

IF2240 - Basis Data

Praktikum 3

DDL dan DML

21 Maret 2024

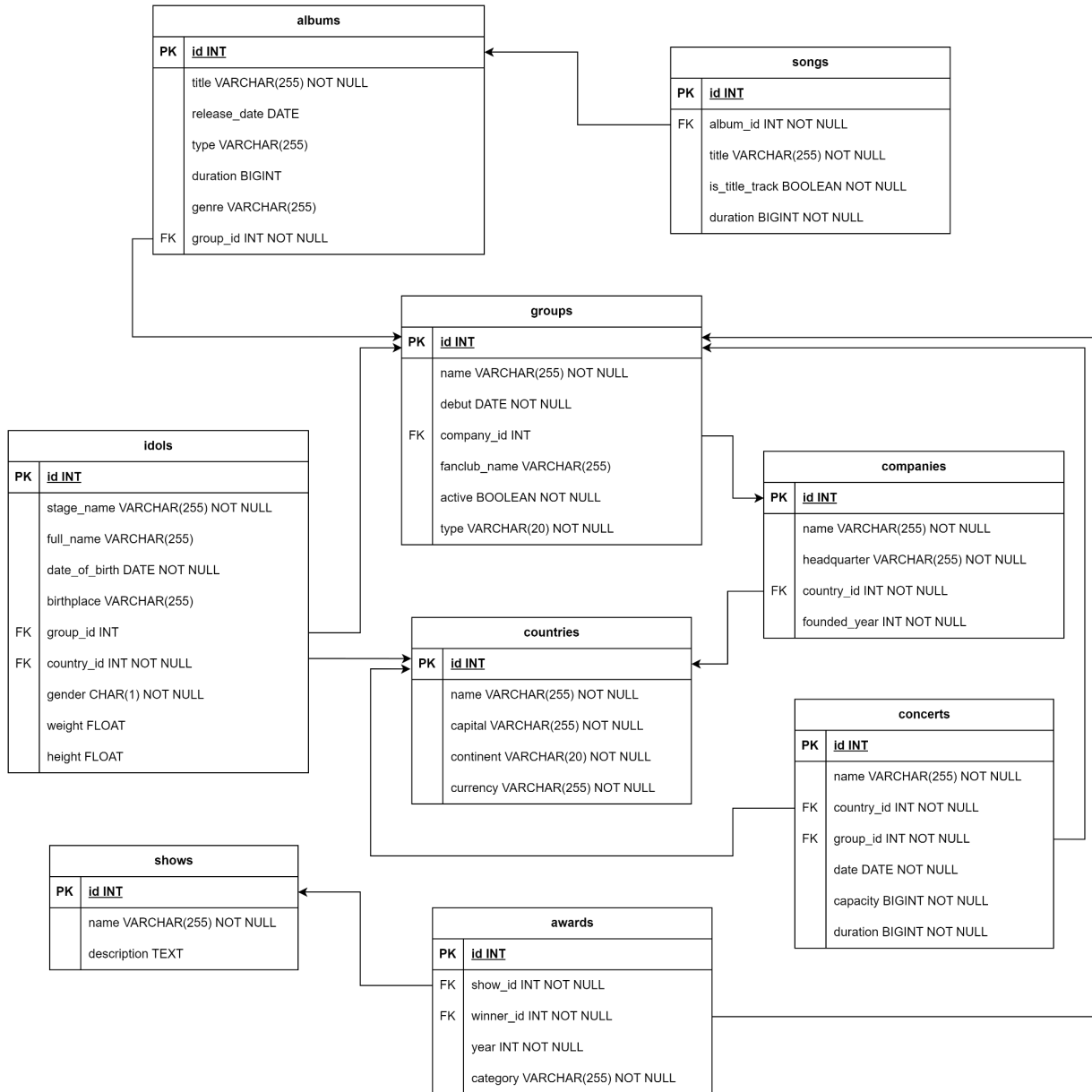


Erdianti Wiga Putri Andini

13522053

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2024**

K-Pop Fanbase



Legenda

No	Nama Tabel	Keterangan
1	albums	Album keluaran <i>Kpop group</i>
2	songs	Lagu <i>K-pop</i> setiap grup
3	groups	<i>K-pop group</i>
4	idols	Idola <i>K-pop</i>
5	companies	Perusahaan <i>agency K-pop</i>
6	countries	Negara
7	concerts	Konser <i>Kpop</i>
8	awards	Penghargaan untuk grup <i>K-pop</i> di setiap <i>shows</i>
9	shows	<i>K-pop music program</i>

Petunjuk

1. Praktikum ini wajib dikerjakan di dalam Laboratorium Teknik Informatika ITB.
2. Tidak ada restriksi penggunaan keyword pada praktikum ini. Anda dapat menggunakan keyword **apapun** untuk menjawab soal praktikum.
3. Terdapat 2 query yang digunakan pada setiap soal,
 - a. Query jawaban
Seluruh query yang Anda tulis pada bagian ini merupakan jawaban Anda untuk persoalan terkait. Query ini akan menjadi komponen penilaian dari praktikum ini.
 - b. Query testing
Query yang digunakan untuk membuktikan jawaban Anda. Query ini sudah disediakan oleh Tim Asisten. Anda hanya perlu menjalankannya pada perangkat Laboratorium untuk mendapatkan tangkapan layar sebelum dan sesudah Anda menjalankan query jawaban Anda.
 - c. Query panduan
Query yang digunakan untuk membantu Anda dalam memanipulasi data. Query ini sudah disediakan oleh Tim Asisten. Anda hanya perlu menjalankannya pada perangkat Laboratorium.

