

Бизнес-область: Сервис потоковой передачи музыки

Этап 1: Определение конкретный бизнес-процесс для выбранной области.

Что нас интересует? В музыкальном сервисе главным событием является прослушивание музыки, так как именно оно генерирует данные для анализа(что слушают, кто слушает, когда слушают, где слушают).

Таким образом, в качестве конкретного бизнес-процесса я выбрала "Добавление песни в плейлист".

Так как частота добавления песни в плейлист показывает заинтересованность и вовлеченность пользователей. Это дает возможность измерять количество добавлений и популярность песен.

Этап 2: Определение уровень детализации (grain) бизнес-процесса.

"Что представляет собой ОДНА запись в нашей фактовой таблице?" - одно добавление одной конкретной песни в один конкретный плейлист одним пользователем в определенный момент времени.

Этап 3: Определение таблицы измерений (dimension tables).

Исходя из нашего grain, можно выделить основные сущности, которые его описывают. Они отвечают на вопросы "Кто?", "Что?", "Куда?", "Когда?".

1. dim_user ("кто?") – пользователь, которому принадлежит плейлист.
 - user_sk – PK
 - source_user_id
 - username
 - country
2. dim_song ("что?") – песня, которая была добавлена.
 - song_sk – PK
 - source_song_id
 - title
 - artist_name
 - genre
3. dim_playlist ("куда?") – плейлист, в который добавили песню.
 - playlist_sk – PK
 - source_playlist_id
 - playlist_name
 - is_public – признак публичности артиста
4. dim_date ("когда?") – дата добавления песни в плейлист.
 - date_sk – PK
 - source_date_id
 - year

- month
- day
- week

Этап 4: Определение таблицы фактов (fact table) с её метриками и атрибутами.

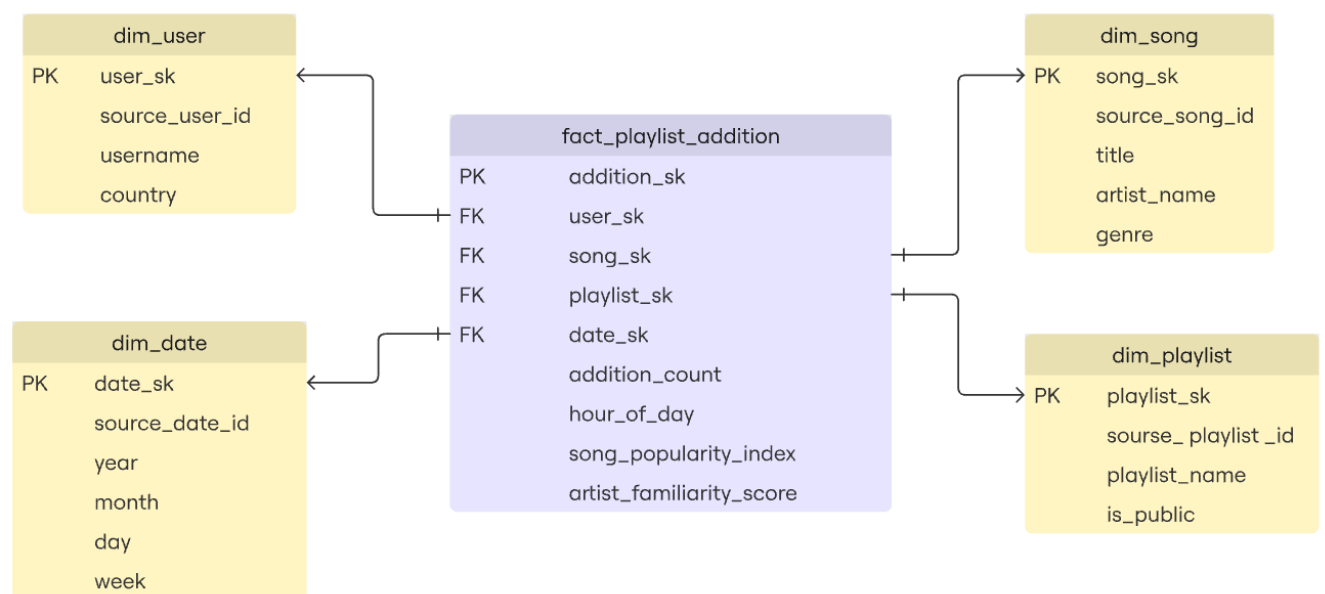
Таблица фактов fact_playlist_addition:

- addition_sk – PK
- user_sk – FK, ссылка на dim_user. "кто добавил?"
- song_sk – FK, ссылка на dim_song. "что добавили?"
- playlist_sk – FK, ссылка на dim_playlist. "куда добавили?"
- date_sk – FK, ссылка на dim_date. "когда добавили?"
- addition_count – счетчик события
- hour_of_day – час суток, когда произошло добавление
- song_popularity_index – индекс популярности песни
- artist_familiarity_score – насколько известен артист

Метрики: addition_count, hour_of_day, song_popularity_index, artist_familiarity_score.

