NEW YORK UNIVERSITY

School of Professional Studies

DATABASE DESIGN &

MANAGEMENT MASY1-GC

3500_1_101 | Spring 2024

Individual Project Assignment 3

SUBMITTED BY:

Saumay Killa – KS – sk10882@nyu.edu

SUBMITTED ON:

May 6th, 2024

UNDER THE GUIDANCE OF:
Prof. Amit Patel

Table Of Contents

Executive Summary

Spotify, a leading music streaming platform, faces significant challenges in managing its vast and dynamic database infrastructure. This executive summary provides an overview of key considerations and strategies for addressing these critical aspects of Spotify's database management.

Data Security:

Spotify recognizes the importance of safeguarding user data against potential threats, including unauthorized access, data breaches, and cyber-attacks. To enhance data security:

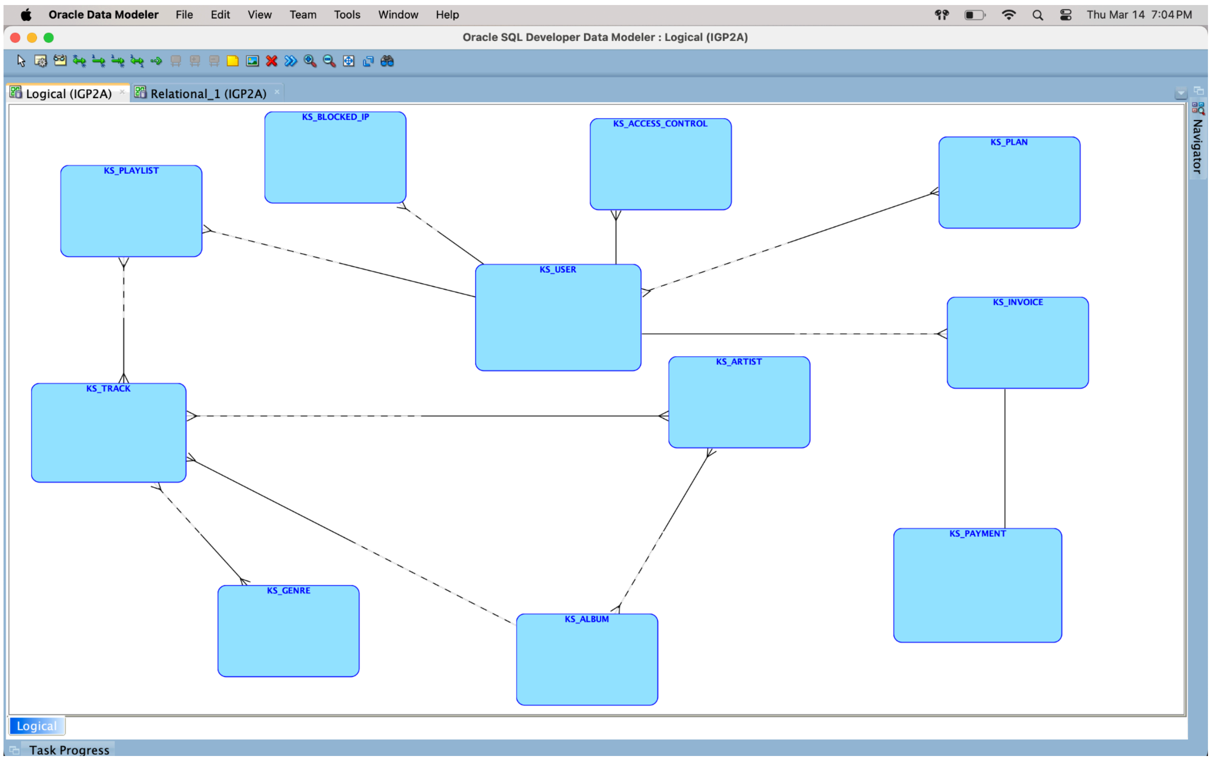
- **Encryption:** Implement end-to-end encryption protocols to protect user data both in transit and at rest, ensuring confidentiality and integrity.
- **Access Controls:** Enforce strict access controls and authentication mechanisms to limit access to sensitive data, ensuring that only authorized personnel can access and modify database resources.
- **Monitoring and Auditing:** Implement comprehensive monitoring and auditing tools to detect suspicious activities, unauthorized access attempts, and unusual data access patterns in real-time.
- **Regular Security Audits:** Conduct regular security audits and penetration testing to identify vulnerabilities and weaknesses in the database infrastructure, enabling proactive mitigation of security risks.

Scalability:

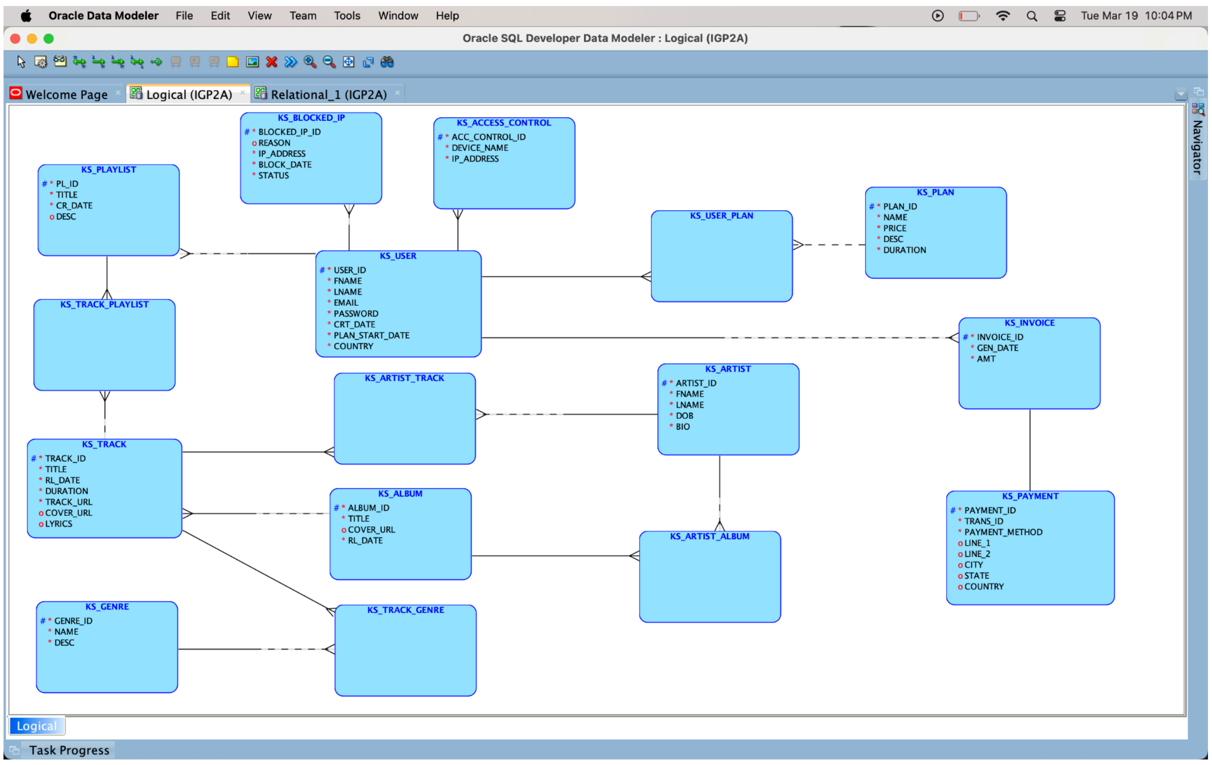
As Spotify's user base continues to grow, maintaining scalability in database management is essential to ensure optimal performance and responsiveness. To achieve scalability:

- **Horizontal Scaling:** Implement a distributed database architecture that supports horizontal scaling, enabling Spotify to add additional database nodes and resources dynamically to handle increased user demand.
- **Load Balancing:** Utilize load balancing techniques to distribute incoming traffic evenly across multiple database servers, preventing bottlenecks and ensuring efficient resource utilization.
- **Caching Mechanisms:** Implement caching mechanisms to store frequently accessed data in memory, reducing the load on the database servers and improving overall system performance.

