

DEDS Assignment 1

Author: Vincent Itucal

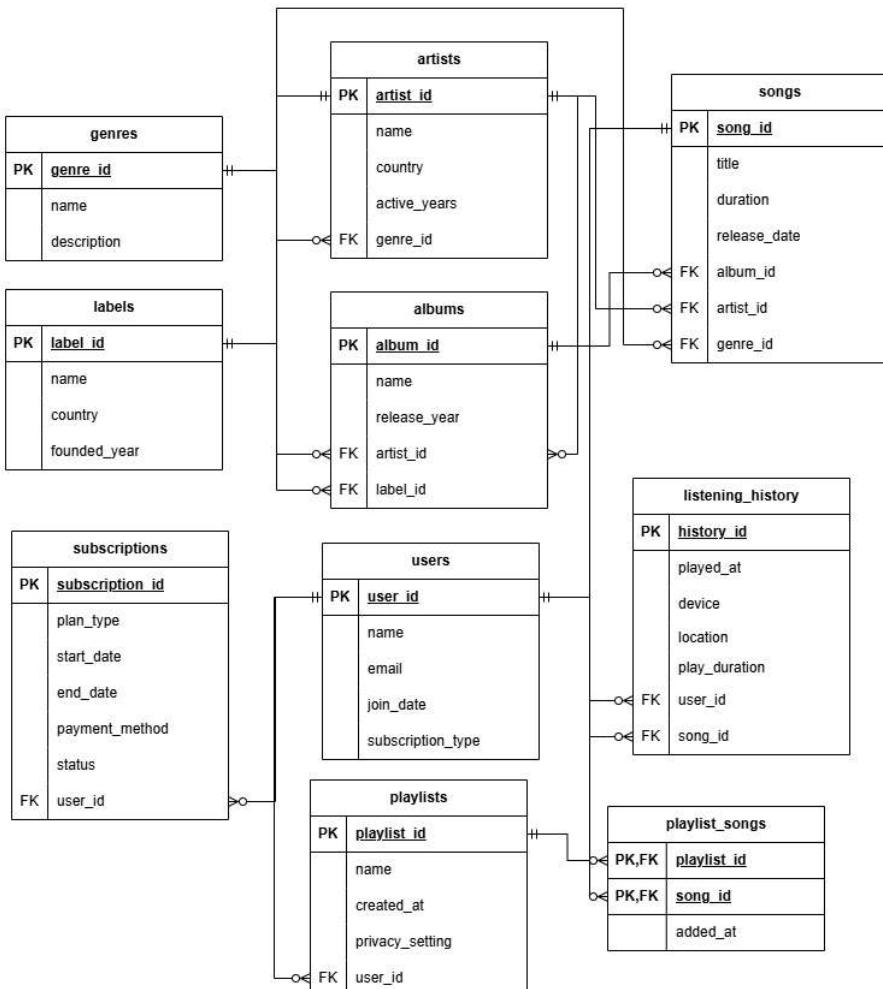
```
In [3]: from faker import Faker
import random
import redshift_connector
import pandas as pd
from sqlalchemy import create_engine
import psycopg2
from decouple import Config, RepositoryEnv
```

```
In [4]: import warnings
warnings.filterwarnings('ignore')
```

```
In [5]: %load_ext sql
```

OLTP Implementation of Music Streaming Service

OLTP ERD



RDS Console Details

sampledb

Summary

| | | | | |
|---------------------------|-------------------------------|------------------|---------------------------|------------------------------------|
| DB identifier sampledb | Status Stopped temporarily | Role Instance | Engine PostgreSQL | Recommendations 1 Informational |
| CPU - | Class db.t4g.micro | Current activity | Region & AZ us-east-1c | |

Connectivity & security Monitoring Logs & events Configuration Maintenance & backups Data migrations - new Tags Recommendations

Connectivity & security

Endpoint & port

Endpoint
 sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com

Port
5432

Networking

Availability Zone
us-east-1c

VPC
vpc-09d0858797f7926b1

Subnet group
default-vpc-09d0858797f7926b1

Subnets

- subnet-0374b24e73f4dc15b
- subnet-0fc4232211cad391c
- subnet-008c0fdc97b79899
- subnet-0e00c95f7a9a2bd3c
- subnet-0538daef62385216
- subnet-0119fdd68db5a1888

Security

VPC security groups

- default (sg-09d47fc95789e98e4)
- Active
- rds-ec2-1 (sg-058391001a33bcd9b)
- Active

Publicly accessible
No

Certificate authority [Info](#)
rds-ca-rsa2048-1

Certificate authority date
May 26, 2061, 07:34 (UTC+08:00)

DB instance certificate expiration date
January 13, 2026, 11:30 (UTC+08:00)

In [6]: config = Config(RepositoryEnv("../db_pass.txt"))
db_oltp_password = config("oltp_pass")
oltp_connection = config("oltp_connection")

In [7]: connection_string = "postgresql://vincent:{}@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service"

In [8]: engine = create_engine(connection_string)

In [12]: get_ipython().run_line_magic('sql', connection_string)

Connecting to 'postgresql://vincent:**@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'

Create Table Scripts

```
%sql
CREATE TABLE IF NOT EXISTS genres (
    genre_id SERIAL PRIMARY KEY,
    name TEXT,
    description TEXT
);

CREATE TABLE IF NOT EXISTS labels (
    label_id SERIAL PRIMARY KEY,
    name TEXT,
    country TEXT,
    founded_year INT
);

CREATE TABLE IF NOT EXISTS artists (
    artist_id SERIAL PRIMARY KEY,
    name TEXT,
    country TEXT,
    active_years TEXT,
    genre_id INT,
    FOREIGN KEY (genre_id) REFERENCES genres(genre_id)
);

CREATE TABLE IF NOT EXISTS albums (
    album_id SERIAL PRIMARY KEY,
    name TEXT,
    release_year INT,
    artist_id INT,
    label_id INT,
    FOREIGN KEY (artist_id) REFERENCES artists(artist_id),
    FOREIGN KEY (label_id) REFERENCES labels(label_id)
);

CREATE TABLE IF NOT EXISTS songs (
    song_id SERIAL PRIMARY KEY,
    title TEXT,
    duration INT,
    album_id INT,
    artist_id INT,
    genre_id INT,
    release_date DATE,
    FOREIGN KEY (album_id) REFERENCES albums(album_id),
    FOREIGN KEY (artist_id) REFERENCES artists(artist_id)
);
```

```

    FOREIGN KEY (artist_id) REFERENCES artists(artist_id),
    FOREIGN KEY (genre_id) REFERENCES genres(genre_id)
);

CREATE TABLE IF NOT EXISTS users (
    user_id SERIAL PRIMARY KEY,
    name TEXT,
    email TEXT,
    join_date DATE,
    subscription_type TEXT
);

CREATE TABLE IF NOT EXISTS playlists (
    playlist_id SERIAL PRIMARY KEY,
    name TEXT,
    created_at TIMESTAMP,
    privacy_setting TEXT,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE IF NOT EXISTS playlist_songs (
    playlist_id INT,
    song_id INT,
    PRIMARY KEY (playlist_id, song_id),
    added_at TIMESTAMP,
    FOREIGN KEY (playlist_id) REFERENCES playlists(playlist_id),
    FOREIGN KEY (song_id) REFERENCES songs(song_id)
);

CREATE TABLE IF NOT EXISTS listening_history (
    history_id SERIAL PRIMARY KEY,
    played_at TIMESTAMP,
    device TEXT,
    location TEXT,
    play_duration INT,
    user_id INT,
    song_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (song_id) REFERENCES songs(song_id)
);

CREATE TABLE IF NOT EXISTS subscriptions (
    subscription_id SERIAL PRIMARY KEY,
    plan_type TEXT,
    start_date DATE,
    end_date DATE,
    payment_method TEXT,
    status TEXT,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

```

Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'

Out[13]:

List of Databases

In [14]: `%sql \l`

Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'

Out[14]:

| Name | Owner | Encoding | Collate | Ctype | Access privileges |
|-------------------------|----------|----------|-------------|-------------|--------------------------------------|
| mlm | vincent | UTF8 | en_US.UTF-8 | en_US.UTF-8 | None |
| music_streaming_service | vincent | UTF8 | en_US.UTF-8 | en_US.UTF-8 | None |
| postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | None |
| rdsadmin | rdsadmin | UTF8 | en_US.UTF-8 | en_US.UTF-8 | rdsadmin=CTc/rdsadmin |
| template0 | rdsadmin | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =C/rdsadmin rdsadmin=CTc/rdsadmin |
| template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =C/postgres postgres=CTc/postgres |

List of tables in oltp music_streaming_service database

In [15]: `%sql \dt`

Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'

| Out[15]: | Schema | Name | Type | Owner |
|----------|--------|-------------------|-------|---------|
| | public | albums | table | vincent |
| | public | artists | table | vincent |
| | public | genres | table | vincent |
| | public | labels | table | vincent |
| | public | listening_history | table | vincent |
| | public | playlist_songs | table | vincent |
| | public | playlists | table | vincent |
| | public | songs | table | vincent |
| | public | subscriptions | table | vincent |
| | public | users | table | vincent |

Truncated to [displaylimit](#) of 10.

```
In [16]: table_list = [
    "genres", "labels", "artists", "albums",
    "songs", "users", "playlists", "playlist_songs",
    "listening_history", "subscriptions"
]
```

