CSIT115 Data Management and Security

CSIT882 Data Management Systems

SELECT statement (5)

Subject Coordinators: Dr Chen Chen, Dr Thanh Le

School of Computing and Information Technology -
University of Wollongong

SELECT statement (5)

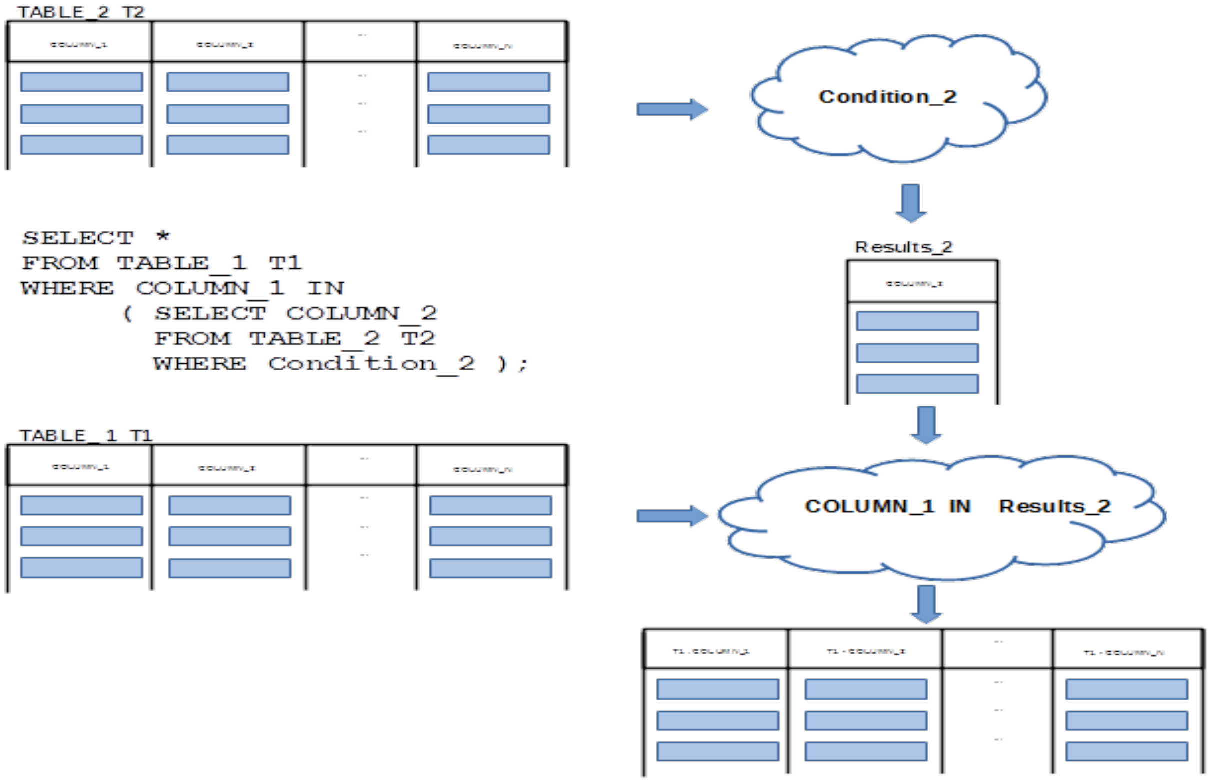
Outline

Nested queries

Correlated nested queries

Nested queries

Nested query is a query with another query embedded in `WHERE` or `SELECT` or `FROM` clauses