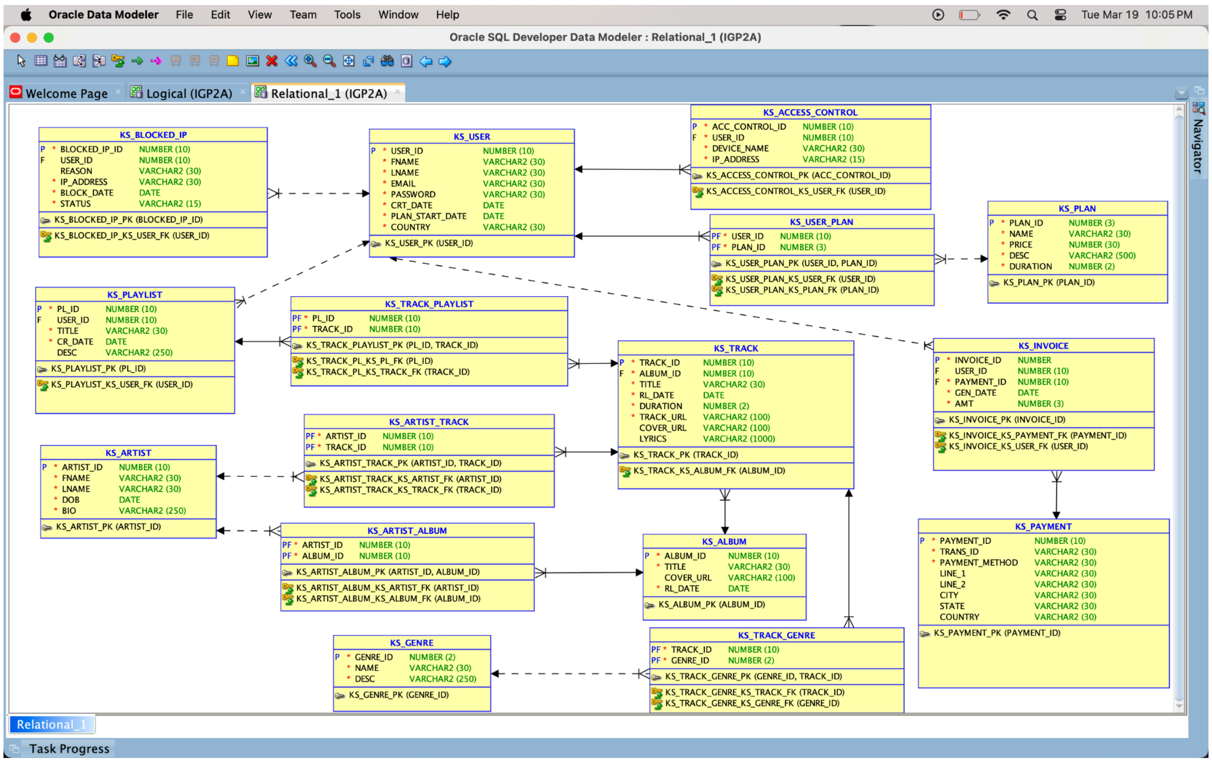
Enterprise Model



Logical Model



Relational Model



```
KS_BLOCKED_IP
P  * BLOCKED_IP_ID     NUMBER (10)
F    USER_ID           NUMBER (10)
     REASON            VARCHAR2 (30)
   * IP_ADDRESS        VARCHAR2 (30)
   * BLOCK_DATE        DATE
   * STATUS            VARCHAR2 (15)
KS_BLOCKED_IP_PK (BLOCKED_IP_ID)
KS_BLOCKED_IP_KS_USER_FK (USER_ID)

KS_USER
P  * USER_ID           NUMBER (10)
   * FNAME             VARCHAR2 (30)
   * LNAME             VARCHAR2 (30)
   * EMAIL             VARCHAR2 (30)
   * PASSWORD          VARCHAR2 (30)
   * CRT_DATE          DATE
   * PLAN_START_DATE   DATE
   * COUNTRY           VARCHAR2 (30)
KS_USER_PK (USER_ID)

KS_ACCESS_CONTROL
P  * ACC_CONTROL_ID    NUMBER (10)
F  * USER_ID           NUMBER (10)
   * DEVICE_NAME       VARCHAR2 (30)
   * IP_ADDRESS        VARCHAR2 (15)
KS_ACCESS_CONTROL_PK (ACC_CONTROL_ID)
KS_ACCESS_CONTROL_KS_USER_FK (USER_ID)

KS_USER_PLAN
PF * USER_ID     NUMBER (10)
PF * PLAN_ID     NUMBER (3)
KS_USER_PLAN_PK (USER_ID, PLAN_ID)
KS_USER_PLAN_KS_USER_FK (USER_ID)
KS_USER_PLAN_KS_PLAN_FK (PLAN_ID)

KS_PLAN
P  * PLAN_ID     NUMBER (3)
   * NAME        VARCHAR2 (30)
   * PRICE       NUMBER (30)
   * DESC        VARCHAR2 (500)
   * DURATION    NUMBER (2)
KS_PLAN_PK (PLAN_ID)

KS_PLAYLIST
P  * PL_ID       NUMBER (10)
F    USER_ID     NUMBER (10)
   * TITLE       VARCHAR2 (30)
   * CR_DATE     DATE
     DESC        VARCHAR2 (250)
KS_PLAYLIST_PK (PL_ID)
KS_PLAYLIST_KS_USER_FK (USER_ID)

KS_TRACK_PLAYLIST
PF * PL_ID       NUMBER (10)
PF * TRACK_ID    NUMBER (10)
KS_TRACK_PLAYLIST_PK (PL_ID, TRACK_ID)
KS_TRACK_PL_KS_PL_FK (PL_ID)
KS_TRACK_PL_KS_TRACK_FK (TRACK_ID)

KS_TRACK
P  * TRACK_ID    NUMBER (10)
F  * ALBUM_ID    NUMBER (10)
   * TITLE       VARCHAR2 (30)
   * RL_DATE     DATE
   * DURATION    NUMBER (2)
   * TRACK_URL   VARCHAR2 (100)
     COVER_URL   VARCHAR2 (100)
     LYRICS      VARCHAR2 (1000)
KS_TRACK_PK (TRACK_ID)
KS_TRACK_KS_ALBUM_FK (ALBUM_ID)

KS_INVOICE
P  * INVOICE_ID  NUMBER
F    USER_ID     NUMBER (10)
F  * PAYMENT_ID  NUMBER (10)
   * GEN_DATE    DATE
   * AMT         NUMBER (3)
KS_INVOICE_PK (INVOICE_ID)
KS_INVOICE_KS_PAYMENT_FK (PAYMENT_ID)
KS_INVOICE_KS_USER_FK (USER_ID)

KS_ARTIST_TRACK
PF * ARTIST_ID   NUMBER (10)
PF * TRACK_ID    NUMBER (10)
KS_ARTIST_TRACK_PK (ARTIST_ID, TRACK_ID)
KS_ARTIST_TRACK_KS_ARTIST_FK (ARTIST_ID)
KS_ARTIST_TRACK_KS_TRACK_FK (TRACK_ID)

KS_ARTIST
P  * ARTIST_ID   NUMBER (10)
   * FNAME       VARCHAR2 (30)
   * LNAME       VARCHAR2 (30)
   * DOB         DATE
   * BIO         VARCHAR2 (250)
KS_ARTIST_PK (ARTIST_ID)

KS_ARTIST_ALBUM
PF * ARTIST_ID   NUMBER (10)
PF * ALBUM_ID    NUMBER (10)
KS_ARTIST_ALBUM_PK (ARTIST_ID, ALBUM_ID)
KS_ARTIST_ALBUM_KS_ARTIST_FK (ARTIST_ID)
KS_ARTIST_ALBUM_KS_ALBUM_FK (ALBUM_ID)

KS_ALBUM
P  * ALBUM_ID    NUMBER (10)
   * TITLE       VARCHAR2 (30)
     COVER_URL   VARCHAR2 (100)
   * RL_DATE     DATE
KS_ALBUM_PK (ALBUM_ID)

KS_PAYMENT
P  * PAYMENT_ID       NUMBER (10)
   * TRANS_ID         VARCHAR2 (30)
   * PAYMENT_METHOD   VARCHAR2 (30)
     LINE_1           VARCHAR2 (30)
     LINE_2           VARCHAR2 (30)
     CITY             VARCHAR2 (30)
     STATE            VARCHAR2 (30)
     COUNTRY          VARCHAR2 (30)
KS_PAYMENT_PK (PAYMENT_ID)

KS_GENRE
P  * GENRE_ID    NUMBER (2)
   * NAME        VARCHAR2 (30)
   * DESC        VARCHAR2 (250)
KS_GENRE_PK (GENRE_ID)

KS_TRACK_GENRE
PF * TRACK_ID    NUMBER (10)
PF * GENRE_ID    NUMBER (2)
KS_TRACK_GENRE_PK (GENRE_ID, TRACK_ID)
KS_TRACK_GENRE_KS_TRACK_FK (TRACK_ID)
KS_TRACK_GENRE_KS_GENRE_FK (GENRE_ID)
```

DDL CODE

```
-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806
--   at:        2024-04-18 15:34:24 EDT
--   site:      Oracle Database 21c
--   type:      Oracle Database 21c



-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE ks_access_control (
    acc_control_id NUMBER(10) NOT NULL,
    user_id        NUMBER(10) NOT NULL,
    device_name    VARCHAR2(30) NOT NULL,
    ip_address     VARCHAR2(15) NOT NULL
);

COMMENT ON COLUMN ks_access_control.acc_control_id IS
    'Unique Access Control ID';

COMMENT ON COLUMN ks_access_control.device_name IS
    'Name of the Device';

COMMENT ON COLUMN ks_access_control.ip_address IS
    'IP address of the device connected';

ALTER TABLE ks_access_control ADD CONSTRAINT ks_access_control_pk PRIMARY
KEY ( acc_control_id );

CREATE TABLE ks_album (
    album_id  NUMBER(10) NOT NULL,
    title     VARCHAR2(30) NOT NULL,
    cover_url VARCHAR2(100),
    rl_date   DATE NOT NULL
);

COMMENT ON COLUMN ks_album.album_id IS
    'Unique Album ID';
```

```
COMMENT ON COLUMN ks_album.title IS
    'Album name';

COMMENT ON COLUMN ks_album.cover_url IS
    'Album cover art url';

COMMENT ON COLUMN ks_album.rl_date IS
    'Album release date';

ALTER TABLE ks_album ADD CONSTRAINT ks_album_pk PRIMARY KEY ( album_id
);

CREATE TABLE ks_artist (
    artist_id NUMBER(10) NOT NULL,
    fname     VARCHAR2(30) NOT NULL,
    lname     VARCHAR2(30) NOT NULL,
    dob       DATE NOT NULL,
    bio       VARCHAR2(250) NOT NULL
);

COMMENT ON COLUMN ks_artist.artist_id IS
    'Unique Artist ID';

COMMENT ON COLUMN ks_artist.fname IS
    'Artist First Name';

COMMENT ON COLUMN ks_artist.lname IS
    'Artist Last Name';

COMMENT ON COLUMN ks_artist.dob IS
    'Artist Date of Birth';

COMMENT ON COLUMN ks_artist.bio IS
    'Artist Biography';

ALTER TABLE ks_artist ADD CONSTRAINT ks_artist_pk PRIMARY KEY ( artist_id );

CREATE TABLE ks_artist_album (
    artist_id NUMBER(10) NOT NULL,
    album_id  NUMBER(10) NOT NULL
);

ALTER TABLE ks_artist_album ADD CONSTRAINT ks_artist_album_pk PRIMARY KEY
( artist_id,
                                                  album_id );

CREATE TABLE ks_artist_track (
   artist_id NUMBER(10) NOT NULL,
   track_id  NUMBER(10) NOT NULL
);

ALTER TABLE ks_artist_track ADD CONSTRAINT ks_artist_track_pk PRIMARY KEY (
artist_id,
                                                  track_id );

CREATE TABLE ks_blocked_ip (
   blocked_ip_id NUMBER(10) NOT NULL,
   user_id       NUMBER(10),
   reason        VARCHAR2(30),
   ip_address    VARCHAR2(30) NOT NULL,
   block_date    DATE NOT NULL,
   status        VARCHAR2(15) NOT NULL
);

COMMENT ON COLUMN ks_blocked_ip.blocked_ip_id IS
   'UNIQUE Blocked IP ID';

COMMENT ON COLUMN ks_blocked_ip.ip_address IS
   'IP Address of the blocked device';

COMMENT ON COLUMN ks_blocked_ip.block_date IS
   'Date the device was blocked';

COMMENT ON COLUMN ks_blocked_ip.status IS
   'Current status of the blocked device';

ALTER TABLE ks_blocked_ip ADD CONSTRAINT ks_blocked_ip_pk PRIMARY KEY (
blocked_ip_id );

CREATE TABLE ks_genre (
   genre_id    NUMBER(2) NOT NULL,
   name        VARCHAR2(30) NOT NULL,
   description VARCHAR2(250) NOT NULL
);

COMMENT ON COLUMN ks_genre.genre_id IS
   'Unique Genre ID';
```

```
COMMENT ON COLUMN ks_genre.name IS
    'Genre name';

COMMENT ON COLUMN ks_genre.description IS
    'Description of the genre';

ALTER TABLE ks_genre ADD CONSTRAINT ks_genre_pk PRIMARY KEY ( genre_id );

CREATE TABLE ks_invoice (
    invoice_id NUMBER NOT NULL,
    user_id    NUMBER(10),
    payment_id NUMBER(10) NOT NULL,
    gen_date   DATE NOT NULL,
    amt        NUMBER(3) NOT NULL
);

COMMENT ON COLUMN ks_invoice.invoice_id IS
    'Unique Invoice ID';

COMMENT ON COLUMN ks_invoice.gen_date IS
    'Invoice Generation Date';

COMMENT ON COLUMN ks_invoice.amt IS
    'Amount of the Invoice';

ALTER TABLE ks_invoice ADD CONSTRAINT ks_invoice_pk PRIMARY KEY (
invoice_id );

CREATE TABLE ks_payment (
    payment_id     NUMBER(10) NOT NULL,
    trans_id       VARCHAR2(30) NOT NULL,
    payment_method VARCHAR2(30) NOT NULL,
    line_1         VARCHAR2(30),
    line_2         VARCHAR2(30),
    city           VARCHAR2(30),
    state          VARCHAR2(30),
    country        VARCHAR2(30)
);

COMMENT ON COLUMN ks_payment.payment_id IS
    'Unique Payment ID';

COMMENT ON COLUMN ks_payment.trans_id IS
    'Transaction ID';
```

```
COMMENT ON COLUMN ks_payment.payment_method IS
    'Method of Payment';

COMMENT ON COLUMN ks_payment.line_1 IS
    'Address for billing ';

COMMENT ON COLUMN ks_payment.line_2 IS
    'Address for billing ';

COMMENT ON COLUMN ks_payment.city IS
    'Billing city';

COMMENT ON COLUMN ks_payment.state IS
    'Billing State';

COMMENT ON COLUMN ks_payment.country IS
    'Billing Country';

ALTER TABLE ks_payment ADD CONSTRAINT ks_payment_pk PRIMARY KEY (
payment_id );

CREATE TABLE ks_plan (
    plan_id     NUMBER(3) NOT NULL,
    name        VARCHAR2(30) NOT NULL,
    price       NUMBER(30) NOT NULL,
    description VARCHAR2(500) NOT NULL,
    duration    NUMBER(2) NOT NULL
);

COMMENT ON COLUMN ks_plan.plan_id IS
    'UNIQUE Plan ID';

COMMENT ON COLUMN ks_plan.name IS
    'Plan Name';

COMMENT ON COLUMN ks_plan.price IS
    'Price of the plan';

COMMENT ON COLUMN ks_plan.description IS
    'Description about the plan ';

COMMENT ON COLUMN ks_plan.duration IS
    'Duration of the Plan';
```

```
ALTER TABLE ks_plan ADD CONSTRAINT ks_plan_pk PRIMARY KEY ( plan_id );

CREATE TABLE ks_playlist (
    pl_id       NUMBER(10) NOT NULL,
    user_id     NUMBER(10),
    title       VARCHAR2(30) NOT NULL,
    cr_date     DATE NOT NULL,
    description VARCHAR2(250)
);

COMMENT ON COLUMN ks_playlist.pl_id IS
    'Unique Playlist ID';

COMMENT ON COLUMN ks_playlist.title IS
    'Title of the Playlist';

COMMENT ON COLUMN ks_playlist.cr_date IS
    'Playlist Creation Date';

COMMENT ON COLUMN ks_playlist.description IS
    'Playlist Description';

ALTER TABLE ks_playlist ADD CONSTRAINT ks_playlist_pk PRIMARY KEY ( pl_id );

CREATE TABLE ks_track (
    track_id  NUMBER(10) NOT NULL,
    album_id  NUMBER(10) NOT NULL,
    title     VARCHAR2(30) NOT NULL,
    rl_date   DATE NOT NULL,
    duration  NUMBER(2) NOT NULL,
    track_url VARCHAR2(100) NOT NULL,
    cover_url VARCHAR2(100),
    lyrics    VARCHAR2(1000)
);

COMMENT ON COLUMN ks_track.track_id IS
    'Unique track id';

COMMENT ON COLUMN ks_track.title IS
    'Track name';

COMMENT ON COLUMN ks_track.rl_date IS
    'Release Date';
```

```
COMMENT ON COLUMN ks_track.duration IS
    'Duration of the track';

COMMENT ON COLUMN ks_track.track_url IS
    'Track URL';

COMMENT ON COLUMN ks_track.cover_url IS
    'Cover Image of the track URL';

COMMENT ON COLUMN ks_track.lyrics IS
    'Lyrics of the track';

ALTER TABLE ks_track ADD CONSTRAINT ks_track_pk PRIMARY KEY ( track_id );

CREATE TABLE ks_track_genre (
    track_id NUMBER(10) NOT NULL,
    genre_id NUMBER(2) NOT NULL
);

ALTER TABLE ks_track_genre ADD CONSTRAINT ks_track_genre_pk PRIMARY KEY (
genre_id,
                                                                          track_id );

CREATE TABLE ks_track_playlist (
    pl_id    NUMBER(10) NOT NULL,
    track_id NUMBER(10) NOT NULL
);

ALTER TABLE ks_track_playlist ADD CONSTRAINT ks_track_playlist_pk PRIMARY
KEY ( pl_id,
                                                                              track_id );

CREATE TABLE ks_user (
    user_id         NUMBER(10) NOT NULL,
    fname           VARCHAR2(30) NOT NULL,
    lname           VARCHAR2(30) NOT NULL,
    email           VARCHAR2(30) NOT NULL,
    password        VARCHAR2(30) NOT NULL,
    crt_date        DATE NOT NULL,
    plan_start_date DATE NOT NULL,
    country         VARCHAR2(30) NOT NULL
);
```

```
COMMENT ON COLUMN ks_user.user_id IS
    'Unique User ID';

COMMENT ON COLUMN ks_user.fname IS
    'First name of User';

COMMENT ON COLUMN ks_user.lname IS
    'Last name of User';

COMMENT ON COLUMN ks_user.email IS
    'Email of the user';

COMMENT ON COLUMN ks_user.password IS
    'Password of the User';

COMMENT ON COLUMN ks_user.crt_date IS
    'User Creation Date';

COMMENT ON COLUMN ks_user.plan_start_date IS
    'Start Date of Plan';

COMMENT ON COLUMN ks_user.country IS
    'Country of the User';

ALTER TABLE ks_user ADD CONSTRAINT ks_user_pk PRIMARY KEY ( user_id );

CREATE TABLE ks_user_plan (
    user_id NUMBER(10) NOT NULL,
    plan_id NUMBER(3) NOT NULL
);

ALTER TABLE ks_user_plan ADD CONSTRAINT ks_user_plan_pk PRIMARY KEY (
user_id,
                                                                    plan_id );

ALTER TABLE ks_access_control
    ADD CONSTRAINT ks_access_control_ks_user_fk FOREIGN KEY ( user_id )
        REFERENCES ks_user ( user_id );

ALTER TABLE ks_artist_album
    ADD CONSTRAINT ks_artist_album_ks_album_fk FOREIGN KEY ( album_id )
        REFERENCES ks_album ( album_id );

ALTER TABLE ks_artist_album
```