1. Kerja Sambilan

Database administrator kesayangan kalian, Julala, sedang kekurangan uang. Lantas, ia mengambil kerja sambilan untuk membuat *music videos* bagi para *idols*. Dengan begitu, ia ingin membuat sebuah tabel tambahan untuk mendaftarkan *music videos* apa saja yang dapat dibuatnya. Tabel tersebut dinamakan ***musicvideos*** dengan spesifikasi atribut sebagai berikut,

1. Atribut ***id***, bertipe *integer*, merupakan *primary key*,
2. Atribut ***song_id***, bertipe *integer*,
3. Atribut ***song***, bertipe *varchar* dengan batas maksimum 255 karakter,
4. Atribut ***duration***, bertipe *bigint*,
5. Atribut ***release_date***, bertipe tanggal, *nullable*,
6. Atribut ***idol_group_id***, bertipe *integer*, *nullable*, dan
7. Atribut ***idol_group***, bertipe *varchar* dengan batas maksimum 255 karakter, *nullable*.

Atribut yang tidak ditandai dengan *nullable* perlu ditambahkan *constraint* tidak boleh bernilai NULL. Perlu juga ditambahkan *constraint* sebagai berikut:

1. Atribut *idol_group_id* memiliki *foreign key constraint* ke atribut *id* pada relasi *groups*, dan
2. Atribut *song_id* memiliki *foreign key constraint* ke atribut *id* pada relasi *songs*.

Setelah tabel selesai dibuat, Julala memintamu melakukan *testing* untuk mengecek apakah tabel yang kamu buat sudah benar. Gunakan *query* ***show*** dan ***describe*** untuk menampilkan hasil kerjamu.

[HINT] Gunakan tambahan tag ***auto_increment*** sebagai fungsi initial default value pada primary key

Query Jawaban	<pre>CREATE TABLE musicvideos(id INT NOT NULL AUTO_INCREMENT, song_id INT NOT NULL,</pre>
---------------	--

	<pre> song VARCHAR(255) NOT NULL, duration BIGINT NOT NULL, release_date DATE, idol_group_id INT, idol_group VARCHAR(255), PRIMARY KEY(id), FOREIGN KEY(idol_group_id) REFERENCES groups(id), FOREIGN KEY(song_id) REFERENCES songs(id)); </pre>
Query Testing	<pre> show tables; desc musicvideos; </pre>

SS Query Sebelum Manipulasi Menggunakan Query Testing

```

MariaDB [prak3]> show tables;
+-----+
| Tables_in_prak3 |
+-----+
| albums          |
| awards          |
| companies       |
| concerts        |
| countries       |
| groups          |
| idols           |
| shows          |
| songs           |
+-----+
9 rows in set (0,001 sec)

```

SS Query Saat Manipulasi Menggunakan Query Jawaban

```

MariaDB [prak3]> CREATE TABLE musicvideos(
-> id INT NOT NULL AUTO_INCREMENT,
-> song_id INT NOT NULL,
-> song VARCHAR(255) NOT NULL,
-> duration BIGINT NOT NULL,
-> release_date DATE,
-> idol_group_id INT,
-> idol_group VARCHAR(255),
-> PRIMARY KEY(id),
-> FOREIGN KEY(idol_group_id) REFERENCES groups(id),
-> FOREIGN KEY(song_id) REFERENCES songs(id));
Query OK, 0 rows affected (0,021 sec)

```

SS Query Setelah Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> show tables;
```

```
+-----+  
| Tables_in_prak3 |  
+-----+  
| albums          |  
| awards          |  
| companies       |  
| concerts        |  
| countries       |  
| groups          |  
| idols           |  
| musicvideos     |  
| shows           |  
| songs           |  
+-----+
```

```
10 rows in set (0,001 sec)
```

```
MariaDB [prak3]> desc musicvideos;
```

```
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |  
| song_id    | int(11)       | NO   | MUL | NULL    |                |  
| song       | varchar(255)  | NO   |     | NULL    |                |  
| duration   | bigint(20)    | NO   |     | NULL    |                |  
| release_date | date         | YES  |     | NULL    |                |  
| idol_group_id | int(11)      | YES  | MUL | NULL    |                |  
| idol_group  | varchar(255)  | YES  |     | NULL    |                |  
+-----+-----+-----+-----+-----+-----+
```

```
7 rows in set (0,002 sec)
```

2. Kesalahan Anak Magang

Note: Kerjakan terlebih dahulu No. 1 untuk mengerjakan soal ini.

Pak Jay melihat bahwa seorang database administrator magang membuat sebuah tabel baru secara asal-asalan. Tabel *musicvideos* yang baru saja dibuat memiliki kolom yang seharusnya tidak diekspos, yaitu *duration*. Hal ini karena tipe data kolom tersebut masih salah. Bantulah Pak Jay menghapus kolom *duration* pada tabel *musicvideos*. Gunakan query ***describe*** sebelum dan sesudah penghapusan kolom.

Agar kesalahan anak magang tidak terulang, jawablah pertanyaan berikut:

- Jelaskan perbedaan antara ALTER TABLE MODIFY, ALTER TABLE RENAME, dan ALTER TABLE DROP dalam SQL!
- Sebutkan resiko terkait dengan penggunaan perintah ALTER TABLE DROP!

[HINT] Gunakan [ALTER TABLE](#) dan jangan lupa jawab soal teori

Jawaban Pertanyaan Teori	
a)	Alter table modify untuk mengubah tipe data suatu kolom/atribut. Alter table rename untuk mengubah nama kolom/atribut. Alter table drop untuk menghapus sesuatu tergantung apa yang mau dihapus. Misalkan alter table song drop column a, maka kolom a akan dihapus dari tabel song.
b)	Jika menggunakan alter table drop, bila kolom tersebut menjadi suatu foreign key untuk tabel lain, maka akan mengubah/berpengaruh error bagi atribut/kolom tabel tersebut.

Query Jawaban	ALTER TABLE musicvideos DROP COLUMN duration;
Query Testing	DESCRIBE musicvideos;

SS Query Sebelum Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> describe musicvideos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL    | auto_increment |
| song_id    | int(11)   | NO   | MUL | NULL    |              |
| song       | varchar(255) | NO   |     | NULL    |              |
| duration   | bigint(20) | NO   |     | NULL    |              |
| release_date | date      | YES  |     | NULL    |              |
| idol_group_id | int(11)   | YES  | MUL | NULL    |              |
| idol_group  | varchar(255) | YES  |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0,002 sec)
```

