

## Stage 2 431W Project

### Schema and DDL Commands

```
Tobi Ajayi=# \dt
```

List of relations			
Schema	Name	Type	Owner
public	air_dates	table	Tobi Ajayi
public	created_by_types	table	Tobi Ajayi
public	created_bys	table	Tobi Ajayi
public	genre_types	table	Tobi Ajayi
public	genres	table	Tobi Ajayi
public	language_types	table	Tobi Ajayi
public	languages	table	Tobi Ajayi
public	link_types	table	Tobi Ajayi
public	links	table	Tobi Ajayi
public	network_types	table	Tobi Ajayi
public	networks	table	Tobi Ajayi
public	origin_country_types	table	Tobi Ajayi
public	production_companies	table	Tobi Ajayi
public	production_company_types	table	Tobi Ajayi
public	production_countries	table	Tobi Ajayi
public	production_country_types	table	Tobi Ajayi
public	show_votes	table	Tobi Ajayi
public	shows	table	Tobi Ajayi
public	spoken_language_types	table	Tobi Ajayi
public	spoken_languages	table	Tobi Ajayi
public	status	table	Tobi Ajayi
public	types	table	Tobi Ajayi

(22 rows)

```

Tobi Ajayi=# CREATE TABLE production_company_types (
Tobi Ajayi(# productionCompanyId SERIAL PRIMARY KEY,
Tobi Ajayi(# producerName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE production_country_types (
Tobi Ajayi(# productionCountryTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# countryOfProduction TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE origin_country_types (
Tobi Ajayi(# originCountryTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# originalCountryName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE spoken_language_types (
Tobi Ajayi(# spokenLanguageTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# languageSpoken TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE created_by_types (
Tobi Ajayi(# createdById SERIAL PRIMARY KEY,
Tobi Ajayi(# creatorName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE shows (
Tobi Ajayi(# showId SERIAL PRIMARY KEY,
Tobi Ajayi(# name TEXT NOT NULL,
Tobi Ajayi(# numberOfSeasons INTEGER NOT NULL,
Tobi Ajayi(# numberOfEpisodes INTEGER NOT NULL,
Tobi Ajayi(# overview TEXT,
Tobi Ajayi(# adult BOOLEAN NOT NULL,
Tobi Ajayi(# inProduction BOOLEAN NOT NULL,
Tobi Ajayi(# originalName TEXT NOT NULL,
Tobi Ajayi(# popularity REAL NOT NULL,
Tobi Ajayi(# tagline TEXT,
Tobi Ajayi(# episodeRunTime INTEGER NOT NULL,
Tobi Ajayi(# statusId INTEGER NOT NULL,
Tobi Ajayi(# typeId INTEGER NOT NULL,
Tobi Ajayi(# FOREIGN KEY (statusId) REFERENCES status(statusId),
Tobi Ajayi(# FOREIGN KEY (typeId) REFERENCES types(typeId)
Tobi Ajayi(# );
CREATE TABLE

```

```

Tobi Ajayi=# CREATE TABLE genres (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# genreTypeId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, genreTypeId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (genreTypeId) REFERENCES genre_types(genreTypeId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE languages (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# languageTypeId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, languageTypeId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (languageTypeId) REFERENCES language_types(languageTypeId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE links (
Tobi Ajayi(# linkTypeId INTEGER NOT NULL,
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# linkName TEXT NOT NULL,
Tobi Ajayi(# PRIMARY KEY (linkTypeId, showId),
Tobi Ajayi(# FOREIGN KEY (linkTypeId) REFERENCES link_types(linkTypeId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE networks (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# networkTypeId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, networkTypeId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (networkTypeId) REFERENCES network_types(networkTypeId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE production_companies (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# productionCompanyId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, productionCompanyId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (productionCompanyId) REFERENCES production_company_types(productionCompanyId)
Tobi Ajayi(# );
CREATE TABLE

