

```
CREATE TABLE myemployees(  
    employee_id SERIAL PRIMARY KEY NOT NULL,  
    firstname VARCHAR(15),  
    lastname VARCHAR(15),  
    title CHAR(50) NOT NULL DEFAULT "",  
    age INTEGER CHECK (age >= 0),  
    salary INTEGER CHECK (salary <= 100000)  
);
```

```
INSERT INTO myemployees (employee_id, firstname, lastname, title, age, salary)  
VALUES
```

```
(1,'Jonie', 'Weber', 'Secretary', 28, 19500),  
(2,'Potsy', 'Weber', 'Programmer', 32, 45300),  
(3,'Dirk', 'Smith', 'Programmer II', 45, 75020),  
(4,'Mike', 'Nicols', 'Programmer', 25, 35000),  
(5,'Jim', 'mith', 'Secretary', 24, 17000),  
(6,'Dean', 'Yeager', 'Programmer II', 39, 73000),  
(7,'Mark', 'Middleton',DEFAULT , 21, 10000);
```

```
SELECT * FROM myemployees;
```

```
SELECT * FROM myemployees WHERE salary < 30000;
```

```
SELECT firstname, lastname FROM myemployees WHERE title = 'Programmer';
```

```
SELECT * FROM myemployees WHERE lastname LIKE '%ebe%';
```

```
SELECT *FROM myemployees WHERE firstname ='Potsy';
```

```
SEIECT * FROM myemployees WHERE lastname LIKE '%ith';
```

```
UPDATE myemployees SET lastname = 'Williams' WHERE firstname = 'Jonie' AND  
lastname = 'Weber';
```

```
UPDATE myemployees SET age = age + 1 WHERE firstname = 'Dirk' AND lastname =  
'Smith';
```

```
UPDATE myemployees SET title = 'Administrative Assistant' WHERE title = 'Secretary';
```

```
UPDATE myemployees SET salary = salary + 3500 WHERE salary < 30000;
```

```
UPDATE myemployees SET salary = salary + 4500 WHERE salary < 33500;
```

```
UPDATE myemployees SET title ='Programmer III' WHERE title = 'Programmer II';
```

```
UPDATE myemployees SET title ='Programmer II' WHERE title = 'Programmer';
```

```
DELETE FROM myemployees WHERE lastname = 'Jonie' AND firstname = 'Williams';
```

```
DELETE FROM myemployees WHERE salary > 70000;
```

```
CREATE DATABASE music
```

```
WITH OWNER postgres
```

```
ENCODING 'UTF8';
```

```
CREATE TABLE album (  
    album_id SERIAL PRIMARY KEY,  
    title VARCHAR(255)  
);
```

```
CREATE TABLE artist (  
    artist_id SERIAL PRIMARY KEY,  
    name VARCHAR(255)  
);
```

```
CREATE TABLE track (  
    track_id SERIAL PRIMARY KEY,  
    title VARCHAR(100),  
    len INTEGER ,  
    rating INTEGER,  
    count INTEGER,  
    album_id INTEGER,  
    artist_id INTEGER,  
    FOREIGN KEY (album_id) REFERENCES album(album_id) ON DELETE  
    CASCADE,  
    FOREIGN KEY (artist_id) REFERENCES artist(artist_id) ON DELETE  
    CASCADE  
);
```

```
COPY album(title)  
FROM 'C:/csv_data/album.csv'  
DELIMITER ','  
CSV HEADER;
```

```
COPY artist(name)
FROM 'C:/csv_data/artist.csv'
DELIMITER ','
CSV HEADER;
```

```
COPY track(title, len, rating, count, album_id, artist_id)
FROM 'C:/csv_data/track.csv'
DELIMITER ','
CSV HEADER;
```

```
SELECT * FROM album;
```

```
SELECT * FROM artist;
```

```
SELECT * FROM track;
```