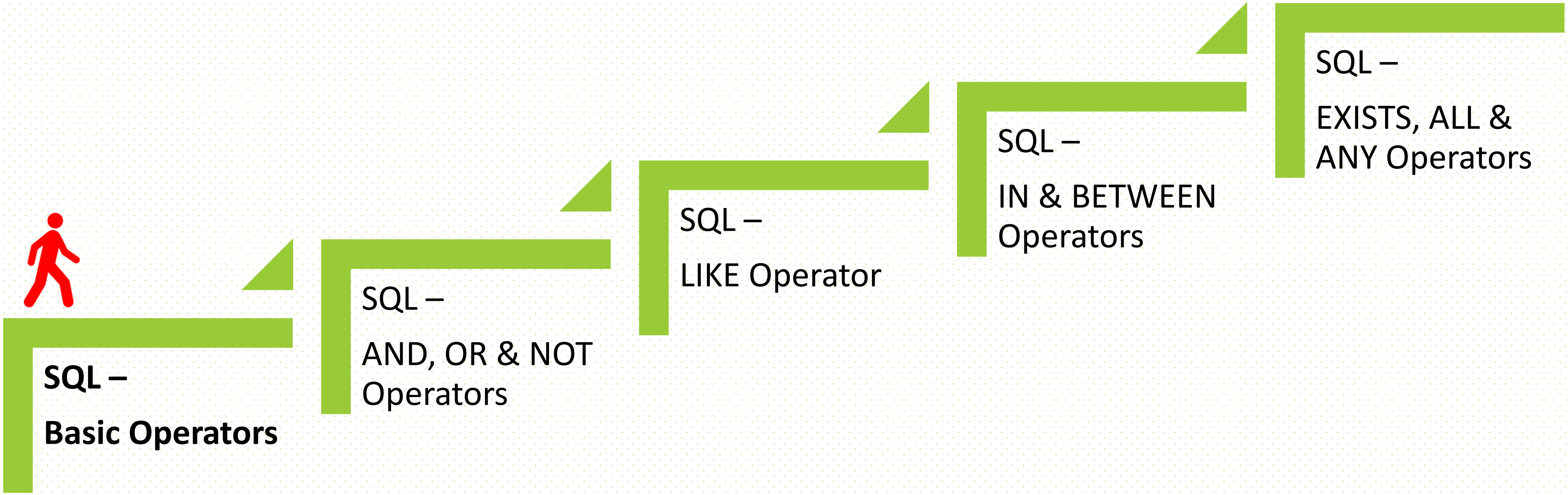
SQL
Session - 04

Learning Objectives



SQL –

Basic Operators

SQL –

AND, OR & NOT Operators

SQL –

LIKE Operator

SQL –

IN & BETWEEN Operators

SQL –

EXISTS, ALL & ANY Operators

SQL – Operators

Arithmetic Operators	
Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulo

Comparison Operators	
Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

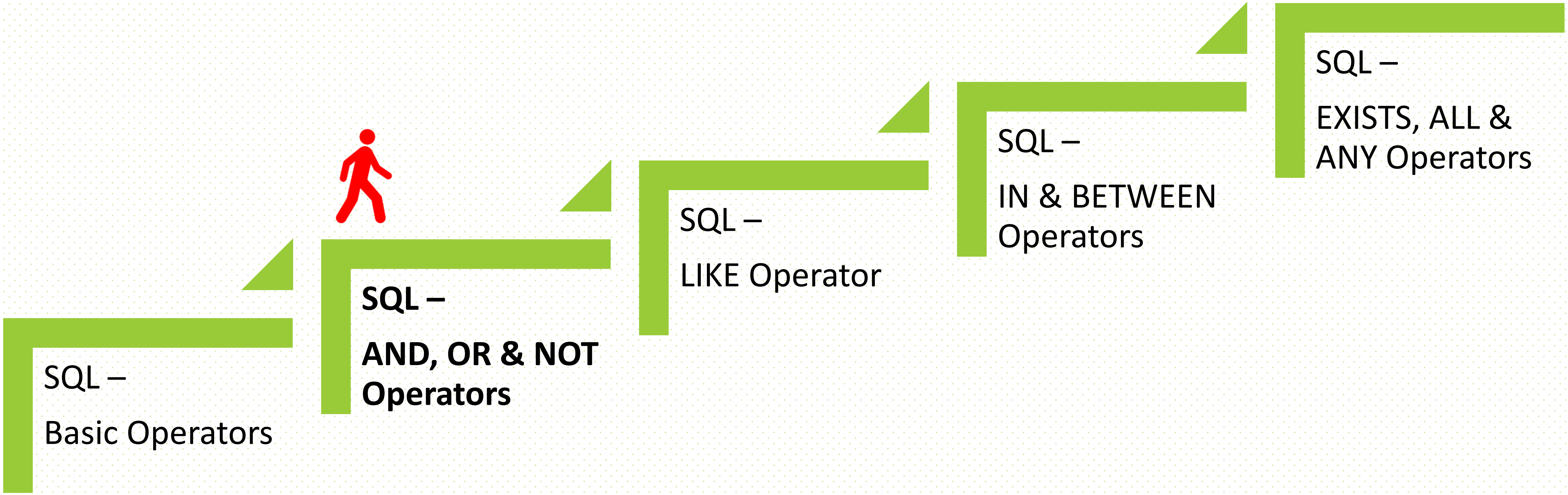
Logical Operators	
Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to one of a list of expressions
LIKE	TRUE if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE
OR	TRUE if any of the conditions separated by OR is TRUE
SOME	TRUE if any of the subquery values meet the condition

SQL – Operators

Bitwise Operators	
Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR

Compound Operators	
Operator	Description
+=	Add equals
-=	Subtract equals
*=	Multiply equals
/=	Divide equals
%=	Modulo equals
&=	Bitwise AND equals
^-=	Bitwise exclusive equals
|*=	Bitwise OR equals

Learning Objectives



SQL –

Basic Operators

SQL –

AND, OR & NOT Operators

SQL –

LIKE Operator

SQL –

IN & BETWEEN Operators

SQL –

EXISTS, ALL & ANY Operators

SQL – AND, OR and NOT Operator

Syntax

```
SELECT <column-name(s)>
FROM <table-name>
WHERE condition1 AND condition2 AND condition3 ...;
```

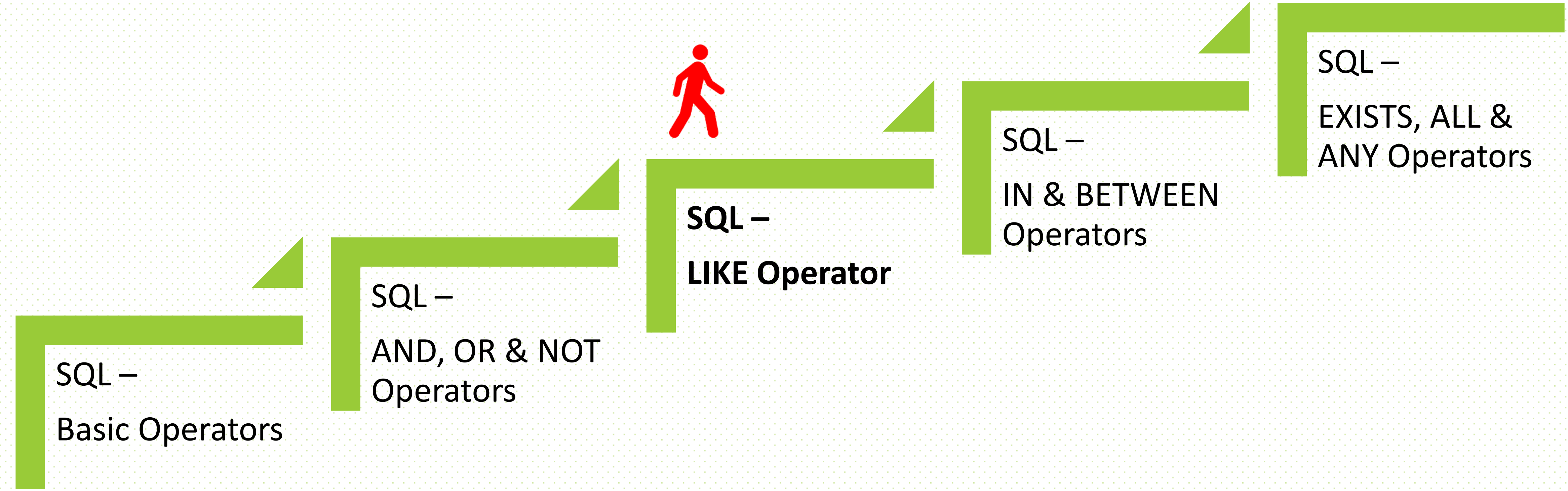
```
SELECT <column-name(s)>
FROM <table-name>
WHERE condition1 OR condition2 OR condition3 ...;
```

```
SELECT <column-name(s)>
FROM <table-name>
WHERE NOT condition;
```