Schema of tables

```
In [17]: for table in table_list:  
    print(f"({table})")  
    schema_query = f"columns --table {table}"  
    display(get_ipython().run_line_magic('sqlcmd', schema_query))
```

| name | type | nullable | | default | autoincrement | comment |
|-------------------|-------------|-----------------|--|----------------------|----------------------|----------------|
| song_id | INTEGER | False | nextval('songs_song_id_seq'::regclass) | True | None | |
| title | TEXT | True | | None | False | None |
| duration | INTEGER | True | | None | False | None |
| album_id | INTEGER | True | | None | False | None |
| artist_id | INTEGER | True | | None | False | None |
| genre_id | INTEGER | True | | None | False | None |
| release_date | DATE | True | | None | False | None |
| users | | | | | | |
| name | type | nullable | | default | autoincrement | comment |
| user_id | INTEGER | False | nextval('users_user_id_seq'::regclass) | True | None | |
| name | TEXT | True | | None | False | None |
| email | TEXT | True | | None | False | None |
| join_date | DATE | True | | None | False | None |
| subscription_type | TEXT | True | | None | False | None |
| playlists | | | | | | |
| name | type | nullable | | default | autoincrement | comment |
| playlist_id | INTEGER | False | nextval('playlists_playlist_id_seq'::regclass) | True | None | |
| name | TEXT | True | | None | False | None |
| created_at | TIMESTAMP | True | | None | False | None |
| privacy_setting | TEXT | True | | None | False | None |
| user_id | INTEGER | True | | None | False | None |
| playlist_songs | | | | | | |
| name | type | nullable | default | autoincrement | comment | |
| playlist_id | INTEGER | False | None | False | None | |
| song_id | INTEGER | False | None | False | None | |
| added_at | TIMESTAMP | True | None | False | None | |
| listening_history | | | | | | |
| name | type | nullable | | default | autoincrement | comment |
| history_id | INTEGER | False | nextval('listening_history_history_id_seq'::regclass) | True | None | |
| played_at | TIMESTAMP | True | | None | False | None |
| device | TEXT | True | | None | False | None |
| location | TEXT | True | | None | False | None |
| play_duration | INTEGER | True | | None | False | None |
| user_id | INTEGER | True | | None | False | None |
| song_id | INTEGER | True | | None | False | None |
| subscriptions | | | | | | |
| name | type | nullable | | default | autoincrement | comment |
| subscription_id | INTEGER | False | nextval('subscriptions_subscription_id_seq'::regclass) | True | None | |
| plan_type | TEXT | True | | None | False | None |
| start_date | DATE | True | | None | False | None |
| end_date | DATE | True | | None | False | None |
| payment_method | TEXT | True | | None | False | None |
| status | TEXT | True | | None | False | None |
| user_id | INTEGER | True | | None | False | None |

Faker to create dummy data

```
In [18]: fake = Faker()

music_genres = [
    "Rock", "Jazz", "Hip-Hop", "Electronic", "Classical",
    "Pop", "Blues", "Reggae", "Country", "Metal",
    "Funk", "R&B", "Soul", "Punk", "Disco"
]

sub_type = ["Free", "Basic", "Premium"]

def generate_oltp_genre(n):
```

```

data = []
for i in range(n):
    data.append((
        i,
        music_genres[i-1],
        fake.text()
    ))
return data

def generate.oltp_labels(n):
    data = []
    for i in range(n):
        data.append((
            i,
            fake.company(),
            fake.country(),
            fake.year()
        ))
    return data

def generate.oltp_artists(n):
    data = []
    for i in range(n):
        fk_yr = fake.year()
        data.append((
            i,
            fake.name(),
            fake.country(),
            f'{int(fk_yr)-3}-{int(fk_yr)+3}',
            fake.random_int(0, 15)
        ))
    return data

def generate.oltp_albums(n):
    data = []
    for i in range(n):
        data.append((
            i,
            f'{fake.word()} {fake.word()} {fake.word()}',
            fake.year(),
            fake.random_int(0, 25),
            fake.random_int(0, 25)
        ))
    return data

def generate.oltp_songs(n):
    data = []
    for i in range(n):
        data.append((
            i,
            f'{fake.word()} {fake.word()} {fake.word()}',
            fake.random_int(120, 320),
            fake.random_int(0, 25),
            fake.random_int(0, 25),
            fake.date()
        ))
    return data

def generate.oltp_users(n):
    data = []
    for i in range(n):
        data.append((
            i,
            fake.name(),
            fake.email(),
            fake.date(),
            random.choice(sub_type)
        ))
    return data

def generate.oltp_playlist(n):
    data = []
    for i in range(n):
        data.append((
            i,
            f'{fake.word()} {fake.word()} {fake.word()}',
            fake.date_time(),
            random.choice(["private", "public"]),
            fake.random_int(0, 99)
        ))
    return data

def generate.oltp_playlist_songs(n):
    data = []
    for i in range(n):
        data.append((
            fake.random_int(0, 99),
            fake.random_int(0, 99),
            fake.date_time()
        ))
    return data

def generate.oltp_listening_history(n):
    data = []

```

```

for i in range(n):
    data.append((
        i,
        fake.date_time(),
        random.choice(["mobile", "desktop"]),
        fake.city(),
        fake.random_int(1000, 3000),
        fake.random_int(0, 99),
        fake.random_int(0, 99)
    ))
return data

def generate.oltp_subscriptions(n):
    data = []
    for i in range(n):
        data.append((
            i,
            random.choice(sub_type),
            fake.date_time(),
            fake.date_time(),
            random.choice(["debit card", "mobile wallet", "credit card"]),
            random.choice(["expired", "active", "inactive"]),
            fake.random_int(0, 99)
        ))
    return data

def save_to_postgres(df, table_name, engine):
    df.to_sql(table_name, engine, if_exists='append', index=False)

```

In [19]:

```

df_genre = pd.DataFrame(
    generate.oltp_genre(16),
    columns=["genre_id", "name", "description"]
)
df_label = pd.DataFrame(
    generate.oltp_labels(26),
    columns=["label_id", "name", "country", "founded_year"]
)
df_artist = pd.DataFrame(
    generate.oltp_artists(26),
    columns=["artist_id", "name", "country", "active_years", "genre_id"]
)
df_album = pd.DataFrame(
    generate.oltp_albums(26),
    columns=["album_id", "name", "release_year", "artist_id", "label_id"]
)
df_song = pd.DataFrame(
    generate.oltp_songs(100),
    columns=["song_id", "title", "duration", "album_id", "artist_id", "release_date"]
)
df_user = pd.DataFrame(
    generate.oltp_users(100),
    columns=["user_id", "name", "email", "join_date", "subscription_type"]
)
df_playlist = pd.DataFrame(
    generate.oltp_playlist(150),
    columns=["playlist_id", "name", "created_at", "privacy_setting", "user_id"]
)
df_playlist_songs = pd.DataFrame(
    generate.oltp_playlist_songs(5),
    columns=["playlist_id", "song_id", "added_at"]
)
df_listening_history = pd.DataFrame(
    generate.oltp_listening_history(200),
    columns=["history_id", "played_at", "device", "location", "play_duration", "user_id", "song_id"]
)
df_subscription = pd.DataFrame(
    generate.oltp_subscriptions(200),
    columns=["subscription_id", "plan_type", "start_date", "end_date", "payment_method", "status", "user_id"]
)

save_to_postgres(df_genre, "genres", engine)
save_to_postgres(df_label, "labels", engine)
save_to_postgres(df_artist, "artists", engine)
save_to_postgres(df_album, "albums", engine)
save_to_postgres(df_song, "songs", engine)
save_to_postgres(df_user, "users", engine)
save_to_postgres(df_playlist, "playlists", engine)
save_to_postgres(df_playlist_songs, "playlist_songs", engine)
save_to_postgres(df_listening_history, "listening_history", engine)
save_to_postgres(df_subscription, "subscriptions", engine)

```

First 100 rows of each table

In [20]:

```

for table in table_list:
    print(f"{table}")
    display(pd.read_sql(f"select * from {table}", engine))

```

genres

| genre_id | name | description | |
|----------|------|-------------|---|
| 0 | 0 | Disco | Stage hot anything now feel most attention. Go... |
| 1 | 1 | Rock | Stand design according charge turn. Direction ... |
| 2 | 2 | Jazz | Senior but study assume now. Character religio... |
| 3 | 3 | Hip-Hop | Safe between series human. Walk federal summer... |
| 4 | 4 | Electronic | Scientist almost southern next stand. Contain ... |
| 5 | 5 | Classical | Enough vote ball impact often benefit college ... |
| 6 | 6 | Pop | Dark under itself it I. Project study spend pe... |
| 7 | 7 | Blues | Force American read. Decision small can over a... |
| 8 | 8 | Reggae | Include affect hold time statement. Rate reach... |
| 9 | 9 | Country | Wide well decide step visit summer present. So... |
| 10 | 10 | Metal | Thank couple can despite behavior strategy age... |
| 11 | 11 | Funk | Soon home speak detail. Reflect upon group beh... |
| 12 | 12 | R&B | Memory family student both student act police... |
| 13 | 13 | Soul | Him for camera summer study difficult tree. He... |
| 14 | 14 | Punk | Quite my another project may. Answer any green... |
| 15 | 15 | Disco | Evidence bring so. Form consider certainly ite... |

labels

| label_id | name | country | founded_year |
|----------|---------------------------------|------------------------------|--------------|
| 0 | Pham-Guerrero | Puerto Rico | 2010 |
| 1 | Smith-Arellano | Brazil | 2011 |
| 2 | Brown-Hall | French Polynesia | 1976 |
| 3 | Hardin LLC | Eritrea | 1989 |
| 4 | Murphy, Norman and Greene | Jersey | 2001 |
| 5 | Soto Group | Marshall Islands | 1992 |
| 6 | Sanchez-Ferrell | Mali | 1992 |
| 7 | Barnes-Fisher | Honduras | 1981 |
| 8 | Greer, Holland and Mitchell | Austria | 1992 |
| 9 | Hernandez, Cunningham and Mcgee | Christmas Island | 1998 |
| 10 | Fisher Group | Svalbard & Jan Mayen Islands | 1980 |
| 11 | Stevens PLC | Portugal | 2008 |
| 12 | Ingram Group | Egypt | 1995 |
| 13 | Smith-Bowman | Iraq | 1995 |
| 14 | Reynolds-Terry | Haiti | 1977 |
| 15 | Rodriguez LLC | Guyana | 1971 |
| 16 | Hanson-Rivera | Belize | 2018 |
| 17 | Kaiser-Reyes | Australia | 2000 |
| 18 | Hunter, Kemp and Cook | Saint Kitts and Nevis | 2006 |
| 19 | Olson, Moore and Jackson | French Southern Territories | 1982 |
| 20 | Smith-Gonzalez | Jersey | 1987 |
| 21 | Palmer-Williams | United States Virgin Islands | 2020 |
| 22 | Wells-Gray | Grenada | 1995 |
| 23 | Rhodes Inc | Antigua and Barbuda | 1996 |
| 24 | Lewis-Berry | El Salvador | 1971 |
| 25 | Cruz LLC | Northern Mariana Islands | 2004 |

