The seventh topic of this subject, introduces a very important technique while querying a database. The latest technique that is to be used involves the use of *sub-queries*. A sub-query is a query which combines two queries together, placing a query inside another query. The *inner query/sub query* will return a value that is used by the *main query/outer query*. Sub queries are equivalent to the use of two sequential queries, with the result of the first query being used as a search value in the second query.

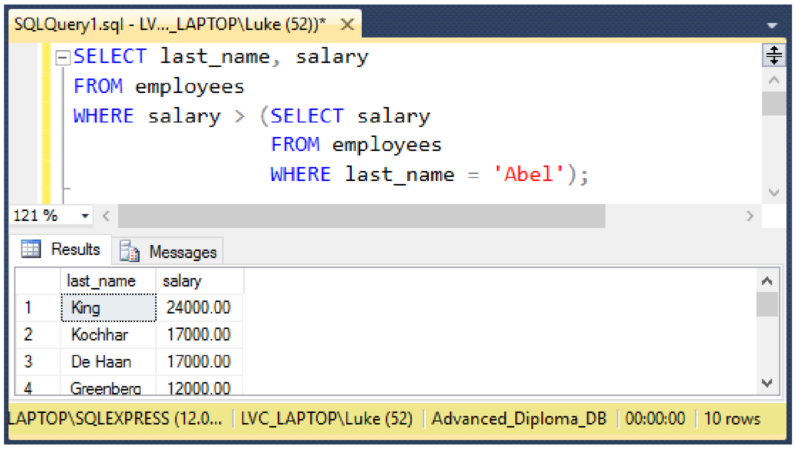
The general syntax for the use of Sub-Query operations is:

```
SELECT select_list
FROM table_name
WHERE expr_operator (SELECT select_list
                     FROM table_name
                     [WHERE condition])
```

Note:

- The inner query will execute first and its result will be used by the outer query to display the final result.
- The inner and outer queries do not need to obtain data from the same table. Data can be extracted from different tables, therefore the FROM clause can have different table_names.
- The inner query can be placed in the FROM, WHERE, or HAVING clauses.

Example 1: Write a query that will display the surname and salary of all the employees who earn more than Abel.


SQLQuery1.sql - LV..._LAPTOP\Luke (52))*

```
SELECT last_name, salary
FROM employees
WHERE salary > (SELECT salary
                FROM employees
                WHERE last_name = 'Abel');
```

	last_name	salary
1	King	24000.00
2	Kochhar	17000.00
3	De Haan	17000.00
4	Greenberg	12000.00

LAPTOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (52) | Advanced_Diploma_DB | 00:00:00 | 10 rows

Figure 1 - Result of example 1: Use of sub-queries to determine salary of employee, then obtain all those who earn more

Sub-queries are usually categorised under two main categories:

- **Single-row sub-queries:** the inner statement returns only one row as a result
- **Multiple-row sub-queries:** the inner statement returns multiple rows as a result

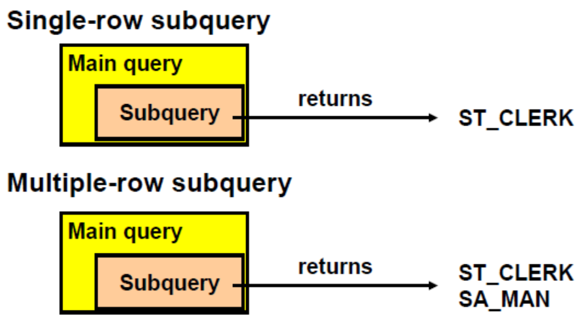


Figure 2 – The two types of possible sub-queries

While using the sub-queries technique, it is very important to follow the below guidelines;

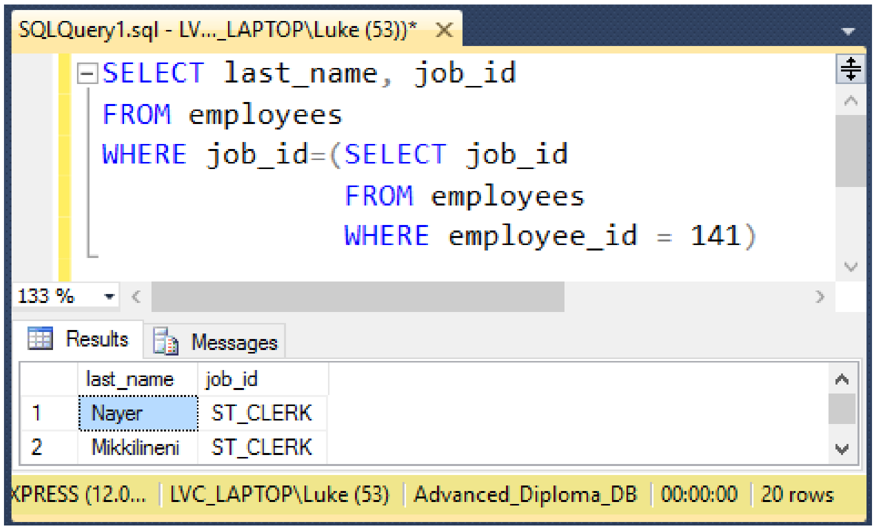
- Sub-queries should be enclosed in parentheses
- Always place sub-queries on the right side of the comparison operators
- Make sure to use single-row operators together with single-row sub-queries and multiple row operators with multiple row sub-queries.