```

```

Tobi Ajayi=# CREATE TABLE production_countries (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# productionCountryTypeId INTEGER NOT NULL,
Tobi Ajayi(# originCountryTypeId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, productionCountryTypeId, originCountryTypeId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (productionCountryTypeId) REFERENCES production_country_types(productionCountryTypeId),
Tobi Ajayi(# FOREIGN KEY (originCountryTypeId) REFERENCES origin_country_types(originCountryTypeId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE spoken_languages (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# spokenLanguageTypeId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, spokenLanguageTypeId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (spokenLanguageTypeId) REFERENCES spoken_language_types(spokenLanguageTypeId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=#
Tobi Ajayi=# CREATE TABLE show_votes (
Tobi Ajayi(# voteCount INTEGER NOT NULL,
Tobi Ajayi(# voteAverage REAL NOT NULL,
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE created_bys (
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# createdById INTEGER NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, createdById),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId),
Tobi Ajayi(# FOREIGN KEY (createdById) REFERENCES created_by_types(createdById)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE air_dates (
Tobi Ajayi(# isFirst BOOLEAN NOT NULL,
Tobi Ajayi(# showId INTEGER NOT NULL,
Tobi Ajayi(# date TEXT NOT NULL,
Tobi Ajayi(# PRIMARY KEY (showId, isFirst),
Tobi Ajayi(# FOREIGN KEY (showId) REFERENCES shows(showId)
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=#

```

```

Tobi Ajayi | Superuser, Create DB, Replication
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS

postgres=# \q

C:\Users\Tobi Ajayi>psql -U "Tobi Ajayi"
Password for user Tobi Ajayi:

psql (16.2)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

Tobi Ajayi=# CREATE TABLE status (
Tobi Ajayi(# statusId SERIAL PRIMARY KEY,
Tobi Ajayi(# statusLabel TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE types (
Tobi Ajayi(# typeId SERIAL PRIMARY KEY,
Tobi Ajayi(# categoryName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE genre_types (
Tobi Ajayi(# genreTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# genreLabel TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE language_types (
Tobi Ajayi(# languageTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# languageLabel TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE link_types (
Tobi Ajayi(# linkTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# linkCategoryName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE network_types (
Tobi Ajayi(# networkTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# networkName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE
Tobi Ajayi=# CREATE TABLE production_company_types (
Tobi Ajayi(# productionCompanyTypeId SERIAL PRIMARY KEY,
Tobi Ajayi(# producerName TEXT NOT NULL
Tobi Ajayi(# );
CREATE TABLE

```

Tobi Ajayi

```
Tobi Ajayi=# SELECT * FROM genres LIMIT 4;
showid | genretypeid
-----+-----
1399 | 7
1399 | 12
1399 | 16
71446 | 4
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM languages LIMIT 4;
showid | languagetypeid
-----+-----
1399 | 66
71446 | 121
66732 | 66
1402 | 66
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM links LIMIT 4;
linktypeid | showid | linkname
-----+-----+-----
1 | 1399 | http://www.hbo.com/game-of-thrones
1 | 71446 | https://www.netflix.com/title/80192098
1 | 66732 | https://www.netflix.com/title/80057281
1 | 1402 | http://www.amc.com/shows/the-walking-dead--1002293
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM networks LIMIT 4;
showid | networktypeid
-----+-----
1399 | 2321
71446 | 2847
71446 | 2111
66732 | 2847
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM production_companies LIMIT 4;
showid | productioncompanytypeid
-----+-----
1399 | 3212
1399 | 1345
1399 | 5460
1399 | 6720
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM production_countries LIMIT 4;
showid | productioncountrytypeid | origincountrytypeid
-----+-----+-----
1399 | 165 | 16
1399 | 30 | 16
71446 | 10 | 150
66732 | 30 | 16
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM spoken_languages LIMIT 4;
showid | spokenlanguagetypeid
-----+-----
1399 | 6
71446 | 73
66732 | 6
1402 | 6
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM show_votes LIMIT 4;
votecount | voteaverage | showid
-----+-----+-----
21857 | 8.442 | 1399
17836 | 8.257 | 71446
16161 | 8.624 | 66732
15432 | 8.121 | 1402
(4 rows)
```

