SQL Subqueries

365√DataScience

SQL Subqueries with IN Nested Inside WHERE

365√DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries

queries embedded in a query

SQL Subqueries with IN Nested Inside WHERE

- subqueries

SQL Subqueries with IN Nested Inside WHERE

subqueries

queries embedded in a query

SQL Subqueries with IN Nested Inside WHERE

- subqueries = inner queries

queries embedded in a query

SQL Subqueries with IN Nested Inside WHERE

- **subqueries** = **inner queries** = **nested queries**

queries embedded in a query

SQL Subqueries with IN Nested Inside WHERE

● <u>subqueries</u> = <u>inner queries</u> = <u>nested queries</u>

queries embedded in a query

- they are part of *another* query, called an <u>outer query</u>

SQL Subqueries with IN Nested Inside WHERE

- subqueries = inner queries = nested queries = inner select

queries embedded in a query

- they are part of *another* query, called an outer query

SQL Subqueries with IN Nested Inside WHERE

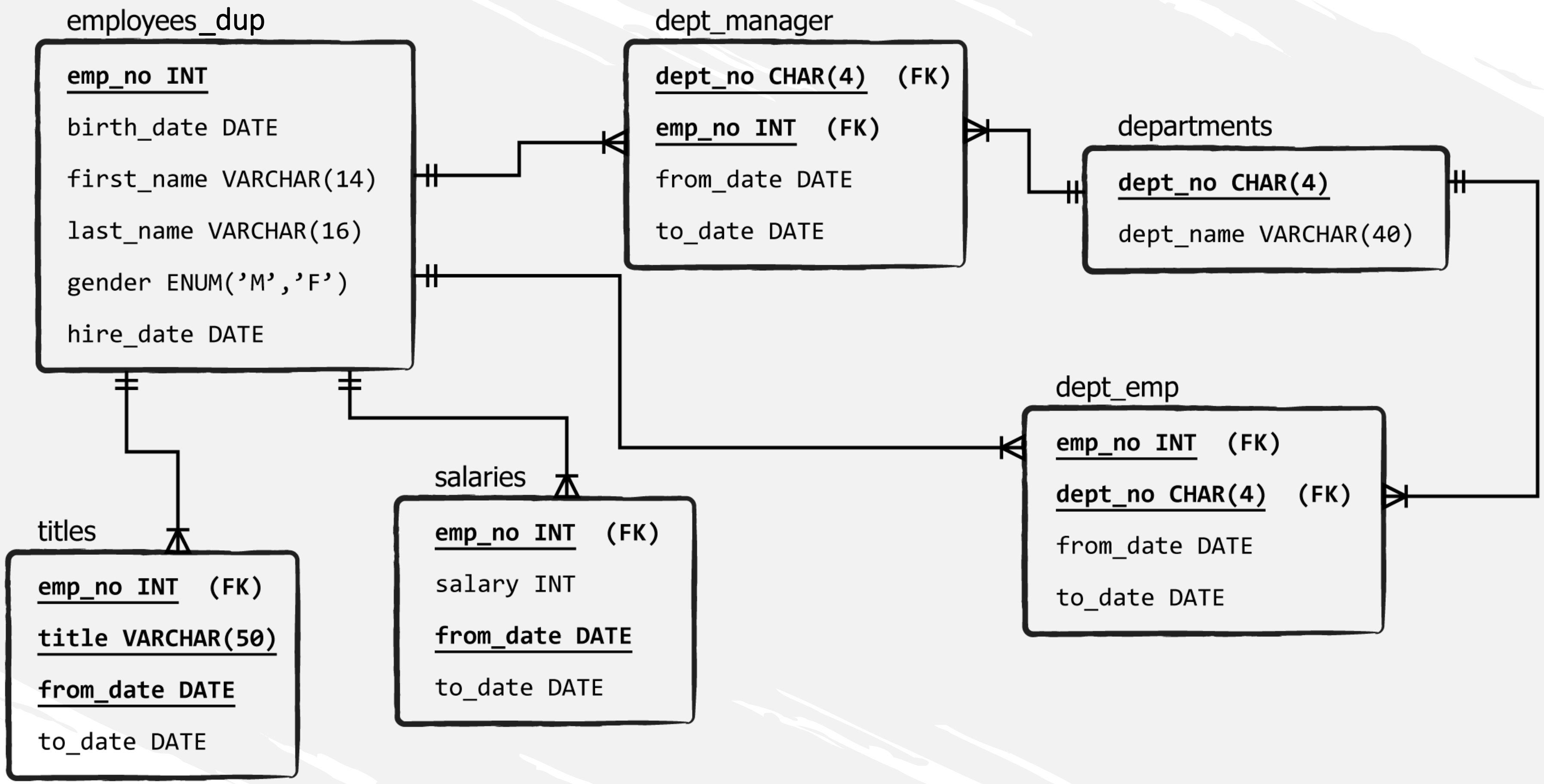
- <u>subqueries</u> = <u>inner queries</u> = <u>nested queries</u> = <u>inner select</u>

