DIVE INTO SQL: UNLEASHING DATA INSIGHTS WITH SELECT, WHERE, AND ORDER BY

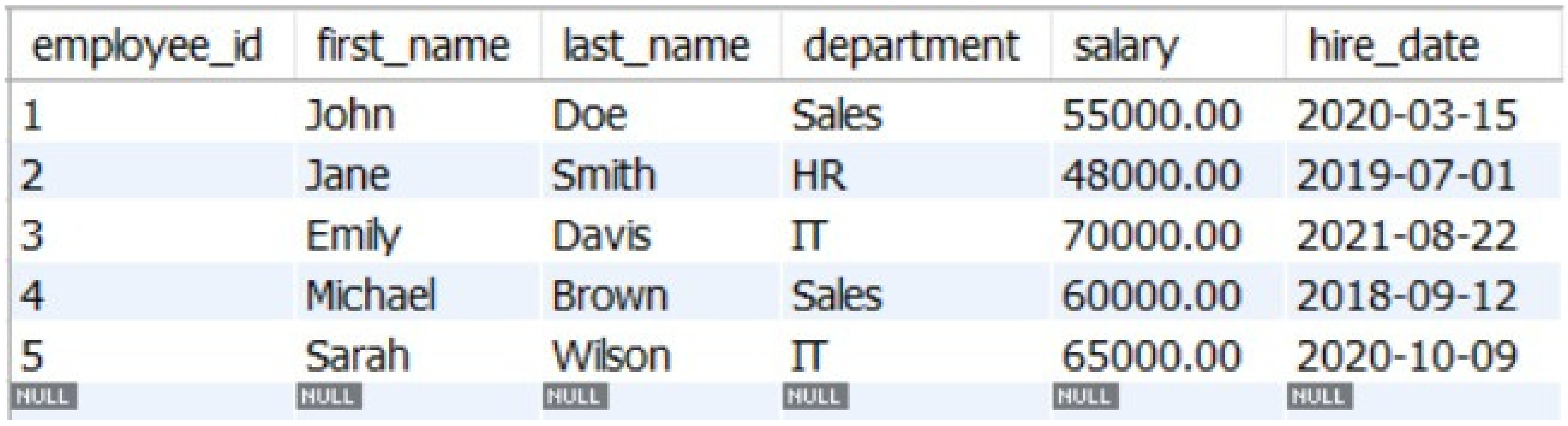
WHAT IS SQL?

- SQL (Structured Query Language) is used to manage and query data in databases. It's a critical tool for anyone working with relational databases.
- Common SQL databases: MySQL, PostgreSQL, SQL Server, etc.
- SQL enables you to retrieve, manipulate, and organize data efficiently.

In this series, we'll dive deep into:

- SELECT: Retrieving specific data.
- WHERE: Filtering based on conditions.
- ORDER BY: Sorting data by categories.

We Generate following data set to execute the codes we study now



employee_id	first_name	last_name	department	salary	hire_date
1	John	Doe	Sales	55000.00	2020-03-15
2	Jane	Smith	HR	48000.00	2019-07-01
3	Emily	Davis	IT	70000.00	2021-08-22
4	Michael	Brown	Sales	60000.00	2018-09-12
5	Sarah	Wilson	IT	65000.00	2020-10-09
NULL	NULL	NULL	NULL	NULL	NULL

MASTERING THE SELECT STATEMENT

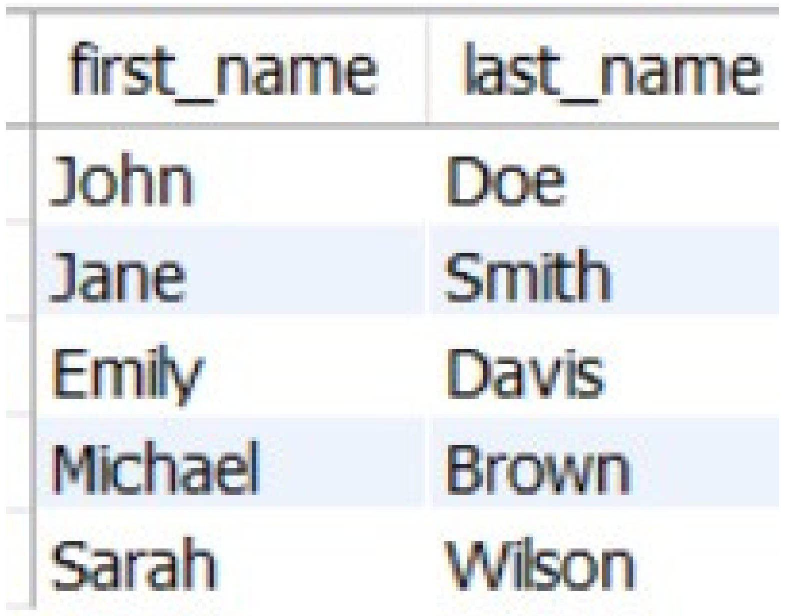
- Pro Tip: Use SELECT * to retrieve all columns.
- SELECT allows you to retrieve specific columns from a table.
- Basic syntax:

```
SELECT column1, column2, ... FROM table_name;
```

