HOW TO Filter SQL Queries

Sometimes you want the result set to be different than the data returned by a simple SELECT statement.

ORDER BY

Allows sorting of result set

After the WHERE clause (if there is one)

Specify one or more columns

Separate columns by commas

ASC (default) or DESC

Who are all the people in my contact list, ordered by last name?

```
SELECT p.last_name,
p.first_name
FROM person p
ORDER BY p.last name;
```

- SELECT CLAUSE
- FROM CLAUSE
- ORDER BY CLAUSE

Set Function

Computes new values from column values

Use in place of columns in SELECT clause

Passes column name to function

Helps us to ask more interesting questions

Often used with the DISTINCT qualifier

Set Functions

Function

COUNT	Count of the column specified (includes NULL values if * is used)
MAX	Maximum value of the column (does not include NULL values)
MIN	Minimum value of the column (does not include NULL values)
AVG	Average of all values of the column (does not include NULL values, only numeric column)
SUM	Sum of all the values of the column (does not include NULL values, only numeric column)

What is the total number of times I’ve contacted my contacts?

```
SELECT
SUM( p.contacted_number)
FROM person p;
```

SELECT CLAUSE WITH THE SUM SET FUNCTION

Set Functions + Qualifiers

Often used together

Add inside of the function

Run against DISTINCT column values

Review the DISTINCT qualifier section from Module 3 if needed

What is the count of unique first names among my contacts?

SELECT CLAUSE WITH THE COUNT SET FUNCTION + DISTINCT QUALIFIER

```
SELECT
COUNT(DISTINCT p.first_name)
FROM person p;
```

GROUP BY

Allows multiple columns with a set function

Breaks result set into subsets

Runs set function against each subset

Result set returns 1 row per subset

Subset is dictated by column in GROUP BY

Column must appear in the SELECT LIST

Appears after FROM and/or WHERE Clauses

What is the count of every unique first name among my contacts?

```
SELECT
COUNT(p.first_name),
p.first_name
FROM person p
GROUP BY p.first_name;
```

- ◀ SELECT CLAUSE WITH THE COUNT SET FUNCTION
- ◀ GROUP BY COLUMN in SELECT LIST
- ◀ GROUP BY CLAUSE

HAVING

Works like WHERE works against SELECT

Restricts the result set

```
SELECT
COUNT(DISTINCT p.first_name),
p.first_name
FROM person p
GROUP BY p.first_name

HAVING COUNT(DISTINCT
p.first_name) >= 5;
```

What is the count of unique first names among my contacts that appear at least 5 times?

- SELECT CLAUSE WITH THE COUNT SET FUNCTION
- GROUP BY COLUMN in SELECT LIST

- HAVING CLAUSE

Demo

WHERE

Constrains the result set

Comes after the FROM clause

Contains boolean expressions

Only matching rows are in the result set

```
SELECT p.last_name
FROM person p;
WHERE p.first_name = 'Jon'
```

What is the last name of all the people I know whose first name is Jon?

- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE

Boolean Operators

Operator

=	Equals	True if values on both sides are equal
<>	Not Equal TO	True if value on both sides are not equal
>	Greater Than	True if left side is larger than right side
<	Less Than	True if left side is smaller than right side
>=	Greater or Equal	True if left side is larger or equal to right
<=	Less Than or Equal	True if left side is smaller or equal to right

A Single Expression is Quite Limiting

We’d like to ask more complex questions

Additional keywords are needed

These can chain multiple expressions

AND

Combines two expressions

If both are TRUE, row is included

If either is FALSE, row is excluded

Who are all the people in my contact list that have the first name Jon and have a birthday later than 1965?

```
SELECT p.first_name,
p.last_name
FROM person p
WHERE p.first_name = ‘Jon’
AND p.birthdate >
’12/31/1965’;
```

◀ SELECT CLAUSE

◀ FROM CLAUSE

◀ WHERE CLAUSE

◀ AND

OR

Also combines two expressions

If either are TRUE, row is included

If both are FALSE, row is excluded

Who are all the people in my contact list that have the first name Jon or a last name of Flanders?

```
 SELECT p.first_name,
 p.last_name
 FROM person p
WHERE p.first_name = ‘Jon’
OR p.last_name = ‘Flanders’;
```

- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE
- OR

Other Operators

- BETWEEN
- LIKE
- IN
- IS
- IS NOT

BETWEEN

Acts on column and two values

TRUE if column value is between two values

Inclusive – includes two values (like >= & <=)

Who are all the people in my contact list that I have contacted at least once but no more than 20 times?

```
SELECT p.first_name,
p.last_name
FROM person p

WHERE p.contacted
BETWEEN 1 AND 20;
```