Tobi Ajayi

```
Tobi Ajayi=# SELECT * FROM created_bys LIMIT 4;
showid | createdbyid
-----+-----
    1399 |          24015
    1399 |           627
   71446 |           481
   66732 |          13781
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM air_dates LIMIT 4;
isfirst | showid | date
-----+-----+-----
t       |    1399 | 17/04/2011
t       |   71446 | 02/05/2017
t       |   66732 | 15/07/2016
t       |    1402 | 31/10/2010
(4 rows)
```

```
Tobi Ajayi=# |
```

```
Tobi Ajayi=# SELECT * FROM status LIMIT 10;
statusid | statuslabel
-----+-----
        1 | Ended
        2 | Returning Series
        3 | Canceled
        4 | In Production
        5 | Planned
        6 | Pilot
        7 | Some Status
(7 rows)
```

```
Tobi Ajayi=# SELECT * FROM types LIMIT 10;
typeid | categoryname
-----+-----
        1 | Scripted
        2 | Documentary
        3 | Miniseries
        4 | Reality
        5 | Talk Show
        6 | Video
        7 | News
(7 rows)
```

```
Tobi Ajayi=# SELECT * FROM genre_types LIMIT 10;
genretypeid | genrelabel
-----+-----
          1 | Romance
          2 | Kids
          3 | Comedy
          4 | Crime
          5 | Musical
          6 | War & Politics
          7 | Sci-Fi & Fantasy
          8 | Soap
          9 | Western
         10 | History
(10 rows)
```

Tobi Ajayi

```
Tobi Ajayi=# SELECT * FROM language_types LIMIT 10;
languageypeid | languagelabel
```

```
-----+-----
          1 | ko
          2 | tn
          3 | tt
          4 | sk
          5 | ru
          6 | nv
          7 | af
          8 | os
          9 | my
         10 | de
```

```
(10 rows)
```

```
Tobi Ajayi=# SELECT * FROM link_types LIMIT 10;
linktypeid | linkcategoryname
```

```
-----+-----
          1 | homepage
          2 | backdrop_path
          3 | poster_path
```

```
(3 rows)
```

```
Tobi Ajayi=# SELECT * FROM network_types LIMIT 4;
networktypeid | networkname
```

```
-----+-----
          1 | SCTV
          2 | Facebook
          3 | NT1
          4 | SHO.com
```

```
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM created_by_types LIMIT 4;
```

```
createdbyid | creatorname
-----+-----
          1 | Diane Kredensor
          2 | Sylvain Gouverneur
          3 | Tim Bullock
          4 | Brad Garrett
```

```
(4 rows)
```

```
Tobi Ajayi=# SELECT * FROM shows LIMIT 4;
```

```
showid | name | numberofseasons | numberofepisodes |
-----+-----+-----+-----+
overview
-----+-----+-----+-----+
| adult | inroduction | originalname | popularity | tagline | episoderuntime | statusid | typeid
-----+-----+-----+-----+-----+-----+-----+-----+
1399 | Game of Thrones | 8 | 73 | Seven noble families fight for control of the mythical land of Westeros. Friction between the houses leads to full-scale war. All while a very ancient evil awakens in the farthest north. Amidst the war a neglected military order of misfits the Night's Watch is all that stands between the realms of men and icy horrors beyond.
71446 | Money Heist | 3 | 41 | To carry out the biggest heist in history a mysterious man called The Professor recruits a band of eight robbers who have a single characteristic: none of them has anything to lose. Five months of seclusion - memorizing every step every detail every probability - culminate in eleven days locked up in the National Coinage and Stamp Factory of Spain surrounded by police forces and with dozens of hostages in their power to find out whether their suicide wager will lead to everything or nothing.
66732 | Stranger Things | 4 | 34 | When a young boy vanishes a small town uncovers a mystery involving secret experiments terrifying supernatural forces and one strange little girl.
1402 | The Walking Dead | 11 | 177 | Sheriff's deputy Rick Grimes awakens from a coma to find a post-apocalyptic world dominated by flesh-eating zombies. He sets out to find his family and encounters many other survivors along the way.
(4 rows)
```

```

Tobi Ajayi=# SELECT * FROM production_company_types LIMIT 4;
productioncompanytypeid | producername
-----+-----
1 | Malcolm Enterprises
2 | Felix Culpa
3 | Mig Beats Sdn. Bhd
4 | Ebru TV
(4 rows)

Tobi Ajayi=# SELECT * FROM production_country_types LIMIT 4;
productioncountrytypeid | countryofproduction
-----+-----
1 | Serbia
2 | Nicaragua
3 | Qatar
4 | Bolivia
(4 rows)

Tobi Ajayi=# SELECT * FROM origin_country_types LIMIT 4;
origincountrytypeid | originalcountryname
-----+-----
1 | PL
2 | MP
3 | UG
4 | FJ
(4 rows)

Tobi Ajayi=# SELECT * FROM spoken_language_types LIMIT 4;
spokenlanguagetypeid | languagespoken
-----+-----
1 | LatvieŅu
2 | Latin
3 | PortuguŅs
4 | Ņslenska
(4 rows)