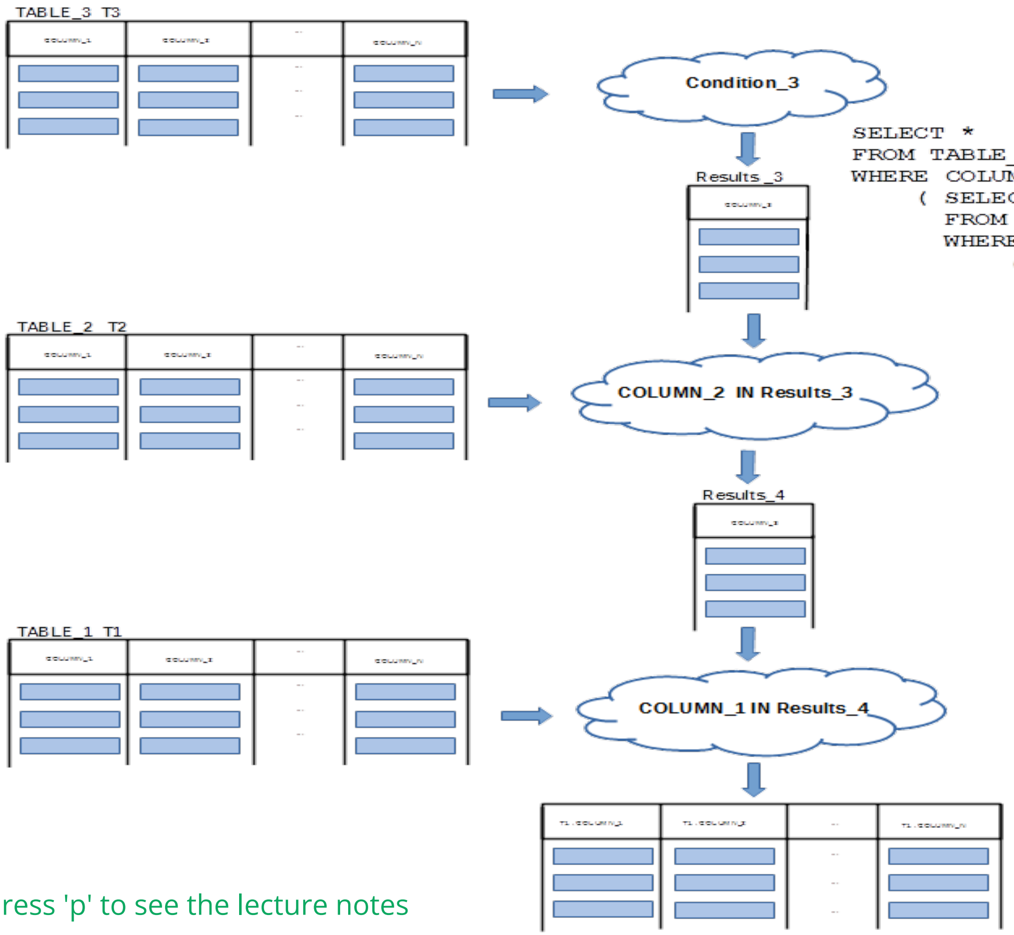


```
SELECT *
FROM TABLE_1 T1
WHERE COLUMN_1 IN
      ( SELECT COLUMN_2
        FROM TABLE_2 T2
        WHERE Condition_2 );
```

In HTML view press 'p' to see the lecture notes

Nested queries

Nested query is a query with another query embedded in `WHERE` or `SELECT` or `FROM` clauses



```
SELECT *
FROM TABLE_1 T1
WHERE COLUMN_1 IN
     ( SELECT COLUMN_4
       FROM TABLE_2 T2
       WHERE COLUMN_2 IN
            ( SELECT COLUMN_3
              FROM TABLE_3 T3
              WHERE Condition_3 ) );
```

In HTML view press 'p' to see the lecture notes

Nested queries

Sample database

CREATE TABLKE statement

```
CREATE TABLE DEPARTMENT(
 name              VARCHAR(50)       NOT NULL,
 code              CHAR(5)           NOT NULL,
 total_staff_number DECIMAL(2)       NOT NULL,
 chair             VARCHAR(50)           NULL,
 budget            DECIMAL(9,1)      NOT NULL,
  CONSTRAINT dept_pkey PRIMARY KEY(name),
  CONSTRAINT dept_ckey1 UNIQUE(code),
  CONSTRAINT dept_ckey2 UNIQUE(chair),
  CONSTRAINT dept_check1 CHECK (total_staff_number BETWEEN 1 AND 50) );
```

