
SQL Subqueries

SQL Subqueries with IN Nested Inside WHERE

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries

queries embedded in a query

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

- subqueries

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries

queries embedded in a query

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries = inner queries

queries embedded in a query

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries = inner queries = nested queries

queries embedded in a query

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries = inner queries = nested queries

queries embedded in a query

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

365 DataScience

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**

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries = inner queries = nested queries = inner select

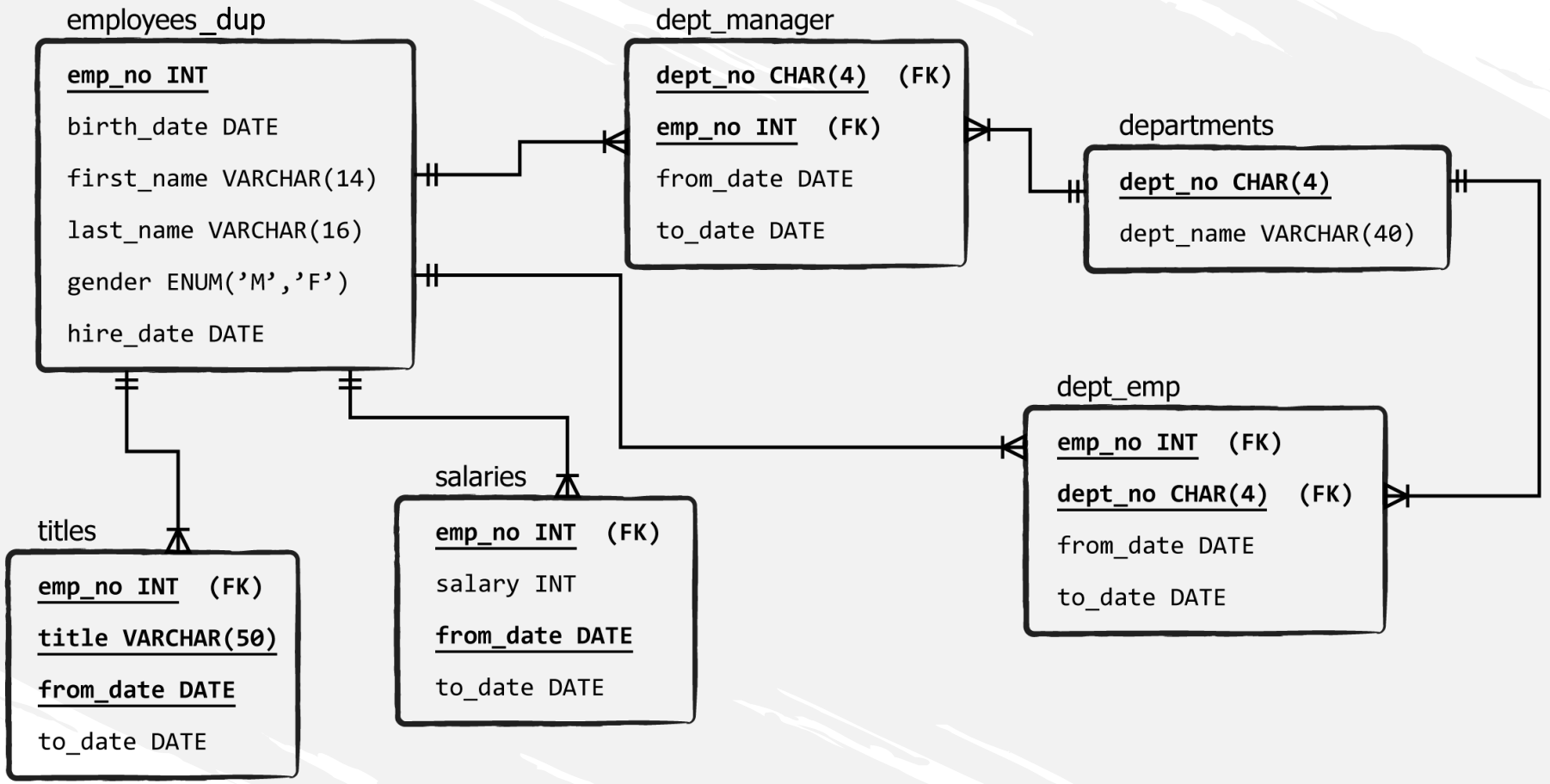
queries embedded in a query

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

  = outer select

365 DataScience

SQL Subqueries with IN Nested Inside WHERE



365 DataScience

SQL Subqueries with IN Nested Inside WHERE

- subqueries

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

subqueries

- a subquery should *always* be placed within parentheses

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

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

365 DataScience

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*

