



2.01 Labs | SQL Intro 1+2

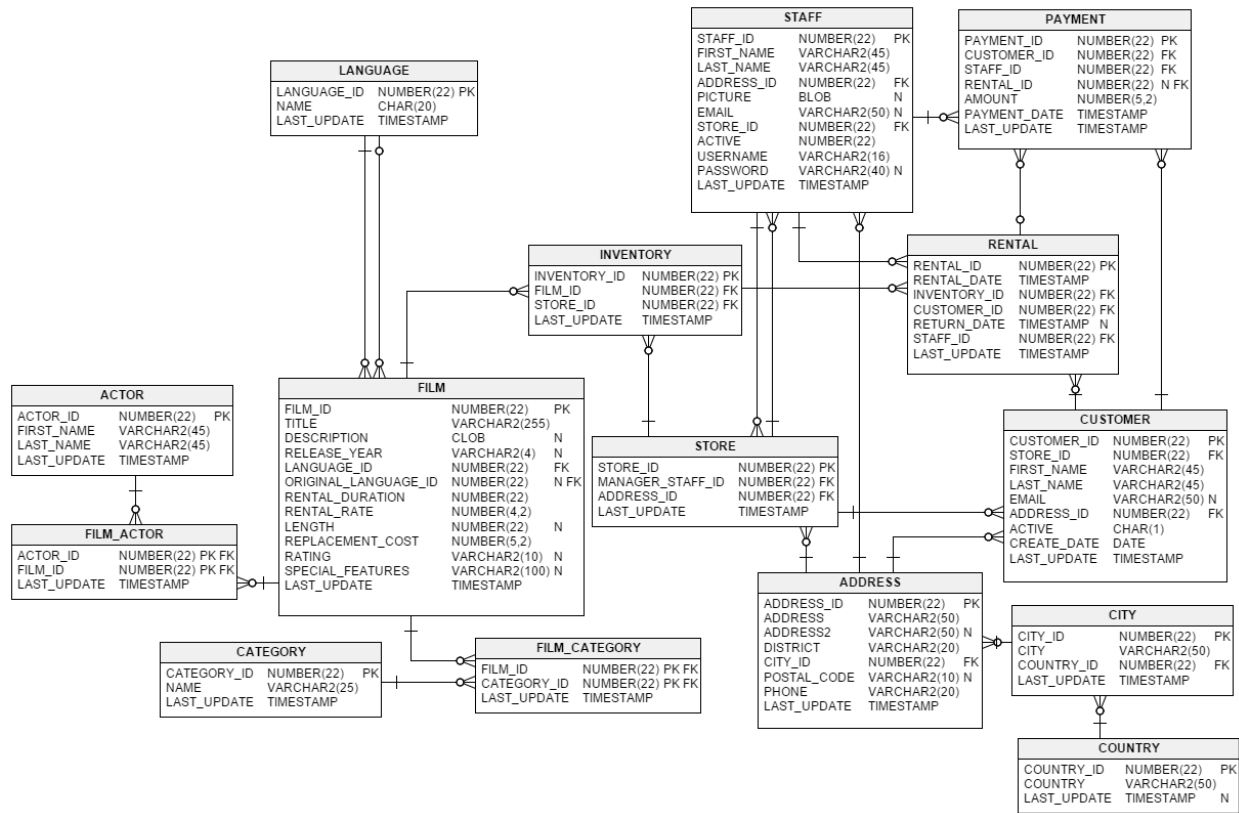
📅 Completed Date	@Oct 26, 2020
☰ Description	Two super fun and easy labs on mySQL.
▼ Module	Unit 2
▼ Status	Complete
☰ Type	Lab



Install the file [here](#) → refresh sakila. First data → schema.

In this lab, you will be using the Sakila database of movie rentals. You can follow the steps listed here to get the data locally: Sakila sample database - installation. You can work with two sql query files - `sakila-schema.sql` (creates the schema) + `sakila-data.sql` which inserts the data.

The ERD is pictured below - not all tables are shown, but many of the key fields you will be using are visible:



Instructions

- Review the tables in the database.
- Explore tables by selecting all columns from each table or using the in built review features for your client.
- Select one column from a table. Get film titles.

```
select title from film
```

TABLES & VIEWS

- actor
- actor_info
- address
- category
- city
- country
- customer
- customer_list
- film**
- film_actor
- film_category
- film_list
- film_text
- inventory
- language
- nicer_but_slower_film_list
- payment
- rental
- sales_by_film_category
- sales_by_store
- staff
- staff_list
- store

PROCS & FUNCS

- film_in_stock
- film_not_in_stock
- get_customer_balance
- inventory held by customer