artists

| | artist_id | name | country | active_years | genre_id |
|----|-----------|-------------------|-----------------------------|--------------|----------|
| 0 | 0 | Brad Acevedo | Korea | 2017-2023 | 8 |
| 1 | 1 | Michael Snyder | Mayotte | 2004-2010 | 2 |
| 2 | 2 | Brian Sweeney | Papua New Guinea | 1996-2002 | 4 |
| 3 | 3 | Jennifer Shea | French Southern Territories | 2004-2010 | 14 |
| 4 | 4 | Kristina Francis | New Caledonia | 2006-2012 | 11 |
| 5 | 5 | Adam Hayes | Pakistan | 1982-1988 | 11 |
| 6 | 6 | Melissa Riley PhD | Grenada | 1972-1978 | 15 |
| 7 | 7 | Derek Hamilton | Malta | 2000-2006 | 1 |
| 8 | 8 | Michael Hernandez | Algeria | 1995-2001 | 6 |
| 9 | 9 | Brett Jackson MD | Bahrain | 1996-2002 | 11 |
| 10 | 10 | Anthony Alvarez | Iraq | 1978-1984 | 3 |
| 11 | 11 | James Howell | Niger | 1979-1985 | 15 |
| 12 | 12 | Ricky Jackson | Congo | 1986-1992 | 3 |
| 13 | 13 | William Duran | Paraguay | 1987-1993 | 9 |
| 14 | 14 | Emily King | Jersey | 1985-1991 | 13 |
| 15 | 15 | Andrea Myers | Rwanda | 2000-2006 | 8 |
| 16 | 16 | Charles Krause | Andorra | 1970-1976 | 10 |
| 17 | 17 | Barbara Gibson | Liberia | 1978-1984 | 14 |
| 18 | 18 | Gary Thomas | Burkina Faso | 1997-2003 | 9 |
| 19 | 19 | Thomas Gentry | Mauritius | 2002-2008 | 11 |
| 20 | 20 | Tanya Leblanc | Spain | 1982-1988 | 15 |
| 21 | 21 | Courtney Montes | Egypt | 1983-1989 | 8 |
| 22 | 22 | Cynthia Miller | Anguilla | 2003-2009 | 0 |
| 23 | 23 | Adrienne Stevens | Nepal | 1998-2004 | 12 |
| 24 | 24 | Megan Sullivan | Malta | 2014-2020 | 7 |
| 25 | 25 | Ashley Jones | El Salvador | 2002-2008 | 10 |

albums

| | album_id | name | release_year | artist_id | label_id |
|----|----------|---------------------------|--------------|-----------|----------|
| 0 | 0 | many many together | 1991 | 22 | 20 |
| 1 | 1 | budget population blue | 2012 | 7 | 9 |
| 2 | 2 | think history check | 1974 | 14 | 4 |
| 3 | 3 | work town property | 2021 | 20 | 21 |
| 4 | 4 | hot west quickly | 1978 | 24 | 19 |
| 5 | 5 | star box special | 1992 | 17 | 15 |
| 6 | 6 | focus both bring | 2005 | 11 | 12 |
| 7 | 7 | poor itself father | 2000 | 19 | 3 |
| 8 | 8 | network site place | 1972 | 14 | 17 |
| 9 | 9 | behavior population close | 1992 | 6 | 17 |
| 10 | 10 | war international order | 1971 | 25 | 19 |
| 11 | 11 | site scene fly | 2022 | 25 | 5 |
| 12 | 12 | image model along | 2013 | 10 | 18 |
| 13 | 13 | guess choose bad | 1982 | 1 | 15 |
| 14 | 14 | case team girl | 2000 | 4 | 1 |
| 15 | 15 | time marriage stop | 1974 | 3 | 7 |
| 16 | 16 | reason store enough | 2012 | 19 | 1 |
| 17 | 17 | car help ten | 1984 | 3 | 5 |
| 18 | 18 | to time open | 2002 | 24 | 4 |
| 19 | 19 | without actually simply | 2004 | 6 | 1 |
| 20 | 20 | need mother just | 1995 | 1 | 0 |
| 21 | 21 | maintain various let | 1973 | 14 | 17 |
| 22 | 22 | daughter democratic exist | 1986 | 4 | 5 |
| 23 | 23 | decade indicate save | 2021 | 15 | 21 |
| 24 | 24 | computer chair fill | 1983 | 18 | 10 |
| 25 | 25 | kid report score | 2006 | 24 | 1 |

songs

| | song_id | title | duration | album_id | artist_id | genre_id | release_date |
|-----|---------|----------------------------|----------|----------|-----------|----------|--------------|
| 0 | 0 | town nature spend | 129 | 1 | 22 | None | 2022-01-04 |
| 1 | 1 | western model class | 171 | 1 | 15 | None | 2012-03-04 |
| 2 | 2 | other work production | 317 | 22 | 19 | None | 1972-04-06 |
| 3 | 3 | partner wife enjoy | 224 | 8 | 11 | None | 2021-01-02 |
| 4 | 4 | animal establish treatment | 155 | 18 | 10 | None | 2020-05-19 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 95 | 95 | building worker same | 246 | 19 | 0 | None | 1978-12-18 |
| 96 | 96 | natural often response | 233 | 2 | 13 | None | 1994-05-12 |
| 97 | 97 | to establish fish | 263 | 17 | 1 | None | 1998-09-07 |
| 98 | 98 | notice share pretty | 295 | 24 | 21 | None | 1986-02-08 |
| 99 | 99 | magazine down cold | 317 | 19 | 15 | None | 1977-03-18 |

100 rows × 7 columns

users

| | user_id | name | | email | join_date | subscription_type |
|-----|---------|-----------------|--|----------------------------------|------------|-------------------|
| 0 | 0 | Taylor Hayes | | robinsonjimmy@example.org | 1997-03-13 | Free |
| 1 | 1 | Jacob Harris | | stephanie16@example.com | 2011-07-19 | Basic |
| 2 | 2 | Karen Pitts | | jamesleon@example.org | 2003-11-18 | Basic |
| 3 | 3 | Anthony Coleman | | robert00@example.com | 2001-05-16 | Premium |
| 4 | 4 | Robert Smith | | jacquelinestrickland@example.net | 1992-11-01 | Free |
| ... | ... | ... | | ... | ... | ... |
| 95 | 95 | Randy Weiss | | andersonemily@example.com | 1989-03-25 | Premium |
| 96 | 96 | Jocelyn Soto | | dawn24@example.com | 1985-01-15 | Free |
| 97 | 97 | Ashley Evans | | brian24@example.org | 1995-10-07 | Free |
| 98 | 98 | Susan Yu | | bfowler@example.com | 2015-07-22 | Basic |
| 99 | 99 | Colin Woodward | | victoriabanks@example.com | 1999-01-05 | Basic |

100 rows × 5 columns

playlists

| | playlist_id | name | | created_at | privacy_setting | user_id |
|-----|-------------|------------------------|--|----------------------------|-----------------|---------|
| 0 | 0 | different recent tend | | 2012-09-27 00:39:45.186647 | public | 35 |
| 1 | 1 | wind accept whatever | | 1984-08-23 19:36:10.904152 | private | 12 |
| 2 | 2 | value concern activity | | 2023-12-01 09:43:41.120748 | private | 30 |
| 3 | 3 | police care generation | | 2009-12-02 18:48:30.521773 | private | 98 |
| 4 | 4 | at woman include | | 1976-05-02 17:53:42.015936 | public | 31 |
| ... | ... | ... | | ... | ... | ... |
| 145 | 145 | certainly exist appear | | 2009-10-28 04:29:52.841553 | public | 1 |
| 146 | 146 | fire gas attention | | 1971-01-13 13:23:39.979594 | public | 22 |
| 147 | 147 | about training reduce | | 2013-03-13 04:11:47.639313 | private | 68 |
| 148 | 148 | develop source ahead | | 1975-12-02 04:47:03.517481 | private | 8 |
| 149 | 149 | radio upon short | | 1986-08-08 02:24:36.003504 | private | 92 |

150 rows × 5 columns

playlist_songs

| | playlist_id | song_id | added_at |
|---|-------------|---------|----------------------------|
| 0 | 0 | 53 | 2015-06-13 20:24:09.341197 |
| 1 | 21 | 30 | 1991-02-25 12:00:39.253939 |
| 2 | 60 | 41 | 2012-04-07 20:18:22.715635 |
| 3 | 59 | 26 | 2012-10-13 04:14:54.246029 |
| 4 | 12 | 28 | 2000-06-21 10:20:19.747239 |

listening_history

| | history_id | played_at | device | location | play_duration | user_id | song_id |
|-----|------------|----------------------------|---------|-------------------|---------------|---------|---------|
| 0 | 0 | 1983-05-16 14:34:49.329634 | desktop | South Debbiehaven | 2542 | 53 | 21 |
| 1 | 1 | 2020-04-09 06:07:10.953801 | mobile | Margaretland | 2883 | 5 | 5 |
| 2 | 2 | 1984-12-08 02:41:49.131716 | mobile | Yuland | 2823 | 10 | 34 |
| 3 | 3 | 2006-10-11 11:01:03.043540 | desktop | Martinshire | 2493 | 37 | 20 |
| 4 | 4 | 1986-03-06 12:42:10.827136 | mobile | Port Moniqueburgh | 2612 | 73 | 58 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 195 | 195 | 1984-07-03 01:00:20.568242 | desktop | Combsshire | 2796 | 54 | 12 |
| 196 | 196 | 2009-12-22 19:40:19.925599 | mobile | Farrellchester | 1062 | 4 | 23 |
| 197 | 197 | 1993-12-09 10:59:51.210013 | mobile | Codyshire | 2341 | 73 | 81 |
| 198 | 198 | 1981-02-12 04:32:34.317493 | mobile | Stevenston | 1878 | 49 | 57 |
| 199 | 199 | 2003-11-20 10:08:56.820486 | desktop | North Jeanette | 1200 | 98 | 31 |

200 rows × 7 columns

subscriptions

| | subscription_id | plan_type | start_date | end_date | payment_method | status | user_id |
|-----|-----------------|-----------|------------|------------|----------------|----------|---------|
| 0 | 0 | Basic | 1998-06-30 | 2024-01-12 | credit card | active | 87 |
| 1 | 1 | Premium | 2011-08-09 | 2007-02-17 | mobile wallet | inactive | 33 |
| 2 | 2 | Free | 2007-10-20 | 1988-07-09 | mobile wallet | active | 43 |
| 3 | 3 | Basic | 1991-12-02 | 1978-05-06 | debit card | active | 67 |
| 4 | 4 | Free | 1979-06-22 | 2011-05-01 | credit card | active | 53 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 195 | 195 | Free | 1997-05-13 | 1998-04-15 | credit card | inactive | 28 |
| 196 | 196 | Basic | 2006-07-13 | 2001-06-27 | credit card | active | 41 |
| 197 | 197 | Premium | 1996-04-04 | 1986-12-16 | mobile wallet | active | 33 |
| 198 | 198 | Free | 1993-05-06 | 1979-04-15 | mobile wallet | inactive | 94 |
| 199 | 199 | Premium | 2017-06-14 | 1982-02-07 | debit card | expired | 96 |