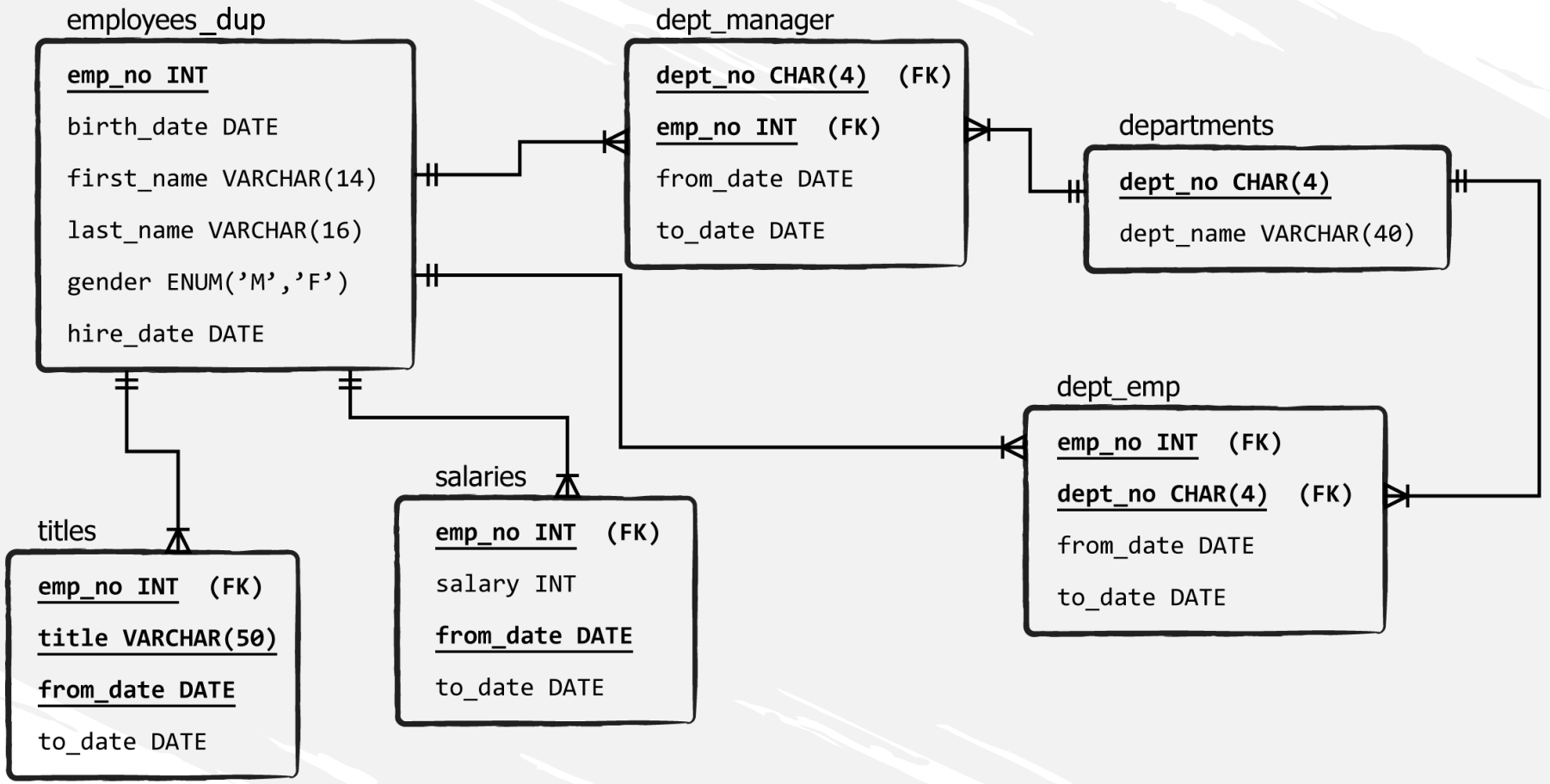
365 DataScience

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*

365 DataScience

SQL Subqueries with IN Nested Inside WHERE



365 DataScience

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*

365 DataScience

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

365 DataScience

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*

365 DataScience

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*

365 DataScience

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*

365 DataScience

SQL Subqueries with IN Nested Inside WHERE

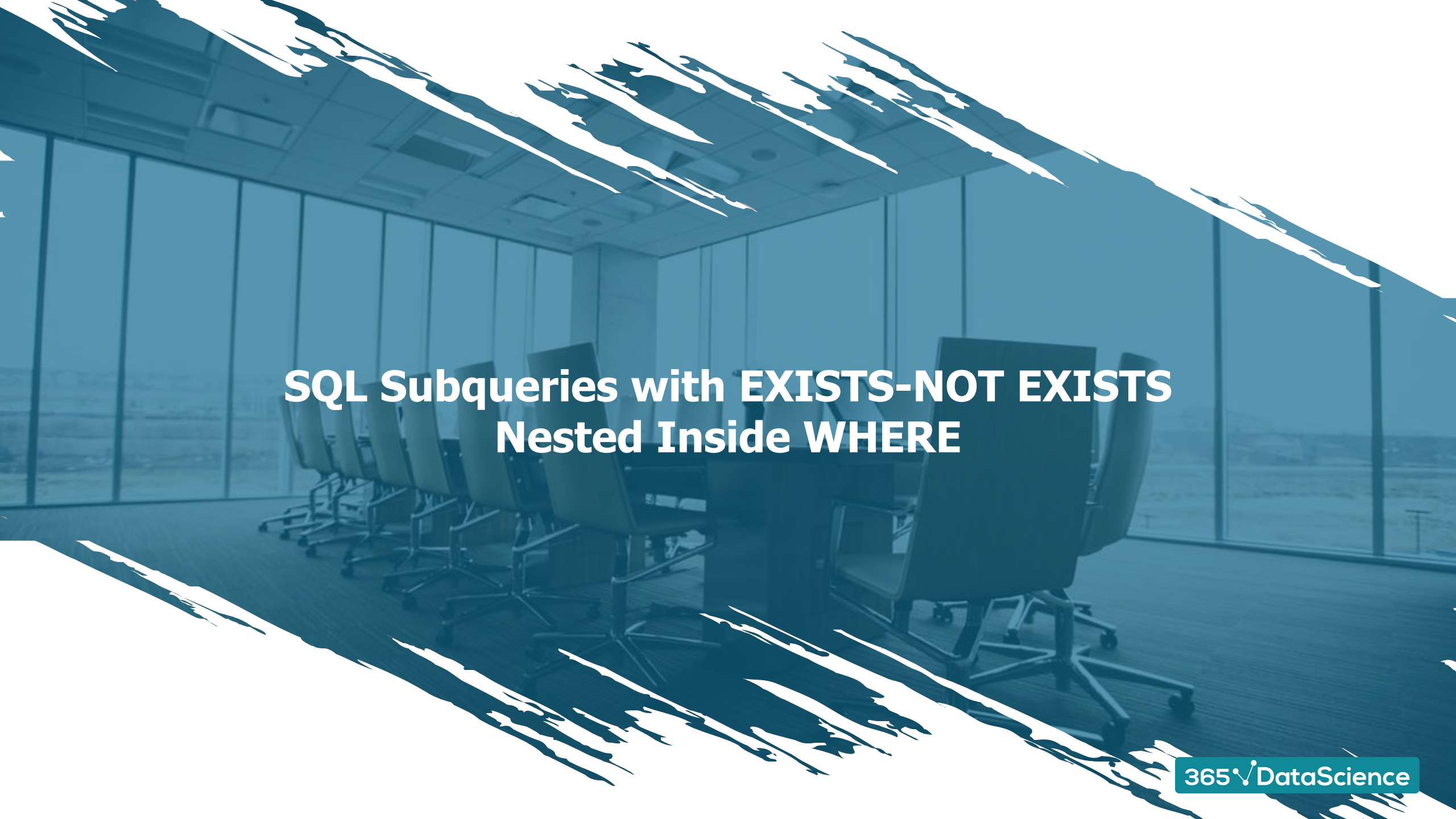
- a subquery may return *a single value (a scalar)*, *a single row*, *a single column*, or *an entire table*