queries embedded in a query

- they are part of *another* query, called an <u>outer query</u>

= <u>outer select</u>

SQL Subqueries with IN Nested Inside WHERE



employees_dup

emp_no INT
birth_date DATE
first_name VARCHAR(14)
last_name VARCHAR(16)
gender ENUM('M','F')
hire_date DATE

dept_manager

dept_no CHAR(4) (FK)
emp_no INT (FK)
from_date DATE
to_date DATE

departments

dept_no CHAR(4)
dept_name VARCHAR(40)

dept_emp

emp_no INT (FK)
dept_no CHAR(4) (FK)
from_date DATE
to_date DATE

salaries

emp_no INT (FK)
salary INT
from_date DATE
to_date DATE

titles

emp_no INT (FK)
title VARCHAR(50)
from_date DATE
to_date DATE

SQL Subqueries with IN Nested Inside WHERE

- subqueries

SQL Subqueries with IN Nested Inside WHERE

- <u>subqueries</u>

  - a subquery should *always* be placed within parentheses

SQL Subqueries with IN Nested Inside WHERE

1) the SQL engine starts by running the *inner query*

SQL Subqueries with IN Nested Inside WHERE

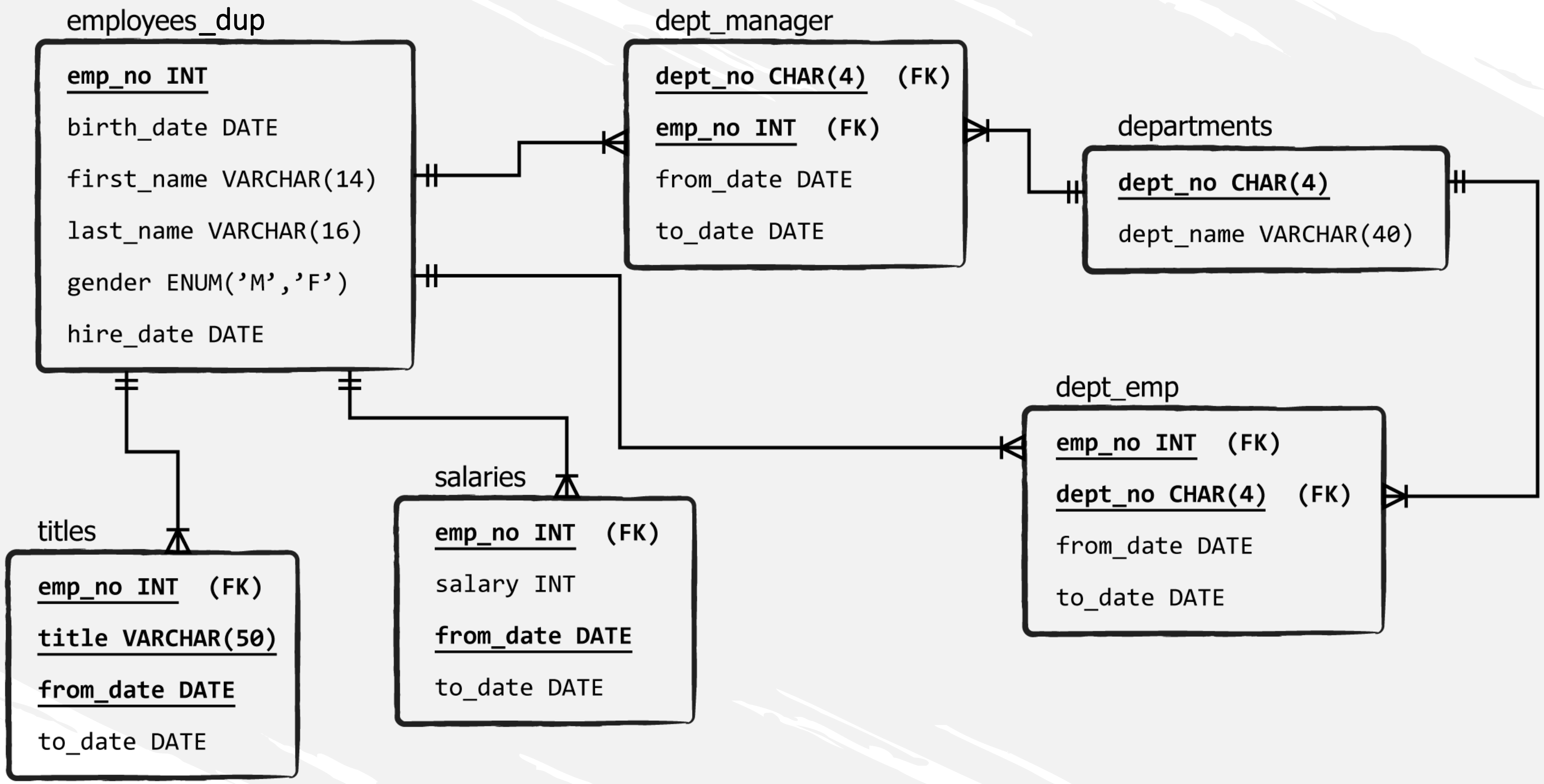
- 1) the SQL engine starts by running the *inner query*