SS Query Saat Manipulasi Menggunakan Query Jawaban

```
MariaDB [prak3]> ALTER TABLE musicvideos DROP COLUMN duration;
Query OK, 0 rows affected (0,024 sec)
Records: 0 Duplicates: 0 Warnings: 0
```


SS Query Setelah Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> describe musicvideos;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | PRI | NULL    | auto_increment |
| song_id        | int(11)       | NO   | MUL | NULL    |                |
| song           | varchar(255)  | NO   |     | NULL    |                |
| release_date   | date          | YES  |     | NULL    |                |
| idol_group_id  | int(11)       | YES  | MUL | NULL    |                |
| idol_group     | varchar(255)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0,003 sec)
```

3. Security Patch

MyHand ditangkap basah seorang pakar IT, Ard Neham, saat MyHand memasukkan gender pada tabel Idols selain 'M' atau 'F'. Perbaiki atribut gender agar hanya dapat diisi menggunakan 'M' atau 'F' saja!

[HINT] Tipe data ENUM(string1, ...) memastikan bahwa nilainya adalah salah satu dari string-string yang telah didefinisikan

Query Jawaban	ALTER TABLE idols MODIFY COLUMN gender ENUM('M', 'F');
Query Testing	INSERT INTO idols VALUES (3000, 'Antena', 'Athena', '2024-02-27', 'Timur Jauh', 1, 1, 'L', 1, 1);

SS Query Sebelum Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> INSERT INTO idols VALUES (3000, 'Antena', 'Athena', '2024-02-27', 'Timur Jauh', 1, 1, 'L', 1, 1);
Query OK, 1 row affected (0,011 sec)
```

```
MariaDB [prak3]> SELECT DISTINCT gender FROM idols;
+-----+
| gender |
+-----+
| M      |
| F      |
| L      |
+-----+
3 rows in set (0,004 sec)
```

```
MariaDB [prak3]> describe idols;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| stage_name | varchar(255)  | NO   |     | NULL    |                |
| full_name  | varchar(255)  | YES  |     | NULL    |                |
| date_of_birth | date         | NO   |     | NULL    |                |
| birthplace | varchar(255)  | YES  |     | NULL    |                |
| group_id   | int(11)       | YES  | MUL | NULL    |                |
| country_id | int(11)       | NO   | MUL | NULL    |                |
| gender     | char(1)       | YES  |     | NULL    |                |
| weight     | float         | YES  |     | NULL    |                |
| height     | float         | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0,001 sec)
```

SS Query Saat Manipulasi Menggunakan Query Jawaban

```
MariaDB [prak3]> DELETE FROM idols WHERE id = 3000;
Query OK, 1 row affected (0,008 sec)
```

```
MariaDB [prak3]> ALTER TABLE idols MODIFY COLUMN gender ENUM('M', 'F');
Query OK, 1945 rows affected (0,080 sec)
Records: 1945 Duplicates: 0 Warnings: 0
```

SS Query Setelah Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> INSERT INTO idols VALUES (3000, 'Antena', 'Athena', '2024-02-27',
, 'Timur Jauh', 1, 1, 'L', 1, 1);
ERROR 1265 (01000): Data truncated for column 'gender' at row 1
```

```
MariaDB [prak3]> describe idols;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
stage_name	varchar(255)	NO		NULL	
full_name	varchar(255)	YES		NULL	
date_of_birth	date	NO		NULL	
birthplace	varchar(255)	YES		NULL	
group_id	int(11)	YES	MUL	NULL	
country_id	int(11)	NO	MUL	NULL	
gender	enum('M','F')	YES		NULL	
weight	float	YES		NULL	
height	float	YES		NULL	

10 rows in set (0,002 sec)

4. Akses Idol

Perusahaan MNH ingin mendapatkan data dari semua idol, namun setiap perusahaan hanya diperbolehkan untuk mengakses idol yang tergabung di grup yang dimiliki oleh perusahaan. Data idol yang diperlukan oleh perusahaan adalah nama panggung, nama lengkap, tanggal lahir, dan nama grup idol tersebut bergabung.

1. Sebagai database admin, bantulah menerapkan aturan tersebut pada pengaksesan data oleh *company* **MNH** dengan membuat **view** dengan nama '*IdolMNH*' sesuai dengan spesifikasi!
2. Sebagai database admin, apakah memberikan view kepada user lebih baik daripada memberikan seluruh tabel hanya dengan query kepada user? Jelaskan alasan Anda.

JAWABAN TEORI

Menurut saya lebih baik menggunakan view untuk memberikan akses kepada user daripada memberi seluruh tabel dengan query. Karena pada intinya seorang user hanya membutuhkan atribut-atribut yang dia mau dan lebih mudah untuk dicerna. Selain itu juga karena user merupakan seorang yang

diluar database, sehingga cukup diberikan akses sebagian terhadap skema basis data yang tersimpan.

