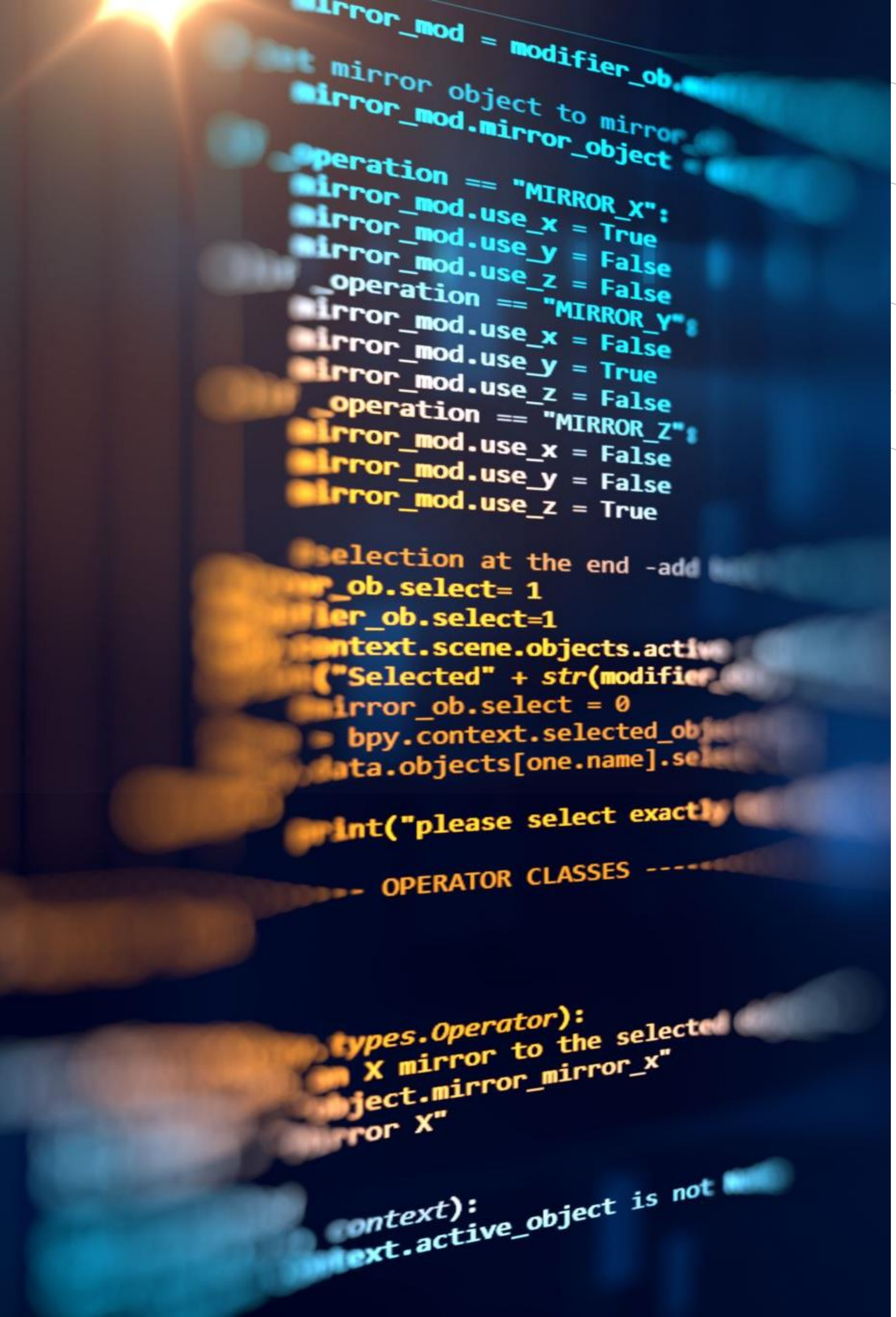

# Intro to DBeaver + SQL (with Chinook DB)

Install → Tour → Basic → Intermediate → Advanced SQL

## Install DBeaver

- 1. Go to https://dbeaver.io/download
2. Download the Community Edition
3. Install with default settings
4. Launch DBeaver