```

## Normalization to BCNF

The initial dataset was represented in a single table with the following attributes:

- id, name, number\_of\_seasons, number\_of\_episodes, original\_language, vote\_count, vote\_average, overview, adult, backdrop\_path, first\_air\_date, last\_air\_date, homepage, in\_production, original\_name, popularity, poster\_path, type, status, tagline, genres, created\_by, languages, networks, origin\_country, spoken\_languages, production\_companies, production\_countries, episode\_run\_time.

This dataset comprised a vast array of information about TV shows, encompassing fields such as identifiers, descriptive data, and categorical attributes. This original table, while comprehensive, suffered from significant redundancy, repetition, and potential update anomalies. To enhance data integrity and optimize query performance, a normalization process was necessary to restructure this dataset into a more manageable and robust schema.

Normalization began by establishing the First Normal Form (1NF), which involved segregating data into atomic elements and eliminating repeating groups. Attributes such as genres, languages, and production countries, which previously contained multiple values within single records, were split into independent tables. This step was crucial to removing duplications and ensuring that each field contained the smallest possible data unit.

Subsequently, the dataset was transitioned into the Second Normal Form (2NF). During this phase, tables were restructured to remove partial dependencies; attributes that did not depend solely on the primary key but instead on portions of composite keys were identified and isolated. This involved detaching various show-related attributes into standalone tables, each linked to the primary show table via foreign keys, thereby reducing redundancy and enhancing data accessibility.

Progressing to the Third Normal Form (3NF), the schema was refined to eliminate transitive dependencies. Attributes that depended on other non-primary attributes were moved into separate

Tobi Ajayi

tables. This step ensured that non-key attributes in each table depended only on the primary key, further enhancing the logical separation and independence of data segments.

The final step towards achieving Boyce-Codd Normal Form (BCNF) involved scrutinizing each table for remaining anomalies and ensuring that every determinant was a candidate key. This stringent form of normalization was necessary to handle remaining irregularities not addressed by 3NF, primarily focusing on ensuring that every non-trivial functional dependency in the table was on a superkey. The resulting database schema includes multiple interlinked tables such as shows, genre\_types, language\_types, network\_types, and various associative tables that manage relationships like show-genres and show-languages.

