



Zoals we hierboven zien, is er na het aanmaken van de database nog geen enkele tabel aangemaakt.

```
1 DROP TABLE IF EXISTS television;
2 DROP TABLE IF EXISTS remotecontroller;
3 DROP TABLE IF EXISTS cimodule;
4 DROP TABLE IF EXISTS wallbracket;
5
```

Je gaat query's schrijven voor het aanmaken van de tabellen voor television, remotecontroller, cimodule en wallbracket. Hiervoor is het noodzakelijk dat deze tabellen niet al bestaan in je database en als ze wel bestaan, je deze eerst verwijdert door bovenstaande query's.

```

5
6 CREATE TABLE television(
7   id INT GENERATED ALWAYS AS IDENTITY,
8   name TEXT NOT NULL UNIQUE,
9   brand TEXT NOT NULL UNIQUE,
10  price DOUBLE PRECISION NOT NULL,
11  currentStock INT NOT NULL,
12  sold INT NOT NULL,
13  typeTelevision TEXT NOT NULL,
14  availableSize INT NOT NULL,
15  refreshRate INT NOT NULL,
16  screenType TEXT NOT NULL
17 );
18

```

We hebben nu de query voor de tabel van de television geschreven.

```

18 |
19 CREATE TABLE remotecontroller(
20   id INT GENERATED ALWAYS AS IDENTITY,
21   name TEXT NOT NULL UNIQUE,
22   brand TEXT NOT NULL UNIQUE,
23   price DOUBLE PRECISION NOT NULL,
24   currentStock INT NOT NULL,
25   sold INT NOT NULL,
26   compatibleWith TEXT,
27   batteryType TEXT NOT NULL,
28   PRIMARY KEY (id)
29 );
30

```

```

30
31 CREATE TABLE cimodule(
32   id INT GENERATED ALWAYS AS IDENTITY,
33   name TEXT NOT NULL UNIQUE,
34   brand TEXT NOT NULL UNIQUE,
35   price DOUBLE PRECISION NOT NULL,
36   currentStock INT NOT NULL,
37   sold INT NOT NULL,
38   PRIMARY KEY (id)
39 );
40

```

```

40
41 CREATE TABLE wallbracket(
42   id INT GENERATED ALWAYS AS IDENTITY,
43   name TEXT NOT NULL UNIQUE,
44   brand TEXT NOT NULL UNIQUE,
45   price DOUBLE PRECISION NOT NULL,
46   currentStock INT NOT NULL,
47   sold INT NOT NULL,
48   adjustable BOOLEAN NOT NULL,
49   PRIMARY KEY (id)
50 );
51

```

Zo hebben we ook de query's voor de andere tabellen geschreven.

```

39 CREATE TABLE television(
40   id INT GENERATED ALWAYS AS IDENTITY,
41   name TEXT NOT NULL UNIQUE,
42   brand TEXT NOT NULL UNIQUE,
43   price DOUBLE PRECISION NOT NULL,
44   currentStock INT NOT NULL,
45   sold INT NOT NULL,
46   typeTelevision TEXT NOT NULL,
47   availableSize INT NOT NULL,
48   refreshRate INT NOT NULL,
49   screenType TEXT NOT NULL,
50   remote_id INT,
51   PRIMARY KEY (id),
52   CONSTRAINT fk_remotecontroller FOREIGN KEY (remote_id) REFERENCES remotecontroller(id)
53 );
54

```

Daarna hebben we de query van de television tabel moeten aanpassen voor de relatie met remotecontroller, omdat er anders geen relatie tussen de tabellen gelegd kan worden. Hieronder doen we dat voor de relatie met de tabel van de cimodule.

```

39 CREATE TABLE television(
40   id INT GENERATED ALWAYS AS IDENTITY,
41   name TEXT NOT NULL UNIQUE,
42   brand TEXT NOT NULL UNIQUE,
43   price DOUBLE PRECISION NOT NULL,
44   currentStock INT NOT NULL,
45   sold INT NOT NULL,
46   typeTelevision TEXT NOT NULL,
47   availableSize INT NOT NULL,
48   refreshRate INT NOT NULL,
49   screenType TEXT NOT NULL,
50   remote_id INT,
51   ci_module_id INT,
52   PRIMARY KEY (id),
53   CONSTRAINT fk_remotecontroller FOREIGN KEY (remote_id) REFERENCES remotecontroller (id),
54   CONSTRAINT fk_cimodule FOREIGN KEY (ci_module_id) REFERENCES cimodule (id)
55 );

```

Nu moeten we alleen nog de many-to-many relatie tussen de tabellen television en wallbracket plaatsen. Echter kunnen we maar één waarde in een cel opslaan. Dat dwingt ons om een koppeltabel aan te maken voor deze relatie. Deze noemen television_wallbracket. Dit doen we hieronder.

```
CREATE TABLE television_wallbracket(  
  television_id INT,  
  wallbracket_id INT,  
  CONSTRAINT fk_television FOREIGN KEY (television_id) REFERENCES television (id),  
  CONSTRAINT fk_wallbracket FOREIGN KEY (wallbracket_id) REFERENCES wallbracket (id)  
);|
```

Aangezien we hier weer een nieuwe tabel aanmaken, moeten we de drop table query bovenaan toevoegen.

```
1 DROP TABLE IF EXISTS television;  
2 DROP TABLE IF EXISTS remotecontroller;  
3 DROP TABLE IF EXISTS cimodule;  
4 DROP TABLE IF EXISTS wallbracket;  
5 DROP TABLE IF EXISTS television_wallbracket;|
```

Omdat we nu relaties hebben aangebracht op de tabellen, kunnen de tabellen niet verwijderd worden zonder eerst de relatie te verwijderen. Dit doen we met de volgende aanpassing:

```
1 DROP TABLE IF EXISTS television CASCADE;  
2 DROP TABLE IF EXISTS remotecontroller;  
3 DROP TABLE IF EXISTS cimodule;  
4 DROP TABLE IF EXISTS wallbracket CASCADE;  
5 DROP TABLE IF EXISTS television_wallbracket;|
```

En zo hebben we de complete opdracht in query's uitgeschreven en kunnen we alle tabellen aanmaken. Probeer nu zelf de tabellen te vullen met data, data te bewerken en data te verwijderen.