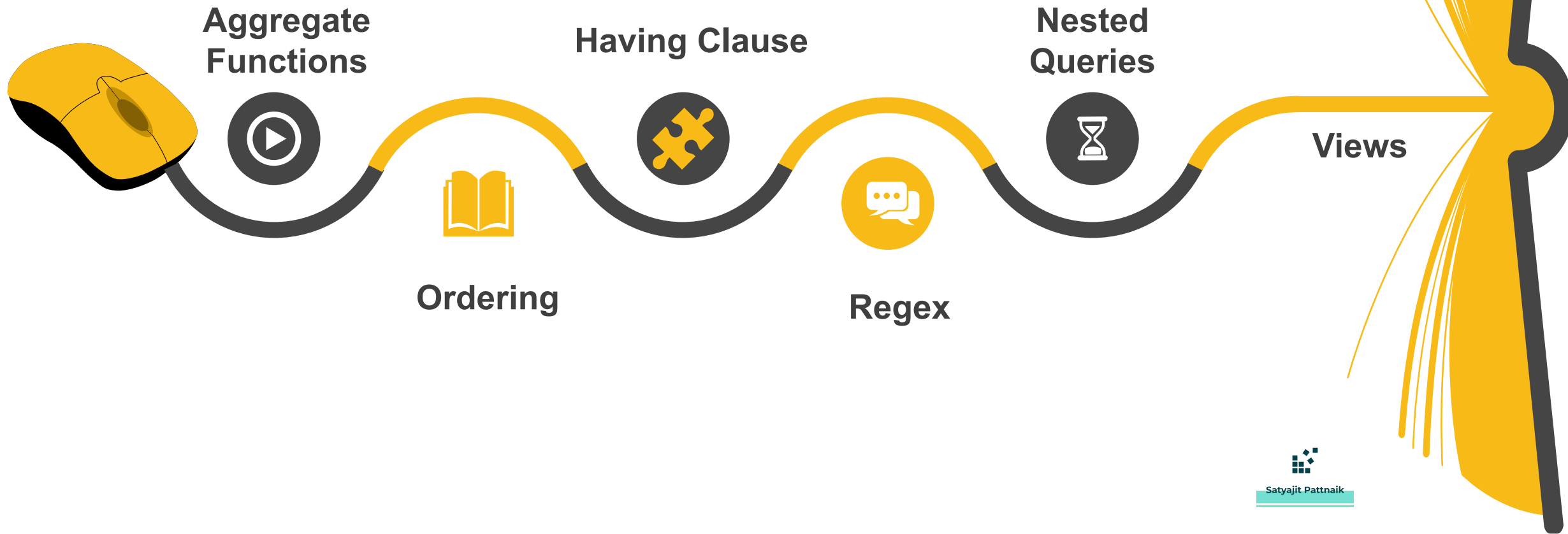


***SQL FOR***

***DATA SCIENCE***

# SQL Timeline Continued\*\*





# SQL Continued\*

## Recap

- ✓ Introduction to Data Science, Machine Learning
- ✓ What is SQL & Basics of SQL
- ✓ SELECT, UPDATE, INSERT, DELETE Queries
- ✓ Joins





# SQL Continued\*



Satyajit Pattnaik

## Aggregate Functions

- ✓ Sometimes we examine & analyse data of varying magnitudes, hence we realise the need of grouping similar types of values together & look them at as one bunch.

**For example:** Consider a table containing data consisting of the marks scored by students in their 12th board exams. While you would want to know how the students performed in all the subjects put together, it is equally important to see how they performed in each subject. You can gain even further insights if you group these students by state. Hence, it is imperative that you learn the usage of aggregate functions in your queries.

- ✓ `groupby()` → To aggregate values of a column C1 'grouped by' a certain column C2.
- ✓ `count()` → Count the number of rows.
- ✓ `min()`, `max()` → Finding the minimum & maximum values for a particular column

**For example:** `select min(score) from employee;`

- ✓ `avg()` → Find the average

# ORDERING



Satyajit Pattnaik

Quite often, you would want to display the retrieved records in a particular order, for example, in increasing order of income, joining date, alphabetical order, etc. This is commonly useful when you are making a report or presenting the data to someone else.



```
select firstName from  
employees order by  
firstName asc limit 3;
```

# HAVING CLAUSE



Satyajit Pattnaik

You have already learnt how to filter individual values based on a given condition.

But how do you do this on grouped values?

Suppose your manager asks you to count all the employees whose salaries are more than the average salary in that particular department.