Этап 5: Моделирование физической модели.

Схема “Звезда”:



Этап 6: Необходимо привести 3–5 примеров аналитических запросов, которые помогут ответить на ключевые вопросы вашего бизнес-процесса.

Код с таблицами и их содержимым можно найти в приложении А.

ПРИЛОЖЕНИЕ А

```
CREATE TABLE dim_user (  
    user_sk SERIAL PRIMARY KEY,  
    source_user_id INTEGER NOT NULL,  
    username VARCHAR(100) NOT NULL,  
    country VARCHAR(50)  
);
```

```
CREATE TABLE dim_song (  
    song_sk SERIAL PRIMARY KEY,  
    source_song_id INTEGER NOT NULL,  
    title VARCHAR(255) NOT NULL,  
    artist_name VARCHAR(100) NOT NULL,  
    genre VARCHAR(50)  
);
```

```
CREATE TABLE dim_playlist (  
    playlist_sk SERIAL PRIMARY KEY,  
    source_playlist_id INTEGER NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    is_public BOOLEAN NOT NULL DEFAULT FALSE  
);
```

```
CREATE TABLE dim_date (  
    date_sk SERIAL PRIMARY KEY,  
    source_date_id DATE NOT NULL UNIQUE,  
    year INTEGER NOT NULL,  
    month INTEGER NOT NULL,  
    day INTEGER NOT NULL,  
    week INTEGER NOT NULL  
);
```

```
CREATE TABLE fact_playlist_addition (  
    addition_sk SERIAL PRIMARY KEY,  
    user_sk INTEGER NOT NULL REFERENCES dim_user(user_sk),  
    song_sk INTEGER NOT NULL REFERENCES dim_song(song_sk),  
    playlist_sk INTEGER NOT NULL REFERENCES dim_playlist(playlist_sk),  
    date_sk INTEGER NOT NULL REFERENCES dim_date(date_sk),  
    addition_count INTEGER NOT NULL DEFAULT 1,  
    hour_of_day INTEGER NOT NULL,  
    song_popularity_index DECIMAL(10,4) NOT NULL,  
    artist_familiarity_score DECIMAL(10,4) NOT NULL  
);
```

```

INSERT INTO dim_user (source_user_id, username, country) VALUES
(1, 'Ranunculus', 'Belarus'),
(2, 'rrilp', 'UK'),
(3, 'fngdof', 'Canada'),
(4, 'gestapo', 'Germany');

```

```

INSERT INTO dim_song (source_song_id, title, artist_name, genre) VALUES
(1, 'Blinding Lights', 'The Weeknd', 'R&B'),
(2, 'Save Your Tears', 'The Weeknd', 'R&B'),
(3, 'Dont Start Now', 'Dua Lipa', 'Pop'),
(4, 'Levitating', 'Dua Lipa', 'Pop'),
(5, 'Do I Wanna Know?', 'Arctic Monkeys', 'Rock'),
(6, 'R U Mine?', 'Arctic Monkeys', 'Rock'),
(7, 'Therefore I Am', 'Billie Eilish', 'Pop'),
(8, 'My Future', 'Billie Eilish', 'Pop'),
(9, 'Follow You', 'Imagine Dragons', 'Rock'),
(10, 'Wrecked', 'Imagine Dragons', 'Rock');

```

```

INSERT INTO dim_playlist (source_playlist_id, name, is_public) VALUES
(1, 'Workout Mix', true),
(2, 'Chill Vibes', false),
(3, 'Party Time', true),
(4, 'Driving Songs', true);

```

```

INSERT INTO dim_date (source_date_id, year, month, day, week) VALUES
('2023-05-01', 2023, 5, 1, 18),
('2023-05-02', 2023, 5, 2, 18),
('2023-05-03', 2023, 5, 3, 18),
('2023-05-04', 2023, 5, 4, 18),
('2023-05-05', 2023, 5, 5, 18);

```

```

INSERT INTO fact_playlist_addition
(user_sk, song_sk, playlist_sk, date_sk, addition_count, hour_of_day,
song_popularity_index, artist_familiarity_score)
VALUES
(1, 1, 1, 1, 1, 10, 0.95, 0.92),
(1, 3, 1, 1, 1, 10, 0.88, 0.89),
(2, 5, 2, 2, 1, 15, 0.82, 0.85),
(2, 9, 2, 2, 1, 15, 0.78, 0.80),
(3, 2, 3, 3, 1, 20, 0.90, 0.92),
(3, 4, 3, 3, 1, 20, 0.85, 0.89),
(4, 7, 4, 4, 1, 9, 0.75, 0.88),
(4, 10, 4, 4, 1, 9, 0.72, 0.80),
(1, 6, 1, 5, 1, 11, 0.80, 0.85),

```

(3, 8, 3, 5, 1, 21, 0.70, 0.88);