Query Jawaban	<pre>CREATE VIEW IdolMNH AS (SELECT i.stage_name, i.full_name, i.date_of_birth, g.name FROM idols i, groups g, companies c WHERE i.group_id = g.id AND g.company_id = c.id AND c.name = "MNH");</pre>
Query Testing	<pre>DESCRIBE IdolMNH; SELECT * FROM IdolMNH;</pre>
SS Query Sebelum Manipulasi Menggunakan Query Testing	
<pre>MariaDB [prak3]> DESCRIBE IdolMNH; ERROR 1146 (42S02): Table 'prak3.IdolMNH' doesn't exist MariaDB [prak3]> SELECT * FROM IdolMNH; ERROR 1146 (42S02): Table 'prak3.IdolMNH' doesn't exist</pre>	
SS Query Saat Manipulasi Menggunakan Query Jawaban	
<pre>MariaDB [prak3]> CREATE VIEW IdolMNH AS -> (SELECT i.stage_name, i.full_name, i.date_of_birth, g.name -> FROM idols i, groups g, companies c -> WHERE i.group_id = g.id -> AND g.company_id = c.id -> AND c.name = "MNH"); Query OK, 0 rows affected (0,015 sec)</pre>	
SS Query Setelah Manipulasi Menggunakan Query Testing	

```

MariaDB [prak3]> DESCRIBE IdolMNH;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| stage_name     | varchar(255)  | NO   |     | NULL    |       |
| full_name      | varchar(255)  | YES  |     | NULL    |       |
| date_of_birth  | date          | NO   |     | NULL    |       |
| name           | varchar(255)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0,003 sec)

MariaDB [prak3]> SELECT * FROM IdolMNH;
+-----+-----+-----+-----+
| stage_name | full_name      | date_of_birth | name      |
+-----+-----+-----+-----+
| Haemin     | Jung Haemin    | 2004-03-16    | 8TURN     |
| Kyungmin   | Jo Kyungmin    | 2004-10-28    | 8TURN     |
| Minho      | Yang Minho     | 2002-10-14    | 8TURN     |
| Myunggho   | Ji Myunggho    | 2001-10-12    | 8TURN     |
| Seungheon  | Lee Seungheon  | 2007-05-15    | 8TURN     |
| Yoonsung   | Cho Yoonsung   | 2003-11-13    | 8TURN     |
| Yungyu     | Lee Yungyu     | 2005-11-07    | 8TURN     |
| Jaeyun     | Moon Jaeyun    | 2002-07-02    | 8TURN     |
| Jungwoo    | Uhm Jungwoo    | 1999-04-02    | BVNDIT    |
| Seungeun   | Shim Seungeun  | 2000-12-27    | BVNDIT    |
| Simyeong   | Lee Simyeong   | 1999-05-27    | BVNDIT    |
| Songhee    | Yoon Songhee   | 1998-11-08    | BVNDIT    |
| Yiyeon     | Jung Dasol     | 1995-05-28    | BVNDIT    |
+-----+-----+-----+-----+
13 rows in set (0,003 sec)

```

5. Info Audisi

Member-member dari *fanbase K-pop*, terutama anak laki-laki pak Jay, ingin sekali mendaftar audisi di agensi-agensi ternama. Namun, ia tidak tahu cara untuk menghubungi agensi (*company*) tersebut. Sebagai member setia fanbase K-Pop ia menemukan bahwa e-mail dari setiap agensi memiliki pola yang sama:

“<company_name><founded_year>@gmail.com”

Bantulah anak pak Jay dengan menambahkan **kolom `email`** pada tabel **companies** dengan tipe data varchar(255) dan nilai default NULL. Lalu **ubah semua nilai kolom email** dengan pola di atas.

Buktikan dengan screenshot eksekusi query penambahan kolom **`email`** dan isi dari kolom tersebut sebelum & setelah dilakukan perubahan!

[HINT] Pastikan alamat email tidak mengandung spasi dan semuanya huruf kecil. Gunakan fungsi REPLACE(), LOWER(), CONCAT(), dan LIMIT()

Query Jawaban (Tambah Kolom)	ALTER TABLE companies ADD COLUMN email VARCHAR(255) DEFAULT NULL;
Query Jawaban (Ubah Kolom)	UPDATE companies SET email = REPLACE(LOWER(CONCAT(name, founded_year, "@gmail.com")), " ", "");
Query Testing	SELECT * FROM companies LIMIT 5;

SS Query Sebelum Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> SELECT *
-> FROM companies
-> LIMIT 5;
```

id	name	headquarter	country_id	founded_year
1	101 Doors	Seoul	91	2012
2	10x	Seoul	91	2020
3	131 Online	Seoul	91	2021
4	143	Seoul	91	2020
5	153/Joombas Music Group	Hollywood, New York, Seoul	91	2012

```
5 rows in set (0,001 sec)
```

SS Query Saat Manipulasi Menggunakan Query Jawaban

```
MariaDB [prak3]> ALTER TABLE companies ADD COLUMN email VARCHAR(255) DEFAULT NULL;
Query OK, 0 rows affected (0,018 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [prak3]> UPDATE companies SET email = REPLACE(LOWER(CONCAT(name, founded_year, "@gmail.com")), " ", "");
Query OK, 1125 rows affected (0,024 sec)
Rows matched: 1125  Changed: 1125  Warnings: 0
```

SS Query Setelah Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> SELECT * FROM companies LIMIT 5;
```

id	name	headquarter	country_id	founded_year	email
1	101 Doors	Seoul	91	2012	NULL
2	10x	Seoul	91	2020	NULL
3	131 Online	Seoul	91	2021	NULL
4	143	Seoul	91	2020	NULL
5	153/Jombas Music Group	Hollywood, New York, Seoul	91	2012	NULL

5 rows in set (0,001 sec)

```
MariaDB [prak3]> SELECT *
-> FROM companies
-> LIMIT 5;
```

id	name	headquarter	country_id	founded_year	email
1	101 Doors	Seoul	91	2012	101doors2012@gmail.com
2	10x	Seoul	91	2020	10x2020@gmail.com
3	131 Online	Seoul	91	2021	131online2021@gmail.com
4	143	Seoul	91	2020	1432020@gmail.com
5	153/Jombas Music Group	Hollywood, New York, Seoul	91	2012	153/jombasmusicgroup2012@gmail.com

