nvarchar(255) NOT NULL
            CHECK ([reaction_type] IN ('like', 'dislike', 'love')),
        [reaction_date] datetime NOT NULL DEFAULT (GETDATE())
    )
    GO

    CREATE TABLE [y_subscription] (
        [subscriber_id] bigint,
        [channel_id] bigint,
        [notification_preference] bit DEFAULT (1),
        [subscription_date] datetime NOT NULL DEFAULT (GETDATE()),
        PRIMARY KEY ([subscriber_id], [channel_id])
    )
    GO

    CREATE TABLE [y_video_view] (
        [video_view_id] bigint PRIMARY KEY IDENTITY(1, 1),
        [user_id] bigint,
        [video_id] bigint,
        [view_date] datetime NOT NULL DEFAULT (GETDATE()),
        [duration_watched] bigint NOT NULL
    )
    GO

    CREATE TABLE [y_video_category] (
        [video_id] bigint,
        [category_id] bigint,
        PRIMARY KEY ([video_id], [category_id])
    )
    GO

    CREATE TABLE [y_category] (
        [category_id] bigint PRIMARY KEY IDENTITY(1, 1),
        [parent_category_id] bigint,
        [category_name] nvarchar(50) UNIQUE NOT NULL
    )
    GO

    CREATE TABLE [y_advertisement] (
        [advertisement_id] bigint PRIMARY KEY IDENTITY(1, 1),
        [title] nvarchar(255) NOT NULL,
        [content] nvarchar(255) NOT NULL,
        [image_url] nvarchar(1000),
        [cta_link] nvarchar(1000),
        [target_audience] nvarchar(255) NOT NULL,
        [status] nvarchar(255) NOT NULL
            CHECK ([status] IN ('active', 'inactive', 'expired')) DEFAULT 'active',
        [click_rate] float NOT NULL,
        [revenue] MONEY NOT NULL,
        [budget] MONEY NOT NULL,
        [created_date] datetime NOT NULL DEFAULT (GETDATE()),
        [last_updated] datetime DEFAULT (GETDATE())
    )
    GO

    CREATE TABLE [y_video_advertisement] (
        [video_ad_id] bigint PRIMARY KEY IDENTITY(1, 1),
        [video_id] bigint,
        [advertisement_id] bigint,
        [ad_type] nvarchar(255) NOT NULL
```

```sql
            CHECK ([ad_type] IN ('pre', 'mid', 'post', 'banner')),
        [start_time] integer NOT NULL,
        [end_time] integer NOT NULL
    )
    GO

    CREATE TABLE [y_ad_impression] (
        [ad_impression_id] bigint PRIMARY KEY IDENTITY(1, 1),
        [advertisement_id] bigint,
        [user_id] bigint,
        [video_id] bigint,
        [impression_date] datetime NOT NULL DEFAULT (GETDATE()),
        [clicked] bit NOT NULL DEFAULT (0),
        [device_type] nvarchar(255) NOT NULL
            CHECK ([device_type] IN ('mobile', 'desktop', 'tablet')),
        [impression_duration] integer
    )
    GO




    ALTER TABLE [y_channel]
    ADD FOREIGN KEY ([user_id]) REFERENCES [y_user] ([user_id])
    GO

    ALTER TABLE [y_video]
    ADD FOREIGN KEY ([channel_id]) REFERENCES [y_channel] ([channel_id])
    GO

    ALTER TABLE [y_playlist]
    ADD FOREIGN KEY ([user_id]) REFERENCES [y_user] ([user_id])
    GO

    ALTER TABLE [y_playlist_video]
    ADD FOREIGN KEY ([playlist_id]) REFERENCES [y_playlist] ([playlist_id])
    GO

    ALTER TABLE [y_playlist_video]
    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
    GO

    ALTER TABLE [y_comment]
    ADD FOREIGN KEY ([parent_comment_id]) REFERENCES [y_comment] ([comment_id])
    GO

    ALTER TABLE [y_comment]
    ADD FOREIGN KEY ([user_id]) REFERENCES [y_user] ([user_id])
    GO

    ALTER TABLE [y_comment]
    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
    GO

    ALTER TABLE [y_reaction]
    ADD FOREIGN KEY ([user_id]) REFERENCES [y_user] ([user_id])
    GO

    ALTER TABLE [y_reaction]
    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
    GO

    ALTER TABLE [y_reaction]
    ADD FOREIGN KEY ([comment_id]) REFERENCES [y_comment] ([comment_id])
    GO

    ALTER TABLE [y_subscription]
    ADD FOREIGN KEY ([subscriber_id]) REFERENCES [y_user] ([user_id])
    GO
```

