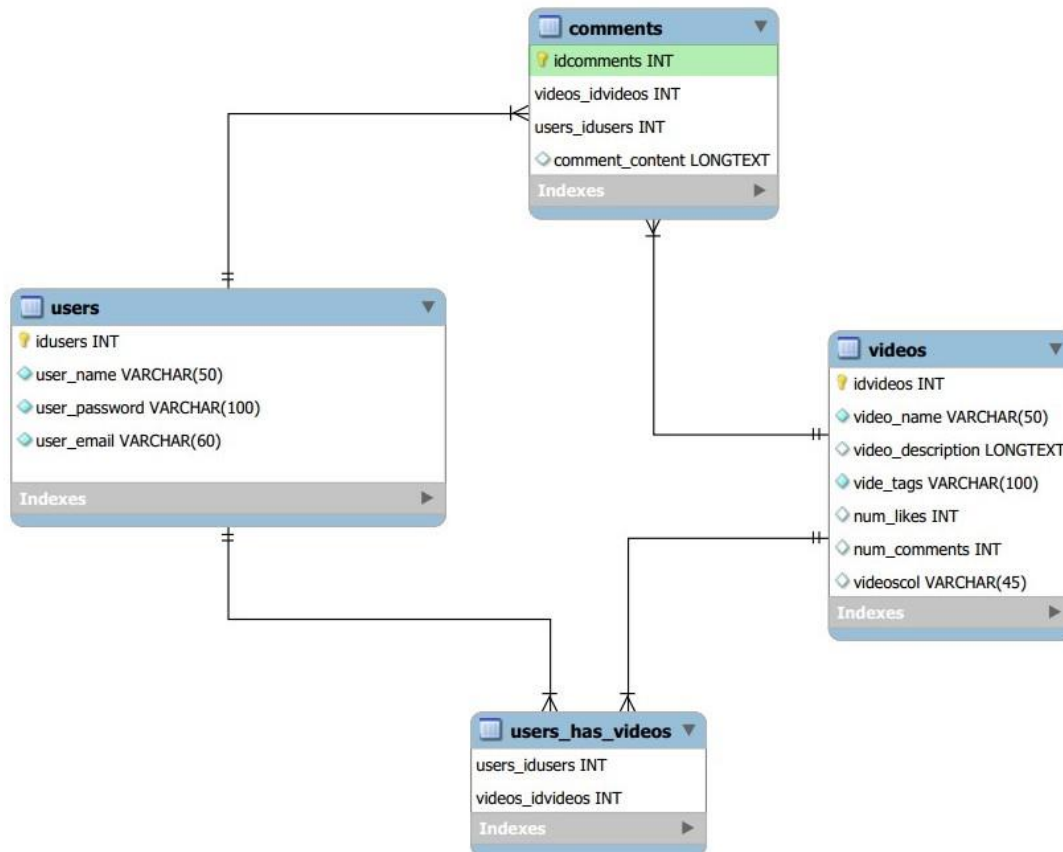


Jakub Persjanow, 167766

Database Diagram

The following diagram presents the database for the project. Consisting of 2 1:n relationships and one n:m relationship. The database represents the video database of streaming services.



Database Creation

The database was created in Oracle infrastructure, using Oracle Database in version 18g Express edition. The database is created using following procedure:

Users table creation:

```
CREATE TABLE "YTADMIN"."USERS"  
    (  
        "USER_ID" NUMBER(*,0) GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE  
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE  
NOKEEP NOSCALE NOT NULL ENABLE,  
        "USER_NAME" VARCHAR2(50 CHAR) NOT NULL ENABLE,  
        "USER_PASSWORD" VARCHAR2(100 CHAR) NOT NULL ENABLE,  
        "USER_EMAIL" VARCHAR2(60 CHAR) NOT NULL ENABLE,  
        "USER_YT_ID" VARCHAR2(50 CHAR) NOT NULL ENABLE,  
        CONSTRAINT "USERS_PK" PRIMARY KEY ("USER_ID")  
    USING INDEX (CREATE UNIQUE INDEX "YTADMIN"."USERS_USER_ID_UINDEX" ON  
"YTADMIN"."USERS" ("USER_ID"))  
    PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
    STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
    PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
    BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "USERS" )    ENABLE
```

```
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

CREATE UNIQUE INDEX "YTADMIN"."USERS_USER_NAME_UINDEX" ON "YTADMIN"."USERS"
("USER_NAME")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

CREATE UNIQUE INDEX "YTADMIN"."USERS_USER_YT_ID_UINDEX" ON "YTADMIN"."USERS"
("USER_YT_ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

Videos table creation:

```
CREATE TABLE "YTADMIN"."VIDEOS"  
(  
    "VIDEO_ID" NUMBER GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE  
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE  
NOKEEP NOSCALE NOT NULL ENABLE,  
    "VIDEO_NAME" VARCHAR2(50 CHAR) NOT NULL ENABLE,  
    "VIDEO_DESCRIPTION" CLOB NOT NULL ENABLE,  
    "VIDEO_TAGS" BLOB,  
    "NUM_LIKES" NUMBER,  
    "NUM_COMMENTS" NUMBER,  
    "VIDEOSCOL" VARCHAR2(45 CHAR),  
    CONSTRAINT "VIDEOS_PK" PRIMARY KEY ("VIDEO_ID")  
USING INDEX (CREATE UNIQUE INDEX "YTADMIN"."VIDEOS_VIDEO_ID_UINDEX" ON  
"YTADMIN"."VIDEOS" ("VIDEO_ID"))  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "USERS" ) ENABLE  
    ) SEGMENT CREATION IMMEDIATE  
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  
NOCOMPRESS LOGGING  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "USERS"  
LOB ("VIDEO_DESCRIPTION") STORE AS SECUREFILE (  
    TABLESPACE "USERS" ENABLE STORAGE IN ROW CHUNK 8192  
    NOCACHE LOGGING NOCOMPRESS KEEP_DUPLICATES  
    STORAGE(INITIAL 106496 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
    PCTINCREASE 0  
    BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT))  
LOB ("VIDEO_TAGS") STORE AS SECUREFILE (  
    TABLESPACE "USERS" ENABLE STORAGE IN ROW CHUNK 8192  
    NOCACHE LOGGING NOCOMPRESS KEEP_DUPLICATES  
    STORAGE(INITIAL 106496 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
    PCTINCREASE 0  
    BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT))) ;
```