Single-Row Sub Queries

As already mentioned in the previous pages, single-row sub-queries include an inner query that will return only one row. Given the fact that the inner query returns only one row, such sub-queries should make use of below single row comparison operators only.

Operator	***Meaning***
=	equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
<>	not equal to

Example 2: Write a query that will display the surname and job of all those employees whose job_id is the same as that of employee 141


SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT last_name, job_id
FROM employees
WHERE job_id=(SELECT job_id
              FROM employees
              WHERE employee_id = 141)
```

133 %

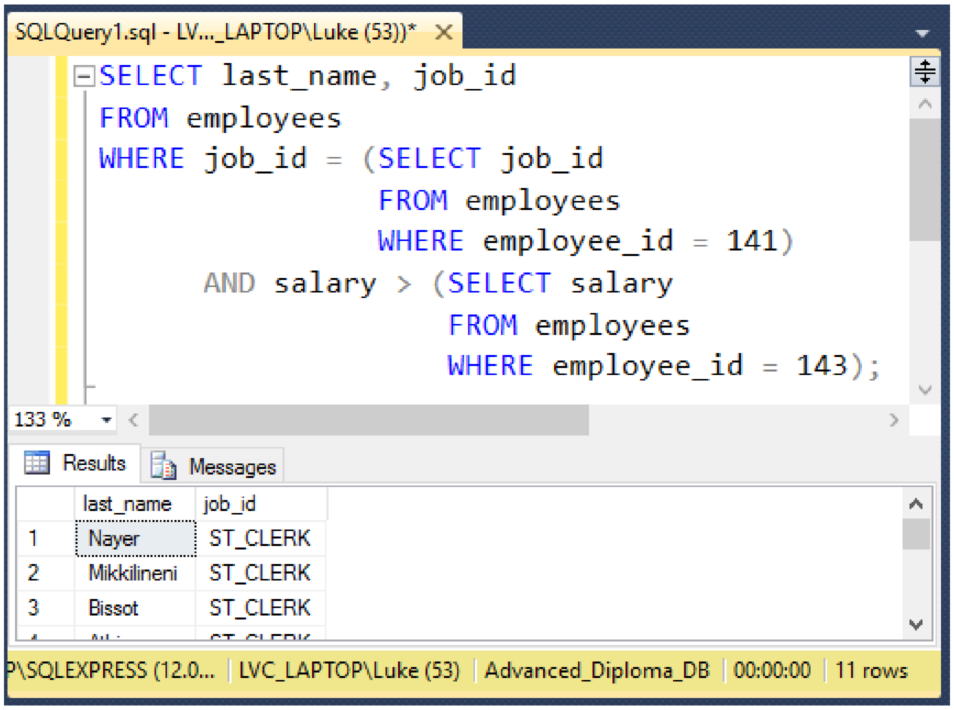
Results | Messages

```
   last_name    job_id
1  Nayer        ST_CLERK
2  Mikkilineni  ST_CLERK
```

XPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 20 rows

Figure 3 – Result of example 2: All employees with same job_id of employee 141

Example 3: Write a query that will display the surname and the job_id of all the employees whose job_id is the same as that of employee 141 and whose salary is the same as that of employee 143


SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT last_name, job_id
FROM employees
WHERE job_id = (SELECT job_id
                FROM employees
                WHERE employee_id = 141)
      AND salary > (SELECT salary
                    FROM employees
                    WHERE employee_id = 143);
```