CREATE TABLE statement

```
CREATE TABLE COURSE(
 cnum              CHAR(7)          NOT NULL,
 title             VARCHAR(200)     NOT NULL,
 credits           DECIMAL(2)       NOT NULL,
 offered_by        VARCHAR(50)          NULL,
  CONSTRAINT course_pkey PRIMARY KEY(cnum),
  CONSTRAINT course_check1 CHECK (credits IN (6, 12)),
  CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
                        REFERENCES DEPARTMENT(name) ON DELETE CASCADE );
```

In HTML view press 'p' to see the lecture notes

Nested queries

A class of nested queries is based on a concept of inline views

Inline views can be used to reduce the complexity of query implementation

An inner query is created first and its outcomes are used as a relational table (inline view) in an outer query

For example, a query find the titles of courses offered by a department chaired by Peter is decomposed into the following two queries:

- `Q1`: Find a department chaired by Peter
- `Q2`: Find the titles of courses offered by a department found in `Q1`

A query `Q1` is implemented first and then it is used in `WHERE` clause of query `Q2`

In HTML view press 'p' to see the lecture notes

Nested queries

In nested queries `SELECT` statements are nested to theoretically unlimited level in `WHERE` clause

Nested query with a set membership operation IN

```
SELECT title
FROM COURSE
WHERE offered_by IN ( SELECT name
                      FROM DEPARTMENT
                      WHERE chair = 'Peter' );
```

If an inner query returns more than one row then we must use `IN` istead of `=`

However, it does not mean that we can always use `IN` instead of `=`

If an inner query returns zero or at most one row then using `=` instead of `IN` allows a query processor to find more efficient query processing plan

Nested queries

Implementation of a query find the titles of courses offered by a department chaired by Peter consists of the following steps

Create a inner query as inline view Q

Inline view

```
( SELECT name
FROM DEPARTMENT
WHERE chair = 'Peter' ) Q
```

Create a temporary outer query, that references an inline view Q created earlier

Query that references an inline view Q

```
SELECT title
FROM COURSE
WHERE offered_by IN Q.name
```

Replace a reference Q to an inline view with the inline view itself

Nested query with a set membership operation IN

```
SELECT title
FROM COURSE
WHERE offered_by IN ( SELECT name
                      FROM DEPARTMENT
                      WHERE chair = 'Peter' );
```

In HTML view press 'p' to see the lecture notes

Nested queries

Another example, find the chairs of all departments, that offer 12 credit point courses

Create an inner query as inline view Q

Inline view

```
( SELECT offered_by
FROM COURSE
WHERE credits = 12) Q;
```

Next, create an intermediate from of outer query, that references an inline view Q created earlier

Query that references an inline view Q

```
SELECT chair
FROM DEPARTMENT
WHERE name IN Q.offered_by
```

Replace a reference to an inline view Q with the inline view itself

Nested query with a set membership operation IN

```
SELECT chair
FROM DEPARTMENT
WHERE name IN ( SELECT offered_by
                FROM COURSE
                WHERE credits = 12 );
```

In HTML view press 'p' to see the lecture notes

Nested queries

A query find the chairs of all departments that offer no courses is an example of ANTIJOIN operation

It is equivalent to a query find all rows in a relational table DEPARTMENT that cannot be joined with any row in a relational table COURSE

An inner query is implemented as an inline view finds all departments, that offer at least one course

Inline view

```
( SELECT offered_by
FROM COURSE ) Q;
```

An outer query finds all chairs of departments that are NOT included in the results of an inner query

Query that references an inline view

```
SELECT chair
FROM DEPARTMENT
WHERE name NOT IN Q.offered_by
```

Nested queries

Finally, we replace a reference to an inline view Q with the inline view itself

Nested query with a negated set membership operation IN

```
SELECT chair
FROM DEPARTMENT
WHERE name NOT IN ( SELECT offered_by
                    FROM COURSE );
```

