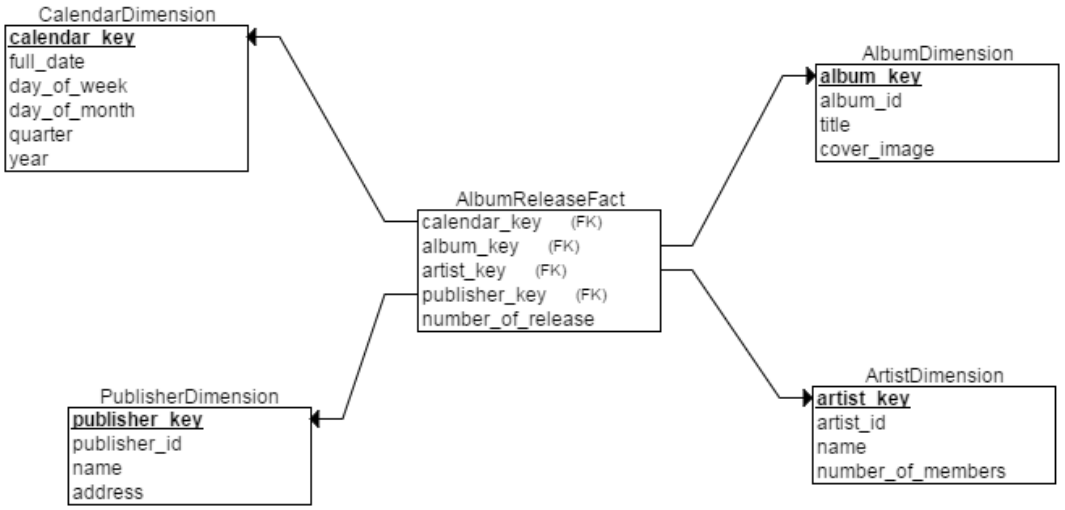
Sample No.1

(a) Dimensional model with a star schema



(b) Query against star schema

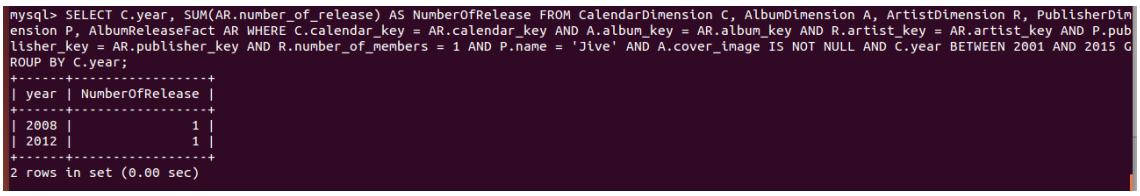
How did the quantities of albums released by artists each of which consists of only one person (member) and published by "Jive" and having a cover image change during the years 2001 to 2015?

(c) SQL and query output

[SQL]

```
SELECT C.year, SUM(AR.number_of_release) AS NumberOfRelease
FROM CalendarDimension C, AlbumDimension A, ArtistDimension R,
PublisherDimension P, AlbumReleaseFact AR
WHERE C.calendar_key = AR.calendar_key
AND A.album_key = AR.album_key
AND R.artist_key = AR.artist_key
AND P.publisher_key = AR.publisher_key
AND R.number_of_members = 1
AND P.name = 'Columbia'
AND A.cover_image IS NOT NULL
AND C.year BETWEEN 2001 AND 2015
GROUP BY C.year;
```

[query output]



(d) How dimension and fact tables are populated?

CalendarDimension table is populated by scanning all data with respect to released_date in Album table in our relational schema, extracting/generating all attributes in CalendarDimension table (full_date, day_of_week, day_of_month, quarter, year) based on the date, and inserting the extracted/generated data to the dimension table.

AlbumDimension table is populated by scanning all data in the original Album table in our schema, extracting/generating all attributes for AlbumDimension table (album_id, title, cover_image), and inserting the extracted/generated data to the dimension table.