  - 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*

365 DataScience



SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- EXISTS

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS

checks whether certain row values are found within a subquery

365 DataScience

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*

365 DataScience

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

365 DataScience

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

---

365 DataScience

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**

365 DataScience

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**

365 DataScience

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

365 DataScience

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**

365 DataScience

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**

365 DataScience

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

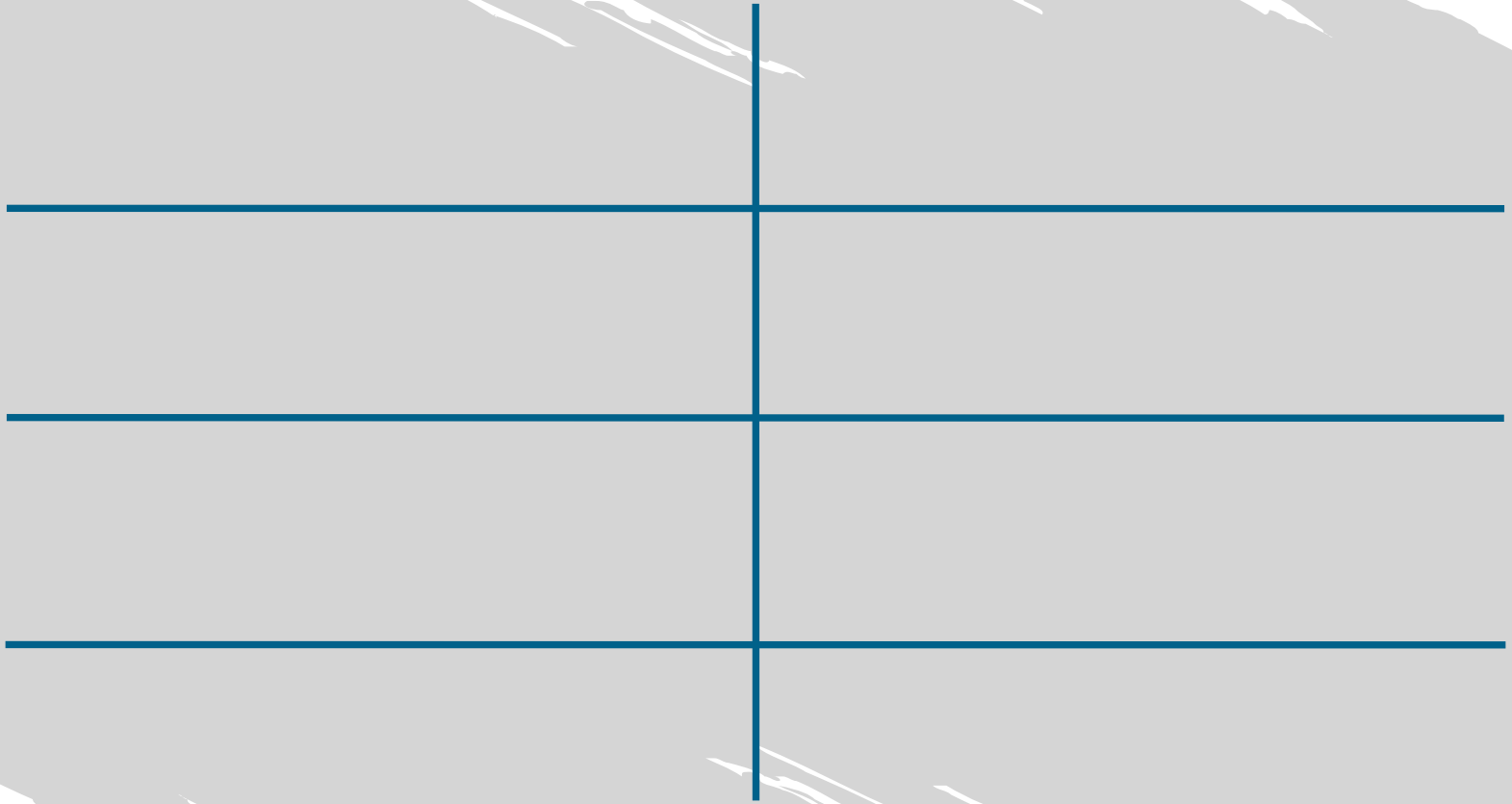
---

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

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

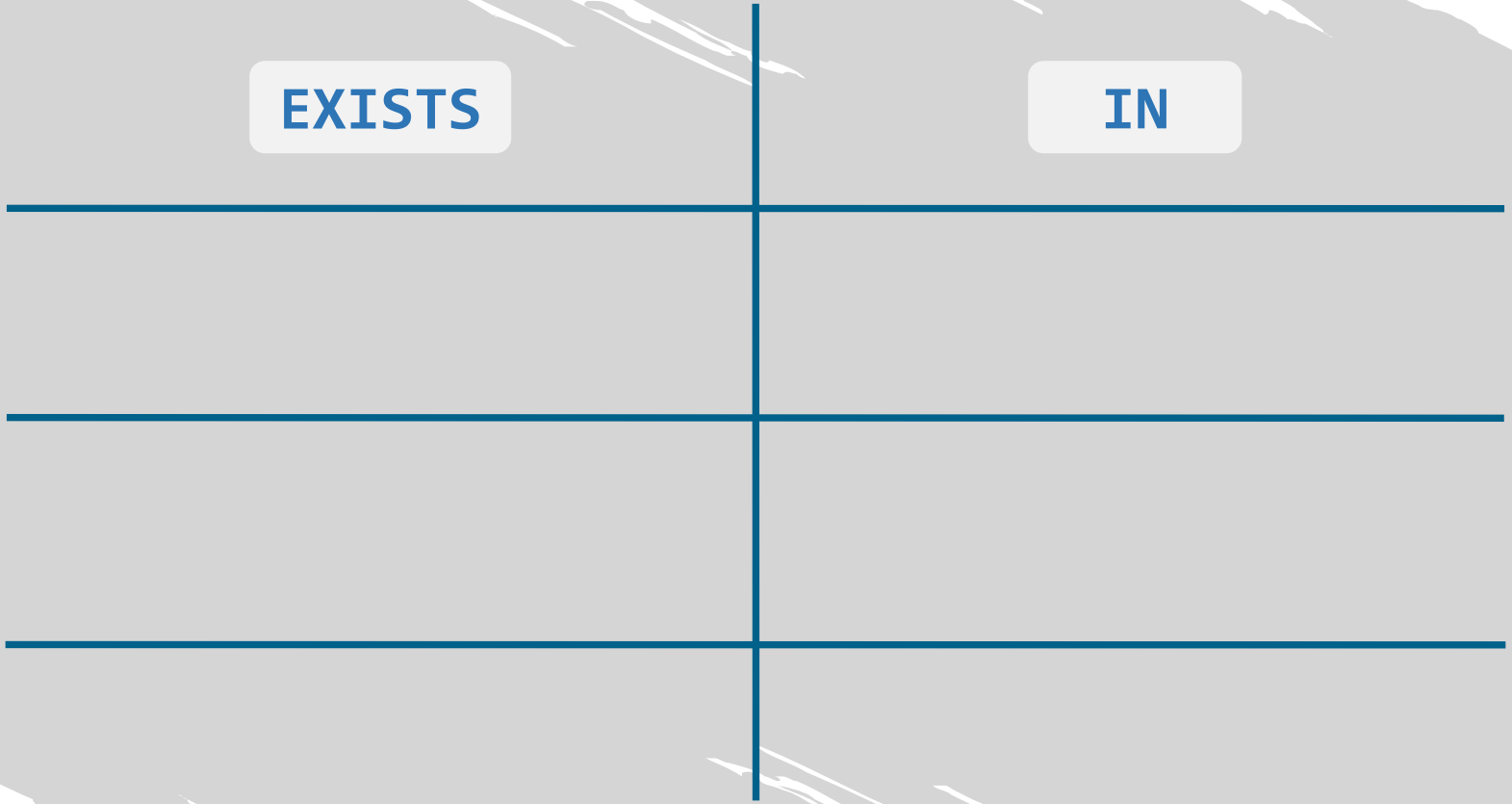


365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE



365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN
tests row values for existence	

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

EXISTS	IN
tests row values for existence	searches among values

365 DataScience

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	

365 DataScience

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

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- ORDER BY (nested queries)

365 DataScience

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

365 DataScience

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

365 DataScience

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

365 DataScience

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

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

- subqueries:

365 DataScience

SQL Subqueries with EXISTS-NOT EXISTS Nested Inside WHERE

subqueries:

- allow for better *structuring* of the outer query

365 DataScience

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

365 DataScience

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!*

365 DataScience

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

365 DataScience

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*

365 DataScience

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!