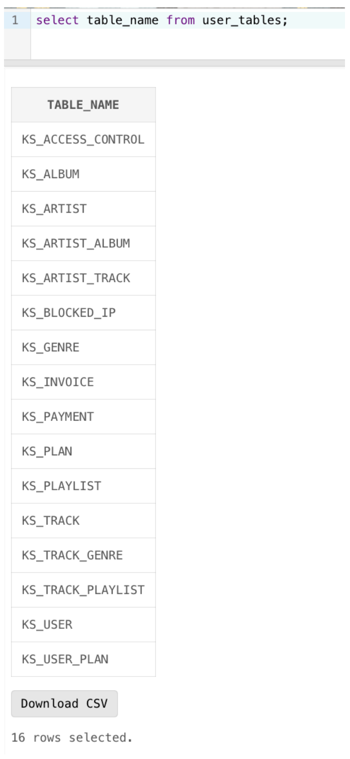
```
    ADD CONSTRAINT ks_artist_album_ks_artist_fk FOREIGN KEY ( artist_id )
        REFERENCES ks_artist ( artist_id );

ALTER TABLE ks_artist_track
    ADD CONSTRAINT ks_artist_track_ks_artist_fk FOREIGN KEY ( artist_id )
        REFERENCES ks_artist ( artist_id );

ALTER TABLE ks_artist_track
    ADD CONSTRAINT ks_artist_track_ks_track_fk FOREIGN KEY ( track_id )
        REFERENCES ks_track ( track_id );

ALTER TABLE ks_blocked_ip
    ADD CONSTRAINT ks_blocked_ip_ks_user_fk FOREIGN KEY ( user_id )
        REFERENCES ks_user ( user_id );

ALTER TABLE ks_invoice
    ADD CONSTRAINT ks_invoice_ks_payment_fk FOREIGN KEY ( payment_id )
        REFERENCES ks_payment ( payment_id );

ALTER TABLE ks_invoice
    ADD CONSTRAINT ks_invoice_ks_user_fk FOREIGN KEY ( user_id )
        REFERENCES ks_user ( user_id );

ALTER TABLE ks_playlist
    ADD CONSTRAINT ks_playlist_ks_user_fk FOREIGN KEY ( user_id )
        REFERENCES ks_user ( user_id );

ALTER TABLE ks_track_genre
    ADD CONSTRAINT ks_track_genre_ks_genre_fk FOREIGN KEY ( genre_id )
        REFERENCES ks_genre ( genre_id );

ALTER TABLE ks_track_genre
    ADD CONSTRAINT ks_track_genre_ks_track_fk FOREIGN KEY ( track_id )
        REFERENCES ks_track ( track_id );

ALTER TABLE ks_track
    ADD CONSTRAINT ks_track_ks_album_fk FOREIGN KEY ( album_id )
        REFERENCES ks_album ( album_id );

ALTER TABLE ks_track_playlist
    ADD CONSTRAINT ks_track_pl_ks_pl_fk FOREIGN KEY ( pl_id )
        REFERENCES ks_playlist ( pl_id );

ALTER TABLE ks_track_playlist
```

```
    ADD CONSTRAINT ks_track_pl_ks_track_fk FOREIGN KEY ( track_id )
        REFERENCES ks_track ( track_id );

ALTER TABLE ks_user_plan
    ADD CONSTRAINT ks_user_plan_ks_plan_fk FOREIGN KEY ( plan_id )
        REFERENCES ks_plan ( plan_id );

ALTER TABLE ks_user_plan
    ADD CONSTRAINT ks_user_plan_ks_user_fk FOREIGN KEY ( user_id )
        REFERENCES ks_user ( user_id );



-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                            16
-- CREATE INDEX                             0
-- ALTER TABLE                             32
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           0
-- ALTER TRIGGER                            0
-- CREATE COLLECTION TYPE                   0
-- CREATE STRUCTURED TYPE                   0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                           0
-- CREATE CONTEXT                           0
-- CREATE DATABASE                          0
-- CREATE DIMENSION                         0
-- CREATE DIRECTORY                         0
-- CREATE DISK GROUP                        0
-- CREATE ROLE                              0
-- CREATE ROLLBACK SEGMENT                  0
-- CREATE SEQUENCE                          0
-- CREATE MATERIALIZED VIEW                 0
-- CREATE MATERIALIZED VIEW LOG             0
-- CREATE SYNONYM                           0
-- CREATE TABLESPACE                        0
-- CREATE USER                              0
--
```

```
-- DROP TABLESPACE                  0
-- DROP DATABASE                    0
--
-- REDACTION POLICY                 0
--
-- ORDS DROP SCHEMA                 0
-- ORDS ENABLE SCHEMA               0
-- ORDS ENABLE OBJECT               0
--
-- ERRORS                           0
-- WARNINGS                         0
```

Dictionary Queries Screen Shots

List of Tables

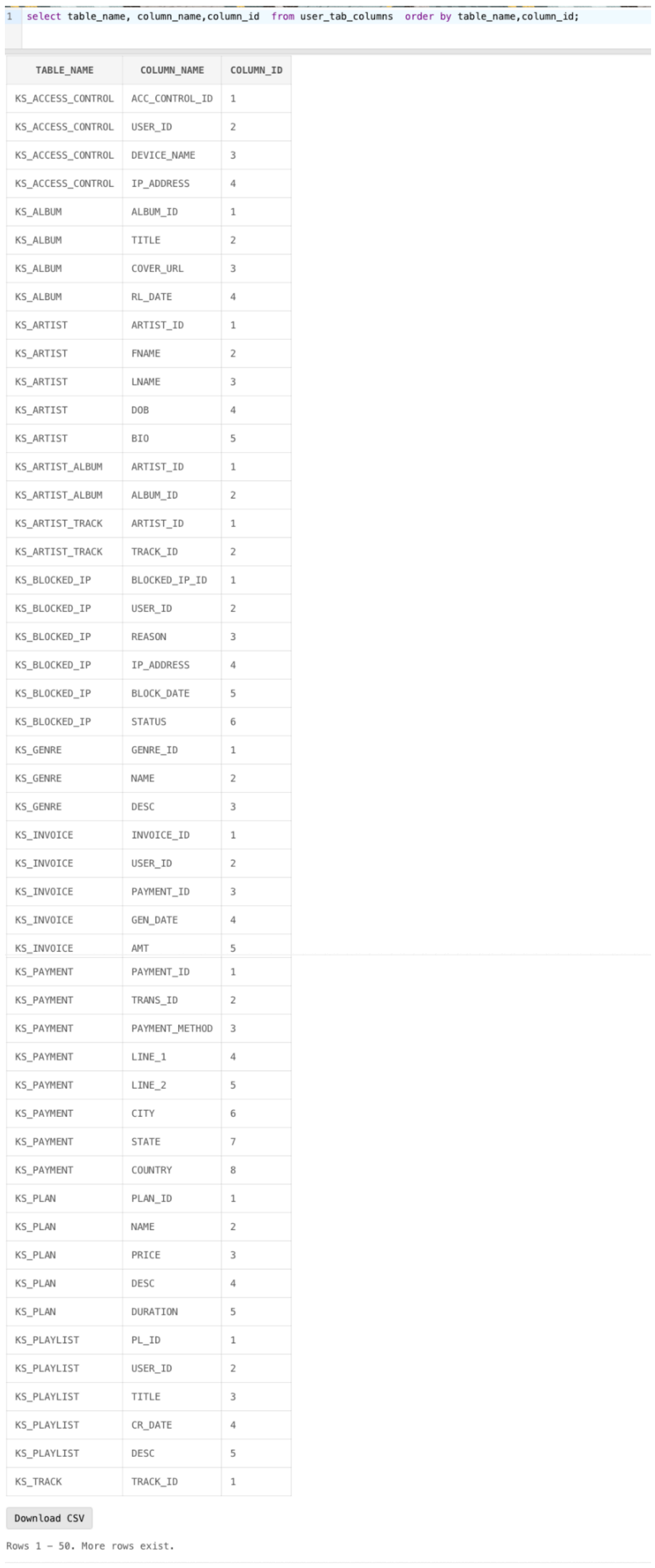


```
select table_name from user_tables;
```

TABLE_NAME
KS_ACCESS_CONTROL
KS_ALBUM
KS_ARTIST
KS_ARTIST_ALBUM
KS_ARTIST_TRACK
KS_BLOCKED_IP
KS_GENRE
KS_INVOICE
KS_PAYMENT
KS_PLAN
KS_PLAYLIST
KS_TRACK
KS_TRACK_GENRE
KS_TRACK_PLAYLIST
KS_USER
KS_USER_PLAN

```
16 rows selected.
```