New DB Schema:

- status: [statusid, statuslabel]
- types: [typeid, categoryname]
- genre\_types: [genretypid, genrelabel]
- language\_types: [languagetypeid, languagelabel]
- link\_types: [linktypeid, linkcategoryname]
- network\_types: [networktypeid, networkname]
- production\_company\_types: [productioncompanytypeid, producername]
- production\_country\_types: [productioncountrytypeid, countryofproduction]
- origin\_country\_types: [origincountrytypeid, originalcountryname]
- spoken\_language\_types: [spokenlanguagetypeid, languagespoken]
- created\_by\_types: [createdbyid, creatorname]
- shows: [showid, name, numberofseasons, numberofepisodes, overview, adult, inproduction, originalname, popularity, tagline, episoderuntime, statusid, typeid ]
- genres: [showid, genretypid]
- languages: [showid, languagetypeid]
- links: [linktypeid, showid, linkname]
- networks: [showid, networktypeid]
- production\_companies: [showid, productioncompanytypeid]
- production\_countries: [showid, productioncountrytypeid, origincountrytypeid]
- spoken\_languages: [showid, spokenlanguagetypeid]
- show\_votes: [votecount, voteaverage, showid]
- created\_bys: [showid, createdbyid]
- air\_dates: [isfirst, showid, date]

Some Resources I found very helpful:

- <https://www.analyticsvidhya.com/blog/2023/02/how-to-normalize-relational-databases-with-sql-code/>
- [https://www.ict.griffith.edu.au/normalization\\_tools/normalization/ind.php](https://www.ict.griffith.edu.au/normalization_tools/normalization/ind.php)
-



Tobi Ajayi

## CLI Functionality and Interface

### CLI Menu:

```
PS C:\Users\Tobi Ajayi> cd .\Downloads\
PS C:\Users\Tobi Ajayi\Downloads> python .\cmpenfp_post.py

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12):
```

### Testing Every Function:

1. Insert Data: ('INSERT INTO {table} ({fields}) VALUES ({values})')

```
Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 1
Which table would you like to insert into?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates
Enter the number of the table: 1
Inserting into status.
Enter statusid: 8
Enter statuslabel: Test Status
Data inserted into status successfully.
```

Tobi Ajayi

## 2. Delete Data: ('DELETE FROM {table\_name} WHERE {condition} = %s')

```

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 2
Which table would you like to delete from?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates
Enter the number of the table: 1
Delete from status.
Enter the condition (e.g., 'column_name = value'): statusid
Enter the value for the condition: 8
Record(s) deleted successfully from status.

```

## 3. Update Data: ('UPDATE {table\_name} SET {set\_clause} = %s WHERE {condition} = %s')

```

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 3
Which table would you like to update?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates
Enter the number of the table: 1
Enter the SET clause (e.g., 'column1 = new_value'): statuslabel
Enter the new value for the set clause: Status Updated
Enter the condition for the update (e.g., 'id = 4'): statusid
Enter the value for the condition: 7
Record(s) updated successfully.

```

Tobi Ajayi

#### 4. Search Data: ('SELECT \* FROM {table\_name} WHERE {column\_name} LIKE %s')

```
Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 4
Which table would you like to search in?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates

Enter the number of the table: 1
Enter the search criteria for the status table.
Enter the column name to search (e.g., 'statuslabel'): statuslabel
Enter the search value (will automatically be wrapped with %): Status Updated
Found 1 record(s):
(7, 'Status Updated')
```

#### 5. Aggregate Functions: ('SELECT {agg\_function}({column}) FROM {table\_name}')

```
Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 5
Which table would you like to perform aggregation on?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates

Enter the number of the table: 1
Enter the column to aggregate: statusid
Which aggregate function would you like to perform?
1. SUM
2. AVG
3. COUNT
4. MIN
5. MAX

Enter your choice (1-5): 3
COUNT of statusid in status is: 7
```

Tobi Ajayi

## 6. Sorting: ('SELECT \* FROM {table\_name} ORDER BY {column} {direction}')

```

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 6
Which table would you like to sort data in?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates

Enter the number of the table: 1
Enter the column to sort by: statuslabel
Choose sort direction:
1. Ascending (ASC)
2. Descending (DESC)
Enter your choice (1-2): 1
(3, 'Canceled')
(1, 'Ended')
(4, 'In Production')
(6, 'Pilot')
(5, 'Planned')
(2, 'Returning Series')
(7, 'Status Updated')

