מסמך פעולות SQL

```
1. DROP database `Music_DB`;
2.CREATE database `Music_DB`;
3.CREATE TABLE `Music_DB`.`musicians` (
                `musician_id` INT  NOT NULL DEFAULT 0,
                `musician_name` VARCHAR(45) NOT NULL,
                `musician_phone` VARCHAR(45) NOT NULL,
                `musician_address` VARCHAR(45) NOT NULL,
                `musician_type` INT NOT NULL,
                PRIMARY KEY (`musician_id`));
4. CREATE TABLE `Music_DB`.`album` (
                `album_id` INT  NOT NULL DEFAULT 0,
                `album_name` VARCHAR(45) NOT NULL,
                `record_begin` VARCHAR(45) NOT NULL,
                `record_end` VARCHAR(45) NOT NULL,
                PRIMARY KEY (`album_id`));
5. CREATE TABLE `Music_DB`.`instrument` (
                `instrument_id` INT  NOT NULL DEFAULT 0,
                `instrument_type` VARCHAR(45) NOT NULL,
                `manufacturer_name` VARCHAR(45) NOT NULL,
                PRIMARY KEY (`instrument_id`,`manufacturer_name`,`instrument_type`));
6.CREATE TABLE `Music_DB`.`producer` (
                `producer_id` INT  NOT NULL DEFAULT 0,
                `producer_name` VARCHAR(45) NOT NULL,
                PRIMARY KEY (`producer_id`));
7. CREATE TABLE `Music_DB`.`recording_technition` (
                `technition_id` INT  NOT NULL DEFAULT 0,
                `technition_name` VARCHAR(45) NOT NULL,
                PRIMARY KEY (`technition_id`));
8. CREATE TABLE `Music_DB`.`album_n_producer` (
                `fk_album` int(11) NOT NULL,
                `fk_producer` int(11) NOT NULL,
                PRIMARY KEY (`fk_album`,`fk_producer`),
                KEY `fk_producer_idx` (`fk_producer`),
                CONSTRAINT `fk_album` FOREIGN KEY (`fk_album`) REFERENCES
`album` (`album_id`) ON DELETE CASCADE ON UPDATE CASCADE,
                CONSTRAINT `fk_producer` FOREIGN KEY (`fk_producer`) REFERENCES
`producer` (`producer_id`) ON DELETE CASCADE ON UPDATE CASCADE
                ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci

9. CREATE TABLE `Music_DB`.`musical_scene` (
                `fk_technition_id` INT NOT NULL,
                `song_id` INT NOT NULL,
                `song_name` VARCHAR(45) NOT NULL,
                `lyrics_author_name` VARCHAR(45) NOT NULL,
                `composure_name` VARCHAR(45) NOT NULL,
                `length` INT NOT NULL,
                `genere` VARCHAR(45) NOT NULL,
                `record_date` VARCHAR(45) NOT NULL,
                 INDEX `fk_technition_id_idx` (`fk_technition_id` ASC) VISIBLE,
                 PRIMARY KEY(`song_id`),
                 CONSTRAINT `fk_technition_id`
                 FOREIGN KEY(`fk_technition_id`)
                 REFERENCES `Music_DB`.`recording_technition` (`technition_id`)
                 ON DELETE NO ACTION
                 ON UPDATE NO ACTION);
```

```
10. CREATE TABLE `musical_scene_and_musician` (
                 `fk_song_id` int(11) NOT NULL,
                `fk_musician_id` int(11) NOT NULL,
                PRIMARY KEY (`fk_song_id`,`fk_musician_id`),
                KEY `fk_musician_id_idx` (`fk_musician_id`),
                CONSTRAINT `fk_musician_id` FOREIGN KEY (`fk_musician_id`)
REFERENCES `musicians` (`musician_id`) ON DELETE CASCADE ON UPDATE
CASCADE,
                CONSTRAINT `fk_song_id` FOREIGN KEY (`fk_song_id`) REFERENCES
`musical_scene` (`song_id`) ON DELETE CASCADE ON UPDATE CASCADE
                ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

```
11. CREATE TABLE `musicians_and_instruments` (
                `fk_musician` int(11) NOT NULL,
                `fk_instrument` int(11) NOT NULL,
                PRIMARY KEY (`fk_musician`,`fk_instrument`),
                KEY `fk_instrument_idx` (`fk_instrument`),
                CONSTRAINT `fk_instrument` FOREIGN KEY (`fk_instrument`)
REFERENCES `instrument` (`instrument_id`) ON DELETE CASCADE ON UPDATE
CASCADE,
                CONSTRAINT `fk_musician` FOREIGN KEY (`fk_musician`) REFERENCES
`musicians` (`musician_id`) ON DELETE CASCADE ON UPDATE CASCADE
                ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