1 `select title from film`

⚙️ Query Favorites ▾ Query History

title
ACADEMY DINOSAUR
ACE GOLDFINGER
ADAPTATION HOLES
AFFAIR PREJUDICE
AFRICAN EGG
AGENT TRUMAN
AIRPLANE SIERRA
AIRPORT POLLOCK
ALABAMA DEVIL
ALADDIN CALENDAR
ALAMO VIDEOTAPE
ALASKA PHANTOM

→ Select one column from a table and alias it. Get unique list of film languages under the alias `language`. Note that we are not asking you to obtain the language per each film, but this is a good time to think about how you might get that information in the future.

```
select distinct name as 'language' from language
```

The screenshot shows a database management interface. On the left, a sidebar titled 'TABLES & VIEWS' lists various tables. The 'language' table is highlighted. The main area displays a SQL query: `select distinct name as 'language' from language`. Below the query, a table shows the results of the query.

language
English
Italian
Japanese
Mandarin
French
German

→ Using the `select` statements and reviewing how many records are returned, can you find out how many stores and staff does the company have? Can you return a list of employee first names only?

```
#how many records are returned?
select COUNT(rental_id) from rental

#stores
select COUNT(store_id) from store-> 2
```

```
#staff
select COUNT(staff_id) from staff
```

TABLES & VIEWS

- actor
- actor_info
- address
- category
- city
- country
- customer
- customer_list
- film
- film_actor
- film_category
- film_list
- film_text
- inventory
- language
- nicer_but_slower_film_list
- payment
- rental

1

```
select COUNT(rental_id) from rental
```

Query Favorites

Query History

COUNT(rental_id)
16044

```
select first_name from staff
```

1

```
select first_name from staff
```

 Query Favorites

 Query History



first_name
Mike
Jon

→ Bonus: How many unique days did customers rent movies in this dataset? → 41

```
select DISTINCT left(rental_date,10) from rental
```

The screenshot shows a database management tool interface. On the left is a sidebar with a search bar labeled 'Filter'. Below it, under the heading 'TABLES & VIEWS', is a list of database tables. The 'rental' table is highlighted. Below this, under the heading 'PROCS & FUNCS', is a list of stored procedures, with 'film_in_stock' visible. The main area on the right is a query editor showing the SQL query: `select DISTINCT left(rental_date,10) from rental`. Below the query editor is a results pane with tabs for 'Query Favorites' and 'Query History'. The 'Query History' tab is active, showing a list of queries. The first query is `left(rental_date,10)`, and the results are a list of dates: 2005-05-24, 2005-05-25, 2005-05-26, 2005-05-27, 2005-05-28, 2005-05-29, 2005-05-30, 2005-05-31, and 2005-06-14.

left(rental_date,10)
2005-05-24
2005-05-25
2005-05-26
2005-05-27
2005-05-28
2005-05-29
2005-05-30
2005-05-31
2005-06-14

Lab 2

Instructions

1. Select all the actors with the first name 'Scarlett'. → 2

```
select first_name from actor
where first_name= 'Scarlett'
```

1. Select all the actors with the last name 'Johansson' → 3

```
select first_name, last_name from actor
where last_name= 'Johansson'
```

The screenshot shows a database management interface. On the left, a sidebar titled 'TABLES & VIEWS' lists various tables, with 'actor' selected. The main area displays a SQL query in a text editor:

```
1 select first_name, last_name from actor
2 where last_name= 'Johansson'
```

Below the query editor, there is a section for 'Query Favorites' and 'Query History'. The 'Query History' section shows the results of the query in a table:

first_name	last_name
MATTHEW	JOHANSSON
RAY	JOHANSSON
ALBERT	JOHANSSON

→ How many films (movies) are available for rent? → 1000

```
select COUNT(title) from film
```


1. How many films have been rented?

→ What is the shortest and longest rental period?

```
MAX-> 7
select max(rental_duration) from film

MIN-> 3
select min(rental_duration) from film
```

→ What are the shortest and longest movie duration? Name the values `max_duration` and `min_duration`.

```
MIN-> 46
select min(length) as min_duration from film

MAX-> 185
select max(length) as max_duration from film
```

→ What's the average movie duration?

```
115.2720

select AVG(length) from film
```

→ What's the average movie duration expressed in format (hours, minutes)?

```
select CONCAT (FLOOR(length/60), 'h', MOD(length,60), 'm') from film

select CONCAT (FLOOR(avg(length)/60), 'h', MOD(avg(length),60), 'm') from film

-> 1.55

OR
```

How many movies longer than 3 hours? → 39

```
select COUNT(length) from film
where length > 180
```

Get the name and email formatted. Example: Mary SMITH
- mary.smith@sakilacustomer.org.

```
select LOWER(email) from customer
```

1. What's the length of the longest film title?

27

```
select max(length(title)) from film
```