

## ACTIVITY 16

Maximillian White

- a) insert into artist(id, name) values (3, 'Maximillian White');

The screenshot shows the pgAdmin 4 interface on the left and a web browser on the right. In pgAdmin, the 'artist' table is selected, and its data is displayed in the 'Data output' pane. The 'Query' pane shows a SQL query: `SELECT * FROM public.artist ORDER BY id ASC`. The web browser displays a document titled 'ProgreSQL Activity' with a description and two tasks. The 'artist table' is shown with three rows: (1, 'Led Zeppelin'), (2, 'AC/DC'), and (3, 'Maximillian White'). The 'album table' is shown with three rows: (1, 'Who made who', 2), (2, 'IV', 1), and (3, 'NOVEMBER RAIN', 3). The 'Join display between artist and album' table shows the results of a join between the two tables.

**ProgreSQL Activity**

Description: Returning to the previous example, perform the following activities

a) Add your name in table artist as the third row/tuple.  
b) Add your favorite song, all in upper case, as the third tuple in table album. Remember that it links to the third tuple in artist table.

**artist table**

id	name
1	Led Zeppelin
2	AC/DC
3	Maximillian White

**album table**

id	title	artist_id
1	Who made who	2
2	IV	1
3	NOVEMBER RAIN	3

c) Display a join table between artist and album: display the column of title from album table, name from artist table, and artist id number:

**Join display between artist and album**

title	name	id
Who made who	AC/DC	2
IV	Led Zeppelin	1
NOVEMBER RAIN	Prof. Huixin Wu	3

- b) insert into album (id, title, artist\_id) values(3, 'Yhap', 3);

The screenshot shows the pgAdmin 4 interface on the left and a web browser on the right. In pgAdmin, the 'album' table is selected, and its data is displayed in the 'Data output' pane. The 'Query' pane shows a SQL query: `SELECT * FROM public.album ORDER BY id ASC`. The web browser displays a document titled 'progreSQLActi...' with a description and two tasks. The 'artist table' is shown with three rows: (1, 'Led Zeppelin'), (2, 'AC/DC'), and (3, 'Prof. Huixin Wu'). The 'album table' is shown with three rows: (1, 'Who made who', 2), (2, 'IV', 1), and (3, 'NOVEMBER RAIN', 3). The 'Join display between artist and album' table shows the results of a join between the two tables.

**progreSQLActi...**

a) Add your name in table artist as the third row/tuple.  
b) Add your favorite song, all in upper case, as the third tuple in table album. Remember that it links to the third tuple in artist table.

**artist table**

id	name
1	Led Zeppelin
2	AC/DC
3	Prof. Huixin Wu

**album table**

id	title	artist_id
1	Who made who	2
2	IV	1
3	NOVEMBER RAIN	3

c) Display a join table between artist and album: display the column of title from album table, name from artist table, and artist id number:

**Join display between artist and album**

title	name	id
Who made who	AC/DC	2
IV	Led Zeppelin	1
NOVEMBER RAIN	Prof. Huixin Wu	3

d) Display a join table between track and genre: display the column of title from the track table and name from the genre table:

c) select album.title, artist.name from album join artist on album.artist\_id = artist.id;

The screenshot shows the pgAdmin 4 interface on the left and a web browser on the right. In pgAdmin, the 'album' table is selected in the left pane, and the 'Query' tab is active. The query editor contains the following SQL statement:

```
1 select album.title, artist.name from album join
2
```

The 'Data output' tab shows the results of the query:

	title	name
1	Who made who	AC/DC
2	IV	Led Zeppelin
3	Yhapp	Maximilian White

The status bar indicates 'Total rows: 3 of 3' and 'Query complete 00:00:00.088'.

The web browser on the right displays a document titled 'progresq[Acti...' with the following content:

album table

id	title	artist_id
1	Who made who	2
2	IV	1
3	NOVEMBER RAIN	3

c) Display a join table between **artist** and **album**: display the column of **title** from album table, **name** from artist table, and artist id number:

Join display between artist and album

title	name	id
Who made who	AC/DC	2
IV	Led Zeppelin	1
NOVEMBER RAIN	Prof. Huixin Wu	3

d) Display a join table between **track** and **genre** table: display the column of **title** from the track table and **name** from the genre table:

title	name
Black Dog	Rock
Stairway	Rock
About to Rock	Metal
WhoMade who	Metal

d) select track.title, genre.name from track join genre on track.genre\_id = genre.id;

The screenshot shows the pgAdmin 4 interface on the left and a web browser on the right. In pgAdmin, the 'track' table is selected in the left pane, and the 'Query' tab is active. The query editor contains the following SQL statement:

```
1 track join genre on track.genre_id = genre.id;
```

The 'Data output' tab shows the results of the query:

	title	name
1	Black Dog	Rock
2	Stairway	Rock
3	About to Rock	Metal
4	Who Made Who	Metal

The status bar indicates 'Total rows: 4 of 4' and 'Query complete 00:00:00.165'.

The web browser on the right displays the same document as in the previous screenshot, showing the 'album table' and the results of the join query between artist and album tables.