- Example Code:

```
SELECT first_name, last_name FROM employees;
```

- Explanation: Retrieves the first and last names from the 'employees' table.



first_name	last_name
John	Doe
Jane	Smith
Emily	Davis
Michael	Brown
Sarah	Wilson

RENAMING COLUMNS WITH ALIASES

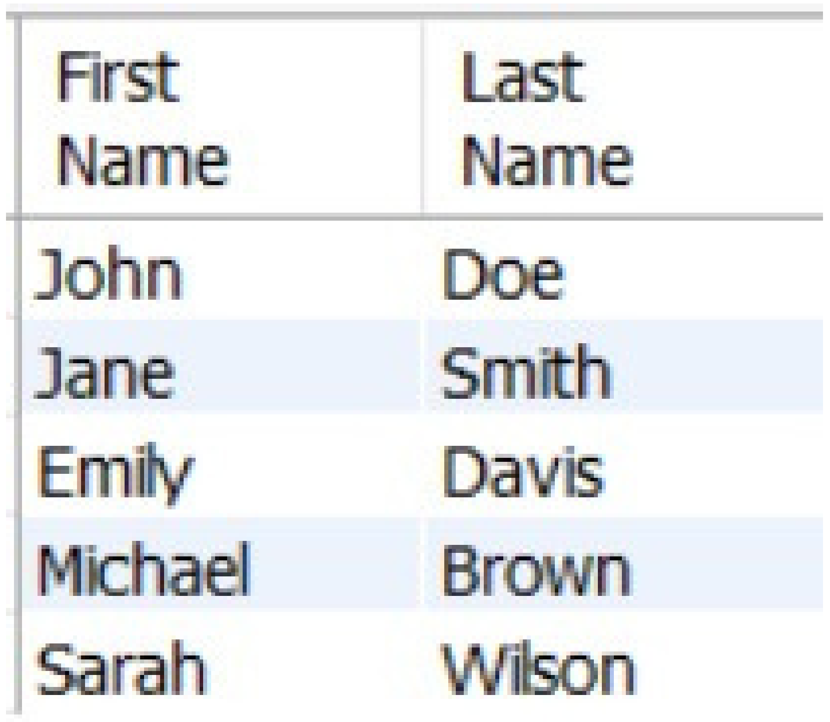
- Aliases allow you to rename columns for better readability.
- Syntax:

```
SELECT column_name AS alias_name FROM table_name;
```

- Example Code:

```
SELECT first_name AS "First Name", last_name AS "Last Name" FROM employees;
```

- Explanation: Renames columns in the output.



First Name	Last Name
John	Doe
Jane	Smith
Emily	Davis
Michael	Brown
Sarah	Wilson

ELIMINATING DUPLICATES WITH DISTINCT

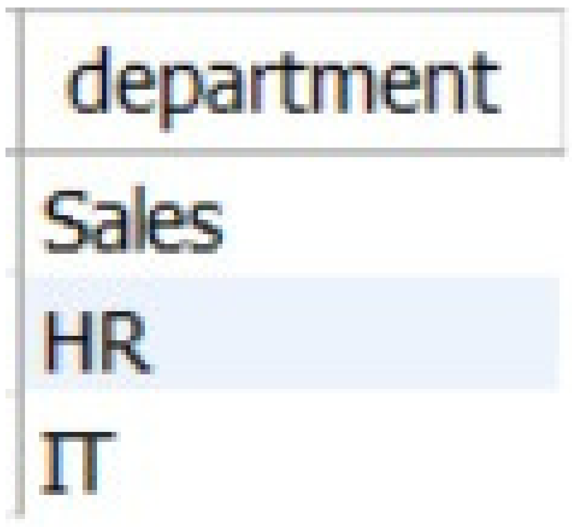
- The DISTINCT keyword removes duplicate rows in the result.
- Syntax:

```
SELECT DISTINCT column1 FROM table_name;
```