```
12. CREATE TABLE `Music_DB`.`album_and_musical_scene` (
                `album_id` INT NOT NULL,
                `musical_scene` INT NOT NULL,
                PRIMARY KEY(`album_id`, `musical_scene`),
                INDEX `fk_ms_idx` (`musical_scene` ASC) VISIBLE,
                CONSTRAINT `fk_album_id_`
                FOREIGN KEY(`album_id`)
```

```
            REFERENCES `Music_DB`.`album` (`album_id`)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION,
            CONSTRAINT `fk_ms`
            FOREIGN KEY(`musical_scene`)
            REFERENCES `Music_DB`.`musical_scene` (`song_id`)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION);
```

```
13. CREATE TABLE `ms_musician_instrument` (
            `fk_musician` int(11) NOT NULL,
            `fk_instrument` int(11) NOT NULL,
            `fk_ms` int(11) NOT NULL,
            PRIMARY KEY(`fk_musician`,`fk_ms`),
            KEY `_fk_ms_idx` (`fk_ms`),
            KEY `_fk_instrument_idx` (`fk_instrument`),
            CONSTRAINT `_fk_instrument` FOREIGN KEY(`fk_instrument`)
            REFERENCES `instrument` (`instrument_id`),
            CONSTRAINT `_fk_ms` FOREIGN KEY(`fk_ms`) REFERENCES
            `musical_scene` (`song_id`),
            CONSTRAINT `_fk_musician` FOREIGN KEY(`fk_musician`) REFERENCES
            `musicians` (`musician_id`)
            ) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE =
            utf8mb4_0900_ai_ci
```

```
14. INSERT INTO musicians(musician_id, musician_name, musician_phone,
musician_address, musician_type)
            VALUES  (0,'Arlene Lippman','050123120','Ana Frank 12',1),
                        (1,'Mia Clover','050123121','777 Brockton Avenue',2),
                        (2,'Joaquin Rexrode','050123122','262 Swansea Mall Dr',0),
                        (3,'Donette Loe','050123123','3005 Cranberry Hwy Rt 6 28',2),
                        (4,'Jackson Mccollum','050123124','352 Palmer Road',0),
                        (5,'Piper Sundstrom','050123125','297 Grant Avenu',1),
                        (6,'Nicolas Mariner','050123126','30 Catskill',1),
                        (7,'Brande Seabaugh','050123127','3018 East Ave',1),
                        (8,'Leana Mejias','050123128','161 Centereach Mall',0),
                        (9,'Mary Naff','050123129','1000 State Route 36',0),
                        (10,'Micheline Shoaf','050123130','100 Thruway Plaza',2);
```

```
15. INSERT INTO album(album_id,album_name,record_begin,record_end)
            VALUES  (0,'Avatar Country','2010/01/01','2011/01/01'),
            (1,'Wrong Creatures','2011/01/02','2012/01/02'),
            (2,'Camila','2014/01/03','2015/08/18'),
            (3,'No Cross No Crown','2019/01/04','2020/11/05'),
            (4,'What Happens Next','2018/01/04','2019/01/05'),
                (5,'Underworld','2019/01/30','2020/12/14'),
                (6,'Mania','2013/06/20','2014/07/03'),
                (7,'Vertigo','2017/04/14','2018/09/05'),
                (8,'Lost on the Road to Eternity','2019/5/11','2020/12/13'),
                (9,'Rise to Glory','2018/07/30','2019/05/20'),
                (10,'Beloved Antichrist','2011/02/27','2013/04/12');

16. INSERT INTO instrument (instrument_id, instrument_type, manufacturer_name)
VALUES (0,'Guitar','Cley Zemer'),
 (1,'Guitar','BassLab'),
 (2,'Guitar','Yamaha'),
 (3,'Piano','BassLab'),
 (4,'Piano','Yamaha'),
 (5,'Piano','intruments INC'),
 (6,'Clarient','BassLab'),
 (7,'Clarient','Cley Zemer'),
 (8,'Bass Guitar','BassLab'),
 (9,'Cello','BassLab'),
 (10,'Violin','BassLab');

17. INSERT INTO producer(producer_id,producer_name)
                VALUES  (0,'Kenneth Hager '),
                (1,'Lino Herder'),
                (2,'Nanci Lichtenberger '),
                (3,'Aimee Pendergrass '),
                (4,'Rosy Hahne'),
                (5,'Haywood Lovett'),
                (6,'Coleman Greenly '),
                (7,'Jerrod Nack'),
                (8,'Larae Cossette'),
                (9,'Olga Behling'),
18. INSERT INTO recording_technition(technition_id,technition_name)