List of Table Columns

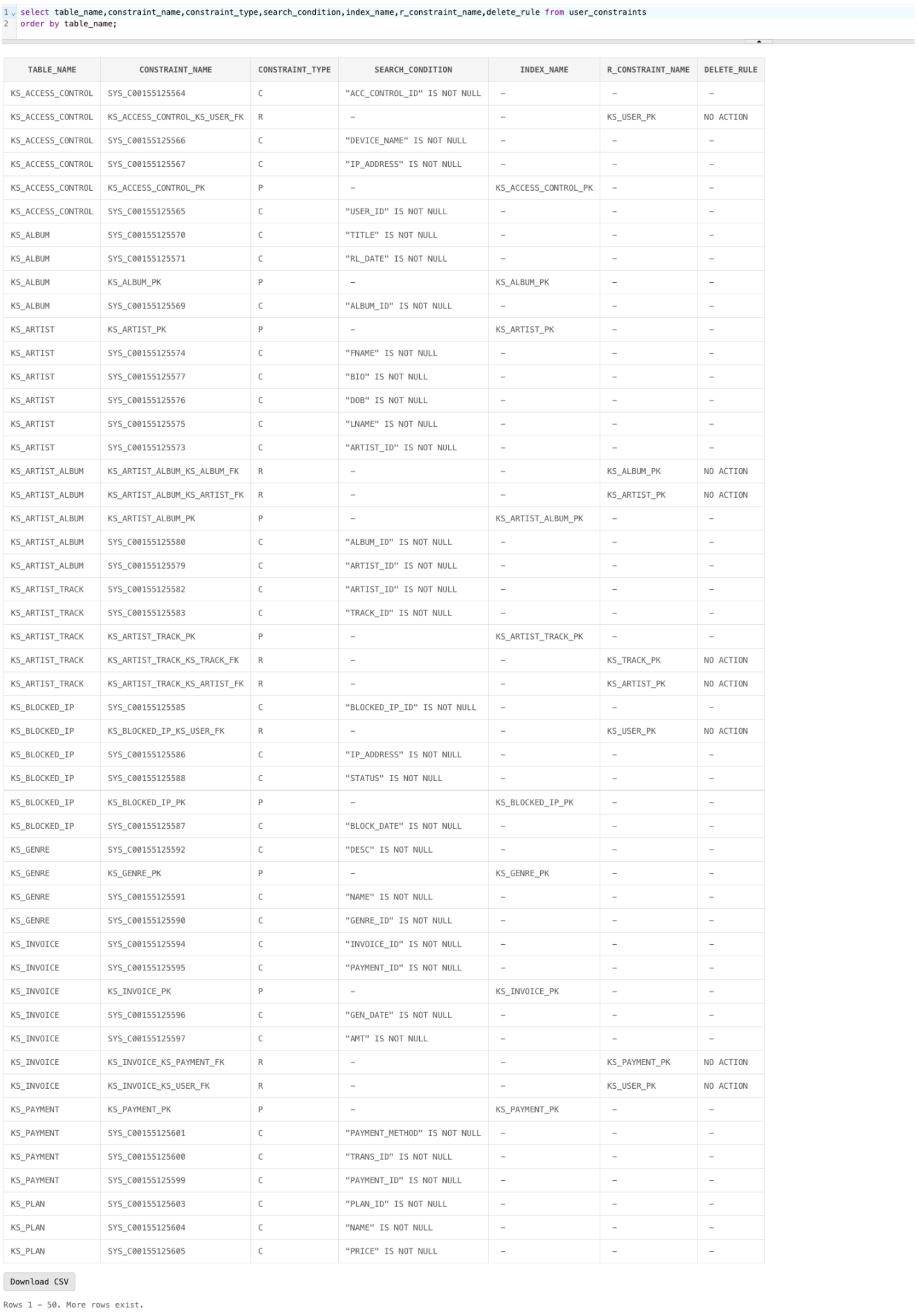


```
select table_name, column_name,column_id  from user_tab_columns  order by table_name,column_id;
```

TABLE_NAME	COLUMN_NAME	COLUMN_ID
KS_ACCESS_CONTROL	ACC_CONTROL_ID	1
KS_ACCESS_CONTROL	USER_ID	2
KS_ACCESS_CONTROL	DEVICE_NAME	3
KS_ACCESS_CONTROL	IP_ADDRESS	4
KS_ALBUM	ALBUM_ID	1
KS_ALBUM	TITLE	2
KS_ALBUM	COVER_URL	3
KS_ALBUM	RL_DATE	4
KS_ARTIST	ARTIST_ID	1
KS_ARTIST	FNAME	2
KS_ARTIST	LNAME	3
KS_ARTIST	DOB	4
KS_ARTIST	BIO	5
KS_ARTIST_ALBUM	ARTIST_ID	1
KS_ARTIST_ALBUM	ALBUM_ID	2
KS_ARTIST_TRACK	ARTIST_ID	1
KS_ARTIST_TRACK	TRACK_ID	2
KS_BLOCKED_IP	BLOCKED_IP_ID	1
KS_BLOCKED_IP	USER_ID	2
KS_BLOCKED_IP	REASON	3
KS_BLOCKED_IP	IP_ADDRESS	4
KS_BLOCKED_IP	BLOCK_DATE	5
KS_BLOCKED_IP	STATUS	6
KS_GENRE	GENRE_ID	1
KS_GENRE	NAME	2
KS_GENRE	DESC	3
KS_INVOICE	INVOICE_ID	1
KS_INVOICE	USER_ID	2
KS_INVOICE	PAYMENT_ID	3
KS_INVOICE	GEN_DATE	4
KS_INVOICE	AMT	5
KS_PAYMENT	PAYMENT_ID	1
KS_PAYMENT	TRANS_ID	2
KS_PAYMENT	PAYMENT_METHOD	3
KS_PAYMENT	LINE_1	4
KS_PAYMENT	LINE_2	5
KS_PAYMENT	CITY	6
KS_PAYMENT	STATE	7
KS_PAYMENT	COUNTRY	8
KS_PLAN	PLAN_ID	1
KS_PLAN	NAME	2
KS_PLAN	PRICE	3
KS_PLAN	DESC	4
KS_PLAN	DURATION	5
KS_PLAYLIST	PL_ID	1
KS_PLAYLIST	USER_ID	2
KS_PLAYLIST	TITLE	3
KS_PLAYLIST	CR_DATE	4
KS_PLAYLIST	DESC	5
KS_TRACK	TRACK_ID	1

Rows 1 - 50. More rows exist.

List of Table Column Constraints



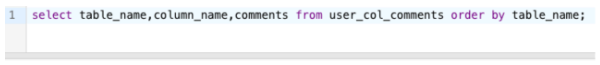
```
select table_name,constraint_name,constraint_type,search_condition,index_name,r_constraint_name,delete_rule from user_constraints
order by table_name;
```

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	INDEX_NAME	R_CONSTRAINT_NAME	DELETE_RULE
KS_ACCESS_CONTROL	SYS_C00155125564	C	"ACC_CONTROL_ID" IS NOT NULL	-	-	-
KS_ACCESS_CONTROL	KS_ACCESS_CONTROL_KS_USER_FK	R	-	-	KS_USER_PK	NO ACTION
KS_ACCESS_CONTROL	SYS_C00155125566	C	"DEVICE_NAME" IS NOT NULL	-	-	-
KS_ACCESS_CONTROL	SYS_C00155125567	C	"IP_ADDRESS" IS NOT NULL	-	-	-
KS_ACCESS_CONTROL	KS_ACCESS_CONTROL_PK	P	-	KS_ACCESS_CONTROL_PK	-	-
KS_ACCESS_CONTROL	SYS_C00155125565	C	"USER_ID" IS NOT NULL	-	-	-
KS_ALBUM	SYS_C00155125570	C	"TITLE" IS NOT NULL	-	-	-
KS_ALBUM	SYS_C00155125571	C	"RL_DATE" IS NOT NULL	-	-	-
KS_ALBUM	KS_ALBUM_PK	P	-	KS_ALBUM_PK	-	-
KS_ALBUM	SYS_C00155125569	C	"ALBUM_ID" IS NOT NULL	-	-	-
KS_ARTIST	KS_ARTIST_PK	P	-	KS_ARTIST_PK	-	-
KS_ARTIST	SYS_C00155125574	C	"FNAME" IS NOT NULL	-	-	-
KS_ARTIST	SYS_C00155125577	C	"BIO" IS NOT NULL	-	-	-
KS_ARTIST	SYS_C00155125576	C	"DOB" IS NOT NULL	-	-	-
KS_ARTIST	SYS_C00155125575	C	"LNAME" IS NOT NULL	-	-	-
KS_ARTIST	SYS_C00155125573	C	"ARTIST_ID" IS NOT NULL	-	-	-
KS_ARTIST_ALBUM	KS_ARTIST_ALBUM_KS_ALBUM_FK	R	-	-	KS_ALBUM_PK	NO ACTION
KS_ARTIST_ALBUM	KS_ARTIST_ALBUM_KS_ARTIST_FK	R	-	-	KS_ARTIST_PK	NO ACTION
KS_ARTIST_ALBUM	KS_ARTIST_ALBUM_PK	P	-	KS_ARTIST_ALBUM_PK	-	-
KS_ARTIST_ALBUM	SYS_C00155125580	C	"ALBUM_ID" IS NOT NULL	-	-	-
KS_ARTIST_ALBUM	SYS_C00155125579	C	"ARTIST_ID" IS NOT NULL	-	-	-
KS_ARTIST_TRACK	SYS_C00155125582	C	"ARTIST_ID" IS NOT NULL	-	-	-
KS_ARTIST_TRACK	SYS_C00155125583	C	"TRACK_ID" IS NOT NULL	-	-	-
KS_ARTIST_TRACK	KS_ARTIST_TRACK_PK	P	-	KS_ARTIST_TRACK_PK	-	-
KS_ARTIST_TRACK	KS_ARTIST_TRACK_KS_TRACK_FK	R	-	-	KS_TRACK_PK	NO ACTION
KS_ARTIST_TRACK	KS_ARTIST_TRACK_KS_ARTIST_FK	R	-	-	KS_ARTIST_PK	NO ACTION
KS_BLOCKED_IP	SYS_C00155125585	C	"BLOCKED_IP_ID" IS NOT NULL	-	-	-
KS_BLOCKED_IP	KS_BLOCKED_IP_KS_USER_FK	R	-	-	KS_USER_PK	NO ACTION
KS_BLOCKED_IP	SYS_C00155125586	C	"IP_ADDRESS" IS NOT NULL	-	-	-
KS_BLOCKED_IP	SYS_C00155125588	C	"STATUS" IS NOT NULL	-	-	-
KS_BLOCKED_IP	KS_BLOCKED_IP_PK	P	-	KS_BLOCKED_IP_PK	-	-
KS_BLOCKED_IP	SYS_C00155125587	C	"BLOCK_DATE" IS NOT NULL	-	-	-
KS_GENRE	SYS_C00155125592	C	"DESC" IS NOT NULL	-	-	-
KS_GENRE	KS_GENRE_PK	P	-	KS_GENRE_PK	-	-
KS_GENRE	SYS_C00155125591	C	"NAME" IS NOT NULL	-	-	-
KS_GENRE	SYS_C00155125590	C	"GENRE_ID" IS NOT NULL	-	-	-
KS_INVOICE	SYS_C00155125594	C	"INVOICE_ID" IS NOT NULL	-	-	-
KS_INVOICE	SYS_C00155125595	C	"PAYMENT_ID" IS NOT NULL	-	-	-
KS_INVOICE	KS_INVOICE_PK	P	-	KS_INVOICE_PK	-	-
KS_INVOICE	SYS_C00155125596	C	"GEN_DATE" IS NOT NULL	-	-	-
KS_INVOICE	SYS_C00155125597	C	"AMT" IS NOT NULL	-	-	-
KS_INVOICE	KS_INVOICE_KS_PAYMENT_FK	R	-	-	KS_PAYMENT_PK	NO ACTION
KS_INVOICE	KS_INVOICE_KS_USER_FK	R	-	-	KS_USER_PK	NO ACTION
KS_PAYMENT	KS_PAYMENT_PK	P	-	KS_PAYMENT_PK	-	-
KS_PAYMENT	SYS_C00155125601	C	"PAYMENT_METHOD" IS NOT NULL	-	-	-
KS_PAYMENT	SYS_C00155125600	C	"TRANS_ID" IS NOT NULL	-	-	-
KS_PAYMENT	SYS_C00155125599	C	"PAYMENT_ID" IS NOT NULL	-	-	-
KS_PLAN	SYS_C00155125603	C	"PLAN_ID" IS NOT NULL	-	-	-
KS_PLAN	SYS_C00155125604	C	"NAME" IS NOT NULL	-	-	-
KS_PLAN	SYS_C00155125605	C	"PRICE" IS NOT NULL	-	-	-

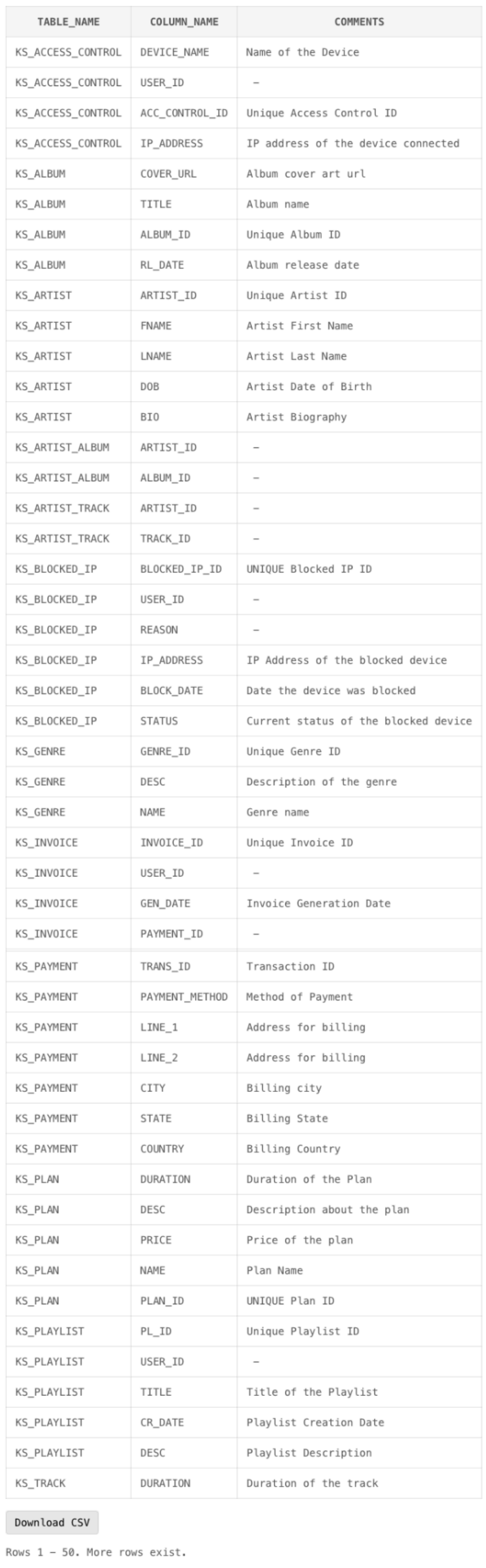
Download CSV

Rows 1 - 50. More rows exist.