- 2) then it uses its returned output, which is intermediate, to execute the *outer query*

SQL Subqueries with IN Nested Inside WHERE

- a subquery may return *a single value (a scalar)*, *a single row*, *a single column*, or *an entire table*

SQL Subqueries with IN Nested Inside WHERE



employees_dup

emp_no INT

birth_date DATE

first_name VARCHAR(14)

last_name VARCHAR(16)

gender ENUM('M','F')

hire_date DATE

dept_manager

dept_no CHAR(4) (FK)

emp_no INT (FK)

from_date DATE

to_date DATE

departments

dept_no CHAR(4)

dept_name VARCHAR(40)

dept_emp

emp_no INT (FK)

dept_no CHAR(4) (FK)

from_date DATE

to_date DATE

salaries

emp_no INT (FK)

salary INT

from_date DATE

to_date DATE

titles

emp_no INT (FK)

title VARCHAR(50)

from_date DATE

to_date DATE

SQL Subqueries with IN Nested Inside WHERE

- a subquery may return *a single value (a scalar), a single row, a single column,* or *an entire table*

SQL Subqueries with IN Nested Inside WHERE

- a subquery may return *a single value (a scalar), a single row, a single column*, or *an entire table*

  - you can have a lot more than one subquery in your outer query

SQL Subqueries with IN Nested Inside WHERE

- a subquery may return *a single value (a scalar)*, *a single row*, *a single column*, or *an entire table*

  - you can have a lot more than one subquery in your outer query

  - it is possible to nest *inner queries within other inner queries*

SQL Subqueries with IN Nested Inside WHERE

- a subquery may return *a single value (a scalar)*, *a single row*, *a single column*, or *an entire table*

- you can have a lot more than one subquery in your outer query

- it is possible to nest *inner queries within other inner queries*

  in that case, the SQL engine would execute the *innermost query first*

SQL Subqueries with IN Nested Inside WHERE

- <u>a subquery</u> may return *a single value (a scalar)*, *a single row*, *a single column*, or *an entire table*

- you can have a lot more than one subquery in your outer query

- it is possible to nest *inner queries within other inner queries*

in that case, the SQL engine would execute the *innermost query first*, and then *each subsequent query*

