Introduction to SQL

LIGHTHOUSE LABS

AGENDA

History

Relational Data

SQL Language

SQL Practice Problems

LIGHTHOUSE LABS

Database History

LIGHTHOUSE LABS

Relational Algebra

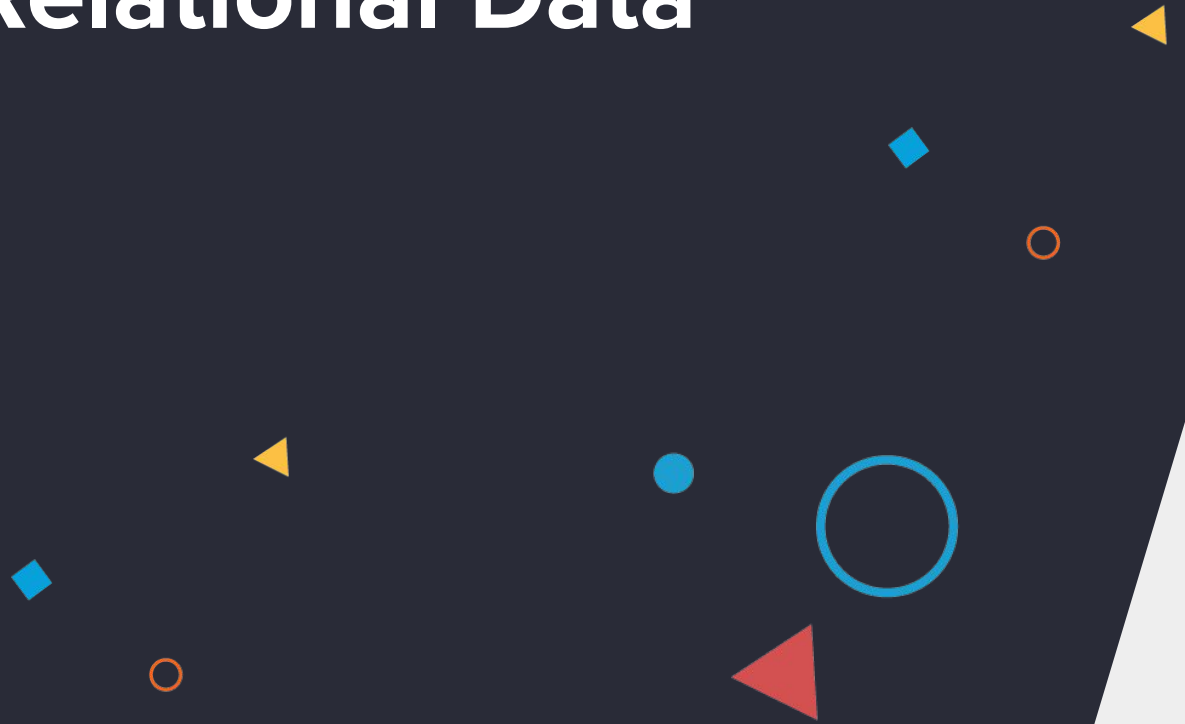
- Mathematical Foundation of all Relational Database Theory
- Pioneered by E.F. Codd in 1970
- A variety of Operations that can be applied on “Relations” Aka “Tables”
- **Projection (Π)**
- **Selection (*σ*)**
- **Rename (*ρ*)**
- **Natural join (⋈)**
- **Semijoin (⋉)(⋊)**
- **Antijoin (▷)**
- Databases are highly optimizable because Relational Algebra provides a mathematical underpinning
    - (a ⋈ b = b ⋈ a)

From SEQUEL to SQL

- Following Codd's Relational Algebra work came: SEQUEL (Structured English Query Language)
- Developed at IBM by Chamberlain and Boyce
- IBM had some of the first big mainframes, and needed to be address data storage needs
- Evolved into SQL (Structured Query Language) because of copyright dispute

- Official "Database Language SQL" ratified by ANSI and ISO standard groups in 1986

Relational Data



LIGHTHOUSE LABS

Data is Related

- The Fundamental concept is a Table
- A Table is a Set of Columns
- Relational Data is organized based around Relationships between Tables
    - One-to-one
    - Many-to-one / One-to-Many
    - Many-to-many