133 %

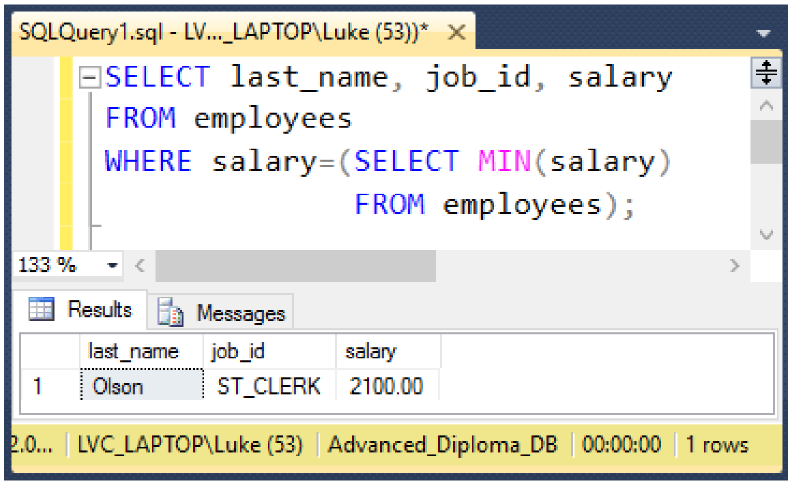
Results | Messages

```
   last_name    job_id
1  Nayer        ST_CLERK
2  Mikkilineni  ST_CLERK
3  Bissot       ST_CLERK
```

P\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 11 rows

Figure 4 - Result of example 4: employees with job_id equal to ST_CLERK and Salary > 2600

Example 5: Write a query that will return the surname, job number and salary of all the employees with the lowest salary. Note that this query makes use of a GROUP function within the sub-query



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary=(SELECT MIN(salary)
              FROM employees);
```

133 %

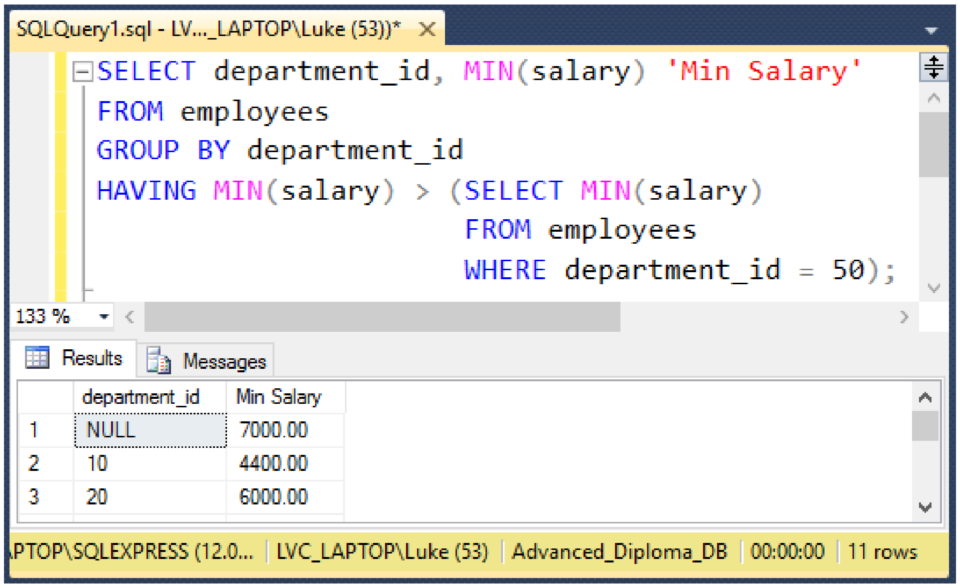
Results | Messages

	last_name	job_id	salary
1	Olson	ST_CLERK	2100.00

2.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 1 rows

Figure 5 - Result of example 5: The lowest paid employees

Example 6: Write a query that will display all the departments that have a minimum salary which is greater than that of department 50



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT department_id, MIN(salary) 'Min Salary'
FROM employees
GROUP BY department_id
HAVING MIN(salary) > (SELECT MIN(salary)
                      FROM employees
                      WHERE department_id = 50);
```

133 %

Results | Messages

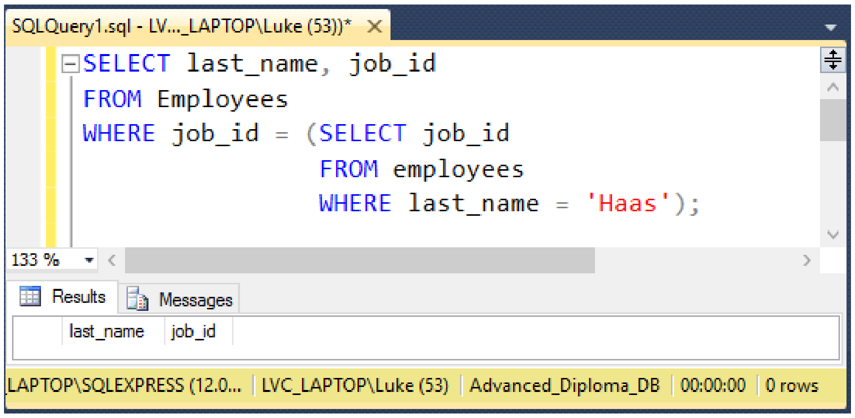
	department_id	Min Salary
1	NULL	7000.00
2	10	4400.00
3	20	6000.00

PTOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 11 rows

Figure 6 - Result of example 6: The minimum salary pf each department which is greater than the minimum in department 50

NOTE: that the above query makes use of the HAVING clause with sub-queries. The sub-query is calculated first and then the result is passed on to the main query

What's wrong with the statements below?


SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```sql
SELECT last_name, job_id
FROM Employees
WHERE job_id = (SELECT job_id
                FROM employees
                WHERE last_name = 'Haas');
```

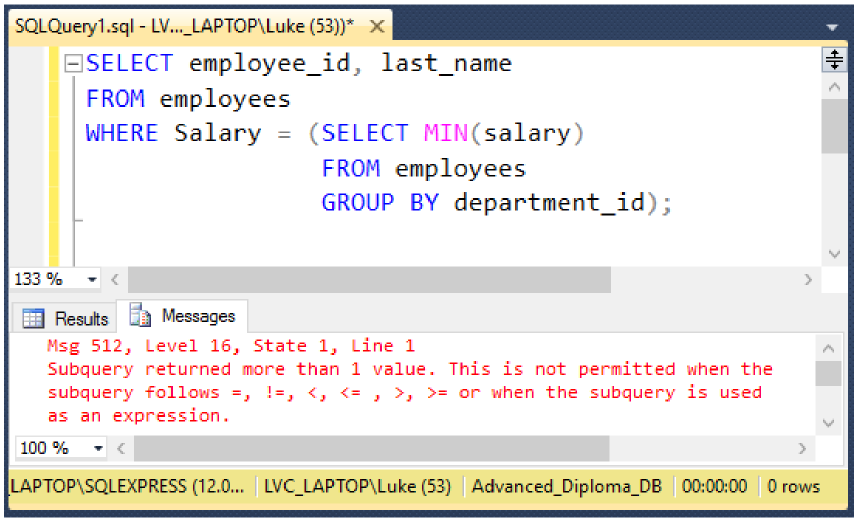
133 %

Results | Messages

last_name	job_id

LAPTOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 0 rows

The above query functions without any errors what so ever. Although such a query does not return any errors, it represents one of the most common problems with sub-queries. The main problem with sub-queries is that the inner query does not return any results. In this case the inner query returns no results as there is no employee with a surname of 'Haas'.


SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```sql
SELECT employee_id, last_name
FROM employees
WHERE Salary = (SELECT MIN(salary)
                FROM employees
                GROUP BY department_id);
```

133 %

Results | Messages

```
Msg 512, Level 16, State 1, Line 1
Subquery returned more than 1 value. This is not permitted when the
subquery follows =, !=, <, <= , >, >= or when the subquery is used
as an expression.
```

100 %

LAPTOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 0 rows

The above query does not return a result but instead it returns an error (shown in red). The error that is returned states that the inner query returns multiple values (2500, 4200, 4400, 6000, 7000, 8300, 17000) and for this reason a single row comparison operator cannot be used. This means that the *equals operator (=)* cannot be used in this case.

NOTE:

- Make sure that the inner query returns the correct result when executed
- Make sure that the inner query does not return multiple columns when a single row comparison operator is used

Multiple-Row Sub Queries

The second type of sub-queries that is available is better known as multiple-row sub-queries. The difference between multiple-row and single-row sub-queries lies primarily in the number of rows that are returned by the inner query. As a result of this the below multiple-row operators are to be used.

Operator	***Meaning***
IN	equal to any member in the list
ANY	compare value to each value returned by the subquery
ALL	compare value to every returned by the subquery

The ANY operator compares each value that is returned by the sub-query and the following apply