SQL - Tasks

1. Display
   a. all customers.
   b. all customers from ‘France’.
   c. all customers from Country ‘France’ or City ‘Berlin’.
   d. all customers from ‘Germany’.
   e. all customers from "Germany" AND "Berlin“.
   f. all customers from city "Berlin" OR "München“
   g. all customers from "Germany" and city "Berlin" OR "München"
   h. all customers who does not belong to ‘Germany’.
   i. all customers whose country is NOT "Germany" and NOT "USA"
2. Display names of all products which are costlier than $100.

Learning Objectives



SQL –

Basic Operators

SQL –

AND, OR & NOT Operators

SQL –

LIKE Operator

SQL –

IN & BETWEEN Operators

SQL –

EXISTS, ALL & ANY Operators

SQL – LIKE Operator

Syntax

SELECT *column1, column2*, ...
FROM <*table-name*>
WHERE *columnN* LIKE pattern;

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column. There are two wildcards used in conjunction with the LIKE operator:

% - The percent sign represents zero, one, or multiple characters

_ - The underscore represents a single character

SQL – LIKE (Examples)

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE CustomerName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

SQL – LIKE (Examples)

LIKE Operator	Description
WHERE CustomerName LIKE '[bsp]%'	Finds any values that start with "b", "s", or "p"
WHERE CustomerName LIKE '[a-c]%'	Finds any values that starts with "a", "b", or "c"

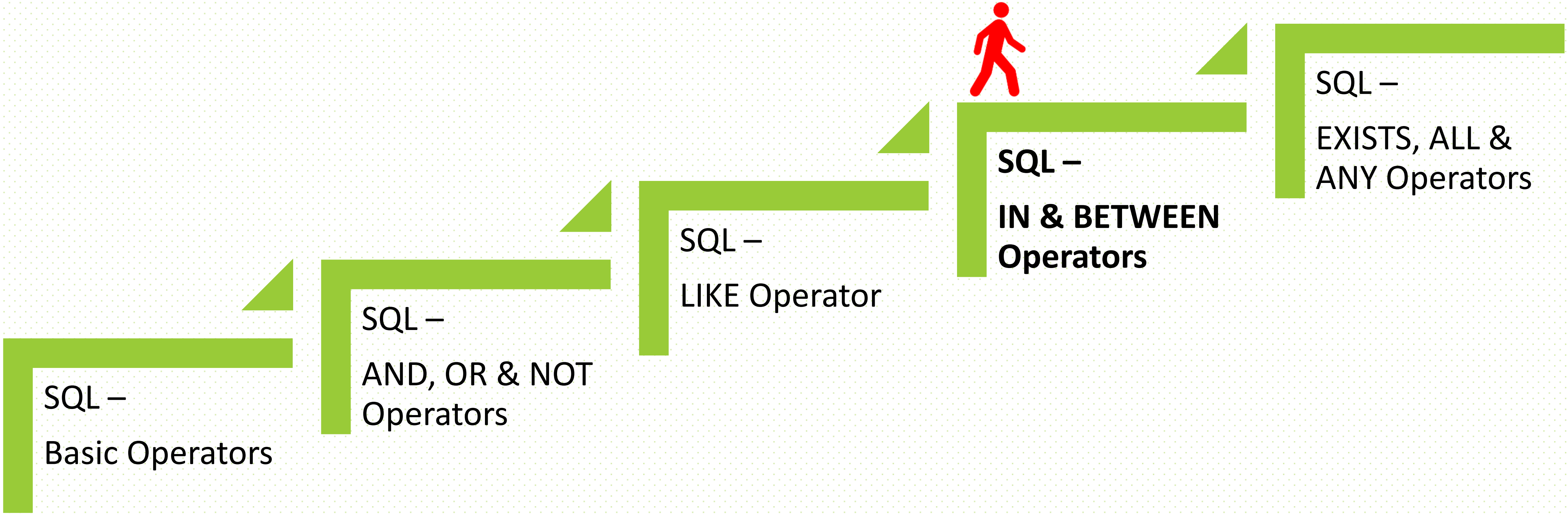
WHERE CustomerName LIKE '[!bsp]%’
WHERE CustomerName NOT LIKE '[bsp]%’
	Finds any values that does not start with "b", "s", or "p"