                VALUES  (0,'Son Shipman'),
                (1,'Anderson Brouwer'),
                (2,'Waldo Mccraney'),
                (3,'Major Yelle'),
                (4,'Wendell Kai'),
                (5,'Stanford Butera'),
                (6,'Trey Taitt'),
                (7,'Cory Perez'),
                (8,'Leonard Shanklin'),
                (9,'Jon Mitts'),
                (10,'Harris Tempel');
19.  INSERT INTO album_n_producer(fk_album, fk_producer)
                VALUES  (0,1),
                (1,1),
                (1,2),
                (2,9),
                (3,3),
                (4,3),
                (5,4),
                (6,1),
                (7,9),
                (8,7),
                (9,5),
                (10,7);

20. INSERT INTO musical_scene (song_id, song_name,
lyrics_author_name,composure_name, length, genere, record_date, fk_technition_id)
                VALUES(1, 'Californication', 'Avi Gabay', 'Arik Nesher', 200, 'Rock',
'2005/02/01', 1),
                (2, 'Zephir song', 'Anthony Kiedis', 'Rick Rubin', 210, 'Pop', '2012/08/13', 0),
                (3, 'Aeroplane', 'Anthony Kiedis', 'Rick Rubin', 200, 'Rock', '2011/07/27', 1),
                (4, 'Annie Wants a Baby', 'Anthony Kiedis', 'Rick Rubin', 180, 'Rock',
'2013/06/28', 2),
                (5, 'By The Way', 'Flea', 'George Clinton', 185, 'Rock', '2011/10/07', 3),
                (6, 'Heaven', 'Flea', 'George Clinton', 222, 'Jazz', '2019/05/15', 4),
                (7, 'Yellow', 'Cliff Martinez', 'Andy Gill', 225, 'Jazz', '2018/09/21', 5),
                (8, 'Yellow light', 'Cliff Martinez', 'Andy Gill', 212, 'Jazz', '2012/02/16', 6),
                (9, 'Cant stop', 'John Frusciante', 'Andy Gill', 192, 'Pop', '2011/01/07', 9),
                (10, 'Dark Necessities', 'John Frusciante', 'Andy Gill', 195, 'Rock',
'2011/12/15', 10);

21. INSERT INTO musical_scene_and_musician(fk_song_id, fk_musician_id)
                VALUES (1,2),
                (1,5),
                (1,6),
                (2,9),
                (3,6),
                (4,7),
                (4,2),
                (5,8),
                (6,1),
                (6,0),
                (7,8),
                (7,10),
                (8,2),
                (8,3),
                (8,7),
                (8,9),
                (9,0),
                (10,5),
                    (10,7);

22. INSERT INTO musical_scene_n_instrument(fk_song_id, fk_instrument_id)
                    VALUES (1,0),
                    (1,2),
                    (1,6),
                    (2,9),
                    (3,6),
                    (4,7),
                    (4,2),
                    (5,8),
                    (6,1),
                    (6,0),
                    (7,8),
                    (7,10),
                    (8,2),
                    (8,3),
                    (8,7),
                    (8,9),
                    (9,0),
                    (10,5),
                    (10,7);
23. INSERT INTO musicians_and_instruments(fk_musician, fk_instrument)
                    VALUES ( 1,2),
                    (1,5),
                    (1,8),
                    (2,1),
                    (3,9),
                    (4,2),
                    (4,8),
                    (8,1),
                    (8,2),
                    (8,8),
                    (9,7),
                    (10,1),
                    (10,2);
24. INSERT INTO album_and_musical_scene(album_id, musical_scene)
                    VALUES (0,1),
                    (1,1),
                    (2,1),
                    (2,9),
                    (2,10),
                    (3,3),
                    (3,6),
                    (4,5),
                    (5,5),
                    (6,4),
                    (7,2),
                    (8,5),
                    (9,7),
                    (10,8);
```

```
25. INSERT INTO ms_musician_instrument(fk_musician, fk_instrument,fk_ms)
                    VALUES(1, 2, 6),
                    (2, 1, 1),
                    (2, 1, 4),
                    (2, 1, 8),