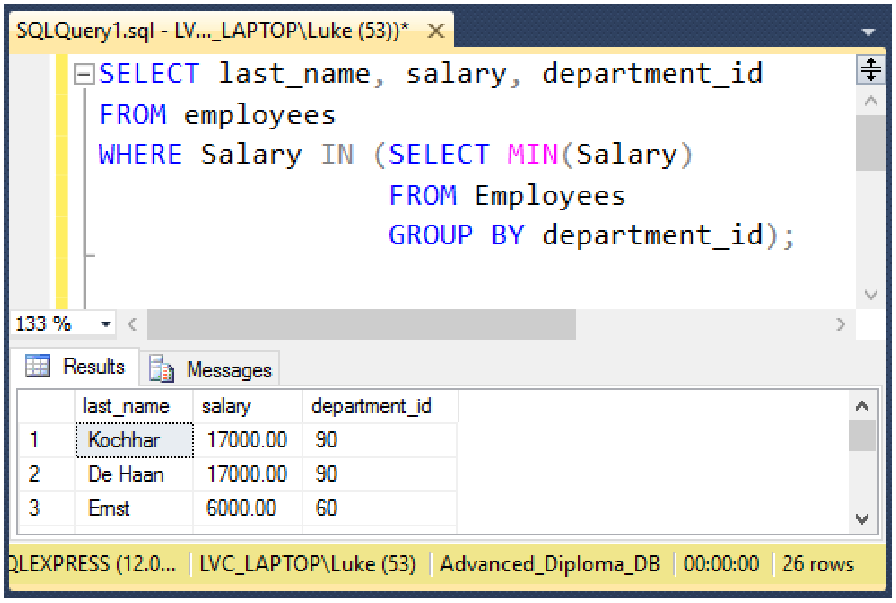
<ANY	means less than the maximum
>ANY	means more than the minimum
=ANY	is equivalent to IN

The ALL operator compares a value to every value returned by a subquery and the following apply

<ALL	means less than the minimum
>ALL	means more than the maximum

The above operators can be used with the NOT operator as well. This is done in order to be able to negate the result.

Example 7: Write a query that will return the surname, salary and department number of all the employees who earn the minimum salary for each department. The inner query will return 2500, 4200, 4400, 6000, 7000, 8300, 8600 and 17000



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT last_name, salary, department_id
FROM employees
WHERE Salary IN (SELECT MIN(Salary)
                 FROM Employees
                 GROUP BY department_id);
```

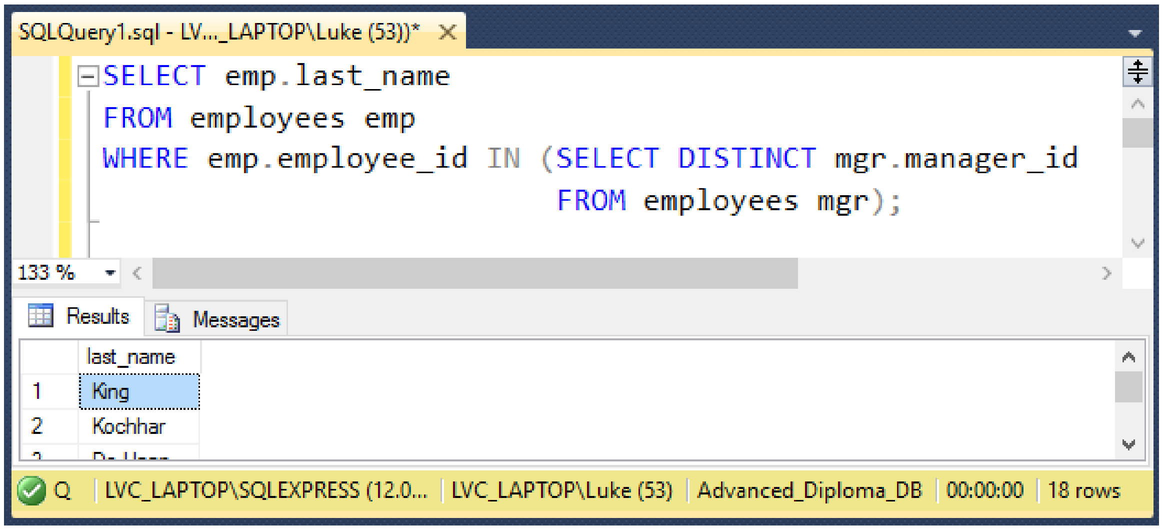
133 %

Results | Messages

	last_name	salary	department_id
1	Kochhar	17000.00	90
2	De Haan	17000.00	90
3	Ernst	6000.00	60

QLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 26 rows

Example 8: Write a query that will display all the employees who are managers using a sub-query



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT emp.last_name
FROM employees emp
WHERE emp.employee_id IN (SELECT DISTINCT mgr.manager_id
                          FROM employees mgr);
```

