УДК 004.65

5.2.2. – Математические, статистические и инструментальные методы в экономике (физико-математические науки, экономические науки)

ЗНАЧЕНИЕ ОКОННЫХ ФУНКЦИЙ И ИХ ИСПОЛЬЗОВАНИЕ ПРИ ПРИНЯТИИ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ

Рыбянцева Мария Сергеевна
Кандидат экономических наук, доцент
РИНЦ SPIN-код: 7874-8981
email: Riban1@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Авакимян Наталья Николаевна
К.ф.-м.н., доцент
РИНЦ SPIN-код: 6082-4770
email: avnatali@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Оконные функции широко применяются при формировании информации в различных сферах деятельности компаний. В данной статье рассмотрено их использование для получения данных о персонале компании (в качестве примера условно выделены три отдела: технический отдел, отдел продаж, отдел снабжения). На конкретных примерах описано применение ранжирующих функций и агрегатных функций. Приведено назначение каждой функции, SQL-запрос, а также результаты его применения. Были подробно описаны следующие функции: ROW_NUMBER(); DENSE_RANK() и RANK(); NTILE(); AVG(); SUM(); COUNT(); MIN() и MAX(). Получена информация о: сотрудниках, пронумерованных в порядке повышения зарплаты по компании в целом, а также в рамках своего отдела; описаны отличия в ранжировании при использовании функций ROW_NUMBER(), DENSE_RANK() и RANK(); подходах к делению персонала на группы; средней, минимальной и максимальной заработной плате в разрезе каждого отдела; разнице между заработной платой сотрудника и средней величиной показателя; численности сотрудников каждого отдела и др. Описано назначение функций смещения

Ключевые слова: ОКОННЫЕ ФУНКЦИИ, РЕЗУЛЬТИРУЮЩИЙ НАБОР, РАНЖИРУЮЩИЕ ФУНКЦИИ, АГРЕГАТНЫЕ ФУНКЦИИ, ФУНКЦИИ СМЕЩЕНИЯ

http://dx.doi.org/10.21515/1990-4665-207-032

UDC 004.65

5.2.2. "Mathematical, statistical and instrumental methods in economics" (physical and mathematical sciences, economic sciences)

THE IMPORTANCE OF WINDOW FUNCTIONS AND THEIR APPLICATION IN MANAGEMENT DECISION MAKING

Rybyantseva Maria Sergeevna
Candidate of Economic Sciences associate professor
RSCI SPIN-code: 7874-8981
email: Riban1@mail.ru
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Avakimyan Natalia Nikolaevna
Cand.Phys.-Math.Sci., associate professor,
RSCI SPIN-code: 6082-4770
email: avnatali@mail.ru.
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Window functions are widely used to generate information in various areas of company activity. This article discusses their use to obtain data on a company's personnel (three departments are distinguished as an example: technical department, sales department, supply department). Specific examples are used to describe the application of ranking functions and aggregate functions. The purpose of each function, the SQL query, and the results of its application are provided. The following functions were described in detail: OW_NUMBER(); DENSE_RANK() and RANK(); NTILE(); AVG(); SUM(); COUNT(); MIN() and MAX(). Information was obtained on: employees numbered in order of salary increase for the company as a whole, as well as within their department; differences in ranking when using the ROW_NUMBER(), DENSE_RANK() and RANK() functions are described; approaches to dividing personnel into groups; average, minimum and maximum wages for each department; the difference between the employee's salary and the average value of the indicator; the number of employees in each department, etc. The purpose of the offset functions is described

Keywords: WINDOW FUNCTIONS, RESULT SET, RANKING FUNCTIONS, AGGREGATE FUNCTIONS, OFFSET FUNCTIONS

Оконная функция – это функция, которая выполняет вычисления на основе определенного набора записей и возвращает одиночное значение. Набор записей, с которым работает оконная функция, называют окном. По умолчанию содержимое окна полностью совпадает с результирующим набором запроса, в рамках которого определено окно.

Оконные функции обычно подразделяют на три основные группы (таблица 1).