200 rows × 7 columns

Number of rows of each table

```
In [21]: for table in table_list:
    print(F'{table}')
    count_query = f"select COUNT(*) from {table}"
    display(get_ipython().run_line_magic('sql', count_query))

genres
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
16

labels
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
26

artists
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
26

albums
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
26

songs
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
100

users
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
100

playlists
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
150

playlist_songs
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
_____
5
```

```

listening_history
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
200

subscriptions
Running query in 'postgresql://vincent***@sampledb.c34wykcu8liz.us-east-1.rds.amazonaws.com/music_streaming_service'
1 rows affected.

count
200

```

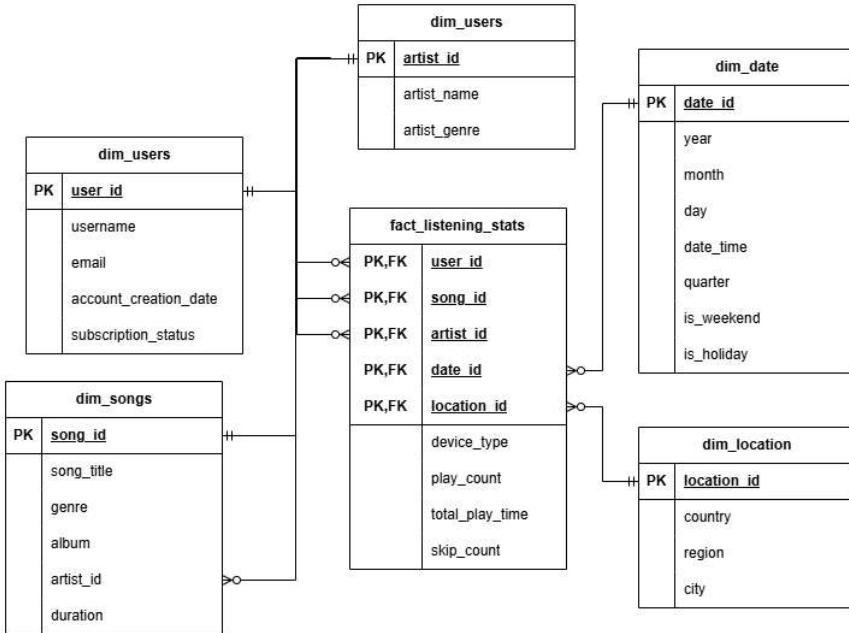
Rollback/Start from scratch queries

```
In [22]: # %%sql
# DROP TABLE IF EXISTS genres, labels, artists, albums,
#       songs, users, playlists, playlist_songs,
#       listening_history, subscriptions
```

OLAP Implementation of Music Streaming Service

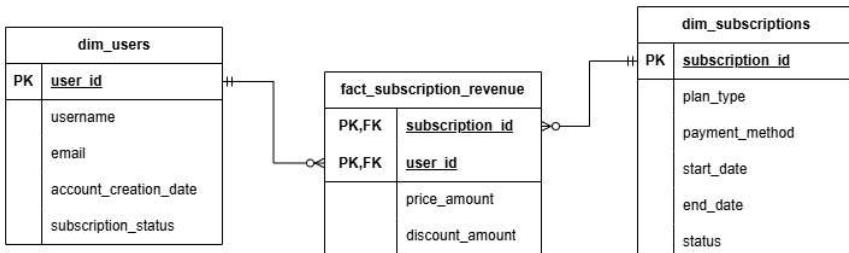
Business Process 1: Analyze song play patterns, user engagement and device usage

- Grain: Each record represents a song play event.



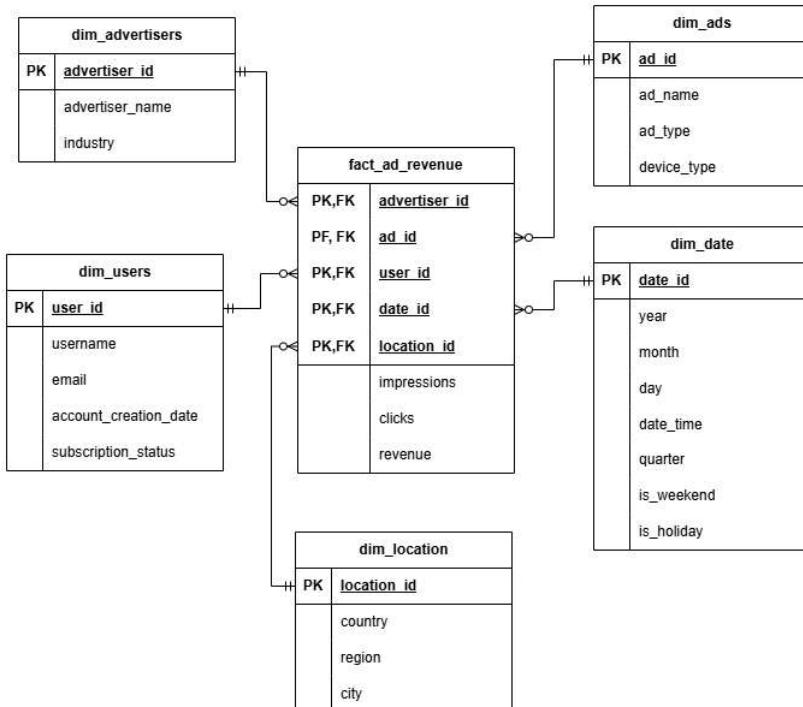
Business Process 2: Analyze revenue trends, customer retention, and churn rates

- Grain: Each record represents a subscription transaction.



Business Process 3: Analyze ad revenue trends, advertiser performance, and impressions

- Grain: Each record represents an ad impression event



Redshift Console Details

Amazon Redshift > Clusters > samplecluster

samplecluster

Actions ▾ Edit Add partner integration Query data ▾

| General information Info | |
|--|--|
| Cluster identifier samplecluster | Status Modifying |
| Custom domain name - | Date created January 27, 2025, 11:32 (UTC+08:00) |
| Cluster namespace ARN arn:aws:redshift:us-east-1:27707127732:namespace:0181b750-d9ed-4002-8527-29702b7d1524 | Multi-AZ No |
| Namespace register status Deregistered | Cluster configuration Production |
| | Patch version - |
| | Storage used 0.31 of 160 GB used (0.20%) |
| Endpoint samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/dev | JDBC URL jdbc:redshift://samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/dev |
| ODBC URL Driver={Amazon Redshift (x64)};Server=samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com;Database=dev | |

```
In [23]: db_olap_password = config("olap_pass")
olap_connection = config("olap_connection")
```

```
In [24]: connection_string = f"postgresql://vincent:{db_olap_password}@{olap_connection}:5439/music_streaming_service"
```

```
In [25]: engine = redshift_connector.connect(
    host=olap_connection,
    port=5439,
    database='music_streaming_service',
    user='vincent',
    password=db_olap_password
)
```

```
In [26]: get_ipython().run_line_magic('sql', connection_string)
```

Connecting and switching to connection 'postgresql://vincent:**@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'

Create Table Scripts

```
In [27]: %%sql
CREATE TABLE IF NOT EXISTS dim_users (
    user_id INT,
    username TEXT,
```

```

    email TEXT,
    account_creation_date DATE,
    subscription_status TEXT
);

CREATE TABLE IF NOT EXISTS dim_artists (
    artist_id INT,
    artist_name TEXT,
    artist_genre TEXT
);

CREATE TABLE IF NOT EXISTS dim_songs (
    song_id INT,
    song_title TEXT,
    genre TEXT,
    album TEXT,
    artist_id INT,
    duration INT
);

CREATE TABLE IF NOT EXISTS dim_date (
    date_id INT,
    date DATE,
    year INT,
    month INT,
    day INT,
    quarter INT,
    weekday INT
);

CREATE TABLE IF NOT EXISTS dim_locations (
    location_id INT,
    country TEXT,
    region TEXT,
    city TEXT
);

CREATE TABLE IF NOT EXISTS dim_subscriptions (
    subscription_id INT,
    plan_type TEXT,
    payment_method TEXT,
    start_date DATE,
    end_date DATE,
    status TEXT
);

CREATE TABLE IF NOT EXISTS dim_ads (
    ad_id INT,
    ad_name TEXT,
    ad_type TEXT,
    device_type TEXT
);

CREATE TABLE IF NOT EXISTS dim_advertisers (
    advertiser_id INT,
    advertiser_name TEXT,
    industry TEXT
);

-- Fact Tables
CREATE TABLE IF NOT EXISTS fact_listening_stats (
    user_id INT,
    song_id INT,
    artist_id INT,
    date_id INT,
    location_id INT,
    play_count INT,
    total_play_time NUMERIC(10, 4),
    skip_count INT
);

CREATE TABLE IF NOT EXISTS fact_subscription_revenue (
    subscription_id INT,
    user_id INT,
    price_amount NUMERIC(10, 4),
    discount_amount NUMERIC(10, 4)
);

CREATE TABLE IF NOT EXISTS fact_ad_revenue (
    ad_id INT,
    advertiser_id INT,
    user_id INT,
    date_id INT,
    location_id INT,
    impressions INT,
    clicks INT,
    revenue NUMERIC(10, 4)
);

```

Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'

Out[27]:

List of Databases

In [28]:

```
%%sql
SELECT * FROM pg_database;
```

Running query in 'postgresql://vincent***@samplecluster.cz68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'

8 rows affected.