133 %

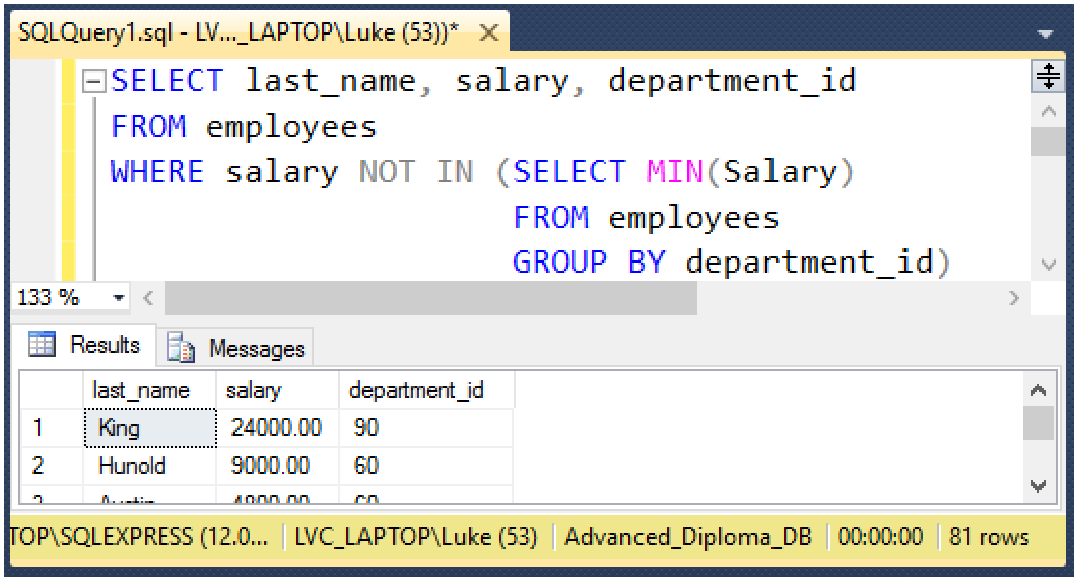
Results | Messages

	last_name
1	King
2	Kochhar

Q | LVC_LAPTOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 18 rows

Figure 7 - Result of example 8: All employees who are managers

Example 9: Write a query that the surname, salary and department number of all the employees who do not earn the minimum salary for each department.



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT last_name, salary, department_id
FROM employees
WHERE salary NOT IN (SELECT MIN(Salary)
                     FROM employees
                     GROUP BY department_id)
```

133 %

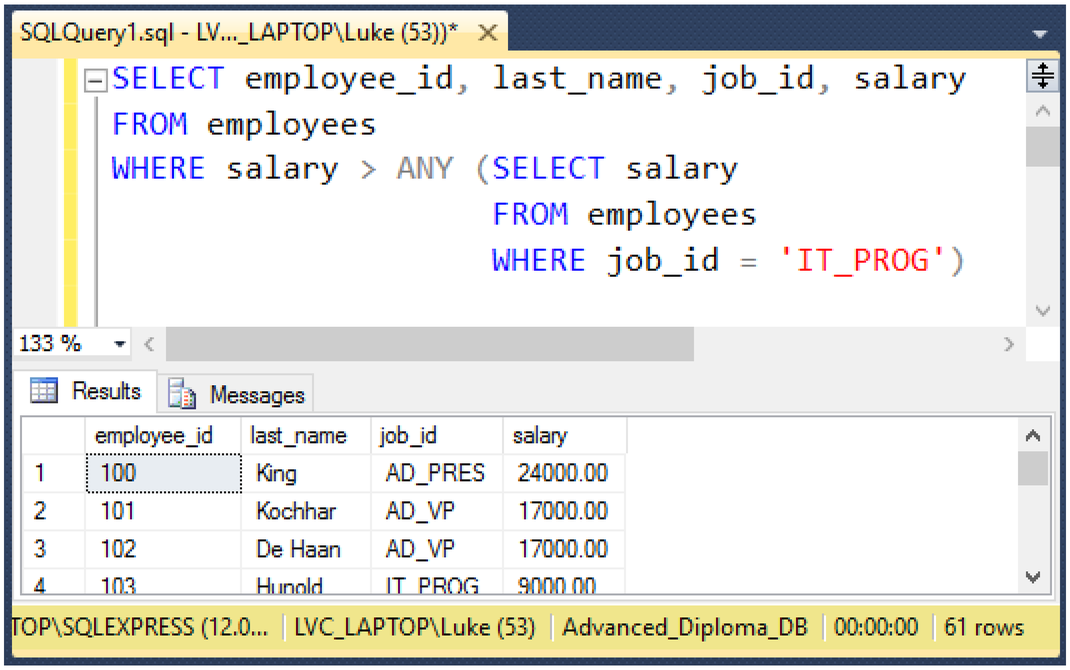
Results | Messages

	last_name	salary	department_id
1	King	24000.00	90
2	Hunold	9000.00	60

TOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 81 rows

Figure 8 - Result of example 9: people who earn more than the minimum salary in their department

Example 10: Write a query that will display all the employees that earn a salary which is more than the minimum for an IT_PROG



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary > ANY (SELECT salary
                    FROM employees
                    WHERE job_id = 'IT_PROG')
```