In HTML view press 'p' to see the lecture notes

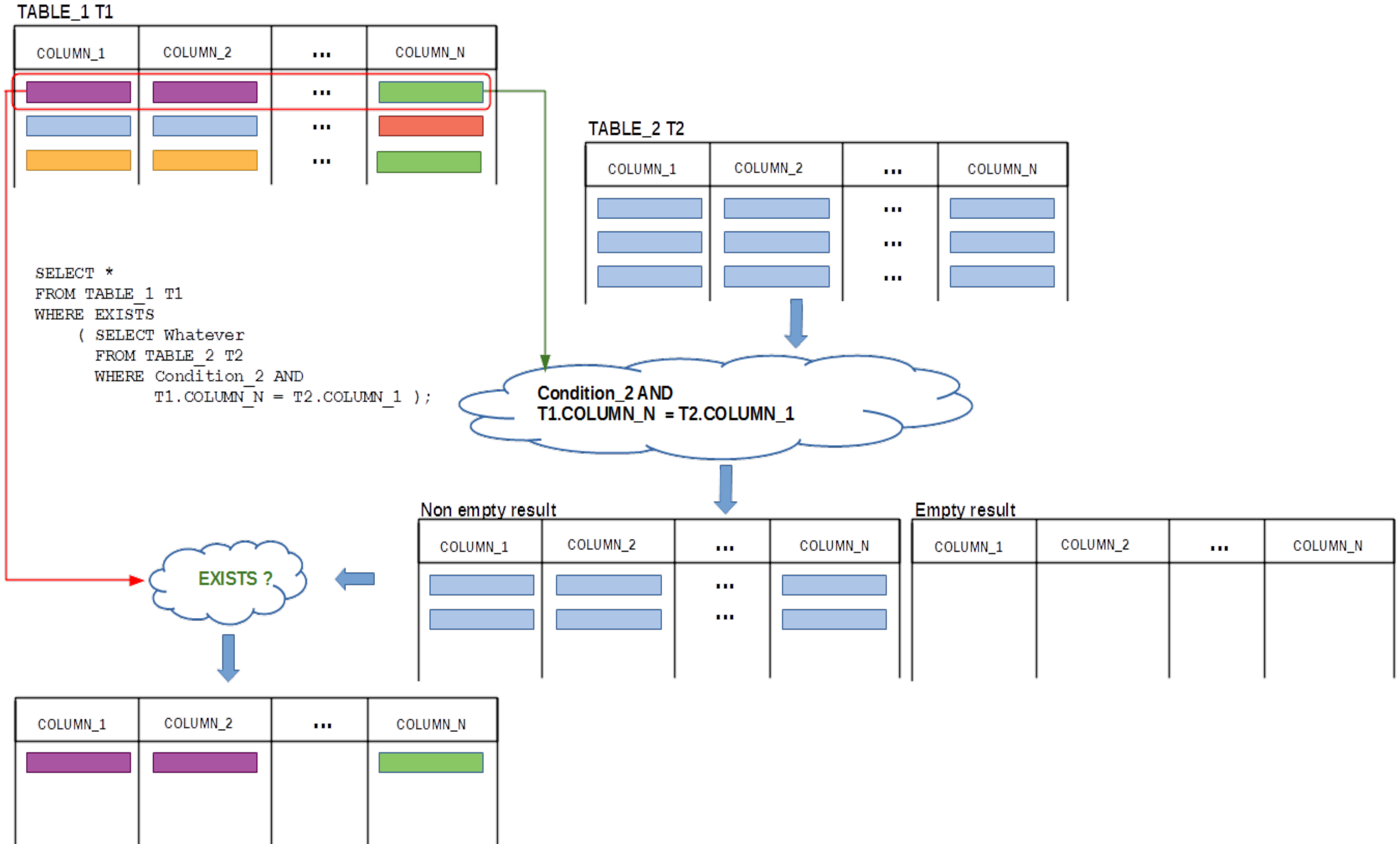
SELECT statement (5)

Outline

- Nested queries
- Correlated nested queries

Correlated nested queries

In a correlated nested query an inline view may reference the names of relational tables used in `SELECT` statements outer to the inline view



```
SELECT *
FROM TABLE_1 T1
WHERE EXISTS
     ( SELECT Whatever
       FROM TABLE_2 T2
       WHERE Condition_2 AND
             T1.COLUMN_N = T2.COLUMN_1 );
```

In HTML view press 'p' to see the lecture notes

Correlated nested queries

In a correlated nested query an inline view may reference the names of relational tables used in `SELECT` statemens outer to the inline view