```sql
201
202    ALTER TABLE [y_subscription]
203    ADD FOREIGN KEY ([channel_id]) REFERENCES [y_channel] ([channel_id])
204    GO
205
206    ALTER TABLE [y_video_view]
207    ADD FOREIGN KEY ([user_id]) REFERENCES [y_user] ([user_id])
208    GO
209
210    ALTER TABLE [y_video_view]
211    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
212    GO
213
214    ALTER TABLE [y_video_category]
215    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
216    GO
217
218    ALTER TABLE [y_video_category]
219    ADD FOREIGN KEY ([category_id]) REFERENCES [y_category] ([category_id])
220    GO
221
222    ALTER TABLE [y_category]
223    ADD FOREIGN KEY ([parent_category_id]) REFERENCES [y_category] ([category_id])
224    GO
225
226    ALTER TABLE [y_video_advertisement]
227    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
228    GO
229
230    ALTER TABLE [y_video_advertisement]
231    ADD FOREIGN KEY ([advertisement_id]) REFERENCES [y_advertisement] ([advertisement_id])
232    GO
233
234    ALTER TABLE [y_ad_impression]
235    ADD FOREIGN KEY ([advertisement_id]) REFERENCES [y_advertisement] ([advertisement_id])
236    GO
237
238    ALTER TABLE [y_ad_impression]
239    ADD FOREIGN KEY ([user_id]) REFERENCES [y_user] ([user_id])
240    GO
241
242    ALTER TABLE [y_ad_impression]
243    ADD FOREIGN KEY ([video_id]) REFERENCES [y_video] ([video_id])
244    GO
```

2.2. DML Skript

```sql
1     -- Insert statements for y_user
2     INSERT INTO y_user
3         (username, first_name, last_name, email, about_me, profile_picture_url)
4     VALUES
5         (
6             'johndoe',
7             'John',
8             'Doe',
9             'johndoe@example.com',
10            'Hello, I am John!',
11            'https://example.com/profile/johndoe.jpg'
12        ),
13        (
14            'janedoe',
15            'Jane',
16            'Doe',
17            'janedoe@example.com',
18            'Welcome to my profile!',
19            'https://example.com/profile/janedoe.jpg'
```

```sql
20          )
21      ;
22
23
24      -- Insert statements for y_channel
25      INSERT INTO y_channel
26          (user_id, channel_name, description)
27      VALUES
28          (
29              (SELECT user_id FROM y_user WHERE username = 'johndoe'),
30              'JohnsChannel',
31              'A channel about technology and programming.'
32          ),
33          (
34              (SELECT user_id FROM y_user WHERE username = 'janedoe'),
35              'JanesChannel',
36              'Sharing my passion for arts and crafts.'
37          )
38      ;
39
40
41      -- Insert statements for y_video
42      INSERT INTO y_video
43          (channel_id, thumbnail_url, video_url, title, description, duration, view_count)
44      VALUES
45          (
46              (SELECT channel_id
47                  FROM y_channel
48                WHERE y_channel.channel_name = 'JohnsChannel'
49                  AND y_channel.user_id = (SELECT user_id FROM y_user WHERE username = 'johndoe')),
50              'https://example.com/thumbnail6.jpg',
51              'https://example.com/video5.mp4',
52              'Intro to Programming',
53              'Learn programming basics.',
54              600,
55              1000
56          ),
57          (
58              (SELECT channel_id
59                  FROM y_channel
60                WHERE y_channel.channel_name = 'JanesChannel'
61                  AND y_channel.user_id = (SELECT user_id FROM y_user WHERE username = 'janedoe')),
62              'https://example.com/thumbnail8.jpg',
63              'https://example.com/video6.mp4',
64              'Art Tutorial',
65              'Beginner art techniques.',
66              1200,
67              500
68          )
69      ;
70
71
72      -- Insert statements for y_playlist
73      INSERT INTO y_playlist
74          (title, user_id, visibility)
75      VALUES
76          (
77              'Programming Tutorials',
78              (SELECT user_id FROM y_user WHERE y_user.username = 'johndoe'),
79              'public'
80          ),
81          (
82              'Art Inspirations',
83              (SELECT user_id FROM y_user WHERE y_user.username = 'janedoe'),
84              'private'
85          )
86      ;
87
88
```

```sql
-- Insert statements for y_playlist_video
INSERT INTO y_playlist_video
    (playlist_id, video_id, [order])
VALUES
    (
        (SELECT playlist_id FROM y_playlist WHERE y_playlist.playlist_id = 1),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 2),
        1
    ),
    (
        (SELECT playlist_id FROM y_playlist WHERE y_playlist.playlist_id = 2),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        1
    )
;