List of Table Column Comments



```sql
select table_name,column_name,comments from user_col_comments order by table_name;
```

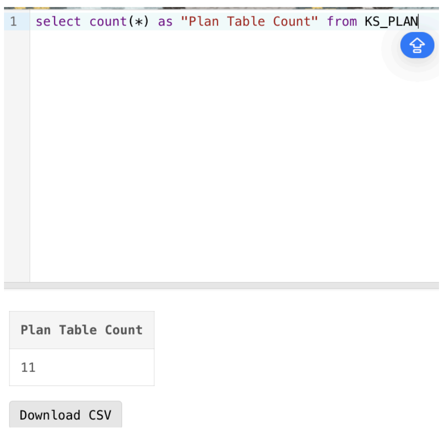


TABLE_NAME	COLUMN_NAME	COMMENTS
KS_ACCESS_CONTROL	DEVICE_NAME	Name of the Device
KS_ACCESS_CONTROL	USER_ID	-
KS_ACCESS_CONTROL	ACC_CONTROL_ID	Unique Access Control ID
KS_ACCESS_CONTROL	IP_ADDRESS	IP address of the device connected
KS_ALBUM	COVER_URL	Album cover art url
KS_ALBUM	TITLE	Album name
KS_ALBUM	ALBUM_ID	Unique Album ID
KS_ALBUM	RL_DATE	Album release date
KS_ARTIST	ARTIST_ID	Unique Artist ID
KS_ARTIST	FNAME	Artist First Name
KS_ARTIST	LNAME	Artist Last Name
KS_ARTIST	DOB	Artist Date of Birth
KS_ARTIST	BIO	Artist Biography
KS_ARTIST_ALBUM	ARTIST_ID	-
KS_ARTIST_ALBUM	ALBUM_ID	-
KS_ARTIST_TRACK	ARTIST_ID	-
KS_ARTIST_TRACK	TRACK_ID	-
KS_BLOCKED_IP	BLOCKED_IP_ID	UNIQUE Blocked IP ID
KS_BLOCKED_IP	USER_ID	-
KS_BLOCKED_IP	REASON	-
KS_BLOCKED_IP	IP_ADDRESS	IP Address of the blocked device
KS_BLOCKED_IP	BLOCK_DATE	Date the device was blocked
KS_BLOCKED_IP	STATUS	Current status of the blocked device
KS_GENRE	GENRE_ID	Unique Genre ID
KS_GENRE	DESC	Description of the genre
KS_GENRE	NAME	Genre name
KS_INVOICE	INVOICE_ID	Unique Invoice ID
KS_INVOICE	USER_ID	-
KS_INVOICE	GEN_DATE	Invoice Generation Date
KS_INVOICE	PAYMENT_ID	-
KS_PAYMENT	TRANS_ID	Transaction ID
KS_PAYMENT	PAYMENT_METHOD	Method of Payment
KS_PAYMENT	LINE_1	Address for billing
KS_PAYMENT	LINE_2	Address for billing
KS_PAYMENT	CITY	Billing city
KS_PAYMENT	STATE	Billing State
KS_PAYMENT	COUNTRY	Billing Country
KS_PLAN	DURATION	Duration of the Plan
KS_PLAN	DESC	Description about the plan
KS_PLAN	PRICE	Price of the plan
KS_PLAN	NAME	Plan Name
KS_PLAN	PLAN_ID	UNIQUE Plan ID
KS_PLAYLIST	PL_ID	Unique Playlist ID
KS_PLAYLIST	USER_ID	-
KS_PLAYLIST	TITLE	Title of the Playlist
KS_PLAYLIST	CR_DATE	Playlist Creation Date
KS_PLAYLIST	DESC	Playlist Description
KS_TRACK	DURATION	Duration of the track

Rows 1 - 50. More rows exist.

Count Query for Each Table Screen Shot

Plan Table

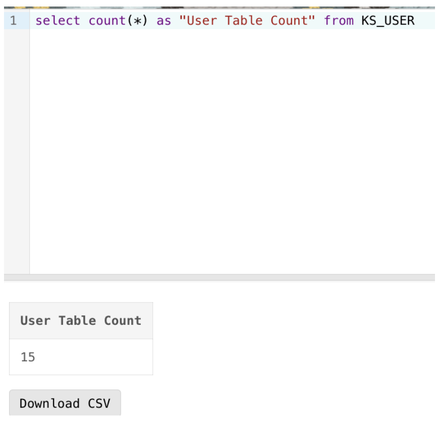


```
select count(*) as "Plan Table Count" from KS_PLAN
```

Plan Table Count
11

Download CSV

User Table

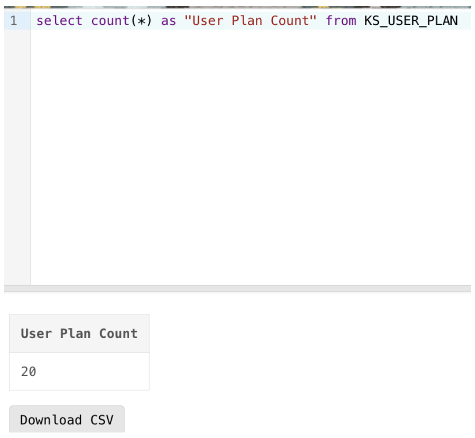


```
select count(*) as "User Table Count" from KS_USER
```

User Table Count
15

Download CSV

User Plan Table

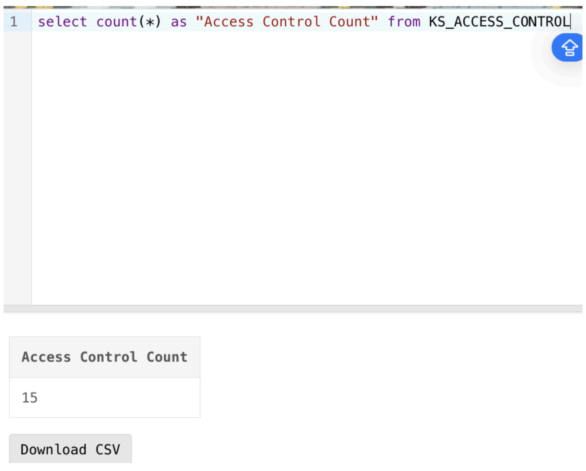


```
select count(*) as "User Plan Count" from KS_USER_PLAN
```

User Plan Count
20

Download CSV

Access Control Table

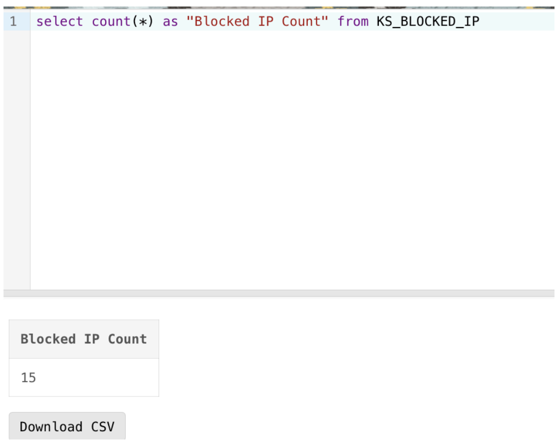


```
select count(*) as "Access Control Count" from KS_ACCESS_CONTROL
```

Access Control Count
15

Download CSV

Blocked IP Table

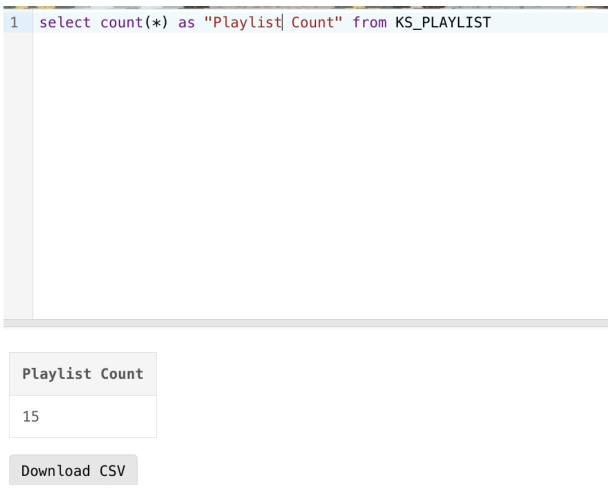


```
select count(*) as "Blocked IP Count" from KS_BLOCKED_IP
```

Blocked IP Count
15

Download CSV

Playlist Table

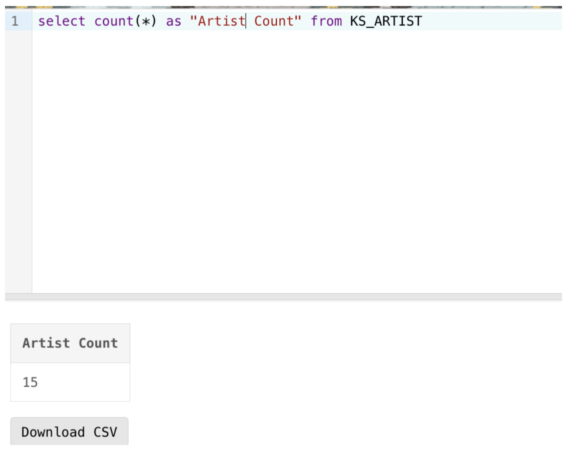


```
select count(*) as "Playlist Count" from KS_PLAYLIST
```

Playlist Count
15

Download CSV

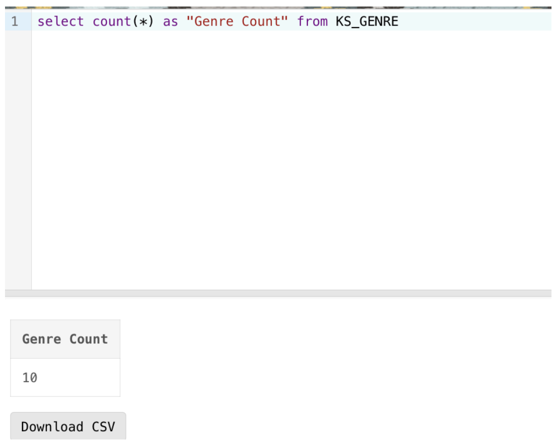
Artist Table



```sql
select count(*) as "Artist Count" from KS_ARTIST
```

Artist Count
15

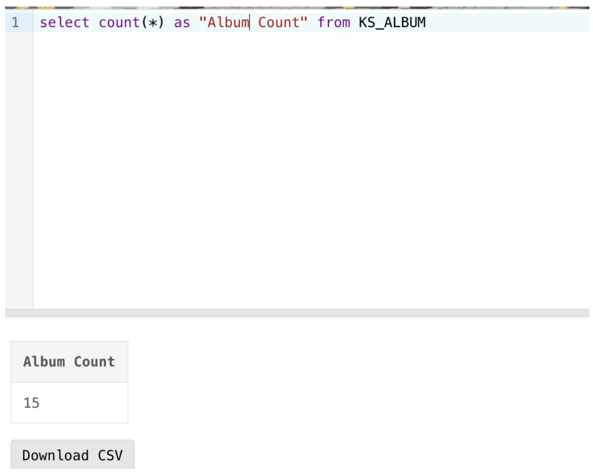
Genre Table



```sql
select count(*) as "Genre Count" from KS_GENRE
```