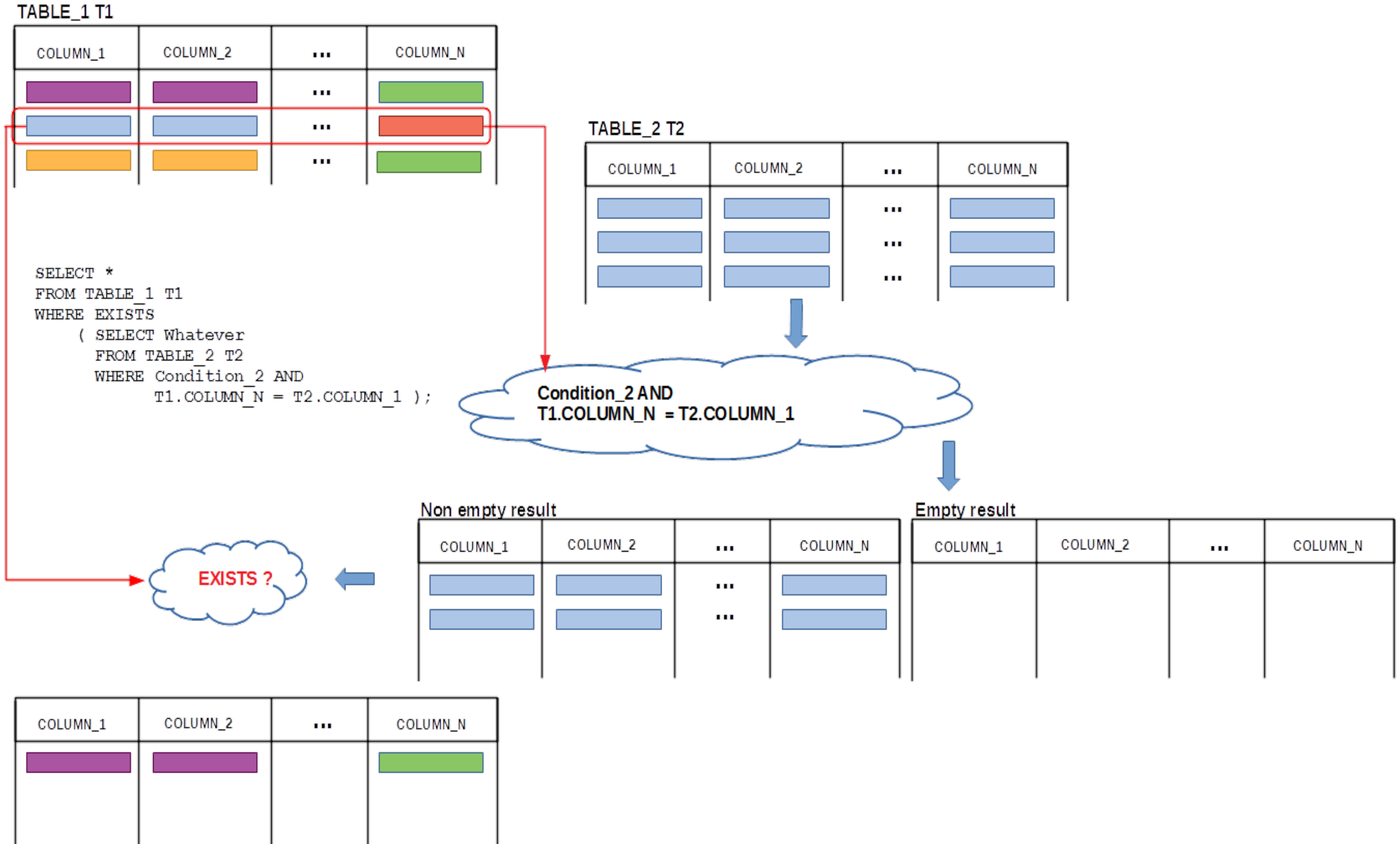


TABLE_1 T1

COLUMN_1	COLUMN_2	...	COLUMN_N
🟪	🟪	...	🟩
🟦	🟦	...	🟥
🟧	🟧	...	🟩

TABLE_2 T2

COLUMN_1	COLUMN_2	...	COLUMN_N
🟦	🟦	...	🟦
🟦	🟦	...	🟦
🟦	🟦	...	🟦

```
SELECT *
FROM TABLE_1 T1
WHERE EXISTS
     ( SELECT Whatever
       FROM TABLE_2 T2
       WHERE Condition_2 AND
             T1.COLUMN_N = T2.COLUMN_1 );
```

