

Intro to SQL!

Brought to you by Galvanize!

Links

- ➔ <http://www.sql-tutorial.net/SQL-Cheat-Sheet.pdf>
- ➔ <http://www.sqltutorial.org/>
- ➔ <https://modern-sql.com/use-case/literate-sql>
- ➔ <http://sqlzoo.net/>

When and Why do we use Data?

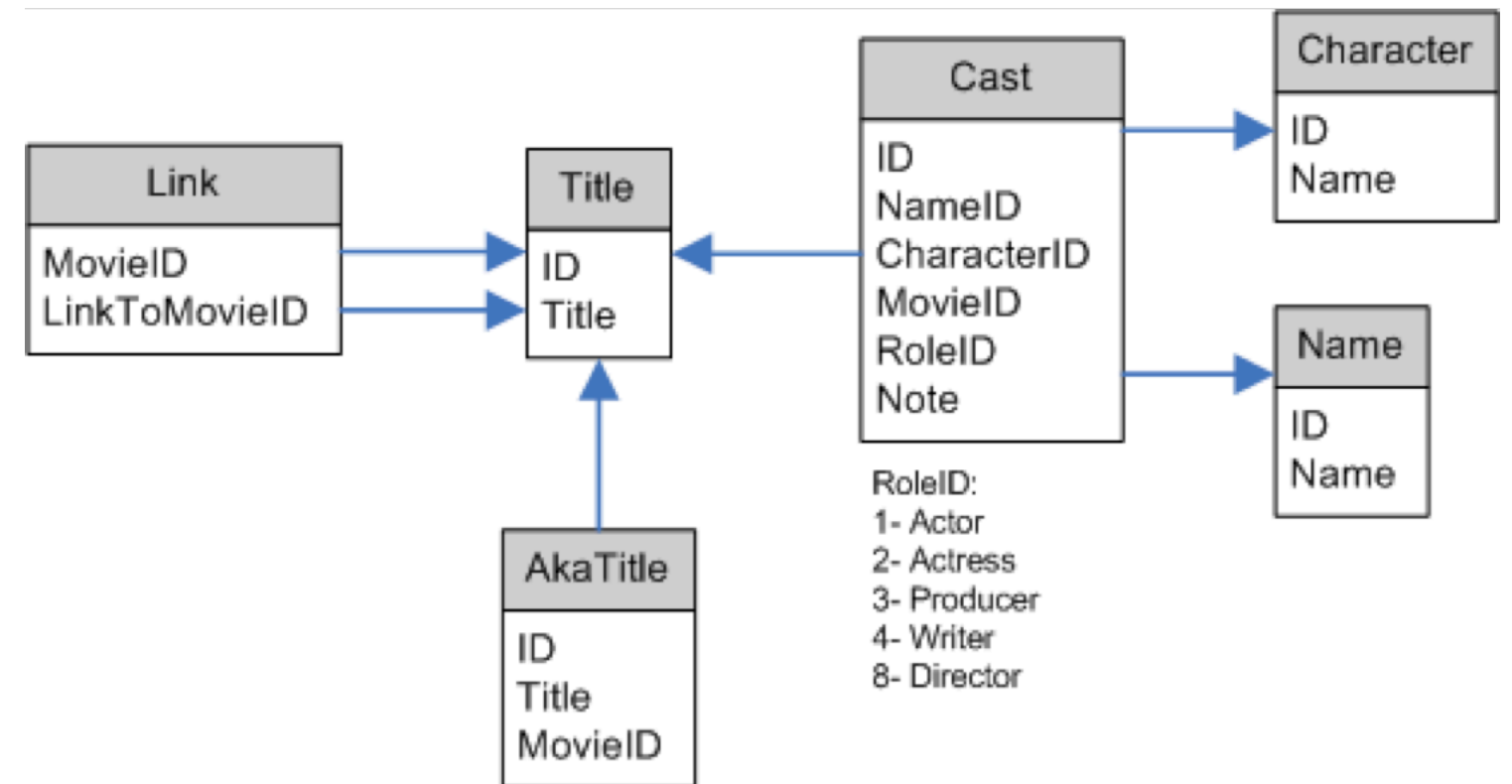
facebook

amazon

Google

What is a Relational Database

Name	FName	City	Age	Salary
Smith	John	3	35	\$280
Doe	Jane	1	28	\$325
Brown	Scott	3	41	\$265
Howard	Shemp	4	48	\$359
Taylor	Tom	2	22	\$250



SQL

Structured Query Language

- DDL - Data Definition Language
- DML - Data Manipulation Language
- DCL - Data Control Language

DML

Data Manipulation Language

SELECT ... FROM

-SQL is structured similar to the English language. The basic command for retrieving data from a database table is to **SELECT** data **FROM** a table. Not surprisingly, the keywords "SELECT" and "FROM" make up the core of a SQL statement.

```
SELECT "COLUMN NAME"  
FROM "TABLE_NAME";
```