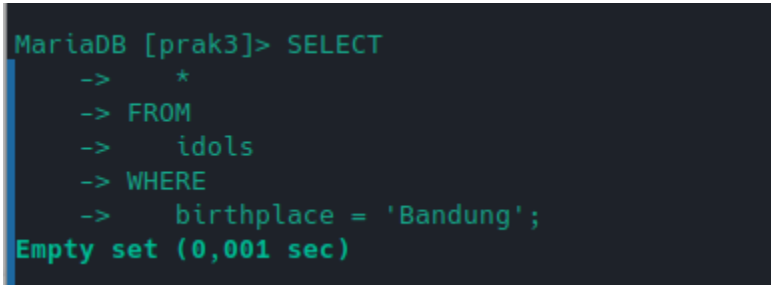
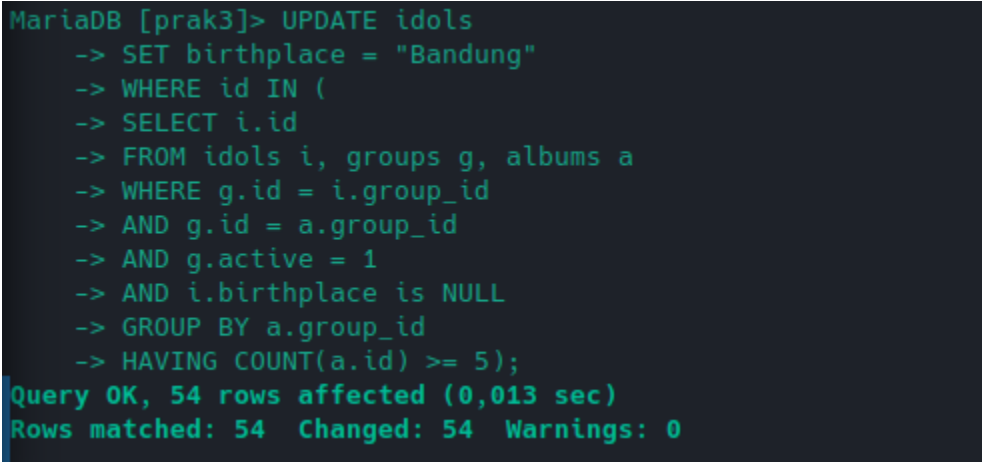
5 rows in set (0,001 sec)

6. Idol Kelahiran Bandung

Pak Cello seorang *database administrator* asal Bandung sangat ingin agar Bandung dapat dikenal oleh lebih banyak orang. Oleh karena itu Pak Cello menemukan ide yang cemerlang yang dapat membuat Bandung lebih terkenal. Bantulah Pak Cello untuk membuatkan *query* yang mengubah tempat lahir idol-idol yang tempat lahirnya tidak diketahui serta berasal dari group aktif yang memiliki album lebih besar atau sama dengan 5 menjadi Bandung!.

Buktikan dengan screenshot eksekusi *query* yang menunjukkan data idol-idol yang bertempat lahir di Bandung sebelum & sesudah dilakukan perubahan!.

Query Jawaban	<pre>UPDATE idols SET birthplace = "Bandung" WHERE id IN (SELECT i.id</pre>
---------------	--

	<pre> FROM idols i, groups g, albums a WHERE g.id = i.group_id AND g.id = a.group_id AND g.active = 1 AND i.birthplace is NULL GROUP BY a.group_id HAVING COUNT(a.id) >= 5); </pre>
Query Testing	<pre> SELECT * FROM idols WHERE birthplace = 'Bandung'; </pre>
SS Query Sebelum Manipulasi Menggunakan Query Testing	
 <pre> MariaDB [prak3]> SELECT -> * -> FROM -> idols -> WHERE -> birthplace = 'Bandung'; Empty set (0,001 sec) </pre>	
SS Query Saat Manipulasi Menggunakan Query Jawaban	
 <pre> MariaDB [prak3]> UPDATE idols -> SET birthplace = "Bandung" -> WHERE id IN (-> SELECT i.id -> FROM idols i, groups g, albums a -> WHERE g.id = i.group_id -> AND g.id = a.group_id -> AND g.active = 1 -> AND i.birthplace is NULL -> GROUP BY a.group_id -> HAVING COUNT(a.id) >= 5); Query OK, 54 rows affected (0,013 sec) Rows matched: 54 Changed: 54 Warnings: 0 </pre>	
SS Query Setelah Manipulasi Menggunakan Query Testing	

MariaDB [prak3]> SELECT

```
--> *
--> FROM
--> idols
--> WHERE
--> birthplace = 'Bandung';
```