Non empty result

COLUMN_1	COLUMN_2	...	COLUMN_N
🟦	🟦	...	🟦
🟦	🟦	...	🟦

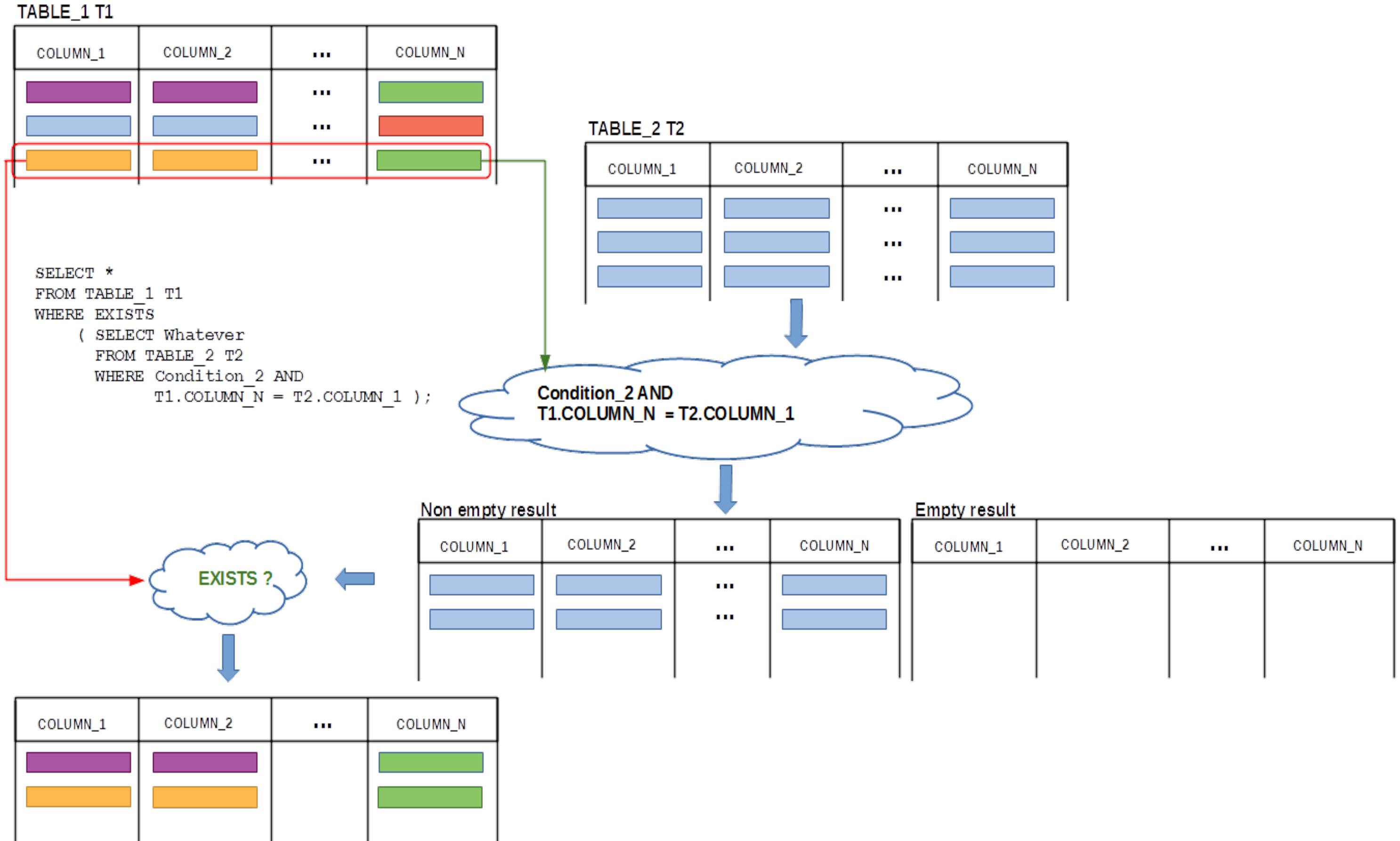
Empty result

COLUMN_1	COLUMN_2	...	COLUMN_N

COLUMN_1	COLUMN_2	...	COLUMN_N
🟪	🟪		🟩

Correlated nested queries

In a correlated nested query an inline view may reference the names of relational tables used in `SELECT` statements outer to the inline view



```
SELECT *
FROM TABLE_1 T1
WHERE EXISTS
     ( SELECT Whatever
       FROM TABLE_2 T2
       WHERE Condition_2 AND
             T1.COLUMN_N = T2.COLUMN_1 );
```

Correlated nested queries

For example, consider a query, that finds the chairs of departments, that offer 12 credit point courses

Such query can be expressed as an equivalent query, that finds the chairs of departments, such that there exists at least one course worth 12 credit points offered by a department we are looking for