# Load the Chinook Database

- 1. Download Chinook_Sqlite.sqlite from GitHub

2. File → New → Database Connection → SQLite

3. Choose Chinook_Sqlite.sqlite file

4. Click Finish

OR use the default install demo DB

## DBeaver UI Tour

- • Database Navigator (left)
- SQL Editor (main pane)
- Results Grid (bottom)
- ER Diagram tab (right-click DB)
- New SQL script: Alt+Insert
- Run SQL: Ctrl+Enter

## Basic SQL: SELECT & LIMIT

-
  ```
  SELECT * FROM Album LIMIT 10;
  ```
  **See first 10 albums**

-
  ```
  SELECT * FROM Artist
  WHERE Name = 'Queen';
  ```
-
  Find a specific artist

-
  ```
  SELECT Name, Milliseconds/60000.0 AS Minutes
  FROM Track
  WHERE Milliseconds > 300000
  ORDER BY Minutes DESC;
  ```
-
  Tracks longer than 5 minutes

## What is an INNER JOIN?

- An INNER JOIN returns **only the rows where there's a match in *both* tables.**
- Think of it like a **Venn diagram's overlapping middle** — you only get the part where the two tables intersect.
- If a row exists in Table A but not in Table B, it's **excluded**.
- If a row exists in Table B but not in Table A, it's **excluded** too.
- Useful when you only care about complete data relationships.

```
SELECT Album.Title AS AlbumTitle,
       Artist.Name AS ArtistName
FROM Album
INNER JOIN Artist
  ON Album.ArtistId = Artist.ArtistId
ORDER BY ArtistName
LIMIT 10;
```

```
SELECT Track.Name, Album.Title
FROM Track
INNER JOIN Album
  ON Track.AlbumId = Album.AlbumId;
```

## What is a LEFT JOIN?

- A LEFT JOIN (aka **LEFT OUTER JOIN**) returns:
- **All rows from the left table**
- **Matching rows from the right table**
- **NULLs where no match exists on the right**

*If there's no match, it* ***still shows the left row*** *— just with blanks from the right side.*

```
SELECT Artist.Name AS ArtistName,
       Album.Title AS AlbumTitle
FROM Artist
LEFT JOIN Album
  ON Artist.ArtistId = Album.ArtistId
ORDER BY ArtistName
LIMIT 15;
```

•Every row from Artist appears.
•If a match exists in Album, its Title is shown.
•If not, AlbumTitle shows NULL.

## Aggregate by Country

```
SELECT Country, COUNT(*) AS CustomerCount
FROM Customer
GROUP BY Country
ORDER BY CustomerCount DESC
LIMIT 5;
```

## Advanced: Subquery (Longest Track)

```
SELECT Name, AlbumId, Milliseconds
FROM Track
WHERE Milliseconds = (
  SELECT MAX(Milliseconds)
  FROM Track t2
  WHERE t2.AlbumId = Track.AlbumId
);
```

## Advanced: CASE Logic

```
SELECT Name,
  CASE
    WHEN Milliseconds < 180000 THEN 'Short'
    WHEN Milliseconds < 300000 THEN 'Medium'
    ELSE 'Long'
  END AS LengthCategory
FROM Track;
```

## Advanced: Create a View

```
CREATE VIEW TopArtists AS
SELECT Artist.Name, COUNT(Album.AlbumId) AS Albums
FROM Artist
JOIN Album ON Artist.ArtistId = Album.ArtistId
GROUP BY Artist.Name;
```

## Advanced: Query a View

```
SELECT * FROM TopArtists
ORDER BY Albums DESC
LIMIT 10;
```

## Advanced: Nested Subquery + JOINs

```
SELECT Artist.Name, COUNT(Track.TrackId) AS
TotalTracks
FROM Artist
JOIN Album ON Artist.ArtistId = Album.ArtistId
JOIN Track ON Album.AlbumId = Track.AlbumId
WHERE Artist.ArtistId IN (
  SELECT ArtistId FROM Album WHERE Title LIKE
'%Greatest%'
)
GROUP BY Artist.Name;
```