Out[28]:

| datname | datdba | encoding | datistemplate | dataallowconn | datalastsysoid | datvacuumxid | datfrozenxid | attablespace | datcollate |
|-------------------------|--------|----------|---------------|---------------|----------------|--------------|--------------|--------------|---|
| awsdatacatalog | 1 | 6 | False | False | 102558 | 967 | 967 | 0 | en_US.UTF-8 |
| template0 | 1 | 6 | True | False | 102558 | 967 | 967 | 1663 | |
| dev | 1 | 6 | False | True | 102558 | 0 | 0 | 1663 | 'enable_query_profiler_instrumentation' |
| padb_harvest | 1 | 6 | False | True | 102558 | 0 | 0 | 1663 | |
| sysinternal | 1 | 6 | False | True | 102558 | 967 | 967 | 1663 | |
| template1 | 1 | 6 | True | True | 102558 | 967 | 967 | 1663 | |
| mlm | 101 | 6 | False | True | 102558 | 967 | 967 | 1663 | |
| music_streaming_service | 101 | 6 | False | True | 102558 | 967 | 967 | 1663 | |

List of tables in olap music_streaming_service database

In [29]:

```
%%sql \dt
```

Running query in 'postgresql://vincent***@samplecluster.cz68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'

Out[29]:

| schema | name | type | owner |
|--------|----------------------|-------|---------|
| public | dim_ads | table | vincent |
| public | dim_advertisers | table | vincent |
| public | dim_artists | table | vincent |
| public | dim_date | table | vincent |
| public | dim_locations | table | vincent |
| public | dim_songs | table | vincent |
| public | dim_subscriptions | table | vincent |
| public | dim_users | table | vincent |
| public | fact_ad_revenue | table | vincent |
| public | fact_listening_stats | table | vincent |

Truncated to displaylimit of 10.

In [30]:

```
table_list = [
    "dim_users", "dim_songs", "dim_artists", "dim_date",
    "dim_locations", "dim_subscriptions", "dim_ads", "dim_advertisers",
    "fact_listening_stats", "fact_subscription_revenue", "fact_ad_revenue"
]
```

Schema of tables

In [31]:

```
for table in table_list:
    print(f"{table}")
    schema_query = f"columns --table {table}"
    display(get_ipython().run_line_magic('sqlcmd', schema_query))
```

dim_users

| name | type | nullable | default | autoincrement | comment |
|-----------------------|--------------|----------|---------|---------------|---------|
| user_id | INTEGER | True | None | False | None |
| username | VARCHAR(256) | True | None | False | None |
| email | VARCHAR(256) | True | None | False | None |
| account_creation_date | DATE | True | None | False | None |
| subscription_status | VARCHAR(256) | True | None | False | None |

dim_songs

| name | type | nullable | default | autoincrement | comment |
|----------------------------------|----------------|----------|---------|---------------|---------|
| user_id | INTEGER | True | None | False | None |
| song_id | INTEGER | True | None | False | None |
| artist_id | INTEGER | True | None | False | None |
| date_id | INTEGER | True | None | False | None |
| location_id | INTEGER | True | None | False | None |
| play_count | INTEGER | True | None | False | None |
| total_play_time | NUMERIC(10, 4) | True | None | False | None |
| skip_count | INTEGER | True | None | False | None |
| fact_subscription_revenue | | | | | |
| name | type | nullable | default | autoincrement | comment |
| subscription_id | INTEGER | True | None | False | None |
| user_id | INTEGER | True | None | False | None |
| price_amount | NUMERIC(10, 4) | True | None | False | None |
| discount_amount | NUMERIC(10, 4) | True | None | False | None |
| fact_ad_revenue | | | | | |
| name | type | nullable | default | autoincrement | comment |
| ad_id | INTEGER | True | None | False | None |
| advertiser_id | INTEGER | True | None | False | None |
| user_id | INTEGER | True | None | False | None |
| date_id | INTEGER | True | None | False | None |
| location_id | INTEGER | True | None | False | None |
| impressions | INTEGER | True | None | False | None |
| clicks | INTEGER | True | None | False | None |
| revenue | NUMERIC(10, 4) | True | None | False | None |

Faker to create dummy data

```
In [32]: industries = [
    "Technology",
    "Finance",
    "Healthcare",
    "Retail",
    "Education",
    "Manufacturing",
    "Real Estate",
    "Transportation",
    "Entertainment",
    "Energy"
]

def generate_olap_dim_users(n):
    data = []
    for i in range(n):
        data.append((
            i,
            fake.user_name(),
            fake.email(),
            fake.date(),
            random.choice(["active", "inactive", "expired"])
        ))
    return data

def generate_olap_dim_songs(n):
    data = []
    for i in range(n):
        data.append((
            i,
            f'{fake.word()} {fake.word()} {fake.word()}',
            random.choice(music_genres),
            f'{fake.word()} {fake.word()} {fake.word()}',
            fake.random_int(0, 50),
            fake.random_int(120, 320)
        ))
    return data

def generate_olap_dim_artists(n):
    data = []
    for i in range(n):
        data.append((
            i,
            fake.name(),
            random.choice(music_genres)
        ))
```

```

        ))
    return data

def generate_olap_dim_date():
    data = []
    for i in range(1, 30):
        dt = fake.date_object()
        data.append((
            dt.strftime("%d%b%Y"),
            dt.strftime("%Y-%m-%d"),
            dt.year,
            dt.month,
            dt.day,
            (dt.month - 1) // 3 + 1,
            dt.weekday()
        ))
    return data

def generate_olap_dim_location(n):
    data = []
    for i in range(n):
        data.append((
            i,
            fake.country(),
            fake.city(),
            fake.city()
        ))
    return data

def generate_olap_dim_subscription(n):
    data = []
    for i in range(n):
        data.append((
            i,
            random.choice(sub_type),
            random.choice(["debit card", "mobile wallet", "credit card"]),
            fake.date(),
            fake.date(),
            random.choice(["active", "inactive", "expired"])
        ))
    return data

def generate_olap_dim_ads(n):
    data = []
    for i in range(n):
        data.append((
            i,
            f'{fake.word()} {fake.word()} {fake.word()}',
            random.choice(["banner", "audio", "video"]),
            random.choice(["mobile", "desktop"])
        ))
    return data

def generate_olap_dim_advertisers(n):
    data = []
    for i in range(n):
        data.append((
            i,
            fake.name(),
            random.choice(industries)
        ))
    return data

def generate_olap_fact_listening_stats(n):
    data = []
    for i in range(n):
        dt = fake.date_object()
        data.append((
            fake.random_int(0, 100),
            fake.random_int(0, 300),
            fake.random_int(0, 50),
            dt.strftime("%d%b%Y"),
            fake.random_int(0, 25),
            fake.random_int(0, 100),
            fake.random_int(1000, 3000),
            fake.random_int(0, 100)
        ))
    return data

def generate_olap_fact_subscription_revenue(n):
    data = []
    for i in range(n):
        data.append((
            fake.random_int(0, 25),
            fake.random_int(0, 100),
            fake.pydecimal(left_digits=3, right_digits=4, positive=True),
            fake.pydecimal(left_digits=3, right_digits=4, positive=True)
        ))
    return data

def generate_olap_fact_ad_revenue(n):
    data = []
    for i in range(n):
        dt = fake.date_object()

```

```

    data.append((
        fake.random_int(0, 25),
        fake.random_int(0, 25),
        fake.random_int(0, 100),
        dt.strftime("%d%b%Y"),
        fake.random_int(0, 25),
        fake.random_int(1000, 3000),
        fake.random_int(1000, 3000),
        fake.pydecimal(left_digits=3, right_digits=4, positive=True)
    ))
    return data

def save_to_redshift(df, table_name, engine):
    with engine.cursor() as cursor:
        cursor.write_dataframe(df, table_name)
    engine.commit()

In [33]: df_dim_users = pd.DataFrame(
    generate_olap_dim_users(100),
    columns=["user_id", "username", "email", "account_creation_date", "subscription_status"]
)
df_dim_songs = pd.DataFrame(
    generate_olap_dim_songs(300),
    columns=["song_id", "song_title", "genre", "album", "artist", "duration"]
)
df_dim_artists = pd.DataFrame(
    generate_olap_dim_artists(50),
    columns=["artist_id", "artist_name", "artist_genre"]
)
df_dim_date = pd.DataFrame(
    generate_olap_dim_date(),
    columns=["date_id", "date", "year", "month", "day", "quarter", "weekday"]
)
df_dim_locations = pd.DataFrame(
    generate_olap_dim_location(25),
    columns=["location_id", "country", "region", "city"]
)
df_dim_subscriptions = pd.DataFrame(
    generate_olap_dim_subscription(25),
    columns=["subscription_id", "plan_type", "payment_method", "start_date", "end_date", "status"]
)
df_dim_ads = pd.DataFrame(
    generate_olap_dim_ads(25),
    columns=["ad_id", "ad_name", "ad_type", "device_type"]
)
df_dim_advertisers = pd.DataFrame(
    generate_olap_dim_advertisers(25),
    columns=["advertiser_id", "advertiser_name", "industry"]
)
df_fact_listening_stats = pd.DataFrame(
    generate_olap_fact_listening_stats(50),
    columns=["user_id", "song_id", "artist_id", "date_id", "location_id", "play_count", "total_play_time", "skip_count"]
)
df_fact_subscription_revenue = pd.DataFrame(
    generate_olap_fact_subscription_revenue(50),
    columns=["subscription_id", "user_id", "price_amount", "discount_amount"]
)
df_fact_ad_revenue = pd.DataFrame(
    generate_olap_fact_ad_revenue(50),
    columns=["ad_id", "advertiser_id", "user_id", "date_id", "location_id", "impressions", "clicks", "revenue"]
)

save_to_redshift(df_dim_users, "dim_users", engine)
save_to_redshift(df_dim_songs, "dim_songs", engine)
save_to_redshift(df_dim_artists, "dim_artists", engine)
save_to_redshift(df_dim_date, "dim_date", engine)
save_to_redshift(df_dim_locations, "dim_locations", engine)
save_to_redshift(df_dim_subscriptions, "dim_subscriptions", engine)
save_to_redshift(df_dim_ads, "dim_ads", engine)
save_to_redshift(df_dim_advertisers, "dim_advertisers", engine)
save_to_redshift(df_fact_listening_stats, "fact_listening_stats", engine)
save_to_redshift(df_fact_subscription_revenue, "fact_subscription_revenue", engine)
save_to_redshift(df_fact_ad_revenue, "fact_ad_revenue", engine)

```

First 100 rows of each table

```

In [34]: for table in table_list:
    print(f"{table}")
    display(pd.read_sql(f"select * from {table}", engine))

dim_users