id	stage_name	full_name	date_of_birth	birthplace	group_id	country_id	gender	weight	height
5	Ace	Jang Wooyoung	1992-08-28	Bandung	411	91	M	63	177
17	Aisha	Heo Yoorim	2000-07-21	Bandung	143	91	F	NULL	NULL
24	Ara	Lee Yunji	2002-01-23	Bandung	211	91	F	NULL	158
33	Arthur	Jang Yunho	2000-04-15	Bandung	232	91	M	57	176
35	Ayden	Kwon Yejun	2005-01-24	Bandung	141	91	M	55	174
44	Bae	Bae Jinsol	2004-12-28	Bandung	293	91	F	NULL	NULL
50	Bain	Song Byeonghee	2001-05-04	Bandung	225	91	M	NULL	NULL
59	Beomgyu	Choi Beomgyu	2001-03-13	Bandung	401	91	M	NULL	NULL
64	Bic	Nam Seungmin	2001-04-25	Bandung	265	91	M	NULL	NULL
74	Boeun	Park Boeun	2008-02-11	Bandung	108	91	F	NULL	155
79	Bomin	Choi Bomin	2000-08-24	Bandung	174	91	M	NULL	NULL
84	Bora	Kim Bora	1999-03-03	Bandung	101	91	F	NULL	NULL
95	Chaebin	Choi Yoobin	1999-07-28	Bandung	287	91	F	NULL	NULL
98	Chaehyun	Kim Chaehyun	2002-04-26	Bandung	230	91	F	NULL	160
99	Chaein	Lee Chaeyoung	2002-12-05	Bandung	321	91	F	NULL	NULL
103	Chael	Lim Junyeop	2001-12-07	Bandung	251	91	M	NULL	186
114	ChaeY	Song Chaeyeon	2003-01-06	Bandung	114	91	F	NULL	NULL
116	Chaeyeon	Kim Chaeyeon	2004-12-04	Bandung	394	91	F	NULL	NULL
131	Chani	Kang Chanhee	2000-01-17	Bandung	345	91	M	NULL	NULL
134	Chanwon	An Chanwon	2008-03-21	Bandung	381	91	M	NULL	NULL
155	Chris	Han Chris	2002-11-19	Bandung	76	91	M	NULL	NULL
166	D1	Jang Dongil	1998-02-15	Bandung	129	91	M	59	174
178	Dahyun	Jeong Dahyun	2005-04-29	Bandung	331	91	F	NULL	160
184	Dajeong	Jung Dajeong	2003-07-31	Bandung	312	91	F	42	157
231	Dohoon	Kim Dohoon	2005-01-30	Bandung	400	91	M	NULL	NULL
242	Dongheon	Lee Dongheon	1995-08-04	Bandung	412	91	M	NULL	NULL
252	Dongpyo	Son Dongpyo	2002-09-09	Bandung	271	91	M	48	166
259	Doryun	Kim Doryun	2000-05-01	Bandung	29	91	M	NULL	NULL
265	Doyoung	Kim Doyoung	2003-12-04	Bandung	389	91	M	NULL	NULL

166	D1	Jang Dongil	1998-02-15	Bandung	129	91	M	59	174
178	Dahyun	Jeong Dahyun	2005-04-29	Bandung	331	91	F	NULL	160
184	Dajeong	Jung Dajeong	2003-07-31	Bandung	312	91	F	42	157
231	Dohoon	Kim Dohoon	2005-01-30	Bandung	400	91	M	NULL	NULL
242	Dongheon	Lee Dongheon	1995-08-04	Bandung	412	91	M	NULL	NULL
252	Dongpyo	Son Dongpyo	2002-09-09	Bandung	271	91	M	48	166
259	Doryun	Kim Doryun	2000-05-01	Bandung	29	91	M	NULL	NULL
265	Doyoung	Kim Doyoung	2003-12-04	Bandung	389	91	M	NULL	NULL
276	Eden	Jae Won	1998-03-27	Bandung	292	91	M	NULL	NULL
283	Eric	Son Youngjae	2000-12-22	Bandung	376	91	M	56	171
300	Eunchan	Choi Byeongseop	2001-02-27	Bandung	374	91	M	NULL	184
306	Eunil	Kim Eunil	2003-11-21	Bandung	391	91	M	NULL	NULL
307	Eunil	Kim Eunil	2003-11-21	Bandung	392	91	M	NULL	NULL
321	Eunseok	Song Eunseok	2001-03-19	Bandung	329	91	M	57	180
377	Gyurae	Kim Gyurae	2009-02-28	Bandung	151	91	M	NULL	183
383	Haechan	Lee Donghyuk	2000-06-06	Bandung	288	91	M	NULL	NULL
423	Hannah	Kang Minkyung	2003-01-04	Bandung	436	91	F	NULL	NULL
427	Haram	Kim Haram	2001-01-13	Bandung	66	91	F	45	161
463	Heeseung	Lee Heeseung	2001-10-15	Bandung	139	91	M	NULL	NULL
480	Hongjoong	Kim Hongjoong	1998-11-07	Bandung	48	91	M	NULL	171
495	Hwanwoong	Yeo Hwanwoong	1998-08-26	Bandung	304	91	M	NULL	NULL
521	Hyeongjun	Song Hyeongjun	2002-11-30	Bandung	113	91	M	55	174
554	Hyunbin	Kim Hyunbin	2004-05-26	Bandung	393	91	F	NULL	NULL
583	Hyunsuk	Yoon Hyunsuk	2001-09-08	Bandung	107	91	M	NULL	188
605	Intak	Hwang Intak	2003-08-31	Bandung	308	91	M	NULL	NULL
626	Jaehee	Lee Jaehee	2004-03-18	Bandung	427	91	F	NULL	166
634	Jaehyun	Myung Jaehyun	2003-12-04	Bandung	85	91	M	NULL	NULL
738	Jinwoo	Lee Jinwoo	2004-09-13	Bandung	166	91	M	57	176
976	Lucy	Kim Daeun	2004-04-09	Bandung	434	91	F	NULL	NULL
1008	Minho	Yang Minho	2002-10-14	Bandung	17	91	M	59	179
1054	Miyeon	Jo Miyeon	1997-01-31	Bandung	2	91	F	NULL	NULL
1111	On	Kim Dongwoo	1969-12-31	Bandung	24	91	M	65	177
1202	Seline	Jung Yeonjeong	2000-06-20	Bandung	105	91	F	NULL	NULL