An inner query ... course worth 12 credit points offered by a department we are looking for is implemented as the following inline view Q

Inline view

```
( SELECT *
FROM COURSE
WHERE credits = 12 AND offered_by = DEPARTMENT.name ) Q
```

A temporary outer query, that finds the chairs of departments such that there exists at least one course found in the inner quer is implemented in the folowing way

Query with an existential quantifier EXISTS that references and inline view

```
SELECT chair
FROM DEPARTMENT
WHERE EXISTS Q
```

Correlated nested queries

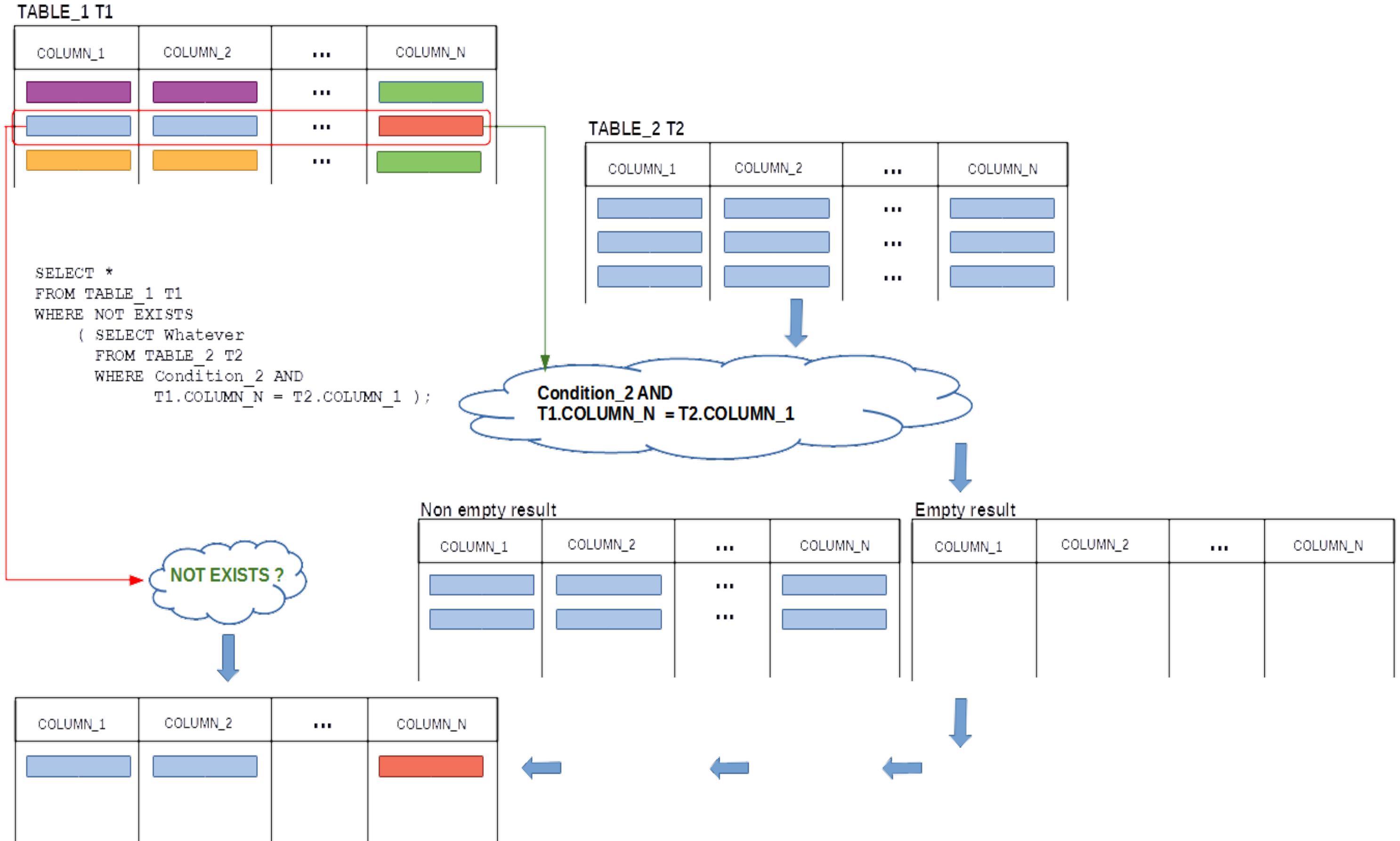
Finally, we replace a reference to an inline view Q with the inline view itself