Genre Count
10

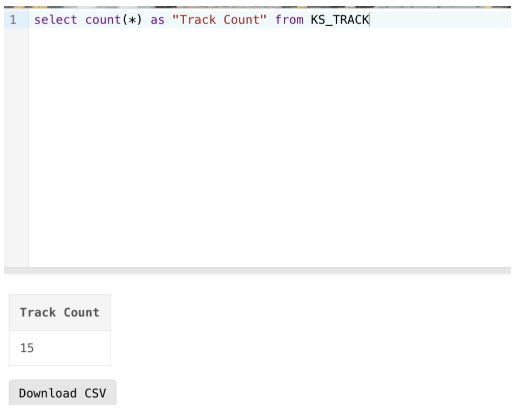
Album Table



```sql
select count(*) as "Album Count" from KS_ALBUM
```

Album Count
15

Track Table

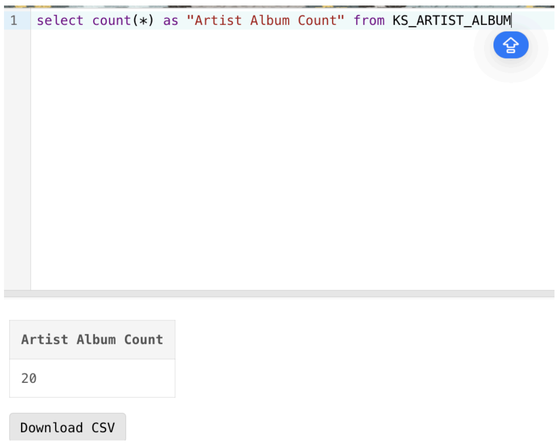


```
select count(*) as "Track Count" from KS_TRACK
```

Track Count
15

Download CSV

Artist Album Table

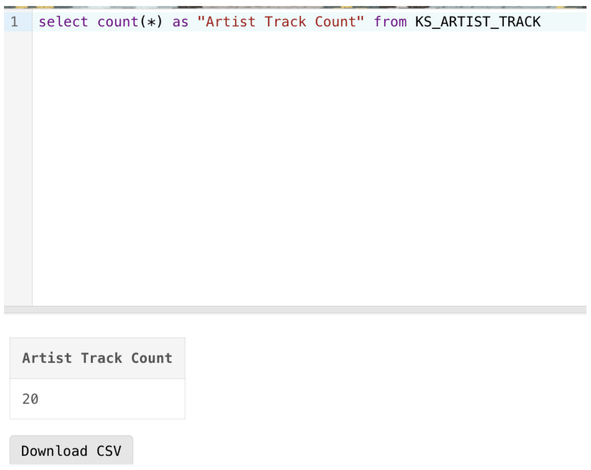


```
select count(*) as "Artist Album Count" from KS_ARTIST_ALBUM
```

Artist Album Count
20

Download CSV

Artist Track Table

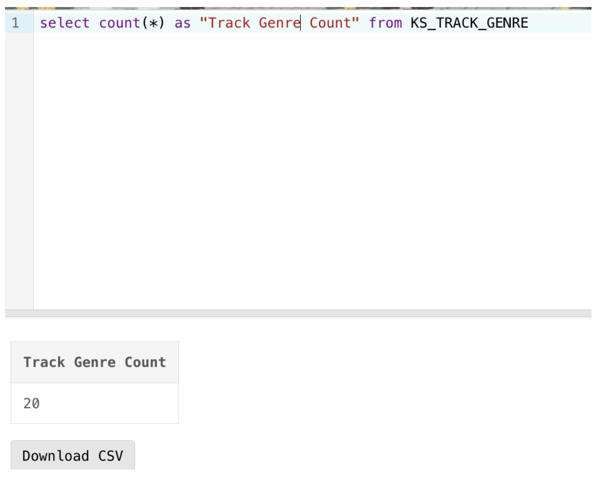


```
select count(*) as "Artist Track Count" from KS_ARTIST_TRACK
```

Artist Track Count
20

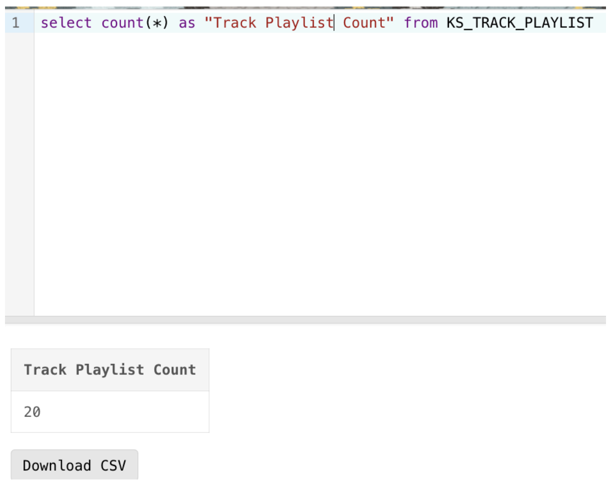
Download CSV

Track Genre Table


```
select count(*) as "Track Genre Count" from KS_TRACK_GENRE
```

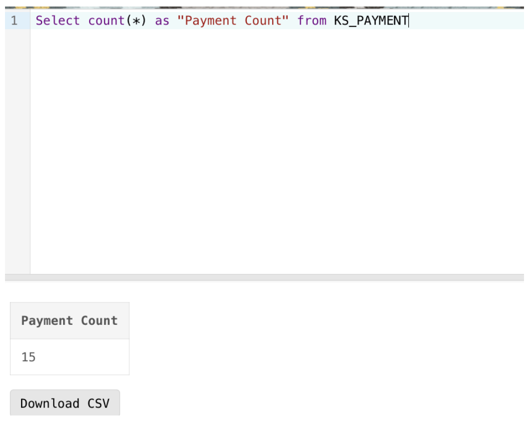
Track Genre Count
20

Track Playlist Table


```
select count(*) as "Track Playlist Count" from KS_TRACK_PLAYLIST
```

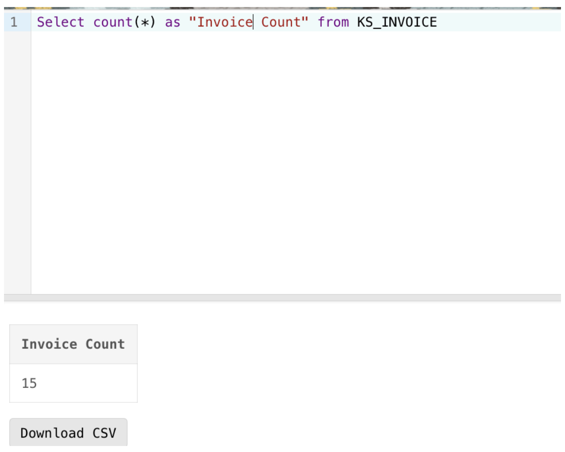
Track Playlist Count
20

Payment Table


```
Select count(*) as "Payment Count" from KS_PAYMENT
```

Payment Count
15

Invoice Table



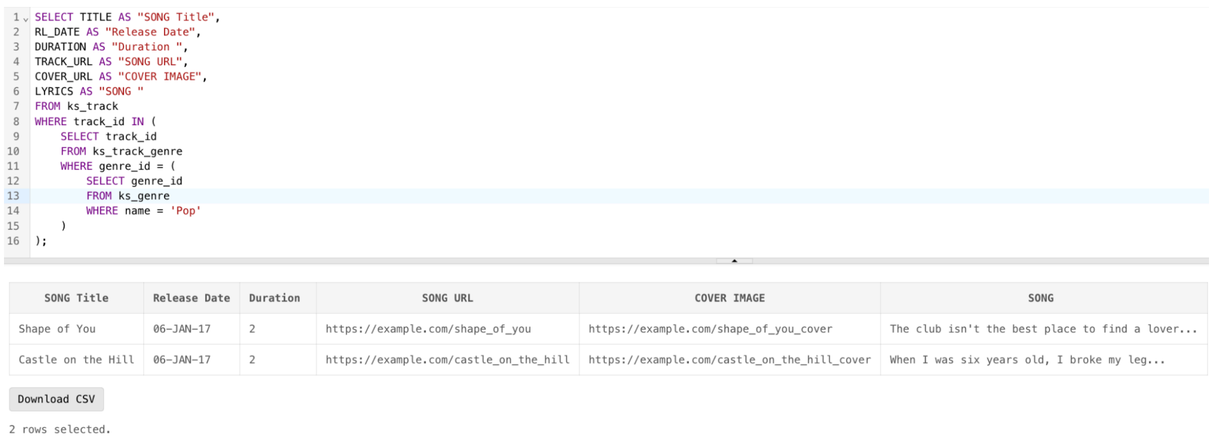
```
Select count(*) as "Invoice Count" from KS_INVOICE
```

Invoice Count
15

SQL Queries

Two queries using Subqueries

Query 1



```sql
SELECT TITLE AS "SONG Title",
RL_DATE AS "Release Date",
DURATION AS "Duration ",
TRACK_URL AS "SONG URL",
COVER_URL AS "COVER IMAGE",
LYRICS AS "SONG "
FROM ks_track
WHERE track_id IN (
    SELECT track_id
    FROM ks_track_genre
    WHERE genre_id = (
        SELECT genre_id
        FROM ks_genre
        WHERE name = 'Pop'
    )
);
```

SONG Title	Release Date	Duration	SONG URL	COVER IMAGE	SONG
Shape of You	06-JAN-17	2	https://example.com/shape_of_you	https://example.com/shape_of_you_cover	The club isn't the best place to find a lover...
Castle on the Hill	06-JAN-17	2	https://example.com/castle_on_the_hill	https://example.com/castle_on_the_hill_cover	When I was six years old, I broke my leg...

```
2 rows selected.
```

Business Purpose of the Query

- **Personalized Recommendations:** By selecting tracks based on specific genres, the platform enhances user experience by offering personalized recommendations tailored to individual preferences.
- **Genre-based Playlists:** Users can easily create or explore genre-based playlists, fostering engagement and discovery within the platform.
- **Targeted Marketing:**Analyzing user interactions with tracks of specific genres allows for targeted marketing campaigns, promoting relevant content to users with similar preferences.

Explanation of the Query

- **Innermost Subquery:**

Selects the genre_id from the ks_genre table where the name of the genre is Pop

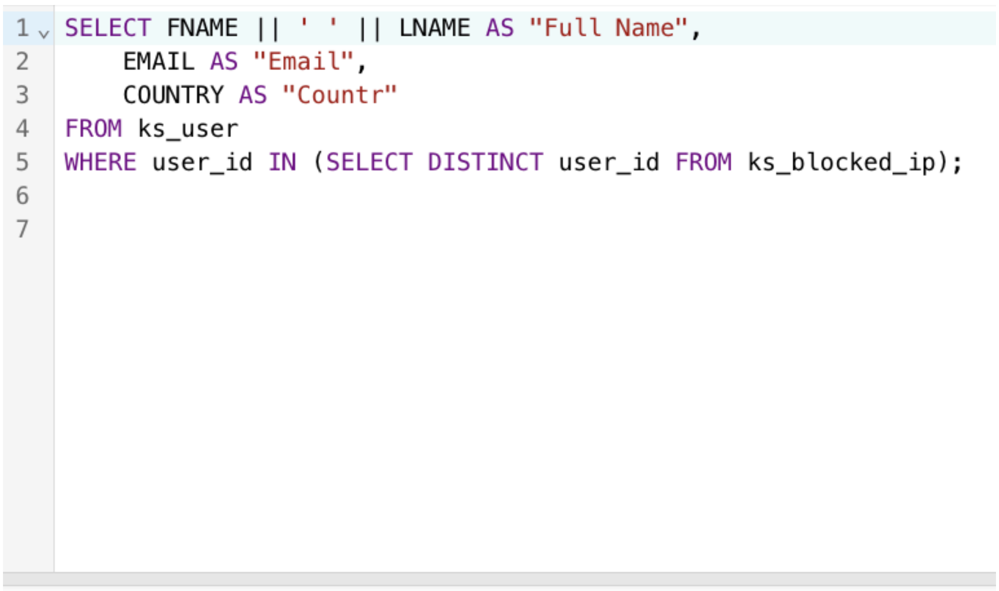
- **Intermediate Subquery:**

Selects track_ids from the ks_track_genre table where the genre_id matches the genre_id selected in the innermost subquery.