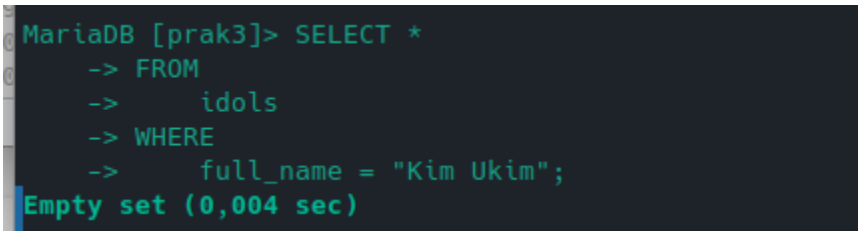
54 rows in set (0.004 sec)

7. Idol Jalur Ordal

Bu Didot adalah seorang database administrator yang ingin anak laki-laki nya bisa menjadi idol. Oleh karena itu, Bu Didot berencana untuk memasukkan data diri anaknya ke dalam tabel idol. Anak Bu Didot bernama Kim Ukim dengan stage name Real, namun ia belum memiliki group debut. Ia lahir di Surabaya pada tanggal 7 April 2003 dan berasal dari negara dengan id 78. Kim Ukim memiliki badan yang proporsional dengan tinggi 180 dan berat 74.

Bantulah Bu Didot untuk menjadikan anaknya sebagai idol!

[HINT] Gunakan CONCAT() untuk melakukan konkatenasi string.

Query Jawaban	<pre>INSERT INTO idols VALUES (1946, 'Real', 'Kim Ukim', '2003-04-07', 'Surabaya', NULL, 78, 'M', 74, 180);</pre>
Query Testing	<pre>SELECT * FROM idols WHERE full_name = "Kim Ukim";</pre>
SS Query Sebelum Manipulasi Menggunakan Query Testing	
	
SS Query Saat Manipulasi Menggunakan Query Jawaban	

```
MariaDB [prak3]> INSERT INTO idols VALUES (1946, "Real", "Kim Ukim", "2003-04-07", "Surabaya", NULL, 78, 'M', 74, 180);
Query OK, 1 row affected (0,007 sec)
```

SS Query Setelah Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> SELECT *
-> FROM
-> idols
-> WHERE
-> full_name = "Kim Ukim";
```

id	stage_name	full_name	date_of_birth	birthplace	group_id	country_id	gender	weight	height
1946	Real	Kim Ukim	2003-04-07	Surabaya	NULL		78 M	74	180

```
1 row in set (0,004 sec)
```

8. Semena-mena

Note: Kerjakan terlebih dahulu No. 1 dan No. 2 untuk mengerjakan soal ini.

Tidak kapok dimarahi atasan atas kesalahan yang dibuatnya, *database administrator* kesayangan kita, Julala, kembali berulah. Dalam tabel *musicvideos* yang telah dibuatnya, ia hanya memasukkan *girl group* yang telah konser setidaknya 10 kali. Julala sudah membuat *query* untuk mendapatkan ID dari seluruh *girl group* tersebut. Berikut adalah *query* yang ia digunakan:

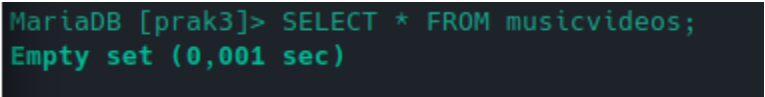
```
SELECT g.id
FROM groups AS g, concerts AS c
WHERE g.id = c.group_id
AND g.type = 'girl groups'
GROUP BY g.id
HAVING COUNT(*) >= 10;
```

Bantu Julala memasukkan seluruh data lagu dari grup-grup tersebut ke dalam tabel *musicvideos* sesuai dengan atribut yang dibutuhkan. Untuk atribut *release_date*, asumsikan *music video* akan rilis 6 bulan setelah data dimasukkan ke dalam tabel.

[HINT] Berikut adalah beberapa fungsi yang dapat kalian gunakan dalam mengerjakan soal ini:

- 1. Gunakan fungsi NOW() untuk mendapatkan waktu sekarang (nilai DATE ketika kalian menggunakan fungsi tersebut).*
- 2. Gunakan fungsi DATE_ADD('DATE', INTERVAL XX DATETYPE) untuk melakukan penambahan pada sebuah nilai DATE.*

e.g. : DATE_ADD('2024-02-04', INTERVAL 2 MONTH) akan mengembalikan nilai '2024-04-04'.

Query Jawaban	
Query Testing	SELECT * FROM musicvideos;
SS Query Sebelum Manipulasi Menggunakan Query Testing	
	
SS Query Saat Manipulasi Menggunakan Query Jawaban	
SS Query Setelah Manipulasi Menggunakan Query Testing	

9. Deprecated Album

Bu Didot telah bosan menjadi database administrator sehingga sekarang ia adalah seorang manajer dari suatu KPOP group. Ia merasa salah satu album musik berjudul CAPTURE THE FLAG memiliki rahasia kelam sehingga akan berbahaya jika diedarkan di pasaran. Oleh karena itu, ia ingin menghapus album tersebut beserta seluruh lagu di dalamnya dari database. Bantulah Bu Didot untuk melakukan hal tersebut.

Query Jawaban (hapus lagu)	DELETE FROM songs WHERE album_id = 1;
Query Jawaban (Hapus album)	DELETE FROM albums WHERE title = "CAPTURE THE FLAG";
Query Testing	<pre>SELECT * FROM albums WHERE title = "CAPTURE THE FLAG"; SELECT * FROM songs WHERE album_id = 1;</pre>

SS Query Sebelum Manipulasi Menggunakan Query Testing

```
MariaDB [prak3]> SELECT * FROM
->   albums
-> WHERE
->   title = "CAPTURE THE FLAG";
+-----+-----+-----+-----+-----+-----+
| id | title           | release_date | type       | duration | genre | group_id |
+-----+-----+-----+-----+-----+-----+
| 1 | CAPTURE THE FLAG | 2024-01-30   | Mini Album | 1086     | Pop   | 409      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,001 sec)
```