```
LOB ("VIDEO TAGS") STORE AS SECUREFILE (
```

Comments table creation:

```
CREATE TABLE "YTADMIN"."COMMENTS"  
(  
    "COMMENT_ID" NUMBER GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE  
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE  
NOKEEP NOSCALE NOT NULL ENABLE,  
    "COMMENT_CONTENT" CLOB NOT NULL ENABLE,  
    "VIDEO_ID" NUMBER NOT NULL ENABLE,  
    "USER_ID" NUMBER NOT NULL ENABLE,  
    CONSTRAINT "COMMENTS_PK" PRIMARY KEY ("COMMENT_ID")  
USING INDEX (CREATE UNIQUE INDEX "YTADMIN"."COMMENTS_COMMENT_ID_UINDEX" ON  
"YTADMIN"."COMMENTS" ("COMMENT_ID"))  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "USERS" ) ENABLE,  
    CONSTRAINT "COMMENTS_VIDEOS_VIDEO_ID_FK" FOREIGN KEY ("VIDEO_ID")  
REFERENCES "YTADMIN"."VIDEOS" ("VIDEO_ID") ENABLE,  
    CONSTRAINT "COMMENTS_USERS_USER_ID_FK" FOREIGN KEY ("USER_ID")  
REFERENCES "YTADMIN"."USERS" ("USER_ID") ENABLE  
  
) SEGMENT CREATION IMMEDIATE  
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  
NOCOMPRESS LOGGING  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "USERS"  
LOB ("COMMENT_CONTENT") STORE AS SECUREFILE (  
TABLESPACE "USERS" ENABLE STORAGE IN ROW CHUNK 8192  
NOCACHE LOGGING NOCOMPRESS KEEP_DUPLICATES  
STORAGE(INITIAL 106496 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT))) ;
```

User_has_videos table creation

```
CREATE TABLE "YTADMIN"."USER_HAVE_VIDEOS"
(
    "USER_ID" NUMBER NOT NULL ENABLE,
    "VIDEO_ID" NUMBER NOT NULL ENABLE,
    CONSTRAINT "USER_HAVE_VIDEOS_USERS_USER_ID_FK" FOREIGN KEY ("USER_ID")
        REFERENCES "YTADMIN"."USERS" ("USER_ID") ON DELETE CASCADE ENABLE,
    CONSTRAINT "USER_HAVE_VIDEOS_VIDEOS_VIDEO_ID_FK" FOREIGN KEY ("VIDEO_ID")
        REFERENCES "YTADMIN"."VIDEOS" ("VIDEO_ID") ENABLE
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

Data Generation

Data for database tables was generated using simple Python scripts that provided the database with simple data for Videos and Comments tables (attached below). The Users table was populated manually. The User_has_videos table was populated using simple Python script (attached below).

Appendix 1 – Python scripts for populating Comments table

```
import cx_Oracle
import random

try:

    con = cx_Oracle.connect('ytadmin/ytadmin@localhost')
    cursor = con.cursor()

    query = f'''
        SELECT VIDEO_ID, NUM_COMMENTS FROM YTADMIN.VIDEOS
        '''

    cursor.execute(query)

    for i in cursor.fetchall():
        for j in range(1, i[1]):
            print(i)
            user = random.randint(1, 10)
            query_user = f'''
                SELECT USER_NAME FROM YTADMIN.USERS WHERE USER_ID={user}
                '''
            cursor.execute(query_user)
            username = cursor.fetchone()
            print(username[0])

            query_comment = f'''
                INSERT INTO YTADMIN.COMMENTS (COMMENT_CONTENT, VIDEO_ID, USER_ID) VALUES
                ('COMMENT from {username[0]}', {i[0]}, {user})
                '''
            print(query_comment)
            cursor.execute(query_comment)
            con.commit()

except cx_Oracle.DatabaseError as e:
    print(e)

finally:
    if cursor:
        cursor.close()
    if con:
        con.close()
```

Appendix 2 – Python scripts for populating User_has_videos table

```
import cx_Oracle
import random

try:

    con = cx_Oracle.connect('ytadmin/ytadmin@localhost')
    cursor = con.cursor()

    for video_id in range(1, 531):
        if video_id != 3:
            user = random.randint(1, 10)

            query = f'''
```

```

        INSERT INTO YTADMIN.USER_HAVE_VIDEOS (USER_ID, VIDEO_ID) VALUES ({user},
{video_id})
        '''

        print(query)
        cursor.execute(query)
        con.commit()

except cx_Oracle.DatabaseError as e:
    print(e)

finally:
    if cursor:
        cursor.close()
    if con:
        con.close()

```

Database Statistics

Table Name	Number of Entries
Users	10
Videos	530
Comments	13 346

*53 is a maximum set inside a query