```

| | user_id | username | email | account_creation_date | subscription_status |
|-----|---------|----------------|----------------------------|-----------------------|---------------------|
| 0 | 0 | samantha98 | steven45@example.org | 1994-03-04 | active |
| 1 | 1 | krogers | zpatrick@example.com | 1997-11-28 | expired |
| 2 | 2 | nancyorozco | virginia09@example.org | 2021-08-11 | active |
| 3 | 3 | ggreen | rthompson@example.com | 1990-05-28 | expired |
| 4 | 4 | petersonholly | johnsondebbie@example.net | 1987-11-07 | inactive |
| ... | ... | ... | ... | ... | ... |
| 95 | 95 | kristengriffin | turnertammy@example.org | 1973-05-04 | inactive |
| 96 | 96 | suzannehuffman | carrieferguson@example.net | 2000-04-09 | active |
| 97 | 97 | smallkenneth | pgoodman@example.org | 1983-03-24 | active |
| 98 | 98 | justinatkinson | sabrinanewton@example.net | 1987-03-14 | active |
| 99 | 99 | kennethcrane | jacksonmelissa@example.org | 2015-08-23 | inactive |

100 rows × 5 columns

| | dim_songs | | | | | |
|-----|-----------|------------------------|------------|-----------------------|-----------|----------|
| | song_id | song_title | genre | album | artist_id | duration |
| 0 | 0 | address reach eat | Disco | program choose art | 12 | 232 |
| 1 | 1 | unit their spring | Blues | war walk relationship | 0 | 219 |
| 2 | 2 | the director financial | Reggae | enter federal various | 6 | 251 |
| 3 | 3 | relationship gas south | Rock | once develop PM | 8 | 282 |
| 4 | 4 | bed throw space | Country | pressure some parent | 27 | 191 |
| ... | ... | ... | ... | ... | ... | ... |
| 295 | 295 | market security but | Jazz | about could family | 20 | 242 |
| 296 | 296 | song likely firm | Country | then senior inside | 1 | 281 |
| 297 | 297 | detail public memory | Electronic | recently single fight | 0 | 216 |
| 298 | 298 | night thank exactly | Jazz | project human worry | 17 | 190 |
| 299 | 299 | save toward line | Rock | reduce trial seem | 33 | 144 |

300 rows × 6 columns

dim_artists

| | artist_id | artist_name | artist_genre |
|----|-----------|--------------------|--------------|
| 0 | 0 | Jamie Frye | Jazz |
| 1 | 1 | Chelsea Mcpherson | Pop |
| 2 | 2 | Lisa Gomez | Hip-Hop |
| 3 | 3 | Anne Jenkins | Rock |
| 4 | 4 | Monique Allen | R&B |
| 5 | 5 | Nicholas Moore | Electronic |
| 6 | 6 | Regina Williams | Blues |
| 7 | 7 | Daniel Kirk | Reggae |
| 8 | 8 | Rodney Sheppard | Funk |
| 9 | 9 | Erika Kim | Soul |
| 10 | 10 | April Norris | Pop |
| 11 | 11 | Kristin Booth | Electronic |
| 12 | 12 | Jesus White | Soul |
| 13 | 13 | Lisa Woodward | Reggae |
| 14 | 14 | Gina Ortiz | Hip-Hop |
| 15 | 15 | Jacob Woods | Punk |
| 16 | 16 | Suzanne Woods | Funk |
| 17 | 17 | Jessica Harris | Electronic |
| 18 | 18 | Michael Reilly | Pop |
| 19 | 19 | Jordan Garcia | R&B |
| 20 | 20 | Timothy Cooper | Rock |
| 21 | 21 | Thomas Brown | Funk |
| 22 | 22 | Lauren Erickson | Punk |
| 23 | 23 | Alexander Nelson | Hip-Hop |
| 24 | 24 | Jennifer Peterson | Electronic |
| 25 | 25 | Megan Ochoa DVM | Pop |
| 26 | 26 | Mrs. Julie Smith | Pop |
| 27 | 27 | David Williams | Blues |
| 28 | 28 | Christopher Clark | Electronic |
| 29 | 29 | Jennifer Fuller | Rock |
| 30 | 30 | Alicia Wood | Country |
| 31 | 31 | Christopher Graham | Soul |
| 32 | 32 | Kenneth Barnett | Country |
| 33 | 33 | Charlotte Jenkins | Blues |
| 34 | 34 | Anthony Lewis | Rock |
| 35 | 35 | Emma Baxter | Disco |
| 36 | 36 | Danny May | Country |
| 37 | 37 | Melissa Terrell | Blues |
| 38 | 38 | Linda Wyatt | Hip-Hop |
| 39 | 39 | Rebecca Snow | Rock |
| 40 | 40 | Nichole Smith | Pop |
| 41 | 41 | Katherine Morales | Reggae |
| 42 | 42 | Kristin Burns | Classical |
| 43 | 43 | Edwin Pierce | Pop |
| 44 | 44 | Elizabeth Miller | R&B |
| 45 | 45 | Crystal Haney | Funk |
| 46 | 46 | Amy Anderson | Disco |
| 47 | 47 | Casey Ho | Country |
| 48 | 48 | Robert Cox | Hip-Hop |
| 49 | 49 | April Williams | Pop |

dim_date

| | date_id | date | year | month | day | quarter | weekday |
|----|----------|------------|------|-------|-----|---------|---------|
| 0 | 1011990 | 1990-01-01 | 1990 | 1 | 1 | 1 | 0 |
| 1 | 6091991 | 1991-09-06 | 1991 | 9 | 6 | 3 | 4 |
| 2 | 28022018 | 2018-02-28 | 2018 | 2 | 28 | 1 | 2 |
| 3 | 16031973 | 1973-03-16 | 1973 | 3 | 16 | 1 | 4 |
| 4 | 21041990 | 1990-04-21 | 1990 | 4 | 21 | 2 | 5 |
| 5 | 11022011 | 2011-02-11 | 2011 | 2 | 11 | 1 | 4 |
| 6 | 6021982 | 1982-02-06 | 1982 | 2 | 6 | 1 | 5 |
| 7 | 20111988 | 1988-11-20 | 1988 | 11 | 20 | 4 | 6 |
| 8 | 21062005 | 2005-06-21 | 2005 | 6 | 21 | 2 | 1 |
| 9 | 26082003 | 2003-08-26 | 2003 | 8 | 26 | 3 | 1 |
| 10 | 3111992 | 1992-11-03 | 1992 | 11 | 3 | 4 | 1 |
| 11 | 23051989 | 1989-05-23 | 1989 | 5 | 23 | 2 | 1 |
| 12 | 21011998 | 1998-01-21 | 1998 | 1 | 21 | 1 | 2 |
| 13 | 1082012 | 2012-08-01 | 2012 | 8 | 1 | 3 | 2 |
| 14 | 5031995 | 1995-03-05 | 1995 | 3 | 5 | 1 | 6 |
| 15 | 10091991 | 1991-09-10 | 1991 | 9 | 10 | 3 | 1 |
| 16 | 8082004 | 2004-08-08 | 2004 | 8 | 8 | 3 | 6 |
| 17 | 28032019 | 2019-03-28 | 2019 | 3 | 28 | 1 | 3 |
| 18 | 21022017 | 2017-02-21 | 2017 | 2 | 21 | 1 | 1 |
| 19 | 11011984 | 1984-01-11 | 1984 | 1 | 11 | 1 | 2 |
| 20 | 4031985 | 1985-03-04 | 1985 | 3 | 4 | 1 | 0 |
| 21 | 9051999 | 1999-05-09 | 1999 | 5 | 9 | 2 | 6 |
| 22 | 9101987 | 1987-10-09 | 1987 | 10 | 9 | 4 | 4 |
| 23 | 21031978 | 1978-03-21 | 1978 | 3 | 21 | 1 | 1 |
| 24 | 18052003 | 2003-05-18 | 2003 | 5 | 18 | 2 | 6 |
| 25 | 14061991 | 1991-06-14 | 1991 | 6 | 14 | 2 | 4 |
| 26 | 21041985 | 1985-04-21 | 1985 | 4 | 21 | 2 | 6 |
| 27 | 26061982 | 1982-06-26 | 1982 | 6 | 26 | 2 | 5 |
| 28 | 20101973 | 1973-10-20 | 1973 | 10 | 20 | 4 | 5 |

dim_locations

| location_id | | country | region | city |
|--------------------|----|-------------------------------|-------------------|---------------------|
| 0 | 0 | Tajikistan | Williamsonborough | Port Joeltown |
| 1 | 1 | Cocos (Keeling) Islands | North Megan | Port Joelhaven |
| 2 | 2 | Guernsey | Nicolemouth | West John |
| 3 | 3 | Turkmenistan | East Michaelmouth | New Cassiemouth |
| 4 | 4 | Ghana | Jessicafurt | New William |
| 5 | 5 | French Southern Territories | Port Geraldview | Mayerstad |
| 6 | 6 | Niger | Michelleburgh | Kathychester |
| 7 | 7 | Saint Pierre and Miquelon | East Timothy | Kellyfurt |
| 8 | 8 | Panama | Jeffreymouth | South Nicholebury |
| 9 | 9 | Taiwan | Lynnborough | Mccarthyburgh |
| 10 | 10 | Taiwan | Mikaylastad | North Michaelton |
| 11 | 11 | Turkey | Keithshire | East Nicholas |
| 12 | 12 | Saint Barthelemy | East Dorothyhaven | Hayesside |
| 13 | 13 | Iceland | Lake Teresa | Port Susan |
| 14 | 14 | Senegal | Normanton | North Baileyborough |
| 15 | 15 | Argentina | East Josephburgh | Aliciaborough |
| 16 | 16 | Holy See (Vatican City State) | Port Angela | Brentport |
| 17 | 17 | Australia | North Joshuafurt | Littleview |
| 18 | 18 | Saint Martin | South Sarahshire | New Debbie |
| 19 | 19 | Korea | Danielland | Taylorport |
| 20 | 20 | French Polynesia | Christopherland | New James |
| 21 | 21 | Tuvalu | Rogerview | Turnerside |
| 22 | 22 | Angola | Phillipport | Michaelville |
| 23 | 23 | Oman | North Annburgh | Christianstad |
| 24 | 24 | Vanuatu | South Johnville | North Lisaview |