```

## 7. Joins: ('SELECT \* FROM {table1\_name} INNER JOIN {table2\_name} ON {table1\_name}.{join\_column} = {table2\_name}.{join\_column} LIMIT 6')

```

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 7
Available tables for JOIN operation:
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates

Enter the number of the first table: 12
Enter the number of the second table: 1
Enter the column name to join on (assumed same in both tables): statusid
Joined data (6 rows):
(1399, 'Game of Thrones', 8, 73, 'Seven noble families fight for control of the mythical land of Westeros. Friction between the houses leads to full-scale war. All while a very ancient evil awakens in the farthest north. Amidst the war, a neglected military order of misfits, the Night's Watch, is all that stands between the realms of men and icy horrors beyond.', False, False, 'Game of Thrones', 1083.917, 'Winter Is Coming', 0, 1, 1, 1, 'Ended')
(72666, 'Money Heist', 3, 43, 'To carry out the biggest heist in history, a mysterious man called The Professor recruits a band of eight robbers who have a single characteristic: none of them has anything to lose. Five months of seclusion - memorizing every step, every detail, every probability - culminates in eleven days locked up in the National Coinage and Stamp Factory of Spain, surrounded by police forces and with dozens of hostages in their power, to find out whether their suicide wager will lead to everything or nothing.', False, False, 'La Casa de Papel', 96.356, 'The perfect robbery.', 70, 1, 1, 1, 'Ended')
(6072, 'Stranger Things', 5, 34, 'When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces, and one strange little girl.', False, True, 'Stranger Things', 185.711, 'Every ending has a beginning.', 6, 1, 2, 1, 'Ended')
(1602, 'The Walking Dead', 11, 177, 'Sheriff's deputy Rick Grimes awakens from a coma to find a post-apocalyptic world dominated by flesh-eating zombies. He sets out to find his family and encounters many other survivors along the way', False, False, 'The Walking Dead', 489.746, 'Fight the dead. Fear the living.', 40, 1, 1, 'Ended')
(63174, 'Lucifer', 8, 91, 'Bored and unhappy as the Lord of Hell, Lucifer Morningstar abandoned his throne and retired to Los Angeles, where he has teamed up with LAPD detective Chloe Decker to take down criminals.\\nBut the longer he's away from the underworld, the greater the threat that the worst of humanity could escape.', False, False, 'Lucifer', 416.668, 'It's good to be bad.', 40, 1, 1, 1, 'Ended')
(60850, 'Riverdale', 7, 137, 'Set in the present, the series offers a bold, subversive take on Archie, Betty, Veronica and their friends, exploring the surrealism of small-town life, the darkness and weirdness bubbling beneath Riverdale's wholesome facade.', False, False, 'Riverdale', 143.75, 'To save the future, they must survive the past.', 40, 1, 1, 1, 'Ended')

```

8. Grouping: ('SELECT {group\_column}, {agg\_function}({group\_column}) FROM {table\_name} GROUP BY {group\_column}')

```
Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 8
Available tables for GROUP BY operation:
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates

Enter the number of the table: 12
Enter the column name to group by: statusid
Choose the aggregate function to apply:
1. COUNT
2. SUM
3. AVG
4. MIN
5. MAX

Enter your choice (1-5): 1
(1, 124518)
(2, 13869)
(3, 18398)
(4, 9333)
(5, 2895)
(6, 1350)
(7, 580)
```