                    (3, 9, 8),
                    (8, 1, 5),
                    (8, 8, 7),
                    (9, 7, 2),
                    (9, 7, 8),
                    (10, 1, 7);
```

```
26. SELECT count(*) as albums_sum FROM album where record_end between '+d1+'
AND '+d2+';
```

```
27. SELECT COUNT(*) AS song_count
 FROM(SELECT record_date
 FROM musical_scene
 WHERE song_id IN(SELECT fk_song_id
 FROM musical_scene_and_musician
 WHERE fk_musician_id = (SELECT musician_id
 FROM musicians
 WHERE  musician_name = '+ musicianName +'))) S
 WHERE S.record_date between + d1 + AND + d2+;
```

```
28. SELECT COUNT(*) AS albums_count
 FROM(SELECT record_begin
 FROM album
 WHERE album_id IN(SELECT album_id
 FROM album_and_musical_scene
 WHERE musical_scene IN(SELECT fk_song_id
 FROM musical_scene_and_musician
 WHERE fk_musician_id = (SELECT musician_id
 FROM musicians
 WHERE  musician_name = '+ musicianName +')))) S
 WHERE record_begin BETWEEN '+d1+' AND '+d2+';
```

```
29. SELECT instrument_type as popular_instrument FROM (SELECT fk_musician,
instrument_type
```

```
 FROM instrument AS i JOIN musicians_and_instruments as mi ON mi.fk_instrument =
i.instrument_id
 WHERE fk_musician IN (SELECT fk_musician_id FROM musical_scene AS ms
 JOIN musical_scene_and_musician as msi ON msi.fk_song_id = ms.song_id)) it
 GROUP by instrument_type ORDER BY COUNT(*) DESC LIMIT 1;

30. SELECT instrument_type
 FROM instrument as i
 JOIN musicians_and_instruments AS mi ON mi.fk_instrument = i.instrument_id
 JOIN musical_scene_and_musician as msm ON msm.fk_musician_id = mi.fk_musician
 JOIN album_and_musical_scene AS ams ON ams.musical_scene = msm.fk_song_id
 JOIN album as a ON a.album_id = ams.album_id
 WHERE a.album_name = '+album+';

31. SELECT producer_name, COUNT(producer_id) AS most_productive_pro
 FROM producer AS p
 JOIN album_n_producer AS ap ON p.producer_id = ap.fk_producer
 GROUP BY producer_id
 order by most_productive_pro DESC LIMIT 1;

32. SELECT manufacturer_name
 FROM instrument
 WHERE instrument_id = (SELECT fk_instrument
 FROM(SELECT fk_instrument, COUNT(*)
 FROM(SELECT fk_instrument
 FROM ms_musician_instrument
 WHERE fk_instrument IN(SELECT fk_instrument
 FROM ms_musician_instrument)
 GROUP BY fk_instrument)fk_ins)ins_cnt);

33. SELECT COUNT(DISTINCT fk_musician_id) AS recording_for_all_years
 FROM musical_scene_and_musician

34. SELECT musician_name FROM musicians
 WHERE musician_id = (SELECT fk_musician_id FROM(SELECT fk_musician_id
 FROM musical_scene_and_musician
 WHERE fk_song_id IN(SELECT fk_song_id
 FROM musical_scene_and_musician
 GROUP BY fk_song_id HAVING COUNT(*)>1
 ORDER BY COUNT(*) DESC))AS mi
 GROUP BY mi.fk_musician_id
 ORDER BY COUNT(*) DESC
 LIMIT 1);

35. SELECT genere,COUNT(*) AS most_pop_genere
 FROM musical_scene
 GROUP BY genere
 ORDER BY most_pop_genere DESC LIMIT 1;

36. SELECT technition_name
 FROM recording_technition
 WHERE technition_id = (SELECT fk_technition_id
 FROM musical_scene
 GROUP BY fk_technition_id HAVING COUNT(*)>1);

37. SELECT album_name, min(record_end)
 FROM album

38. SELECT song_name
 FROM musical_scene
 WHERE song_id IN(SELECT musical_scene
 FROM album_and_musical_scene
 GROUP BY musical_scene HAVING COUNT(*)>1);

39. SELECT technition_name
 FROM recording_technition
 WHERE technition_id IN(SELECT DISTINCT musical_scene.fk_technition_id
 FROM musical_scene
 WHERE musical_scene.song_id IN(SELECT DISTINCT musical_scene
 FROM album_and_musical_scene INNER JOIN(SELECT album_id
 FROM(SELECT album_id, COUNT(*)
 FROM(SELECT DISTINCT fk_technition_id, album_id
 FROM musical_scene
 JOIN album_and_musical_scene ON album_and_musical_scene.musical_scene =
musical_scene.song_id)a
 GROUP BY album_id
 HAVING COUNT(*) = 1)alb)ms ON album_and_musical_scene.album_id = ms.album_id));

40. SELECT musician_name
 FROM(SELECT distinct musician_id, musician_name, genere
 FROM musicians AS mn
 JOIN musical_scene_and_musician AS msm ON msm.fk_musician_id = mn.musician_id
 JOIN musical_scene AS ms ON ms.song_id = msm.fk_song_id) a
 GROUP BY musician_name having count(*)>1;
```