SQL - Tasks

Select all customers

a. with a CustomerName starting with "a“.
b. with a CustomerName ending with "a“.
c. with a CustomerName that have "or" in any position.
d. with a CustomerName that have "r" in the second position.
e. with a CustomerName that starts with "a" and are at least 3 characters in length.
f. with a CustomerName that starts with "a" and ends with "o“.
g. with a CustomerName that does NOT start with "a“.
h. with a City starting with "ber“.
i. with a City containing the pattern "es“.
j. with a City starting with any character, followed by "erlin“.
k. with a City starting with "L", followed by any character, followed by "n", followed by any character, followed by "on“.
l. with a City starting with "b", "s", or "p“.
m. with a City starting with "a", "b", or "c“.
n. with a City NOT starting with "b", "s", or "p“.

Learning Objectives



SQL –

Basic Operators

SQL –

AND, OR & NOT Operators

SQL –

LIKE Operator

SQL –

IN & BETWEEN Operators

SQL –

EXISTS, ALL & ANY Operators

SQL – IN Operator

Syntax

```
SELECT <column-name(s)>
FROM <table-name>
WHERE <column-name> IN (value1, value2, …) ;
```

```
SELECT <column-name(s)>
FROM <table-name>
WHERE <column-name> IN (SELECT STATEMENT);;
```