dim_subscriptions

| | subscription_id | plan_type | payment_method | start_date | end_date | status |
|----|-----------------|-----------|----------------|------------|------------|----------|
| 0 | 0 | Premium | credit card | 2017-01-21 | 2007-01-07 | expired |
| 1 | 1 | Free | mobile wallet | 1999-01-25 | 1995-11-22 | active |
| 2 | 2 | Premium | debit card | 1980-07-17 | 1990-06-13 | inactive |
| 3 | 3 | Free | credit card | 1978-07-09 | 1986-03-27 | expired |
| 4 | 4 | Premium | mobile wallet | 1999-09-06 | 1994-11-11 | inactive |
| 5 | 5 | Basic | debit card | 2023-09-21 | 2005-10-17 | expired |
| 6 | 6 | Premium | credit card | 1999-04-30 | 1995-08-11 | active |
| 7 | 7 | Free | mobile wallet | 2000-04-12 | 2010-07-23 | active |
| 8 | 8 | Basic | mobile wallet | 1976-03-26 | 1996-04-01 | inactive |
| 9 | 9 | Premium | debit card | 2021-02-04 | 2023-03-18 | expired |
| 10 | 10 | Basic | mobile wallet | 1981-02-26 | 1999-07-05 | active |
| 11 | 11 | Free | debit card | 2016-09-07 | 1974-10-13 | active |
| 12 | 12 | Free | debit card | 1972-02-25 | 1991-08-19 | inactive |
| 13 | 13 | Free | mobile wallet | 2012-01-08 | 1987-01-01 | expired |
| 14 | 14 | Premium | mobile wallet | 2008-08-08 | 1994-07-18 | expired |
| 15 | 15 | Premium | credit card | 2004-03-14 | 2015-01-12 | active |
| 16 | 16 | Premium | debit card | 1983-08-14 | 1992-02-14 | inactive |
| 17 | 17 | Basic | debit card | 1971-01-08 | 1986-05-26 | active |
| 18 | 18 | Basic | debit card | 1983-09-01 | 1986-04-14 | inactive |
| 19 | 19 | Basic | credit card | 2011-08-03 | 2009-12-04 | inactive |
| 20 | 20 | Basic | credit card | 2021-07-21 | 1983-08-13 | active |
| 21 | 21 | Basic | debit card | 2020-01-12 | 1972-03-27 | inactive |
| 22 | 22 | Free | mobile wallet | 1987-10-25 | 1974-04-17 | expired |
| 23 | 23 | Free | mobile wallet | 1997-08-23 | 1971-08-17 | active |
| 24 | 24 | Premium | mobile wallet | 2000-05-07 | 1986-07-09 | expired |

dim_ads

| ad_id | | ad_name | ad_type | device_type |
|--------------|----|----------------------------|----------------|--------------------|
| 0 | 0 | special official herself | video | desktop |
| 1 | 1 | site source difficult | video | desktop |
| 2 | 2 | product Republican few | banner | desktop |
| 3 | 3 | sound social finally | audio | mobile |
| 4 | 4 | throughout draw stay | audio | mobile |
| 5 | 5 | paper level talk | audio | desktop |
| 6 | 6 | student direction top | banner | desktop |
| 7 | 7 | carry tend issue | banner | desktop |
| 8 | 8 | example care head | audio | mobile |
| 9 | 9 | cut cup consider | audio | mobile |
| 10 | 10 | charge kind carry | audio | desktop |
| 11 | 11 | two someone size | video | desktop |
| 12 | 12 | skin value organization | audio | desktop |
| 13 | 13 | always dog science | video | desktop |
| 14 | 14 | project point then | banner | desktop |
| 15 | 15 | later that indeed | audio | mobile |
| 16 | 16 | support possible analysis | audio | mobile |
| 17 | 17 | nature top toward | banner | desktop |
| 18 | 18 | candidate able player | audio | mobile |
| 19 | 19 | everybody participant join | video | mobile |
| 20 | 20 | radio speak ahead | video | desktop |
| 21 | 21 | control three personal | video | desktop |
| 22 | 22 | street federal inside | banner | desktop |
| 23 | 23 | child general situation | audio | desktop |
| 24 | 24 | find account state | banner | mobile |

dim_advertisers

| | advertiser_id | advertiser_name | industry |
|----|---------------|---------------------|---------------|
| 0 | 0 | Jodi Arnold | Entertainment |
| 1 | 1 | Maria Harper | Finance |
| 2 | 2 | Kimberly Jefferson | Real Estate |
| 3 | 3 | Tyler Sanford | Manufacturing |
| 4 | 4 | Allison Lawrence MD | Real Estate |
| 5 | 5 | Shawn Norton | Energy |
| 6 | 6 | Cindy Turner | Retail |
| 7 | 7 | Teresa Nielsen | Education |
| 8 | 8 | Darryl Davis | Finance |
| 9 | 9 | Derrick Baker | Real Estate |
| 10 | 10 | Jamie Wilson | Manufacturing |
| 11 | 11 | Scott Mccoy | Healthcare |
| 12 | 12 | Justin Ferguson | Healthcare |
| 13 | 13 | Daniel Roberts | Manufacturing |
| 14 | 14 | Marco Blair | Education |
| 15 | 15 | Joshua Stewart | Entertainment |
| 16 | 16 | Adam Keller | Energy |
| 17 | 17 | Michele Burton | Finance |
| 18 | 18 | Vicki Bell | Real Estate |
| 19 | 19 | Robert Zamora | Retail |
| 20 | 20 | Amanda Pearson | Manufacturing |
| 21 | 21 | Juan Alexander | Real Estate |
| 22 | 22 | Michael Baker | Healthcare |
| 23 | 23 | Christopher Turner | Manufacturing |
| 24 | 24 | Olivia McDaniel | Finance |

fact_listening_stats

| | user_id | song_id | artist_id | date_id | location_id | play_count | total_play_time | skip_count |
|----|---------|---------|-----------|----------|-------------|------------|-----------------|------------|
| 0 | 74 | 155 | 12 | 3122008 | 24 | 27 | 1534.0 | 62 |
| 1 | 84 | 262 | 8 | 5081988 | 21 | 13 | 1097.0 | 45 |
| 2 | 72 | 97 | 36 | 27051987 | 5 | 34 | 2878.0 | 70 |
| 3 | 81 | 99 | 6 | 16091980 | 4 | 74 | 1885.0 | 85 |
| 4 | 47 | 160 | 11 | 5092004 | 24 | 79 | 1705.0 | 29 |
| 5 | 26 | 178 | 13 | 5032002 | 12 | 72 | 1915.0 | 70 |
| 6 | 73 | 48 | 33 | 7041971 | 9 | 96 | 2778.0 | 57 |
| 7 | 15 | 195 | 40 | 4061993 | 1 | 43 | 1537.0 | 41 |
| 8 | 99 | 55 | 31 | 16061975 | 23 | 49 | 1799.0 | 26 |
| 9 | 64 | 240 | 45 | 10091975 | 6 | 98 | 2877.0 | 38 |
| 10 | 85 | 114 | 34 | 20081978 | 17 | 93 | 1386.0 | 49 |
| 11 | 81 | 205 | 41 | 1061996 | 10 | 69 | 2174.0 | 41 |
| 12 | 77 | 58 | 42 | 13082020 | 3 | 33 | 1654.0 | 96 |
| 13 | 89 | 47 | 25 | 22021996 | 18 | 85 | 2816.0 | 68 |
| 14 | 55 | 188 | 34 | 26032021 | 24 | 86 | 1305.0 | 30 |
| 15 | 63 | 234 | 3 | 26091972 | 10 | 76 | 2443.0 | 57 |
| 16 | 61 | 96 | 48 | 17071996 | 4 | 29 | 2770.0 | 70 |
| 17 | 64 | 299 | 37 | 19031977 | 11 | 49 | 2395.0 | 50 |
| 18 | 2 | 167 | 4 | 15102015 | 14 | 73 | 1539.0 | 100 |
| 19 | 25 | 90 | 45 | 20011983 | 21 | 12 | 1036.0 | 53 |
| 20 | 89 | 116 | 5 | 26032001 | 16 | 10 | 1387.0 | 6 |
| 21 | 90 | 188 | 26 | 5092011 | 5 | 84 | 2323.0 | 58 |
| 22 | 51 | 237 | 35 | 12042006 | 6 | 51 | 2972.0 | 54 |
| 23 | 99 | 144 | 45 | 13122023 | 18 | 26 | 1819.0 | 67 |
| 24 | 53 | 18 | 29 | 13091972 | 1 | 80 | 2859.0 | 56 |
| 25 | 48 | 193 | 23 | 20072008 | 14 | 52 | 1750.0 | 53 |
| 26 | 8 | 284 | 41 | 25101978 | 6 | 73 | 2179.0 | 77 |
| 27 | 40 | 124 | 31 | 25081982 | 22 | 16 | 1754.0 | 54 |
| 28 | 34 | 100 | 16 | 14081985 | 23 | 30 | 1809.0 | 5 |
| 29 | 75 | 20 | 9 | 10051999 | 4 | 84 | 1608.0 | 23 |
| 30 | 98 | 182 | 42 | 22011980 | 18 | 66 | 1274.0 | 55 |
| 31 | 45 | 281 | 47 | 15091977 | 4 | 93 | 1442.0 | 99 |
| 32 | 9 | 233 | 42 | 26051979 | 2 | 69 | 2578.0 | 85 |
| 33 | 44 | 166 | 40 | 31051977 | 19 | 22 | 1637.0 | 88 |
| 34 | 31 | 140 | 32 | 3042020 | 25 | 48 | 2154.0 | 58 |
| 35 | 20 | 79 | 28 | 2031973 | 5 | 69 | 1844.0 | 97 |
| 36 | 57 | 157 | 3 | 8071994 | 21 | 7 | 2027.0 | 95 |
| 37 | 16 | 278 | 13 | 25022001 | 23 | 17 | 1112.0 | 24 |
| 38 | 46 | 270 | 37 | 28042002 | 10 | 12 | 2229.0 | 38 |
| 39 | 24 | 73 | 30 | 31011987 | 12 | 32 | 2752.0 | 81 |
| 40 | 4 | 119 | 39 | 28082023 | 11 | 43 | 2055.0 | 27 |
| 41 | 67 | 230 | 31 | 5022021 | 10 | 65 | 1626.0 | 19 |
| 42 | 98 | 27 | 0 | 30091995 | 25 | 0 | 1439.0 | 64 |
| 43 | 3 | 217 | 49 | 8081976 | 8 | 13 | 2260.0 | 99 |
| 44 | 29 | 10 | 6 | 6101975 | 8 | 42 | 1254.0 | 72 |
| 45 | 29 | 66 | 26 | 3041987 | 19 | 39 | 1989.0 | 92 |
| 46 | 17 | 194 | 44 | 27031977 | 13 | 51 | 1108.0 | 76 |
| 47 | 77 | 81 | 26 | 7031977 | 6 | 38 | 1062.0 | 64 |
| 48 | 19 | 23 | 27 | 30042013 | 11 | 48 | 2923.0 | 63 |
| 49 | 64 | 24 | 12 | 11041990 | 0 | 32 | 2483.0 | 16 |