- SELECT CLAUSE
- FROM CLAUSE

- WHERE CLAUSE
- BETWEEN

LIKE

A more fuzzy version of =

String with special characters inside

If the match is true, the row is returned

Who are all the people in my contact list that have a first name that begins with the letter J?

```
SELECT p.first_name,
p.last_name
FROM person p

WHERE p.first_name
LIKE ‘J%’;
```

- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE
- LIKE

IN

Like a multi-value = operator

List of potential values

True if any of the values in the list “hit”

Who are all the people in my contact list that are named Jon or Fritz?

```
SELECT p.first_name,
p.last_name
FROM person p
WHERE p.first_name
IN (‘Jon’,’Fritz’);
```

- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE
- IN

IS

Special operator

Like a equals operator

But just for values that might be NULL

Who are all the people in my contact list that don’t have a last name?

```
SELECT p.first_name,
p.last_name
FROM person p

WHERE p.last_name
IS NULL;
```

- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE
- IS

IS NOT

Also just for NULL

Like a “NOT EQUALS” operator

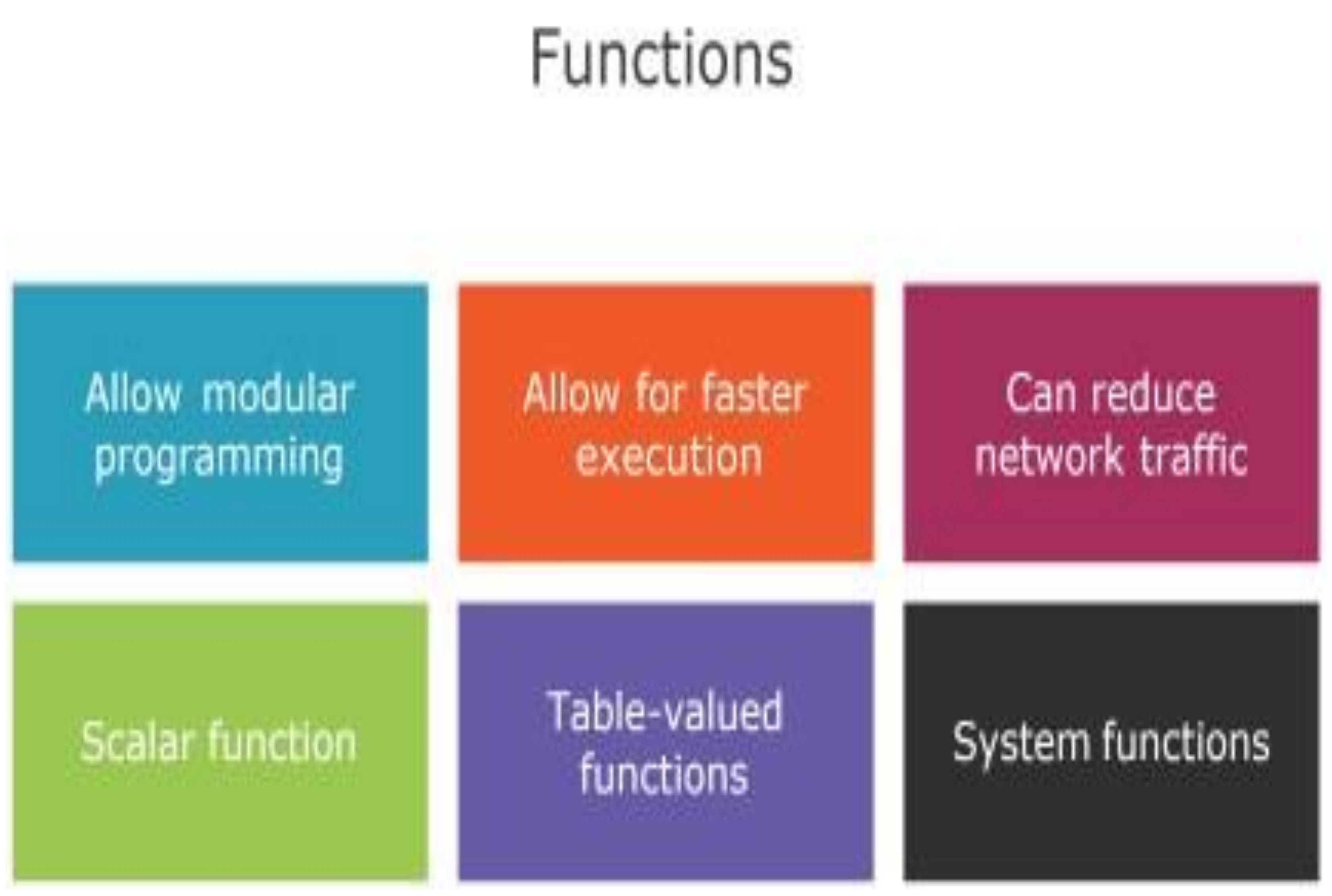
Who are all the people in my contact list that have a last name?

```
SELECT p.first_name,
p.last_name
FROM person p
WHERE p.last_name
IS NOT NULL;
```

- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE
- IS NOT

Functions

Function are routines that accept parameters, perform an action, and return the result of that action as a value



Functions

Allow modular programming

Allow for faster execution

Can reduce network traffic

Scalar function

Table-valued functions

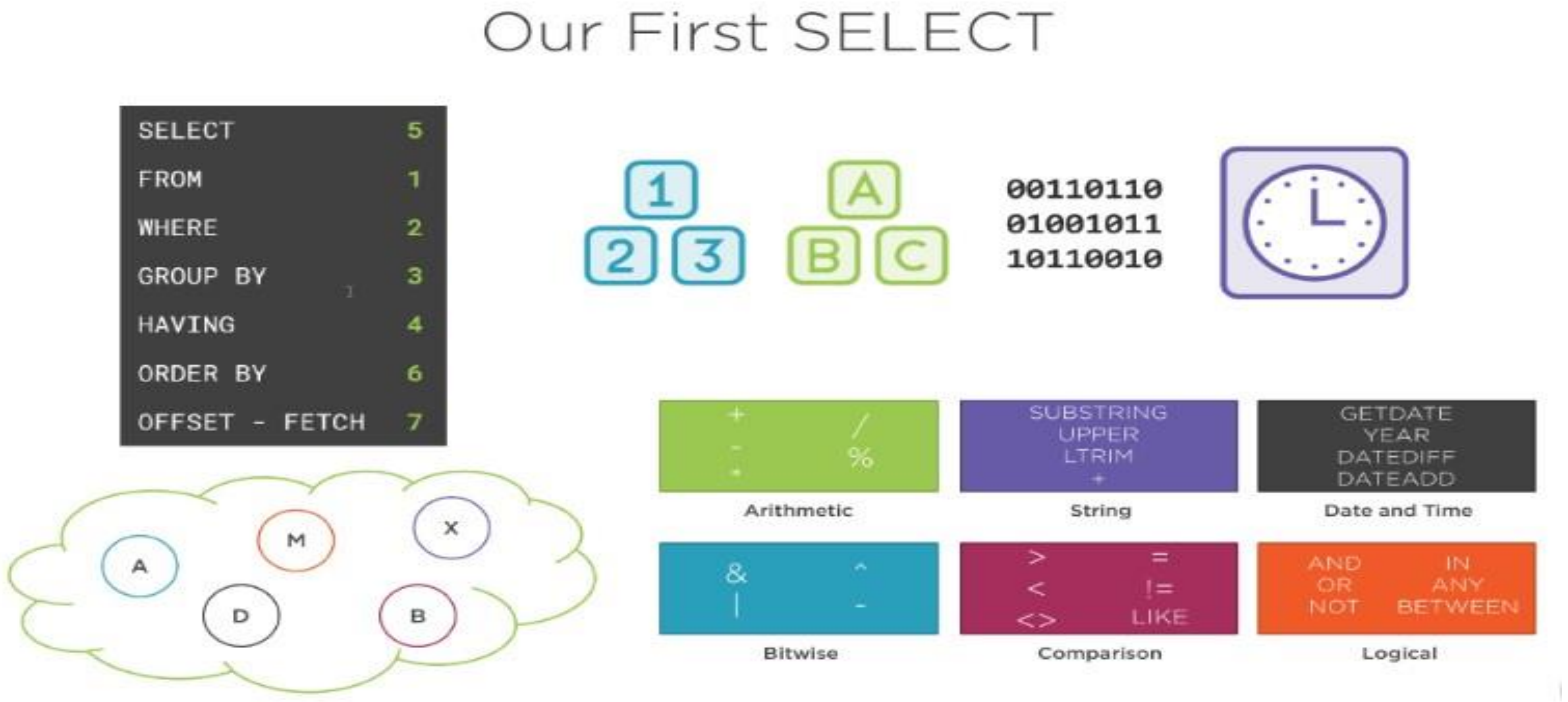
System functions

Aggregate functions operate on a set of elements, and return a single value.