133 %

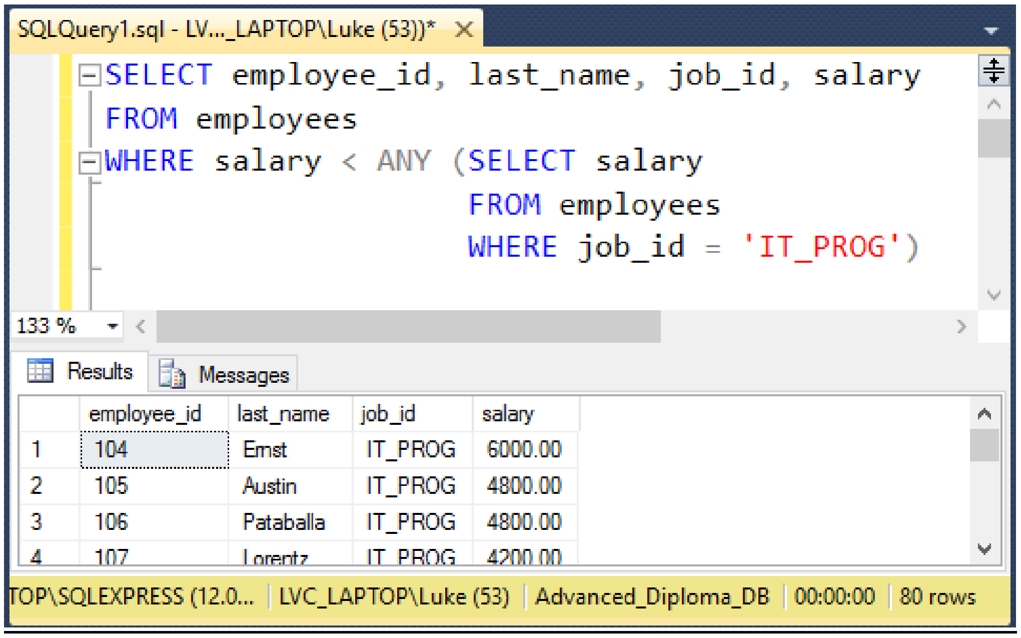
Results | Messages

	employee_id	last_name	job_id	salary
1	100	King	AD_PRES	24000.00
2	101	Kochhar	AD_VP	17000.00
3	102	De Haan	AD_VP	17000.00
4	103	Hunold	IT_PROG	9000.00

TOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 61 rows

Figure 9 - Result of example 10: employees who earn a salary more than the minimum of IT_PROG

Example 11: Write a query that will display all the employees that earn a salary which is less than the maximum for an IT_PROG



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

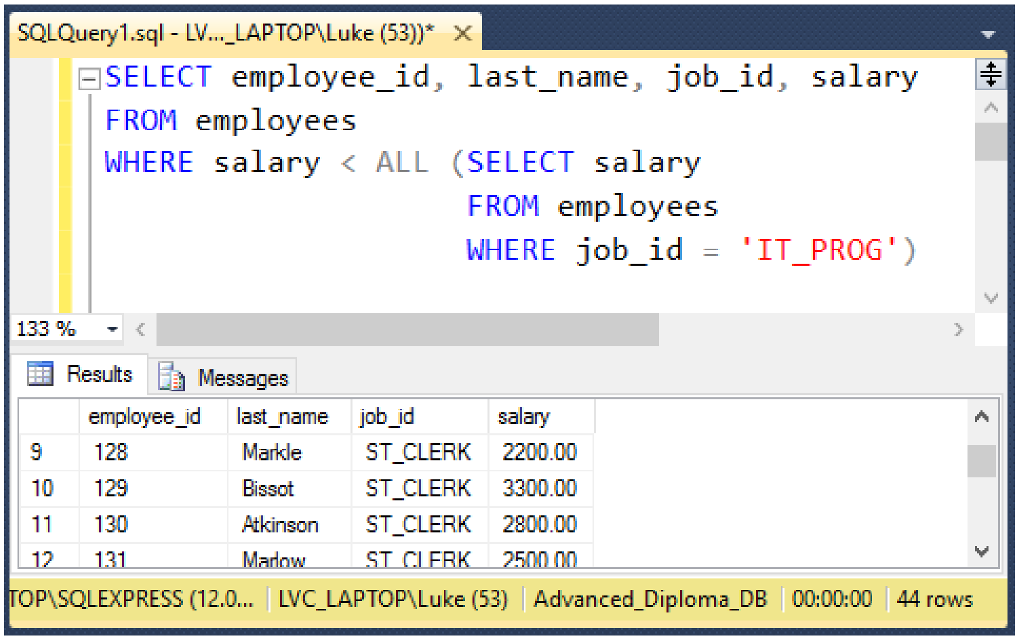
```sql
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ANY (SELECT salary
                    FROM employees
                    WHERE job_id = 'IT_PROG')
```

	employee_id	last_name	job_id	salary
1	104	Ernst	IT_PROG	6000.00
2	105	Austin	IT_PROG	4800.00
3	106	Pataballa	IT_PROG	4800.00
4	107	Lorentz	IT_PROG	4200.00

TOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 80 rows

Figure 10 - Result of example 11: employees who earn a salary less than the maximum of IT_PROG

Example 12: Write a query that will display all the employees that earn a salary which is less than the minimum for an IT_PROG



SQLQuery1.sql - LV..._LAPTOP\Luke (53))*

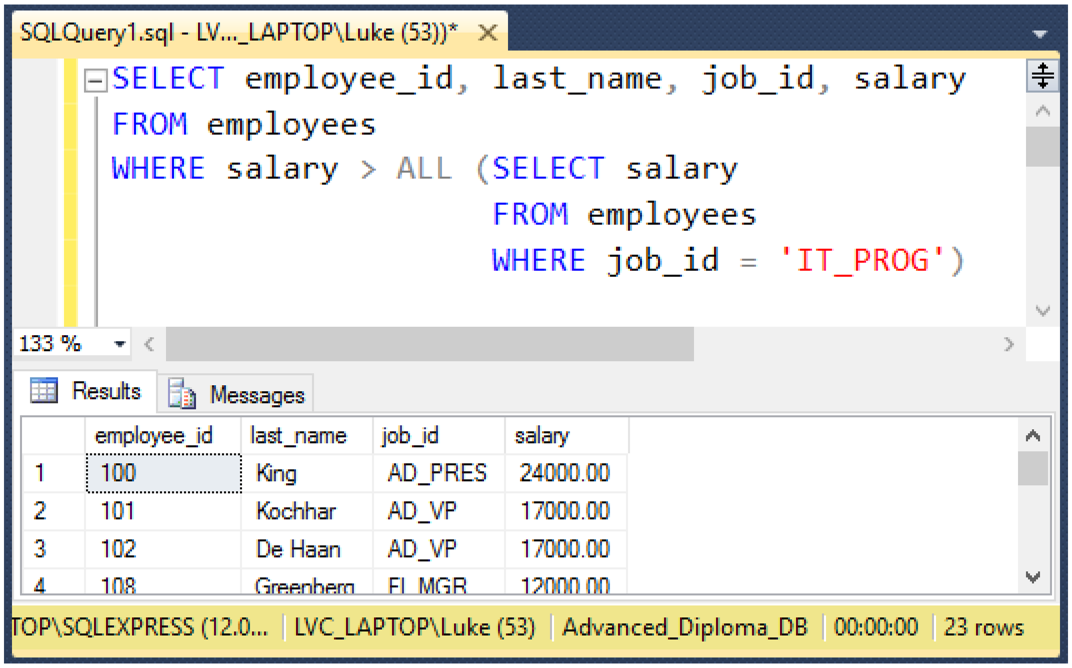
```sql
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL (SELECT salary
                    FROM employees
                    WHERE job_id = 'IT_PROG')
```

	employee_id	last_name	job_id	salary
9	128	Markle	ST_CLERK	2200.00
10	129	Bissot	ST_CLERK	3300.00
11	130	Atkinson	ST_CLERK	2800.00
12	131	Marlow	ST_CLERK	2500.00

TOP\SQLEXPRESS (12.0... | LVC_LAPTOP\Luke (53) | Advanced_Diploma_DB | 00:00:00 | 44 rows

Figure 11 - Result of example 12: employees who earn a salary less than the minimum of IT_PROG

Example 13: Write a query that will display all the employees that earn a salary which is more than the maximum that an IT_PROG can earn



```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary > ALL (SELECT salary
                    FROM employees
                    WHERE job_id = 'IT_PROG')
```

```
   employee_id  last_name  job_id   salary
1  100          King       AD_PRES  24000.00
2  101          Kochhar    AD_VP    17000.00
3  102          De Haan    AD_VP    17000.00
4  108          Greenberg  FI_MGR   12000.00
```

Figure 12 - Result of example 13: employees who earn a salary more than the maximum of IT_PROG

NOTE: If the value of the inner query contains a row which has a null value do not use the NOT IN operator as otherwise the answer would be an empty answer

```
SELECT emp.last_name
FROM employees emp
WHERE emp.employee_id NOT IN (SELECT mgr.manager_id
                              FROM employees mgr);
```