fact_subscription_revenue

| | subscription_id | user_id | price_amount | discount_amount |
|----|-----------------|---------|--------------|-----------------|
| 0 | 15 | 70 | 518.7673 | 803.2624 |
| 1 | 11 | 81 | 949.1920 | 752.3696 |
| 2 | 4 | 45 | 648.7443 | 105.7026 |
| 3 | 15 | 61 | 181.2004 | 291.8732 |
| 4 | 19 | 91 | 285.2516 | 981.8588 |
| 5 | 15 | 73 | 953.4113 | 549.7436 |
| 6 | 18 | 40 | 408.0218 | 251.0908 |
| 7 | 10 | 33 | 753.7369 | 829.6835 |
| 8 | 3 | 90 | 436.5115 | 659.0460 |
| 9 | 18 | 62 | 630.0771 | 115.0403 |
| 10 | 17 | 53 | 302.0301 | 351.0214 |
| 11 | 25 | 9 | 581.1656 | 231.4019 |
| 12 | 3 | 30 | 375.1293 | 276.3093 |
| 13 | 4 | 47 | 436.5566 | 896.5260 |
| 14 | 22 | 90 | 748.0941 | 368.3860 |
| 15 | 10 | 39 | 358.0806 | 286.3828 |
| 16 | 24 | 5 | 383.9800 | 523.8041 |
| 17 | 23 | 50 | 394.4679 | 710.6106 |
| 18 | 5 | 73 | 489.8662 | 673.2034 |
| 19 | 7 | 70 | 370.0843 | 271.6611 |
| 20 | 12 | 58 | 456.6889 | 657.9644 |
| 21 | 9 | 12 | 469.0987 | 911.2790 |
| 22 | 2 | 2 | 510.2275 | 924.2088 |
| 23 | 18 | 7 | 890.1571 | 121.5512 |
| 24 | 21 | 64 | 892.6382 | 937.6598 |
| 25 | 22 | 48 | 443.5143 | 327.0436 |
| 26 | 5 | 12 | 569.4203 | 580.9649 |
| 27 | 25 | 36 | 496.5838 | 757.7221 |
| 28 | 19 | 61 | 208.2748 | 536.4354 |
| 29 | 15 | 0 | 376.3548 | 412.0390 |
| 30 | 24 | 47 | 582.7051 | 961.8875 |
| 31 | 3 | 27 | 474.7708 | 263.5290 |
| 32 | 1 | 77 | 953.4729 | 850.9857 |
| 33 | 21 | 18 | 579.3662 | 651.6234 |
| 34 | 4 | 20 | 825.7200 | 428.2675 |
| 35 | 14 | 18 | 940.3983 | 591.4897 |
| 36 | 16 | 26 | 421.4671 | 889.0238 |
| 37 | 15 | 74 | 364.7276 | 275.3286 |
| 38 | 24 | 32 | 766.0056 | 561.6472 |
| 39 | 10 | 79 | 506.3349 | 471.7380 |
| 40 | 22 | 36 | 433.6494 | 926.0841 |
| 41 | 19 | 4 | 916.6859 | 263.9628 |
| 42 | 22 | 26 | 797.9162 | 309.3800 |
| 43 | 21 | 83 | 958.5758 | 774.4509 |
| 44 | 18 | 52 | 386.1540 | 827.1223 |
| 45 | 19 | 40 | 462.4971 | 374.9078 |
| 46 | 16 | 60 | 107.4240 | 245.5726 |
| 47 | 20 | 11 | 844.3030 | 508.4696 |
| 48 | 1 | 66 | 158.8432 | 445.0723 |
| 49 | 15 | 1 | 949.1685 | 980.0501 |

fact_ad_revenue

| | ad_id | advertiser_id | user_id | date_id | location_id | impressions | clicks | revenue |
|----|-------|---------------|---------|----------|-------------|-------------|--------|----------|
| 0 | 20 | 4 | 2 | 27021997 | 9 | 2345 | 1730 | 274.2529 |
| 1 | 6 | 7 | 30 | 9042016 | 22 | 1318 | 1203 | 790.8953 |
| 2 | 3 | 2 | 35 | 8021990 | 15 | 2090 | 2237 | 292.6663 |
| 3 | 4 | 4 | 20 | 1051987 | 11 | 1967 | 1510 | 736.6041 |
| 4 | 15 | 11 | 52 | 10011972 | 8 | 2517 | 2030 | 631.8490 |
| 5 | 7 | 18 | 14 | 25082009 | 9 | 1003 | 2621 | 141.2121 |
| 6 | 20 | 17 | 84 | 19011990 | 2 | 2184 | 1679 | 702.8577 |
| 7 | 23 | 21 | 31 | 23042005 | 16 | 2419 | 1508 | 160.7831 |
| 8 | 12 | 24 | 98 | 15032009 | 23 | 1977 | 2430 | 196.4188 |
| 9 | 11 | 21 | 45 | 4012000 | 19 | 2242 | 1703 | 636.1875 |
| 10 | 20 | 16 | 67 | 22051989 | 11 | 2154 | 1611 | 322.5803 |
| 11 | 5 | 18 | 49 | 24101971 | 6 | 1434 | 2566 | 300.6670 |
| 12 | 16 | 7 | 9 | 25011985 | 19 | 2027 | 1520 | 764.8835 |
| 13 | 15 | 19 | 88 | 6091976 | 16 | 1538 | 1359 | 914.9163 |
| 14 | 15 | 23 | 75 | 5122003 | 17 | 1106 | 1436 | 389.0408 |
| 15 | 20 | 13 | 10 | 6051971 | 6 | 2022 | 1769 | 924.2461 |
| 16 | 2 | 20 | 33 | 20041973 | 21 | 1681 | 1995 | 508.6197 |
| 17 | 24 | 8 | 92 | 23121996 | 17 | 1432 | 1219 | 309.0372 |
| 18 | 21 | 0 | 66 | 3022016 | 8 | 1281 | 1380 | 861.9808 |
| 19 | 17 | 11 | 85 | 2062004 | 14 | 1571 | 1277 | 178.3686 |
| 20 | 9 | 22 | 88 | 19021995 | 24 | 1948 | 1125 | 970.7580 |
| 21 | 2 | 22 | 47 | 30052020 | 5 | 1485 | 1629 | 227.1406 |
| 22 | 11 | 10 | 2 | 11022022 | 13 | 1533 | 2620 | 417.0904 |
| 23 | 25 | 8 | 10 | 20102006 | 5 | 1224 | 2767 | 811.4493 |
| 24 | 17 | 4 | 28 | 1022009 | 14 | 1945 | 1510 | 170.0899 |
| 25 | 4 | 3 | 96 | 11071994 | 21 | 1391 | 1934 | 826.0754 |
| 26 | 24 | 15 | 0 | 23051973 | 5 | 1899 | 2936 | 802.0627 |
| 27 | 5 | 3 | 3 | 30061987 | 17 | 2215 | 1620 | 345.4974 |
| 28 | 20 | 8 | 43 | 1081970 | 12 | 1562 | 2103 | 918.1179 |
| 29 | 15 | 8 | 41 | 13112024 | 17 | 2811 | 2092 | 608.1872 |
| 30 | 16 | 3 | 85 | 23111979 | 4 | 2471 | 1543 | 861.6342 |
| 31 | 12 | 14 | 31 | 26102012 | 5 | 2667 | 1151 | 864.3059 |
| 32 | 18 | 24 | 46 | 14122020 | 18 | 2120 | 2434 | 623.5364 |
| 33 | 1 | 4 | 84 | 21071977 | 15 | 2005 | 2963 | 316.4525 |
| 34 | 1 | 7 | 35 | 27121989 | 13 | 1749 | 1596 | 542.7510 |
| 35 | 22 | 22 | 82 | 24031994 | 22 | 1769 | 2328 | 710.7824 |
| 36 | 8 | 18 | 31 | 5072013 | 13 | 2413 | 1480 | 909.4243 |
| 37 | 21 | 9 | 47 | 29121978 | 4 | 2609 | 2512 | 487.1959 |
| 38 | 25 | 2 | 32 | 14062007 | 25 | 1241 | 2742 | 488.8024 |
| 39 | 19 | 22 | 62 | 11091993 | 7 | 2495 | 2615 | 427.6466 |
| 40 | 10 | 11 | 93 | 18021971 | 10 | 1347 | 2091 | 958.5965 |
| 41 | 1 | 16 | 11 | 8092013 | 17 | 1773 | 1253 | 252.6162 |
| 42 | 3 | 19 | 11 | 17102014 | 3 | 2447 | 1649 | 794.4534 |
| 43 | 1 | 25 | 3 | 10031983 | 11 | 1254 | 1195 | 138.3930 |
| 44 | 1 | 13 | 98 | 27092008 | 1 | 2824 | 1976 | 848.9353 |
| 45 | 17 | 11 | 48 | 2072015 | 15 | 1561 | 1281 | 330.0617 |
| 46 | 4 | 9 | 96 | 4071995 | 20 | 1186 | 2508 | 171.4955 |
| 47 | 14 | 13 | 16 | 29101986 | 8 | 1901 | 2392 | 815.0404 |
| 48 | 15 | 18 | 74 | 10022006 | 9 | 2752 | 2863 | 726.2236 |
| 49 | 22 | 19 | 87 | 7061977 | 17 | 1532 | 1332 | 480.5669 |

Number of rows of each table

```
In [35]: for table in table_list:
    print(f'{table}')
    count_query = f"select COUNT(*) from {table}"
    display(get_ipython().run_line_magic('sql', count_query))

dim_users
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
100

dim_songs
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
300

dim_artists
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
50

dim_date
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
29

dim_locations
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
25

dim_subscriptions
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
25

dim_ads
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
25

dim_advertisers
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
25

fact_listening_stats
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
50

fact_subscription_revenue
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
50

fact_ad_revenue
Running query in 'postgresql://vincent***@samplecluster.cq68pg38qszs.us-east-1.redshift.amazonaws.com:5439/music_streaming_service'
1 rows affected.

count
_____
50
```

Rollback/Start from scratch queries

```
In [36]: # %%sql
# DROP TABLE IF EXISTS dim_users, dim_songs, dim_artists, dim_date,
#     dim_locations, dim_subscriptions, dim_ads, dim_advertisers,
#     fact_listening_stats, fact_subscription_revenue, fact_ad_revenue
```