Relational Algebra of functional queries that are located in functional.sql

1.

π count(*) (σ End_Date<? ∧End_Date>? Album)

2.

π(*) (σ Name like %?% performer)

π count(*) (π * (σ M_ID=?∧date>?∧date<?
(π (*) ((π (*) Songs) ⋈ performer_songs))))

3.

π(*) (σ Name like %?% performer)

π count(*) (σ End_Date > ? ∧ End_Date < ?
(γ a_ID (π A_ID (π (*) (π * Album_Songs) ⋈
(π (*) (σ M_ID = ? performer_Songs))))) ⋈ Album)

4.

π(*) (inst)

π count(*) (π (*) (σ I_ID=? performer_inst)⋈ performer_Songs)

5.

π(*) (σ Name like %?% album)

π (*) ((π I_ID performer)⋈(π M_ID
(σ a_id = ? Album_Songs)) ⋈ () ⋈ instt)

6.

π(*) (Prod)

π count (*) ((π (*) (σ End_Date > ? ∧ End_Date < ?
((σ p_ID=? album_prod) ⋈ album))))

7.

π(*) (inst)

π count (*) ((π (*)
(σ I_ID=? performer_inst)) ⋈ performer_Songs)

8.

π count (*) (γ M_id (π (*) performer) ⋈ performer_Songs)

9.

π (*) (Songs)

π count (*) (σ T_ID=? performer_songs)

π (*) (σ T_ID=? performer_songs)

π (*) (σ ID_musician = ? performer)

10.

π(*) (songs)

11.

π(*) (σ Date>? ∧ Date <? songs)

12.

π (*) (σ End_Date = (π MIN(End_Date) album) album)

13.

π(*) (songs)

13.

π count (*)(σ t_ID=? album_songs)

14.

π(*) (songs)

π(*) (album)

π count (*)(σ Tech=? ((π (*) (σ A_ID =?)) ⋈ songs))

15.

π(*) (performer)

π count (*)((π* (σ M_ID = ? performer_songs)) ⋈ songs)