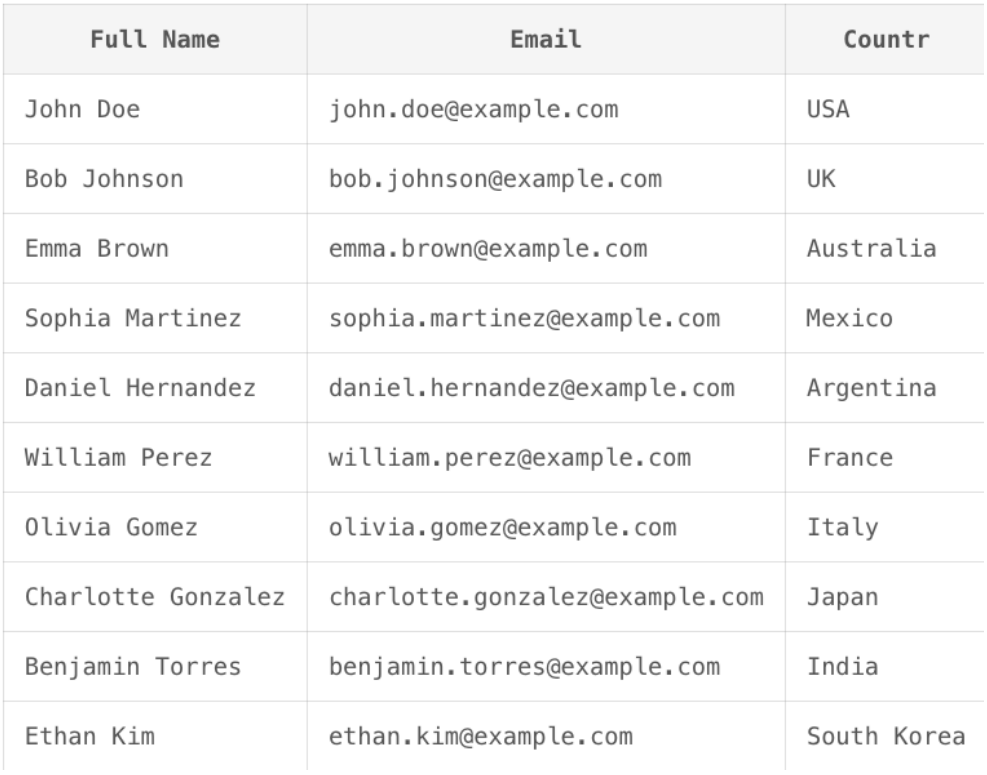
- **Outermost Query:**

Selects all columns from the ks_track table where the track_id matches any of the track_ids selected in the intermediate subquery.

QUERY 2



```sql
SELECT FNAME || ' ' || LNAME AS "Full Name",
     EMAIL AS "Email",
     COUNTRY AS "Countr"
FROM ks_user
WHERE user_id IN (SELECT DISTINCT user_id FROM ks_blocked_ip);
```



Full Name	Email	Countr
John Doe	john.doe@example.com	USA
Bob Johnson	bob.johnson@example.com	UK
Emma Brown	emma.brown@example.com	Australia
Sophia Martinez	sophia.martinez@example.com	Mexico
Daniel Hernandez	daniel.hernandez@example.com	Argentina
William Perez	william.perez@example.com	France
Olivia Gomez	olivia.gomez@example.com	Italy
Charlotte Gonzalez	charlotte.gonzalez@example.com	Japan
Benjamin Torres	benjamin.torres@example.com	India
Ethan Kim	ethan.kim@example.com	South Korea

Business Purpose of the Query

- Identify users who have encountered issues with their IP addresses being blocked.
- Helps customer support or security teams to monitor and manage potentially problematic user accounts.
- Allows for targeted communication or assistance to users experiencing access issues due to blocked IPs.

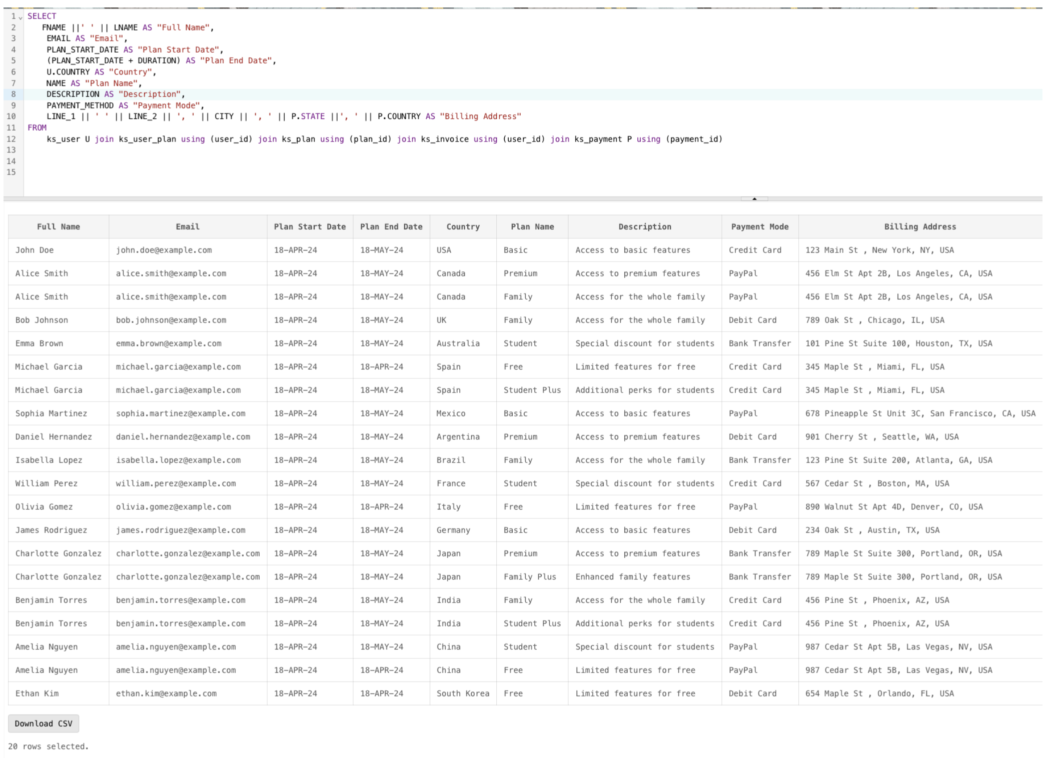
Explanation of the Query

- SELECT FNAME || ' ' || LNAME AS "Full Name", Concatenates the FNAME and LNAME columns to create a new column named "Full Name".
- EMAIL AS "Email", Selects the EMAIL column as it is.
- COUNTRY AS "Country", Selects the COUNTRY column as it is.
- FROM ks_user Specifies the table from which to select data, which is ks_user.
- WHERE user_id IN (SELECT DISTINCT user_id FROM ks_blocked_ip):

  Filters the results to include only users whose user_id exists in the subquery result.The subquery selects distinct user_id values from the ks_blocked_ip table, identifying users with blocked IP issues.

Two queries using Table joins (minimum three table joins)

QUERY 1



```sql
SELECT
   FNAME ||' ' || LNAME AS "Full Name",
    EMAIL AS "Email",
    PLAN_START_DATE AS "Plan Start Date",
    (PLAN_START_DATE + DURATION) AS "Plan End Date",
    U.COUNTRY AS "Country",
    NAME AS "Plan Name",
    DESCRIPTION AS "Description",
    PAYMENT_METHOD AS "Payment Mode",
    LINE_1 || ' ' || LINE_2 || ', ' || CITY || ', ' || P.STATE ||', ' || P.COUNTRY AS "Billing Address"
FROM
    ks_user U join ks_user_plan using (user_id) join ks_plan using (plan_id) join ks_invoice using (user_id) join ks_payment P using (payment_id)
```

Full Name	Email	Plan Start Date	Plan End Date	Country	Plan Name	Description	Payment Mode	Billing Address
John Doe	john.doe@example.com	18-APR-24	18-MAY-24	USA	Basic	Access to basic features	Credit Card	123 Main St , New York, NY, USA
Alice Smith	alice.smith@example.com	18-APR-24	18-MAY-24	Canada	Premium	Access to premium features	PayPal	456 Elm St Apt 2B, Los Angeles, CA, USA
Alice Smith	alice.smith@example.com	18-APR-24	18-MAY-24	Canada	Family	Access for the whole family	PayPal	456 Elm St Apt 2B, Los Angeles, CA, USA
Bob Johnson	bob.johnson@example.com	18-APR-24	18-MAY-24	UK	Family	Access for the whole family	Debit Card	789 Oak St , Chicago, IL, USA
Emma Brown	emma.brown@example.com	18-APR-24	18-MAY-24	Australia	Student	Special discount for students	Bank Transfer	101 Pine St Suite 100, Houston, TX, USA
Michael Garcia	michael.garcia@example.com	18-APR-24	18-APR-24	Spain	Free	Limited features for free	Credit Card	345 Maple St , Miami, FL, USA
Michael Garcia	michael.garcia@example.com	18-APR-24	18-MAY-24	Spain	Student Plus	Additional perks for students	Credit Card	345 Maple St , Miami, FL, USA
Sophia Martinez	sophia.martinez@example.com	18-APR-24	18-MAY-24	Mexico	Basic	Access to basic features	PayPal	678 Pineapple St Unit 3C, San Francisco, CA, USA
Daniel Hernandez	daniel.hernandez@example.com	18-APR-24	18-MAY-24	Argentina	Premium	Access to premium features	Debit Card	901 Cherry St , Seattle, WA, USA
Isabella Lopez	isabella.lopez@example.com	18-APR-24	18-MAY-24	Brazil	Family	Access for the whole family	Bank Transfer	123 Pine St Suite 200, Atlanta, GA, USA
William Perez	william.perez@example.com	18-APR-24	18-MAY-24	France	Student	Special discount for students	Credit Card	567 Cedar St , Boston, MA, USA
Olivia Gomez	olivia.gomez@example.com	18-APR-24	18-APR-24	Italy	Free	Limited features for free	PayPal	890 Walnut St Apt 4D, Denver, CO, USA
James Rodriguez	james.rodriguez@example.com	18-APR-24	18-MAY-24	Germany	Basic	Access to basic features	Debit Card	234 Oak St , Austin, TX, USA
Charlotte Gonzalez	charlotte.gonzalez@example.com	18-APR-24	18-MAY-24	Japan	Premium	Access to premium features	Bank Transfer	789 Maple St Suite 300, Portland, OR, USA
Charlotte Gonzalez	charlotte.gonzalez@example.com	18-APR-24	18-MAY-24	Japan	Family Plus	Enhanced family features	Bank Transfer	789 Maple St Suite 300, Portland, OR, USA
Benjamin Torres	benjamin.torres@example.com	18-APR-24	18-MAY-24	India	Family	Access for the whole family	Credit Card	456 Pine St , Phoenix, AZ, USA
Benjamin Torres	benjamin.torres@example.com	18-APR-24	18-MAY-24	India	Student Plus	Additional perks for students	Credit Card	456 Pine St , Phoenix, AZ, USA
Amelia Nguyen	amelia.nguyen@example.com	18-APR-24	18-MAY-24	China	Student	Special discount for students	PayPal	987 Cedar St Apt 5B, Las Vegas, NV, USA
Amelia Nguyen	amelia.nguyen@example.com	18-APR-24	18-APR-24	China	Free	Limited features for free	PayPal	987 Cedar St Apt 5B, Las Vegas, NV, USA
Ethan Kim	ethan.kim@example.com	18-APR-24	18-APR-24	South Korea	Free	Limited features for free	Debit Card	654 Maple St , Orlando, FL, USA

```
20 rows selected.
```