Common Aggregate Functions

Minimum	Maximum	Average
Count	Standard deviation	Variance

SQL Server Operators & Functions



Our First SELECT

SELECT	5
FROM	1
WHERE	2
GROUP BY	3
HAVING	4
ORDER BY	6
OFFSET - FETCH	7

Arithmetic: + - * / %

String: SUBSTRING UPPER LTRIM +

Date and Time: GETDATE YEAR DATEDIFF DATEADD

Bitwise: & | ^ -

Comparison: > < <> = != LIKE

Logical: AND OR NOT IN ANY BETWEEN

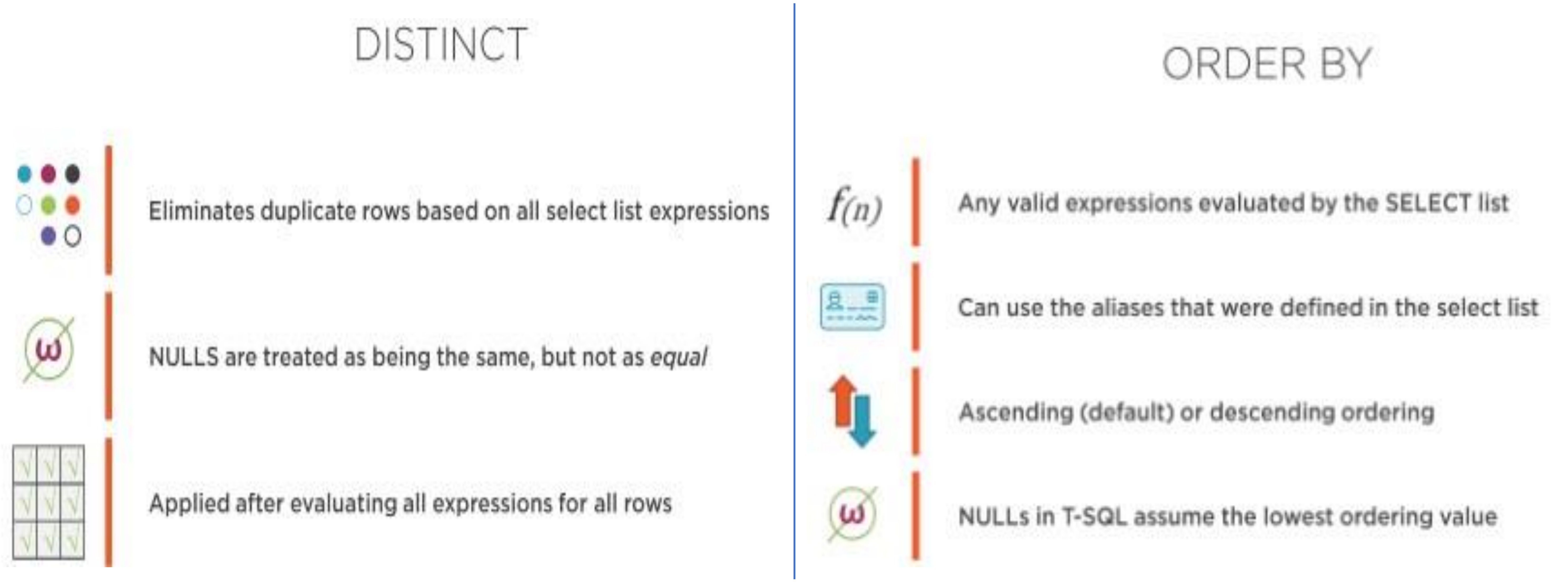
Predicate	True when
`X > ALL (A, B, C)`	X > A AND X > B AND X > C
`X > ANY|SOME (A, B, C)`	X > A OR X > B OR X > C
`X IN (A, B, C)`	X = A OR X = B OR X = C
`X NOT IN (A, B, C)`	X <> A AND X<> B AND X<>C
`X BETWEEN A AND B`	X >= A AND X <= B
`X LIKE ( <pattern> )`	X matches wildcard pattern

NULL Predicates

X = NULL	X <> NULL	X IS NULL	X IS NOT NULL
Always unknown	Always unknown	True if X is null False if not Never unknown	False if X is null True if not Never unknown

Additional Logical Operators

ALL	ANY / SOME	BETWEEN
EXISTS	IN	LIKE



DISTINCT

- Eliminates duplicate rows based on all select list expressions
- NULLS are treated as being the same, but not as *equal*
- Applied after evaluating all expressions for all rows

ORDER BY

- Any valid expressions evaluated by the SELECT list
- Can use the aliases that were defined in the select list
- Ascending (default) or descending ordering
- NULLs in T-SQL assume the lowest ordering value