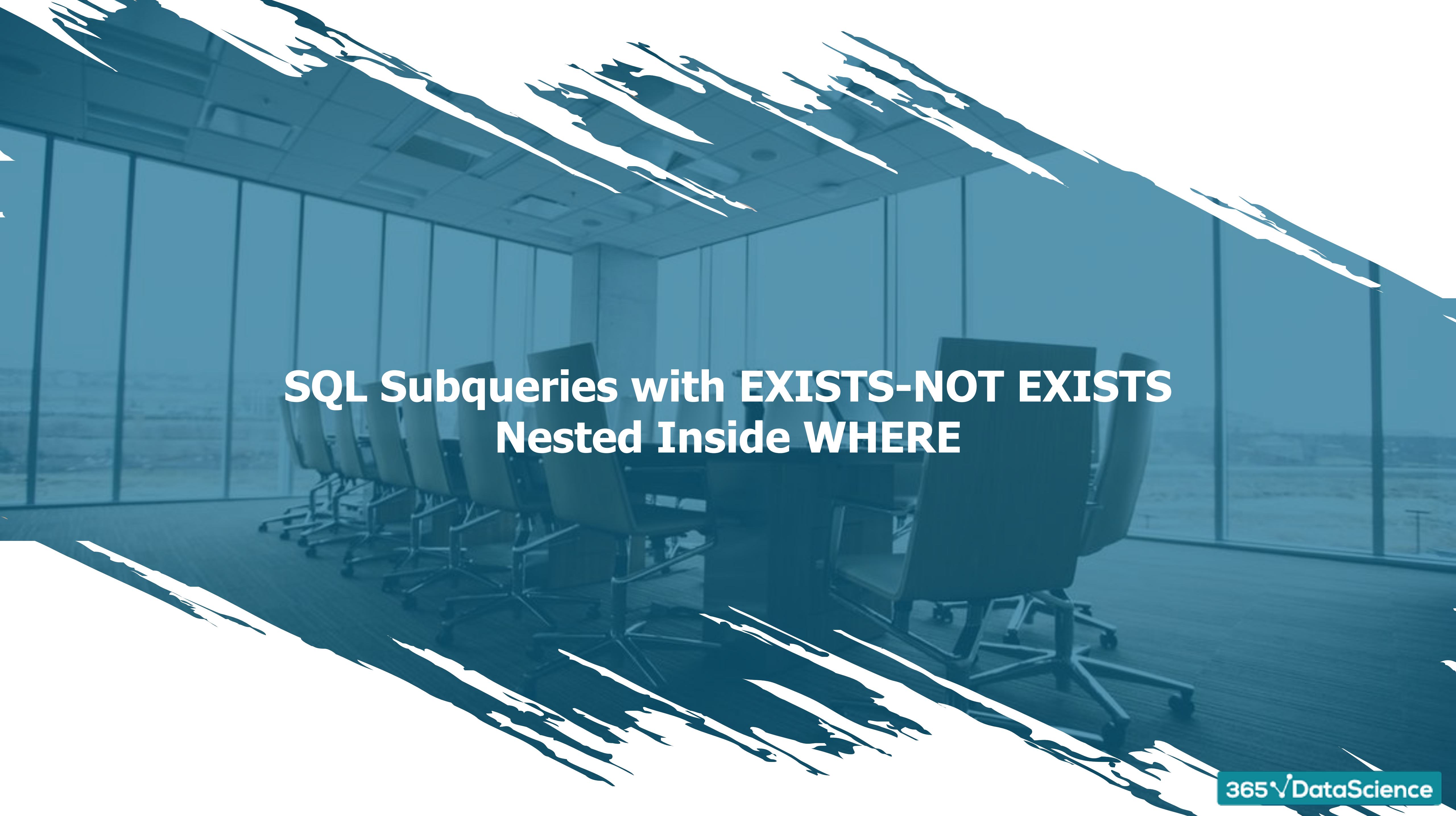
SQL Subqueries with IN Nested Inside WHERE

- a subquery may return *a single value (a scalar)*, *a single row*, *a single column*, or *an entire table*