One to One

- A rather rare type of relationship
- Used mostly for human comprehensibility when there is a set of self-contained data

songs

id	title	album_id
1	Stairway	1
2	Helter Skelter	2
3	Black Dog	1

one to one

songs_dj_info

song_id	key	bpm
1	Am	82
2	Em	168
3	A	162

Many to One

- Many to One or One to Many
- A very common type of relationship
- A book has one author, but each author has many books

songs

id	title	bpm	album_id
1	Stairway	82	1
2	Helter Skelter	168	2
3	Black Dog	162	1

albums

id	name	release
1	Zeppelin 4	1971
2	White Album	1968

LIGHTHOUSE LABS

Many to One

- The table which is owned by the other table contains a *resource_id* column
- This column is called a **foreign key**

songs

id	title	bpm	album_id
1	Stairway	82	1
2	Helter Skelter	168	2
3	Black Dog	162	1

albums

id	name	release
1	Zeppelin 4	1971
2	White Album	1968

Each song is *on* a particular album, so the **songs** table gets an **album_id** foreign key

Many to Many

- A join table with two **foreign keys**

songs

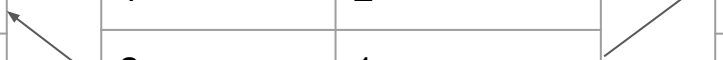
id	title
1	Killer Queen
2	Under Pressure
3	Five Years

songs_artists

song_id	artist_id
1	2
2	1
2	2
3	1

artists

id	name
1	David Bowie
2	Queen



SQL Language

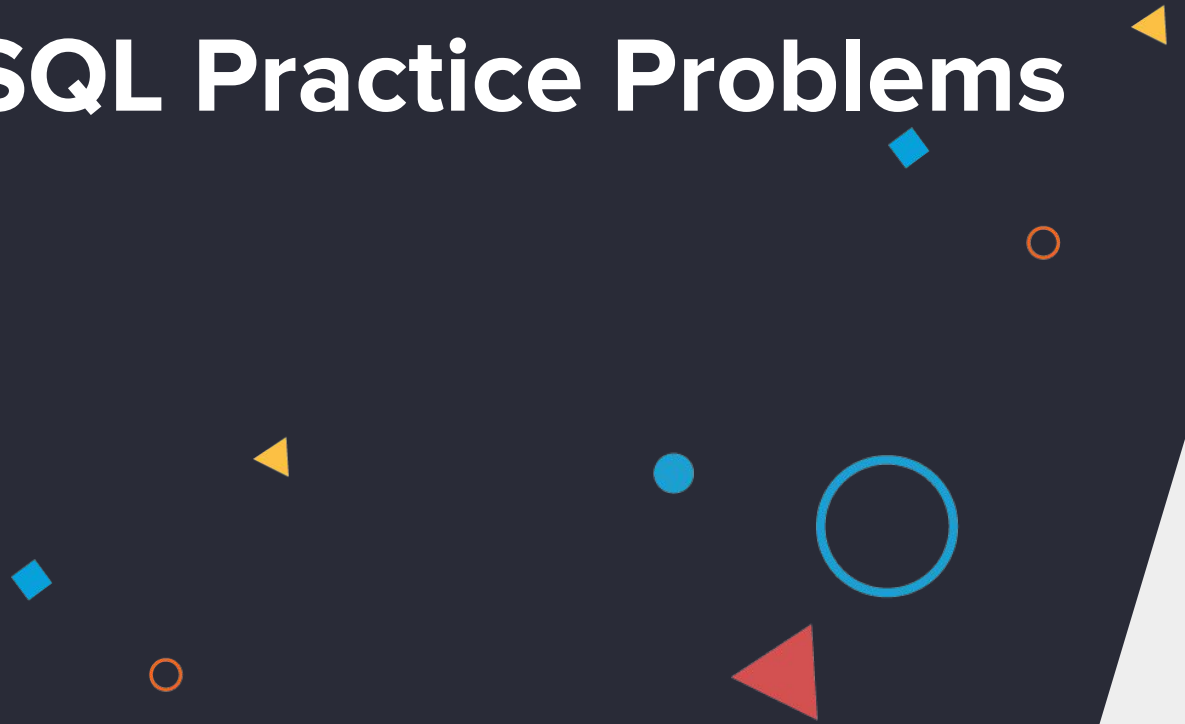
LIGHTHOUSE LABS

SELECT statements

- SELECT statements are used to **query** data
- SELECT statements always have a **column_list**, a **FROM clause** and a **WHERE clause**
- SELECT statements may have one or more **JOIN clauses**
- SELECT statements may modify the order of results using **ORDER BY clauses**
- SELECT statements support aggregate queries using **GROUP BY** and **HAVING clauses**

```
SELECT *
  FROM songs
  JOIN albums on (songs.album_id = albums.id)
WHERE
  songs.title = 'Stairway';
```

SQL Practice Problems



LIGHTHOUSE LABS

Questions?

LIGHTHOUSE LABS