Business Purpose of the Query

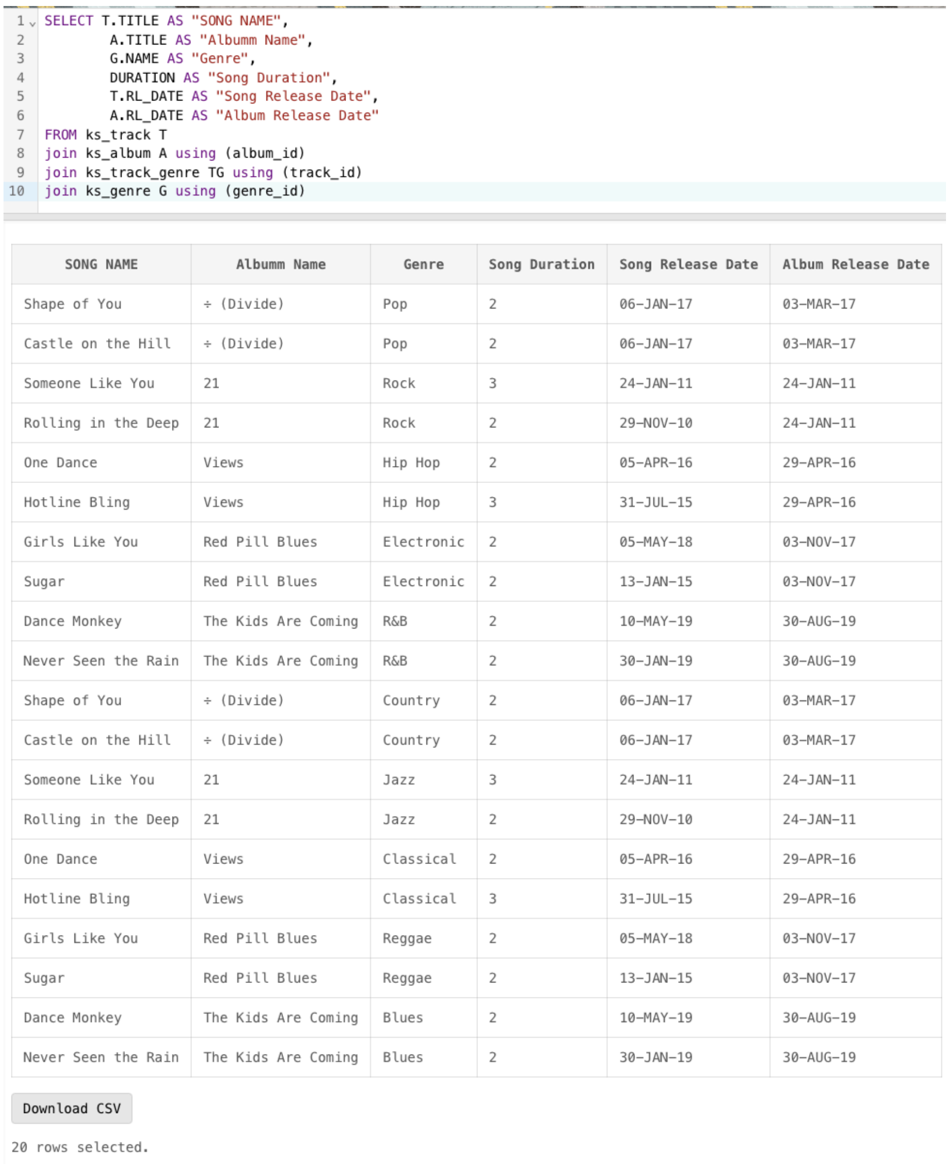
The business purpose of this query is to generate a report containing comprehensive details about users, their subscription plans, payment information, and billing addresses.

- **User Information:** Retrieves the full name, email address, country, and plan start date of each user.
- **Plan Information:** Includes details such as the plan name and description, providing insight into the type of subscription each user has.
- **Subscription Duration:** Calculates the plan end date by adding the plan duration to the plan start date, allowing for easy understanding of when each subscription expires.
- **Payment Details:** Presents the payment method used by each user for their subscription.
- **Billing Address:** Constructs the complete billing address by concatenating the address lines, city, state, and country, facilitating communication or verification processes.

Explanation of the Query

- **SELECT Clause:**
  - Concatenates the user's first name (FNAME) and last name (LNAME) to create a column labeled "Full Name".
  - Selects the user's email address (EMAIL).
  - Retrieves the plan start date (PLAN_START_DATE).
  - Calculates the plan end date by adding the plan duration (DURATION) to the plan start date.
  - Selects the user's country (COUNTRY).
  - Includes the plan name (NAME) and description (DESCRIPTION) from the subscription plan table.
  - Retrieves the payment method (PAYMENT_METHOD) used by the user.
  - Constructs the complete billing address by concatenating address lines (LINE_1 and LINE_2), city (CITY), state (STATE), and country (COUNTRY) from the payment table.
- **FROM Clause:**
  - Specifies the tables from which to retrieve data: ks_user, ks_user_plan, ks_plan, ks_invoice, ks_payment.
  - Joins these tables using common keys (user_id, plan_id, payment_id) to link user information, subscription details, payment information, and billing addresses.
- **JOIN Conditions:**
  - Joins the ks_user table with ks_user_plan using the user_id column.
  - Joins the resulting table with the ks_plan table using the plan_id column.
  - Joins the resulting table with the ks_invoice table using the user_id column.
  - Joins the resulting table with the ks_payment table using the payment_id column.

QUERY 2



```sql
SELECT T.TITLE AS "SONG NAME",
        A.TITLE AS "Albumm Name",
        G.NAME AS "Genre",
        DURATION AS "Song Duration",
        T.RL_DATE AS "Song Release Date",
        A.RL_DATE AS "Album Release Date"
FROM ks_track T
join ks_album A using (album_id)
join ks_track_genre TG using (track_id)
join ks_genre G using (genre_id)
```

SONG NAME	Albumm Name	Genre	Song Duration	Song Release Date	Album Release Date
Shape of You	÷ (Divide)	Pop	2	06-JAN-17	03-MAR-17
Castle on the Hill	÷ (Divide)	Pop	2	06-JAN-17	03-MAR-17
Someone Like You	21	Rock	3	24-JAN-11	24-JAN-11
Rolling in the Deep	21	Rock	2	29-NOV-10	24-JAN-11
One Dance	Views	Hip Hop	2	05-APR-16	29-APR-16
Hotline Bling	Views	Hip Hop	3	31-JUL-15	29-APR-16
Girls Like You	Red Pill Blues	Electronic	2	05-MAY-18	03-NOV-17
Sugar	Red Pill Blues	Electronic	2	13-JAN-15	03-NOV-17
Dance Monkey	The Kids Are Coming	R&B	2	10-MAY-19	30-AUG-19
Never Seen the Rain	The Kids Are Coming	R&B	2	30-JAN-19	30-AUG-19
Shape of You	÷ (Divide)	Country	2	06-JAN-17	03-MAR-17
Castle on the Hill	÷ (Divide)	Country	2	06-JAN-17	03-MAR-17
Someone Like You	21	Jazz	3	24-JAN-11	24-JAN-11
Rolling in the Deep	21	Jazz	2	29-NOV-10	24-JAN-11
One Dance	Views	Classical	2	05-APR-16	29-APR-16
Hotline Bling	Views	Classical	3	31-JUL-15	29-APR-16
Girls Like You	Red Pill Blues	Reggae	2	05-MAY-18	03-NOV-17
Sugar	Red Pill Blues	Reggae	2	13-JAN-15	03-NOV-17
Dance Monkey	The Kids Are Coming	Blues	2	10-MAY-19	30-AUG-19
Never Seen the Rain	The Kids Are Coming	Blues	2	30-JAN-19	30-AUG-19

```
20 rows selected.
```

Business Purpose of the Query

The business purpose of this query is to:

- **Track Information Retrieval:** The query aims to retrieve specific information about tracks stored in the database.
- **Album Association:** It associates each track with its corresponding album by retrieving the album name and release date.
- **Genre Identification:** The query identifies the genre of each track, providing insights into the diversity of music genres available in the database.

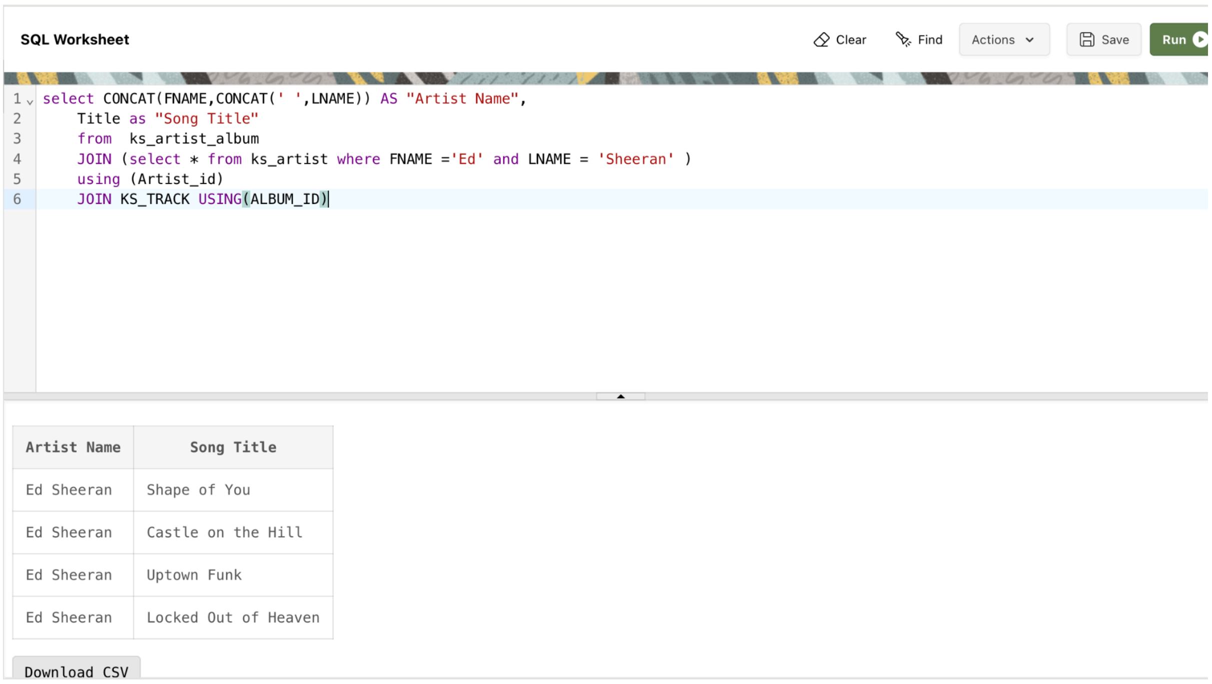
- **Duration and Release Date Details:**It retrieves the duration and release date of each track, allowing users to assess the length and chronological order of tracks.

Explanation of the Query

- **SELECT Clause:**
    - Retrieves specific columns from the tables being queried.
    - T.TITLE AS "SONG NAME": Selects the title of the track and labels it as "SONG NAME".
    - A.TITLE AS "Album Name": Selects the title of the album and labels it as "Album Name".
    - G.NAME AS "Genre": Selects the name of the genre and labels it as "Genre".
    - DURATION AS "Song Duration": Selects the duration of the song and labels it as "Song Duration".
    - T.RL_DATE AS "Song Release Date": Selects the release date of the song and labels it as "Song Release Date".
    - A.RL_DATE AS "Album Release Date": Selects the release date of the album and labels it as "Album Release Date".
- **FROM Clause:**
    - Specifies the tables involved in the query: ks_track, ks_album, ks_track_genre, and ks_genre.
- **JOIN Conditions:**
    - JOIN ks_album A USING (album_id): Joins the ks_track table with the ks_album table using the album_id column to link tracks with their respective albums.
    - JOIN ks_track_genre TG USING (track_id): Joins the result with the ks_track_genre table using the track_id column to associate tracks with their genres.
    - JOIN ks_genre G USING (genre_id): Joins the result with the ks_genre table using the genre_id column to retrieve the genre names.

One query using in-line View

Query 1



```
select CONCAT(FNAME,CONCAT(' ',LNAME)) AS "Artist Name",
    Title as "Song Title"
    from  ks_artist_album
    JOIN (select * from ks_artist where FNAME ='Ed' and LNAME = 'Sheeran' )
    using (Artist_id)
    JOIN KS_TRACK USING(ALBUM_ID)
```

Artist Name	Song Title
Ed Sheeran	Shape of You
Ed Sheeran	Castle on the Hill
Ed Sheeran	Uptown Funk
Ed Sheeran	Locked Out of Heaven

Business Purpose of the Query

The business purpose of this query is to :

- **Retrieve Songs:** Obtain a list of songs.
- **By Artist:** Specifically, those performed by a particular artist.
- **Artist Identification:** Identified by the name "Ed Sheeran".
- **For Display or Analysis:** Likely for displaying on an artist's page or for analytical purposes like popularity assessment or recommendations.

Explanation of the Query

- **SELECT Clause:**
  - We're selecting specific columns to display in the result set.
  - CONCAT(FNAME, ' ', LNAME) AS "Artist Name": Concatenating the first name and last name from the "ks_artist" table to form the artist's full name.
  - Title AS "Song Title": Selecting the song title from the "ks_track" table.

- **FROM Clause:**
  - We're specifying the main tables involved in the query.
  - ks_artist_album: This table likely contains information about albums and the artists associated with them.
  - ks_artist: This table likely stores details about artists, including their first and last names.

  - ks_track: This table presumably holds information about individual tracks/songs.

- **JOIN Clauses:**
  - We're joining multiple tables to retrieve related data.

  - JOIN (SELECT * FROM ks_artist WHERE FNAME ='Ed' AND LNAME = 'Sheeran'): This is an in-line view that filters the "ks_artist" table to only include records where the artist's first name is 'Ed' and last name is 'Sheeran'.

  - USING (Artist_id): This join condition connects the filtered artist data with the "ks_artist_album" table based on the Artist_id column.

  - JOIN KS_TRACK USING(ALBUM_ID): This join connects the "ks_artist_album" table with the "ks_track" table based on the ALBUM_ID column.