SQL – BETWEEN Operator

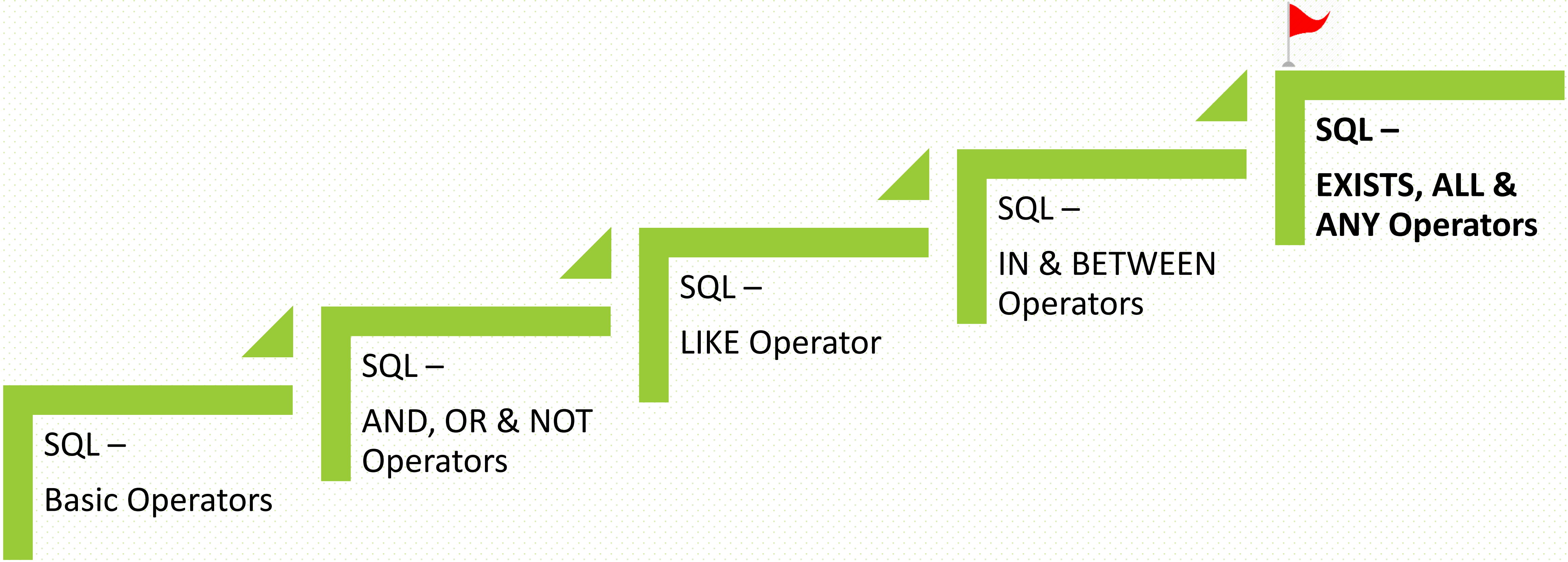
Syntax

```
SELECT <column-name(s)>
FROM <table-name>
WHERE <column-name> BETWEEN value1 AND value2;
```

SQL - Tasks

1. Select all customers
   a. that are located in "Germany", "France" and "UK“.
   b. that are NOT located in "Germany", "France" or "UK“.
   c. that are from the same countries as the suppliers.
2. Select all products with a price BETWEEN 10 and 20.
3. Select all the products outside the range of 10 to 20, use NOT BETWEEN.
4. Select all products with a price BETWEEN 10 and 20. In addition; do not show products with a CategoryID of 1,2, or 3.
5. Select all products with a ProductName BETWEEN 'Carnarvon Tigers' and 'Mozzarella di Giovanni’.
6. Select all products with a ProductName NOT BETWEEN 'Carnarvon Tigers' and 'Mozzarella di Giovanni’.
7. Select all orders with an OrderDate BETWEEN '04-July-1996' and '09-July-1996’.

Learning Objectives



SQL –

Basic Operators

SQL –

AND, OR & NOT Operators

SQL –

LIKE Operator

SQL –

IN & BETWEEN Operators

SQL –

EXISTS, ALL & ANY Operators

SQL – EXISTS Operator

Syntax

```
SELECT <column-name(s)>
FROM <table-name>
WHERE EXISTS
(SELECT <column-name> FROM <table-name> WHERE condition);
```

SQL – ANY and ALL Operator

Syntax

```
SELECT <column-name(s)>
FROM <table-name>
WHERE <column-name> operator ANY
(SELECT <column-name> FROM <table-name> WHERE condition);
```

```
SELECT <column-name(s)>
FROM <table-name>
WHERE <column-name> operator ALL
(SELECT <column-name> FROM <table-name> WHERE condition);
```

Note: The *operator* must be a standard comparison operator (=, <>, !=, >, >=, <, or <=).

SQL - Tasks

1. List the suppliers with a product price less than 20 that are located in "Germany", "France" and "UK“.
2. List the suppliers with a product price equal to 22.
3. List the ProductName if it finds ANY records in the Order Details table that quantity = 10.
4. List the ProductName if it finds ANY records in the Order Details table that quantity > 99.
5. List the ProductName if ALL the records in the Order Details table has quantity = 10.