365 DataScience

SQL Subqueries Nested in SELECT and FROM

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

365 DataScience

SQL Subqueries Nested in SELECT and FROM

In this lecture:

365 DataScience

SQL Subqueries Nested in SELECT and FROM

In this lecture:

challenging task

365 DataScience

SQL Subqueries Nested in SELECT and FROM

In this lecture:

+

challenging task

365 DataScience

SQL Subqueries Nested in SELECT and FROM

In this lecture:

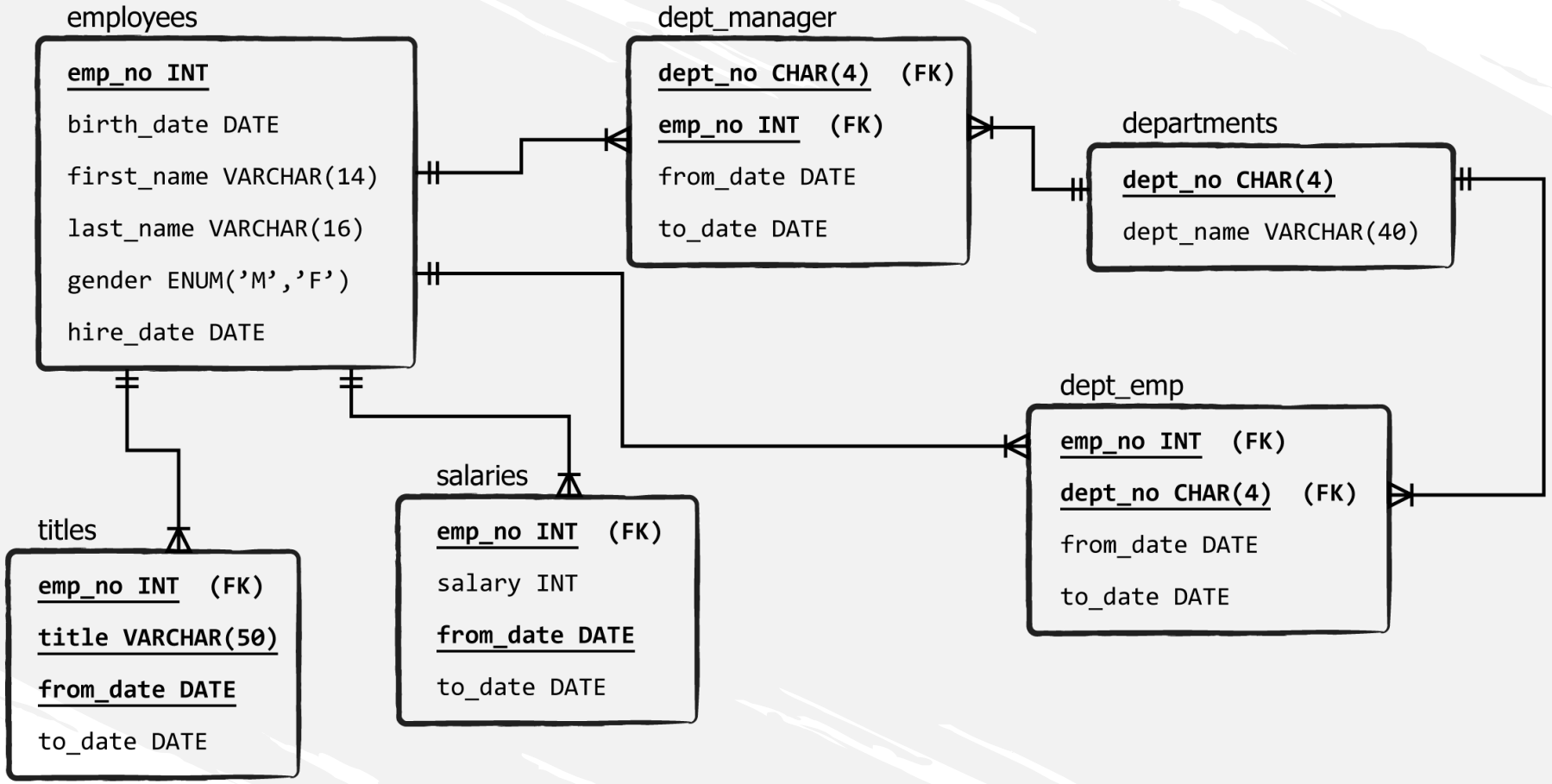
challenging task

+

exercise

365 DataScience

SQL Subqueries Nested in SELECT and FROM



365 DataScience