9. Subqueries: ('SELECT \* FROM {main\_table\_name} WHERE {subquery\_column} IN (SELECT {subquery\_column} FROM {sub\_table\_name}) LIMIT 6')

```
Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 9
Available tables for performing subqueries:
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates

Enter the number of the main table: 12
Enter the number of the table for the subquery: 1
Enter the column used in the subquery condition: statusid
Found 6 record(s) based on the subquery condition:
(119, "Game of Thrones", 6, 7), "Seven noble families fight for control of the mythical land of Westeros. Friction between the houses leads to full-scale war. All while a very ancient evil awakens in the farthest north. Amidst the war, a neglected military order of misfits, the Night's Watch, is all that stands between the realm of men and icy horrors beyond.", False, False, "Game of Thrones", 1863.917, "Winter is Coming", 0, 1, 1)
(119, "Heros Fall", 7, 8), "To carry out the biggest heist in history, a mysterious man called The Professor recruits a band of eight robbers who have a single characteristic: none of them has anything to lose. Five months of seclusion - memorizing every stop, every detail, every probability - culminate in one very day, locked up in the National Goldmine and Stamp Factory of Spain surrounded by police forces and with dozens of hostages in their power, to find out whether their suicide wager will lead to everything or nothing.", False, False, "La Casa de Papel", 96.350, "The perfect robbery.", 98, 1, 1)
(1072, "Stranger Things", 4, 2), "When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces, and one strange little girl.", False, True, "Stranger Things", 186.711, "Every ending has a beginning.", 1, 1, 2)
(1082, "The Walking Dead", 11, 17), "Sheriff's deputy Rick Grimes awakens from a coma to find a post-apocalyptic world dominated by flesh-eating zombies. He sets out to find his family and encounters many other survivors along the way.", False, False, "The Walking Dead", 489.768, "Fight the dead. Fear the living.", 12, 1, 1)
(6176, "Lucifer", 6, 9), "Bored and unhappy as the Lord of Hell, Lucifer Morningstar abandoned his throne and retired to Los Angeles, where he has teamed up with LAPD detective Chloe Decker to take down criminals.\\nBut the longer he's away from the underworld, the greater the threat that the worst of humanity could escape.", False, False, "Lucifer", 116.660, "It's good to be bad.", 98, 1, 1)
(6060, "Riverdale", 7, 13), "Set in the present, the series offers a bold, subversive take on Archie, Betty, Veronica and their friends, exploring the surrealism of small-town life, the darkness and weirdness bubbling beneath Riverdale's wholesome facade.", False, False, "Riverdale", 143.75, "To save the future, they must survive the past.", 45, 1, 1)
```

## 10. Transactions:

```

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 10

Transaction Menu:
1. Begin Transaction
2. Commit Transaction
3. Rollback Transaction
4. Return to Main Menu

Enter your choice: 1
Transaction started. You can now perform operations before committing.

Transaction Menu:
1. Begin Transaction
2. Commit Transaction
3. Rollback Transaction
4. Return to Main Menu

Enter your choice: 4
Returning to main menu...
```

```

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 1
Which table would you like to insert into?
1. status
2. types
3. genre_types
4. language_types
5. link_types
6. network_types
7. production_company_types
8. production_country_types
9. origin_country_types
10. spoken_language_types
11. created_by_types
12. shows
13. genres
14. languages
15. links
16. networks
17. production_companies
18. production_countries
19. spoken_languages
20. show_votes
21. created_bys
22. air_dates
Enter the number of the table: 1
Inserting into status.
Enter statusid: 9
Enter statuslabel: Transaction Test
Data inserted into status successfully.

Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12): 10

Transaction Menu:
1. Begin Transaction
2. Commit Transaction
3. Rollback Transaction
4. Return to Main Menu

Enter your choice: 2
Transaction committed successfully.

Transaction Menu:
1. Begin Transaction
2. Commit Transaction
3. Rollback Transaction
4. Return to Main Menu

Enter your choice: 4
Returning to main menu...

```

Tobi Ajayi

11. Exit:

```
        Welcome to the Database CLI Interface!

        Please select an option:
        1. Insert Data
        2. Delete Data
        3. Update Data
        4. Search Data
        5. Aggregate Functions
        6. Sorting
        7. Joins
        8. Grouping
        9. Subqueries
        10. Transactions
        11. Error Handling
        12. Exit

Enter your choice (1-12): 12
Exiting the application...
Database connection closed.
PS C:\Users\Tobi Ajayi\Downloads> |
```