Now, intuitively, you know that two aggregate functions would be used here, namely, `count()` and `avg()`. You decide to apply the 'where' condition on the average salary of the department, but to your surprise, the query fails. This is exactly what the having clause is for.

The 'having' clause is typically used when you have to apply a filter condition on an 'aggregated value'. This is because the 'where' clause is applied before aggregation takes place, and thus, it is not useful when you want to apply a filter on an aggregated value.



# Exercise



Satyajit Pattnaik

## Keywords in SQL

List the keywords below in order of their occurrence in a query:

1. group by
2. order by
3. select
4. where
5. from
6. limit
7. having

Answer is:

3 - 5 - 4 - 1 - 7 - 2 - 6



# STRING FUNCTIONS



Satyajit Pattnaik

Used to manipulate the string data and make it more understandable for analysis.

For example: amitabhbachchan, or Amitabh Bachchan, which one of them is more readable, obviously the later one right.

concat → Concatenates two strings

substr/substring → Extracts a substring from a string

upper → Converts a string to uppercase

lower → Converts a string to lowercase

character\_length → Calculates the length of the string variable.

[https://www.w3schools.com/sql/sql\\_ref\\_mysql.asp](https://www.w3schools.com/sql/sql_ref_mysql.asp)





# DATE & TIME FUNCTIONS



Satyajit Pattnaik

Used to manipulate the date & time columns

For example: If you want to change the date format, or just want to see the exact day, or so on.

datediff → Return the number of days between the two date values

date\_format → Format a date variable

day → Return the day of the month for a date.

quarter → Return the quarter of the year for a date.

and so on....

[https://www.w3schools.com/sql/func\\_mysql\\_adddate.asp](https://www.w3schools.com/sql/func_mysql_adddate.asp)



# REGEX



Satyajit Pattnaik

So far you have already known about the wildcards like “like” operator, but in cases wildcards may fall short for some advanced use cases, regular expressions comes into picture.

Regex, or Regular expressions, is a sequence of characters, used to search and locate specific sequences of characters that match a pattern.

Example 1: Match beginning of the name

```
select * from customer WHERE email REGEXP '^w'
```



# REGEX



Satyajit Pattnaik

Example 2: Find the customers, which email address having a 'z', 'v' or 'p'

```
SELECT * FROM customer  
WHERE email REGEXP "[zvp]";
```

Example 3: Find the customers, which email address containing characters from 'x' to 'z'

```
SELECT * FROM customer  
WHERE email REGEXP "[x-z]";
```



# NESTED QUERIES



Satyajit Pattnaik

By now, you know that a database is a collection of multiple related tables. Now, while generating insights from data, you may need to refer to these multiple tables in a query. There are two ways to deal with such types of queries. These include the following:

1. Joins
2. Nested queries/Subqueries

A subquery is called an inner query while the query that contains the subquery is called an outer query. A subquery can be used anywhere that expression is used and must be closed in parentheses.

Outer Query

```
SELECT lastName,firstName
FROM employees
WHERE
    office_Code IN (SELECT
        office_Code
        FROM
        offices
        WHERE
        country = 'USA');
```

Inner Query



LEARNING

# Exercise



Satyajit Pattnaik

Write a query to retrieve the names of all employees who have an age greater than what Gus Gray has.

Answer is:



# Views



Satyajit Pattnaik

Views are virtual tables that do not store any data of their own but display data stored in other tables.

Advantages:

- Hide the complexity of data.
- Act as aggregated tables.
- If you are doing an user level access control, you can give an user access to a view without giving access to the tables behind it
- It can allow for massive performance improvements.

```
create view newEmp as
SELECT lastName,firstName
FROM employees
WHERE
    office_Code IN (SELECT
                    office_Code
                    FROM
                    offices
                    WHERE
                    country = 'USA');
```



LEARNING

# Views



Satyajit Pattnaik

## Disadvantages:

- If done wrong, it can result in performance issues.
- You may not be able to update the view, forcing you back to the original tables.

```
create view newEmp as
SELECT lastName,firstName
FROM employees
WHERE
    office_Code IN (SELECT
        office_Code
        FROM
        offices
        WHERE
        country = 'USA');
```



LEARNING

# Exercise



Satyajit Pattnaik

Write a query to retrieve the names of all employees who have an age greater than what Gus Gray has, and store the output in a view.

Answer is:





# Summary



Satyajit Pattnaik

You learnt about the crux of SQL queries. These are the basic to intermediate level concepts of SQL that will be tested rigorously in your interviews. Hence, practice these points again and again to gain a deep understanding of the several SQL statements.

Leave your top three takeaways from this session in the comments section below.





***THANKS***  
***FOR WATCHING!!***