- Example Code:

```
SELECT DISTINCT department FROM employees;
```

- Explanation: Retrieves unique departments from the 'employees' table.



department
Sales
HR
IT

FILTERING ROWS WITH THE WHERE CLAUSE

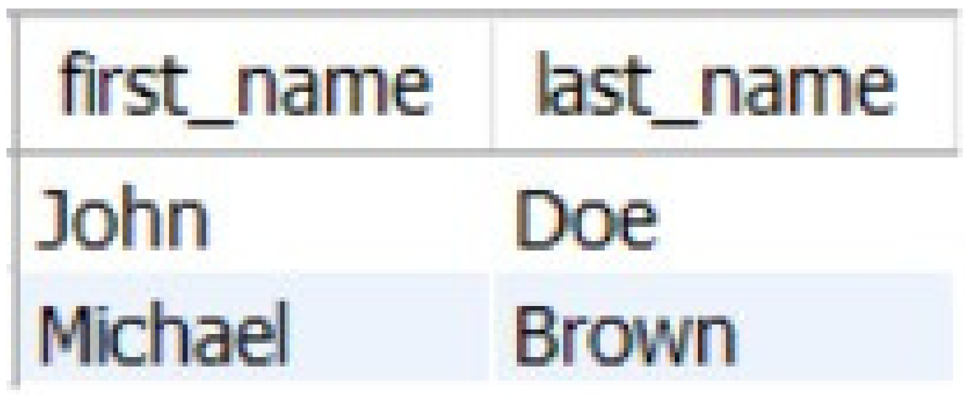
- The WHERE clause filters rows based on a condition.
- Syntax:

```
SELECT column1, column2 FROM table_name WHERE condition;
```

- Example Code:

```
SELECT first_name, last_name FROM employees WHERE department = 'Sales';
```

- Explanation: Filters for employees in the 'Sales' department.



first_name	last_name
John	Doe
Michael	Brown

COMBINING CONDITIONS WITH AND/OR

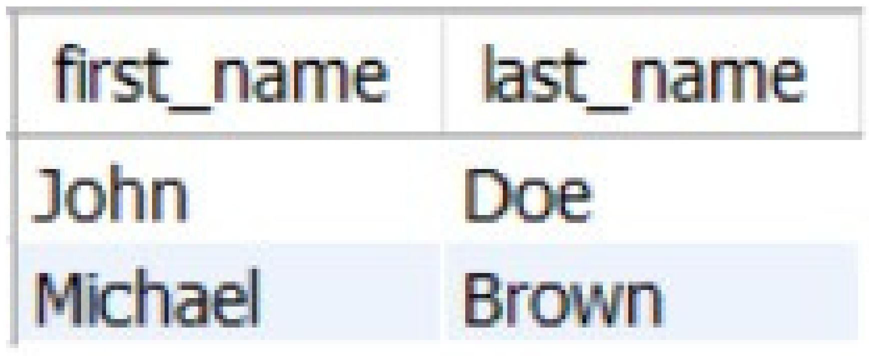
- Combine multiple conditions using AND and OR.
- Syntax:

```
SELECT column1, column2 FROM table_name WHERE condition1 AND/OR condition2;
```

- Example Code:

```
SELECT first_name, last_name
FROM employees
WHERE department = 'Sales' AND salary >
50000;
```

- Explanation: Filters employees in Sales with a salary greater than 50,000.