SELECT ... FROM

→ You can select more than 1 row

```
SELECT "COLUMN_1", "COLUMN_2"  
FROM "TABLE_NAME";
```

→ You can select all columns

```
SELECT *  
FROM "TABLE_NAME";
```

→ You can select unique values as well

```
SELECT DISTINCT "COLUMN_1"  
FROM "TABLE_NAME";
```


WHERE

Sometimes we only want to use a subset of data

```
SELECT *  
FROM "TABLE_NAME"  
WHERE "CONDITION";
```

ORDER BY

When we want to list the data in a particular order

```
SELECT *  
FROM "TABLE_NAME"  
WHERE "CONDITION"  
ORDER BY "COLUMN_NAME" [ASC | DESC] ;
```

MATHEMATICAL FUNCTIONS

SQL has built-in mathematical functions to allow us to perform mathematical operations on the data. Common mathematical functions include:

- SUM
- AVG
- COUNT
- MAX

GROUP BY

Date	Store	Sales_Amount
10/27/2000	Boulder	10.46
10/28/2000	Denver	12.58

```
SELECT MAX (Sales_Amount)  
FROM SALES_HISTORY;
```

→ Finds the Sales Amount from all stores

GROUP BY

Date	Store	Sales_Amount
10/27/2000	Boulder	10.46
10/28/2000	Denver	12.58

```
SELECT Store, MAX (Sales_Amount)  
FROM SALES_HISTORY;
```

→ Finds highest sales for each Store

GROUP BY

- ➔ However, this `SELECT` statement by itself is not enough. To allow SQL to correctly calculate what we want, we need to use the `GROUP BY` keyword. In the following example, the `Store` column after `GROUP BY` tells SQL to apply the `MAX` function for each `Store`

```
SELECT Store, MAX(Sales-Amount)
FROM SALES_HISTORY
GROUP BY Store;
```

GROUP BY

```
SELECT "COLUMN_1", FUNCTION ("COLUMN_2")  
FROM "TABLE_NAME"  
WHERE "CONDITION"  
GROUP BY "COLUMN_1";
```

HAVING

- Previously we had talked about using the WHERE keyword to filter results.
- We cannot use WHERE to filter based on the result of a function, because we need to specify the filtering condition after SQL has calculated the function, and consequently any filtering condition based on the function needs to be specified after the GROUP BY phrase. So we cannot use the WHERE keyword because

HAVING

```
SELECT "COLUMN_NAME", FUNCTION("OTHER_COLUMN_NAME")  
FROM "TABLE_NAME"  
GROUP BY "COLUMN_NAME"  
HAVING (CONDITION based on FUNCTION);
```

HAVING

→ Using the SALES_HISTORY table we had earlier. If we want to sum the sales amount for each store, but only want to see results for stores with total sales amount greater than 100, we use the following SQL:

```
SELECT Store, SUM(Sales_Amount)
FROM SALES_HISTORY
GROUP BY Store
HAVING SUM(Sales_Amount) > 12;
```

Order of SQL Commands

- SELECT
- FROM
- WHERE
- GROUP BY
- HAVING
- ORDER BY

Links

- ➔ <http://www.sql-tutorial.net/SQL-Cheat-Sheet.pdf>
- ➔ <http://www.sqltutorial.org/>
- ➔ <https://modern-sql.com/use-case/literate-sql>
- ➔ <http://sqlzoo.net/>

DDL



Data Definition Language

SQL became the standard for relational databases in the mid-1980's. Since then, it has been revised a few times to include a growing set of features. Despite the existence of such standards, most SQL code is not completely portable between different relational database systems without adjustments. The following are popular open source relational database systems that implement the SQL standard.

Vocab

- Schema
- Entity
- Attribute

What's the difference between Data and Information?

Date	Store	Sales_Amount
10/27/2000	Boulder	10.46
10/28/2000	Denver	12.58

- Row = Entity or Tuples
- Column = Attribute

Data Types

- integer
- text
- char
- varchar
- boolean
- serial
- timestamp

Constraints

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- also many more

CREATE TABLE

```
CREATE TABLE table_name (  
    column_name data_type column_constraint,  
    table_constraint  
);
```

Example

```
CREATE TABLE stores (  
    store_id INT PRIMARY KEY,  
    store_loc VARCHAR(100) NOT NULL,  
    sales    INT NOT NULL  
);
```

ALTER TABLE

```
ALTER TABLE table_name  
    ADD new_colum data_type column_constraint [AFTER existing_column];
```

```
ALTER TABLE table_name  
    MODIFY column_definition;
```

```
ALTER TABLE table_name  
DROP column_name,  
DROP column_name;
```

DROP TABLE

```
DROP TABLE [IF EXISTS] table_name;
```

```
DROP TABLE table_name1, table_name2, ...;
```