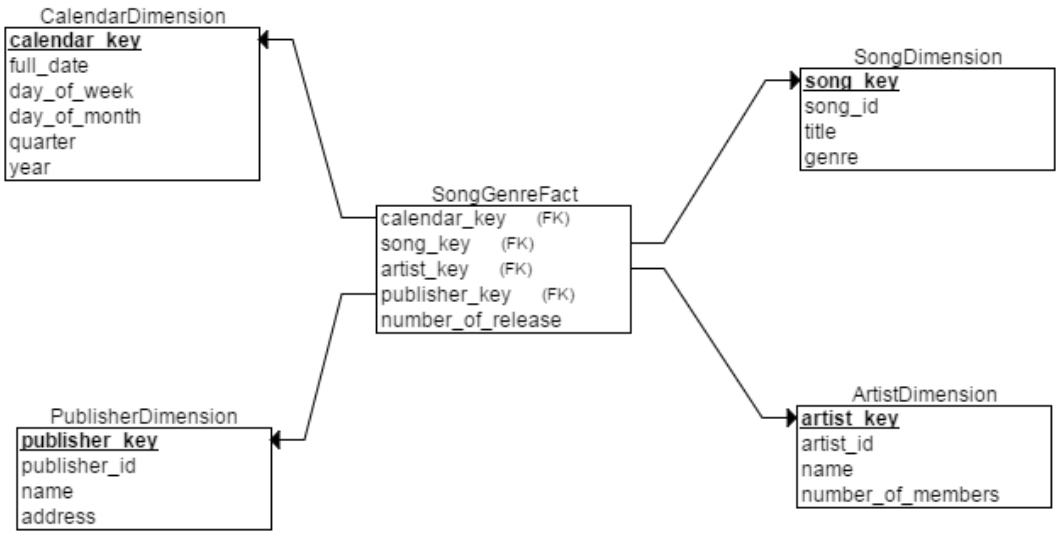
ArtistDimension table is populated by scanning all data in the original Artist table in our schema, extracting/generating all attributes for ArtistDimension table (artist_id, name, number_of_members), and inserting the extracted/generated data to the dimension table.

PublisherDimension table is populated by scanning all data in the original Publisher table in our schema, extracting/generating all attributes for PublisherDimension table (publisher_id, name, address), and inserting the extracted/generated data to the dimension table.

AlbumReleaseFact table is populated by scanning all data with respect to multiple tables, Album, Artist, Publisher, finding all foreign key attributes in AlbumReleaseFact table (calendar_key, album_key, artist_key, publisher_key), summing up all events for the release having the same dimension, and inserting the generated data to the fact table.

Sample No.2

(a) Dimensional model with a star schema



(b) Query against star schema

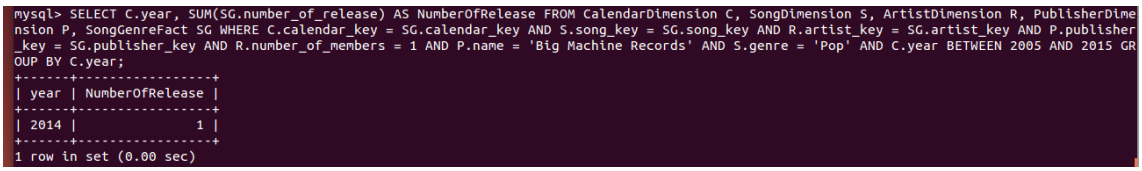
What is the year-based number of songs published by "Big Machine Records" between years 2005 and 2010, made by artists each of which consists of only one person (member), and categorized to the genre, "Pop"?

(c) SQL and query output

[SQL]

```
SELECT C.year, SUM(SG.number_of_release) AS NumberOfRelease
FROM CalendarDimension C, SongDimension S, ArtistDimension R,
PublisherDimension P, SongGenreFact SG
WHERE C.calendar_key = SG.calendar_key
AND S.song_key = SG.song_key
AND R.artist_key = SG.artist_key
AND P.publisher_key = SG.publisher_key
AND R.number_of_members > 1
AND P.name = 'Big Machine Records'
AND S.genre = 'Pop'
AND C.year BETWEEN 2005 and 2010
GROUP BY C.year;
```

[query output]



```
mysql> SELECT C.year, SUM(SG.number_of_release) AS NumberOfRelease FROM CalendarDimension C, SongDimension S, ArtistDimension R, PublisherDime
nsion P, SongGenreFact SG WHERE C.calendar_key = SG.calendar_key AND S.song_key = SG.song_key AND R.artist_key = SG.artist_key AND P.publisher
_key = SG.publisher_key AND R.number_of_members = 1 AND P.name = 'Big Machine Records' AND S.genre = 'Pop' AND C.year BETWEEN 2005 AND 2015 GR
OUP BY C.year;
+------+-----------------+
| year | NumberOfRelease |
+------+-----------------+
| 2014 |               1 |
+------+-----------------+
1 row in set (0.00 sec)
```

(d) How dimension and fact tables are populated?

As for CalendarDimension table, ArtistDimension table , PublisherDimension table, the population scheme is the same as the one shown in Sample 1.

SongDimension table is populated by scanning all data in the original Song table in our schema, extracting/generating all attributes for SongDimension table (song_id, title, genre), and inserting the extracted/generated data to the dimension table.

SongGenreFact table is populated by scanning all data with respect to multiple tables, Song, Album, Artist, Publisher, finding all foreign key attributes in SongGenreFact table (calendar_key, song_key, artist_key, publisher_key), summing up all events for the release having the same dimension, and inserting the generated data to the fact table.

Relational schema