-- Insert statements for y_comment
INSERT INTO y_comment
    (parent_comment_id, user_id, video_id, content)
VALUES
    (
        NULL,
        (SELECT user_id FROM y_user WHERE y_user.username = 'johndoe'),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        'Great video! Thanks for sharing.'
    ),
    (
        (SELECT comment_id FROM y_comment WHERE comment_id = 1 AND video_id = 1),
        (SELECT user_id FROM y_user WHERE y_user.username = 'janedoe'),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        'You're welcome!'
    )
;


-- Insert statements for y_reaction
INSERT INTO y_reaction
    (user_id, video_id, comment_id, reaction_type)
VALUES
    (
        (SELECT user_id FROM y_user WHERE y_user.username = 'johndoe'),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        (SELECT comment_id FROM y_comment WHERE comment_id = 1 AND video_id = 1),
        'like'
    ),
    (
        (SELECT user_id FROM y_user WHERE y_user.username = 'janedoe'),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        (SELECT comment_id FROM y_comment WHERE comment_id = 1 AND video_id = 1),
        'love'
    )
;


-- Insert statements for y_subscription
INSERT INTO y_subscription
    (subscriber_id, channel_id)
VALUES
    (
        (SELECT user_id FROM y_user WHERE y_user.username = 'johndoe'),
        (SELECT channel_id
         FROM y_channel
         WHERE channel_name = 'JanesChannel'
           AND user_id = (SELECT user_id FROM y_user WHERE username = 'janedoe'))
    ),
    (
        (SELECT user_id FROM y_user WHERE y_user.username = 'janedoe'),
        (SELECT channel_id
```

```sql
            FROM y_channel
            WHERE channel_name = 'JohnsChannel'
              AND user_id = (SELECT user_id FROM y_user WHERE username = 'johndoe'))
        )
;


-- Insert statements for y_video_view
INSERT INTO y_video_view
    (user_id, video_id, duration_watched)
VALUES
    (
        (SELECT user_id FROM y_user WHERE y_user.username = 'johndoe'),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        500
    ),
    (
        (SELECT user_id FROM y_user WHERE y_user.username = 'janedoe'),
        (SELECT video_id FROM y_video WHERE y_video.video_id = 2),
        1000
    )
;


-- Insert statements for y_video_category
INSERT INTO y_video_category
    (video_id, category_id)
VALUES
    (
        (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
        (SELECT category_id FROM y_category WHERE y_category.category_name = 'Programming')
    ),
    (
        (SELECT video_id FROM y_video WHERE y_video.video_id = 2),
        (SELECT category_id FROM y_category WHERE y_category.category_name = 'Arts')
    )
;


-- Insert statements for y_category
INSERT INTO y_category
    (parent_category_id, category_name)
VALUES
    (
        NULL,
        'Education'
    ),
    (
        NULL,
        'Arts'
    ),
    (
        (SELECT category_id FROM y_category WHERE category_name = 'Education'),
        'Programming'
    )
;


-- Insert statements for y_advertisement
INSERT INTO y_advertisement
    (title, content, image_url, cta_link, target_audience, click_rate, revenue, budget)
VALUES
    (
        'Tech Ad',
        'Learn coding today!',
        'https://example.com/ad1.jpg',
        'https://example.com/signup',
        'Programmers',
        0.05,
```

```sql
                1000,
                5000
        ),
        (
                'Art Ad',
                'Create amazing art!',
                'https://example.com/ad2.jpg',
                'https://example.com/join',
                'Artists',
                0.03,
                500,
                3000
        )
;


-- Insert statements for y_video_advertisement
INSERT INTO y_video_advertisement
        (video_id, advertisement_id, ad_type, start_time, end_time)
VALUES
        (
                (SELECT video_id FROM y_video WHERE y_video.video_id = 1),
                (SELECT advertisement_id FROM y_advertisement WHERE advertisement_id = 1),
                'pre',
                0,
                30
        ),
        (
                (SELECT video_id FROM y_video WHERE y_video.video_id = 2),
                (SELECT advertisement_id FROM y_advertisement WHERE advertisement_id = 2),
                'mid',
                600,
                630
        )
;


-- Insert statements for y_ad_impression
INSERT INTO y_ad_impression
        (advertisement_id, user_id, video_id, clicked, device_type, impression_duration)
VALUES
        (
                (SELECT advertisement_id FROM y_advertisement WHERE advertisement_id = 1),
                (SELECT user_id FROM y_user WHERE y_user.username = 'johndoe'),
                (SELECT video_id FROM y_video WHERE video_id = 2),
                1,
                'desktop',
                10
        ),
        (
                (SELECT advertisement_id FROM y_advertisement WHERE advertisement_id = 2),
                (SELECT user_id FROM y_user WHERE y_user.username = 'janedoe'),
                (SELECT video_id FROM y_video WHERE video_id = 1),
                0,
                'mobile',
                15
        )
;
```