```
MariaDB [prak3]> SELECT * FROM
->   songs
-> WHERE
->   album_id = 1;
+-----+-----+-----+-----+-----+
| id | album_id | title           | is_title_track | duration |
+-----+-----+-----+-----+-----+
| 1 | 1        | JACKPOT         | 1              | 158      |
| 2 | 1        | PONYTAIL (KOR ver.) | 0              | 169      |
| 3 | 1        | After Party     | 0              | 160      |
| 4 | 1        | Be My Love      | 0              | 205      |
| 5 | 1        | CIRCU           | 0              | 195      |
| 6 | 1        | Across The Stars | 0              | 196      |
+-----+-----+-----+-----+-----+
6 rows in set (0,000 sec)
```

SS Query Saat Manipulasi Menggunakan Query Jawaban	
<pre>MariaDB [prak3]> DELETE FROM songs WHERE album_id = 1; Query OK, 6 rows affected (0,052 sec)</pre> <pre>MariaDB [prak3]> DELETE FROM albums WHERE title = "CAPTURE THE FLAG"; Query OK, 1 row affected (0,008 sec)</pre>	
SS Query Setelah Manipulasi Menggunakan Query Testing	
<pre>MariaDB [prak3]> DELETE FROM albums WHERE title = "CAPTURE THE FLAG"; Query OK, 1 row affected (0,008 sec)</pre> <pre>MariaDB [prak3]> SELECT * FROM -> albums -> WHERE -> title = "CAPTURE THE FLAG"; Empty set (0,000 sec)</pre> <pre>MariaDB [prak3]> SELECT * FROM -> songs -> WHERE -> album_id = 1; Empty set (0,001 sec)</pre>	

10. Yang Korea Korea Aja (YKKA)

Ailbert ingin melakukan penyederhanaan data pada basis data K-Pop Fanbase. Ia hanya ingin berfokus pada data-data company yang berasal dari Korea Selatan saja (ditandai dengan nama negara 'South Korea'). Bantulah Ailbert untuk melakukan hal tersebut dengan langkah langkah berikut,

- Buat tabel baru dengan nama `companies_korea` yang berisikan seluruh data yang ada pada tabel `companies`.

Query Panduan (Diberikan kepada Praktikan)	<pre>CREATE TABLE `companies_korea` (`id` int(11) NOT NULL, `name` varchar(255) NOT NULL, `headquarter` varchar(255) NOT NULL, `country_id` int(11) NOT NULL, `founded_year` int(11) NOT NULL,</pre>
--	---

	<pre>PRIMARY KEY (`id`), KEY `country_id` (`country_id`), CONSTRAINT `companiesk_ibfk_1` FOREIGN KEY (`country_id`) REFERENCES `countries` (`id`)); INSERT INTO companies_korea SELECT id, name, headquarter, country_id, founded_year FROM companies;</pre>
--	--

SS Query Saat Manipulasi Menggunakan Query Panduan

```
MariaDB [prak3]> CREATE TABLE `companies_korea` (
-> `id` int(11) NOT NULL,
-> `name` varchar(255) NOT NULL,
-> `headquarter` varchar(255) NOT NULL,
-> `country_id` int(11) NOT NULL,
-> `founded_year` int(11) NOT NULL,
-> PRIMARY KEY (`id`),
-> KEY `country_id` (`country_id`),
-> CONSTRAINT `companiesk_ibfk_1` FOREIGN KEY (`country_id`) REFERENCES `co
untries` (`id`)
-> );
Query OK, 0 rows affected (0,014 sec)

MariaDB [prak3]> INSERT INTO companies_korea SELECT id, name, headquarter, countr
y_id, founded_year FROM companies;
Query OK, 1125 rows affected (0,026 sec)
Records: 1125 Duplicates: 0 Warnings: 0
```

- b. Pastikan tabel `companies_korea` sudah tersedia. Selanjutnya, pada tabel `companies_korea`, hapus seluruh company yang bukan berasal dari Korea Selatan

Query Jawaban	<pre>DELETE FROM companies_korea WHERE id IN (SELECT ck.id FROM companies_korea ck INNER JOIN countries ct ON ck.country_id = ct.id WHERE ct.name <> 'South Korea');</pre>
Query Testing (Diberikan kepada Praktikan)	<pre>SELECT COUNT(*) FROM companies_korea ck INNER JOIN countries ct ON ck.country_id = ct.id WHERE ct.name <> 'South Korea';</pre>
SS Query Sebelum Manipulasi Menggunakan Query Testing	

```

MariaDB [prak3]> SELECT COUNT(*)
-> FROM companies_korea ck
->     INNER JOIN countries ct ON ck.country_id = ct.id
-> WHERE ct.name <> 'South Korea';
+-----+
| COUNT(*) |
+-----+
|      194 |
+-----+
1 row in set (0,001 sec)

```

SS Query Saat Manipulasi Menggunakan Query Jawaban

```

MariaDB [prak3]> DELETE FROM companies_korea
-> WHERE id IN
-> (SELECT ck.id
-> FROM companies_korea ck
->     INNER JOIN countries ct ON ck.country_id = ct.id
-> WHERE ct.name <> 'South Korea');
Query OK, 194 rows affected (0,014 sec)

```

SS Query Setelah Manipulasi Menggunakan Query Testing

```

MariaDB [prak3]> SELECT COUNT(*)
-> FROM companies_korea ck
->     INNER JOIN countries ct ON ck.country_id = ct.id
-> WHERE ct.name <> 'South Korea';
+-----+
| COUNT(*) |
+-----+
|         0 |
+-----+
1 row in set (0,001 sec)

```