- you can have a lot more than one subquery in your outer query

- it is possible to nest *inner queries within other inner queries*

  in that case, the SQL engine would execute the *innermost query first*, and then *each subsequent query*, until it runs the *outermost query last*



SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- EXISTS

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted ***row by row***

- **it returns a Boolean value**

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*

- it returns a Boolean value

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*

- it returns a Boolean value

if a row value of a subquery **exists**

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*

- it returns a Boolean value

---

if a row value of a subquery **exists** → **TRUE**

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*
- it returns a Boolean value

if a row value of a subquery **exists** → **TRUE** → *the corresponding record of the outer query is extracted*

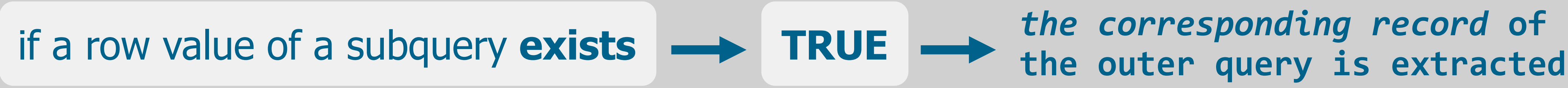
SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*

- it returns a Boolean value



if a row value of a subquery **exists** → **TRUE** → *the corresponding record of the outer query is extracted*

if a row value of a subquery
doesn’t exist

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*
- it returns a Boolean value

if a row value of a subquery **exists** → **TRUE** → *the corresponding record of the outer query is extracted*

if a row value of a subquery **doesn't exist** → **FALSE**

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

- this check is conducted *row by row*

- it returns a Boolean value

---

if a row value of a subquery **exists** → **TRUE** → *the corresponding record* of the outer query is extracted

if a row value of a subquery **doesn’t exist** → **FALSE** → *no row value* from the outer query is extracted

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN
tests row values for existence	

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN
tests row values for existence	searches among values

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN
tests row values for existence	searches among values
quicker in retrieving large amounts of data	

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN
tests row values for existence	searches among values
quicker in retrieving large amounts of data	faster with smaller datasets

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- ORDER BY (nested queries)

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- ORDER BY (nested queries)

it is more professional to apply ORDER BY in the outer query

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- ORDER BY (nested queries)

it is more professional to apply ORDER BY in the outer query

- it is more acceptable logically to sort the *final* version of your dataset

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- some, though not all, nested queries can be rewritten using joins, which are more efficient in general

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- some, though not all, nested queries can be rewritten using joins, which are more efficient in general
- this is true particularly for inner queries using the WHERE clause

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- subqueries:

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

subqueries:

- allow for better *structuring* of the outer query

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

subqueries:

- allow for better *structuring* of the outer query
  - thus, each inner query can be thought of in isolation

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

subqueries:

- allow for better *structuring* of the outer query
  - thus, each inner query can be thought of in isolation
  - *hence the name of SQL – Structured Query Language!*

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

subqueries:

- allow for better *structuring* of the outer query
  - thus, each inner query can be thought of in isolation
  - *hence the name of SQL – Structured Query Language!*

- in some situations, the use of subqueries is much *more intuitive* compared to the use of complex joins and unions

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

subqueries:

- allow for better *structuring* of the outer query
  - thus, each inner query can be thought of in isolation
  - *hence the name of SQL – Structured Query Language!*

- in some situations, the use of subqueries is much *more intuitive* compared to the use of complex joins and unions

- many users prefer subqueries simply because they offer *enhanced code readability*

SQL Subqueries Nested in SELECT and FROM

365√DataScience

SQL Subqueries Nested in SELECT and FROM

You have advanced with SQL *a lot* at this point!

SQL Subqueries Nested in SELECT and FROM

You have advanced with SQL *a lot* at this point!

☺

SQL Subqueries Nested in SELECT and FROM

In this lecture:

SQL Subqueries Nested in SELECT and FROM

In this lecture:

challenging task

SQL Subqueries Nested in SELECT and FROM

In this lecture:

challenging task

+

SQL Subqueries Nested in SELECT and FROM

In this lecture:

challenging task

+

exercise