Correlated nested query with an existential quantifier EXISTS

```
SELECT chair
FROM DEPARTMENT
WHERE EXISTS ( SELECT *
               FROM COURSE
               WHERE credits = 12 AND offered_by = DEPARTMENT.name );
```

Correlated nested queries

In a correlated nested query an inline view may reference the names of relational tables used in `SELECT` statement outer to the inline view



```
SELECT *
FROM TABLE_1 T1
WHERE NOT EXISTS
    ( SELECT Whatever
      FROM TABLE_2 T2
      WHERE Condition_2 AND
            T1.COLUMN_N = T2.COLUMN_1 );
```

Correlated nested queries

Another example where inline view references a name of relational table used in `SELECT` statement outer to the inline view is the following

Find the chairs of all departments, that offer no courses

It is equivalent to a query find the chairs of departments, such that does not exist a course offered by a department we are looking for

An inner query finds ... a course offered by a department we are looking for and it is implemented as an inline view `Q`

Inline view

```
( SELECT *
FROM COURSE
WHERE offered_by = DEPARTMENT.name ) Q;
```

A temporary outer query finds all chairs of deparments such that does not exist a course found by an inner query

Query with a negated existential quantifier EXISTS that references an inline view

```
SELECT chair
FROM DEPARTMENT
WHERE NOT EXISTS Q
```

In HTML view press 'p' to see the lecture notes

Correlated nested queries

Finally, we replace a reference to an inline view Q with the inline view itself

Correlated nested query with a negated existential quantifier EXISTS

```
SELECT chair
FROM DEPARTMENT
WHERE NOT EXISTS( SELECT *
                  FROM COURSE
                  WHERE offered_by = DEPARTMENT.name );
```

In HTML view press 'p' to see the lecture notes

References

C. Coronel, S. Morris, A. Basta, M. Zgola, Data Management and Security, Chapters 5, 7, Cengage Compose eBook, 2018, eBook: Data Management and Security, 1st Edition

T. Connoly, C. Begg, Database Systems, A Practical Approach to Design, Implementation, and Management, Chapter 6.3.5 Subqueries, Chapter 6.3.8 EXISTS and NOT EXISTS, Chapter 7.4.1 Creating a View Pearson Education Ltd, 2015

D. Darmawikarta, SQL for MySQL A Beginner's Tutorial, Chapter 7 Subqueries, Chapter 9 Views, Brainy Software Inc. First Edition: June 2014

How to ... ? Cookbook, How to implement queries in SQL ? (Part 2) Recipe 6.3 How to implement nested queries ?

How to ... ? Cookbook, How to implement queries in SQL ? (Part 2) Recipe 6.4 Recipe 6.4 How to implement correlated nested queries ?