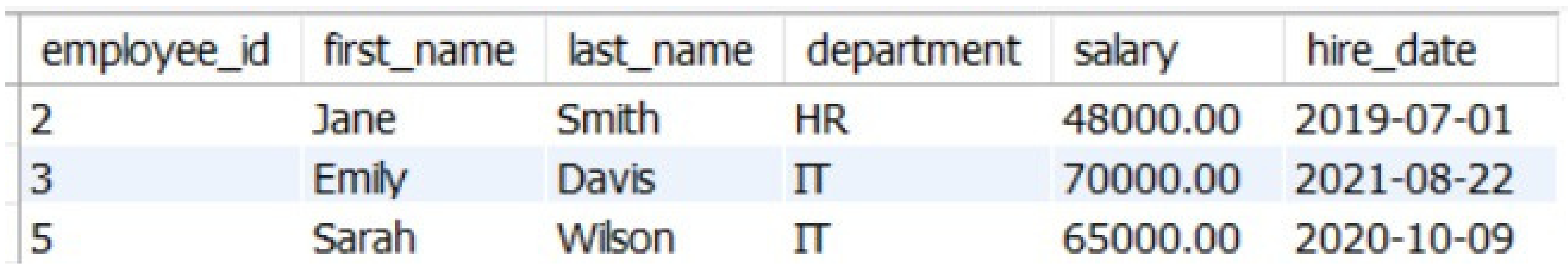


first_name	last_name
John	Doe
Michael	Brown

ADVANCED FILTERING OPTIONS: IN, BETWEEN

- IN: Matches any value in a list.

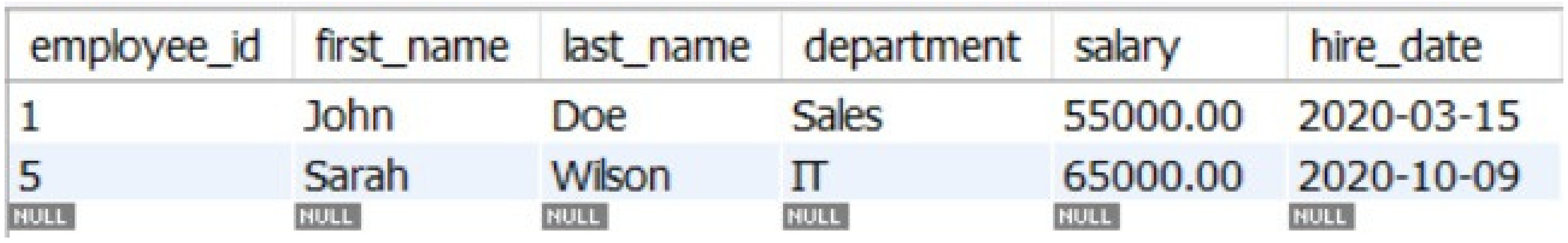
SELECT * FROM employees WHERE department IN ('HR', 'IT');



employee_id	first_name	last_name	department	salary	hire_date
2	Jane	Smith	HR	48000.00	2019-07-01
3	Emily	Davis	IT	70000.00	2021-08-22
5	Sarah	Wilson	IT	65000.00	2020-10-09

- BETWEEN: Filters within a range.

SELECT * FROM employees WHERE hire_date BETWEEN '2020-01-01' AND '2020-12-31';



employee_id	first_name	last_name	department	salary	hire_date
1	John	Doe	Sales	55000.00	2020-03-15
5	Sarah	Wilson	IT	65000.00	2020-10-09
NULL	NULL	NULL	NULL	NULL	NULL

SORTING RESULTS WITH ORDER BY

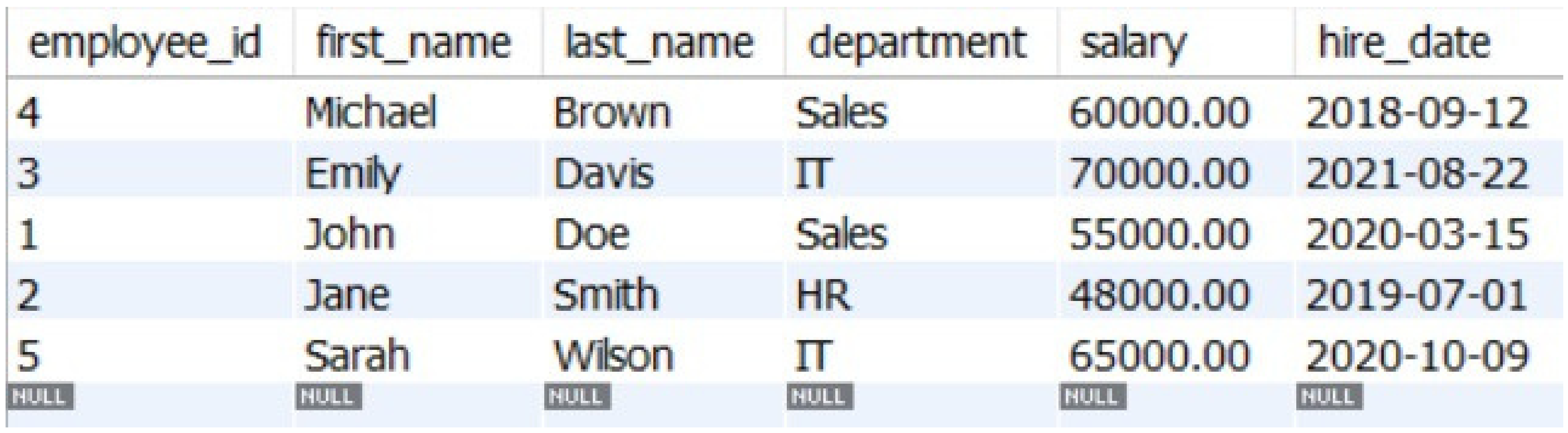
- The ORDER BY clause sorts the result set.
- Syntax:

```
SELECT column1, column2 FROM table_name ORDER BY column_name ASC|DESC;
```

- Example Code:

SELECT * FROM employees ORDER BY last_name ASC; Explanation:

- employees sorted by last name in ascending order.



employee_id	first_name	last_name	department	salary	hire_date
4	Michael	Brown	Sales	60000.00	2018-09-12
3	Emily	Davis	IT	70000.00	2021-08-22
1	John	Doe	Sales	55000.00	2020-03-15
2	Jane	Smith	HR	48000.00	2019-07-01
5	Sarah	Wilson	IT	65000.00	2020-10-09
NULL	NULL	NULL	NULL	NULL	NULL

SORTING MULTIPLE COLUMNS

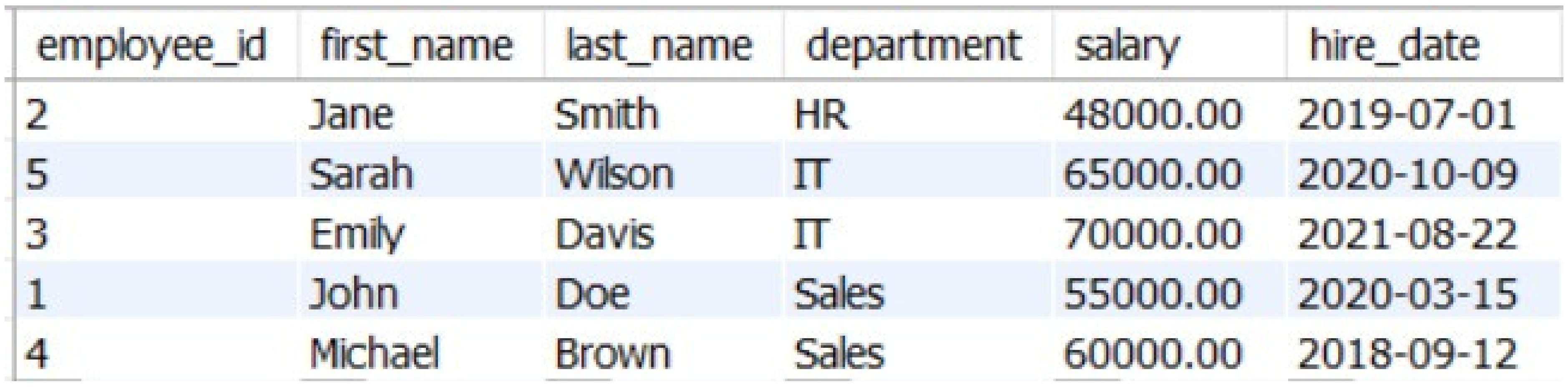
- You can sort by multiple columns.
- Syntax:

```
SELECT column1, column2 FROM table_name ORDER BY column1 ASC, column2 DESC;
```

- Example Code:

```
SELECT * FROM employees ORDER BY department ASC, last_name DESC;
```

- Explanation: Retrieves employees sorted first by department (ascending) and then by last name (descending).



employee_id	first_name	last_name	department	salary	hire_date
2	Jane	Smith	HR	48000.00	2019-07-01
5	Sarah	Wilson	IT	65000.00	2020-10-09
3	Emily	Davis	IT	70000.00	2021-08-22
1	John	Doe	Sales	55000.00	2020-03-15
4	Michael	Brown	Sales	60000.00	2018-09-12