Таблица 1 – Группы оконных функций

Группа	Функции
Ранжирующие функции	а) ROW_NUMBER(); б) DENSE_RANK() и RANK(); в) NTILE();
Агрегатные функции	а) AVG(); б) SUM(); в) COUNT(); г) MIN() и MAX();
Функции смещения	а) FIRST_VALUE(); б) LAST_VALUE(); в) NTH_VALUE(); г) LAG(); д) LEAD()

Результирующий набор – таблица, которая формируется извлекающим запросом в результате выполнения следующих операций: соединение (JOIN), извлечение (FROM), фильтрация (WHERE), группировка (GROUP BY) и фильтрация групп (HAVING). Результат запроса формируется на основе результирующего набора.

Рассмотрим базу данных, которая будет использоваться в последующих примерах, иллюстрирующих информационные возможности оконных функций. Она состоит из одной таблицы с именем Staff (сотрудники отделов продаж, снабжения и технического отдела) (таблица 2).

Функция ROW_NUMBER() вычисляет порядковый номер записи в рамках указанного окна (начиная с 1).

Таблица 2 – Сотрудники компании

id	full_name	department	salary
1	Беседин Матвей Сергеевич	Технический отдел	55000
2	Бородин Ярослав Романович	Отдел продаж	55000
3	Даниленко Максим Андреевич	Технический отдел	45000
4	Ковалев Никита Андреевич	Отдел снабжения	49000
5	Назлуханов Владимир Олегович	Отдел продаж	60000
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000
7	Тихомирова Екатерина Максимовна	Технический отдел	49000
8	Дырко Артем Александрович	Технический отдел	45000
9	Хан Виолетта Владимировна	Отдел снабжения	52000
10	Маркина Мария Олеговна	Технический отдел	55000
11	Цылюрик Артем Александрович	Отдел продаж	49000
12	Меркулов Максим Евгеньевич	Технический отдел	52000

Пример 1. Необходимо получить информацию о сотрудниках, пронумерованных в порядке повышения зарплаты.

С этой целью используем следующий запрос:

```
SELECT id, full_name, department, salary,
    ROW_NUMBER() OVER salary_asc AS row_num
FROM Staff
WINDOW salary_asc AS (ORDER BY salary);
```

Результат выполнения запроса приведен в таблице 3.

Таблица 3 – Результат применения функции ROW_NUMBER()

id	full_name	department	salary	row_ num
3	Даниленко Максим Андреевич	Технический отдел	45000	1
8	Дырко Артем Александрович	Технический отдел	45000	2
4	Ковалев Никита Андреевич	Отдел снабжения	49000	3
7	Тихомирова Екатерина Максимовна	Технический отдел	49000	4
11	Цылюрик Артем Александрович	Отдел продаж	49000	5
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000	6
9	Хан Виолетта Владимировна	Отдел снабжения	52000	7
12	Меркулов Максим Евгеньевич	Технический отдел	52000	8
1	Беседин Матвей Сергеевич	Технический отдел	55000	9
2	Бородин Ярослав Романович	Отдел продаж	55000	10
10	Маркина Мария Олеговна	Технический отдел	55000	11
5	Назлуханов Владимир Олегович	Отдел продаж	60000	12

Секционирование используется для разбиения окна на секции по определенному полю (или нескольким полям). Две записи окна попадают в одну секцию, если их значения по выбранному полю (или нескольким полям) совпадают.

Пример 2. Напишем запрос, который извлекает информацию о сотрудниках, дополнительно пронумеровывая их в рамках своего отдела (сортировка в рамках отдела по возрастанию заработной платы).

Для этого используется следующий запрос:

```
SELECT id, full_name, department, salary,
    ROW_NUMBER() OVER part_by_dep AS row_num
FROM Staff
WINDOW part_by_dep AS (PARTITION BY department ORDER BY salary);
```

Результатом выполнения запроса является таблица 4.

Таблица 4 – Результат выполнения запроса по нумерации сотрудников в рамках отдела

id	full_name	department	salary	row_num
11	Цылюрик Артем Александрович	Отдел продаж	49000	1
2	Бородин Ярослав Романович	Отдел продаж	55000	2
5	Назлуханов Владимир Олегович	Отдел продаж	60000	3
4	Ковалев Никита Андреевич	Отдел снабжения	49000	1
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000	2
9	Хан Виолетта Владимировна	Отдел снабжения	52000	3
3	Даниленко Максим Андреевич	Технический отдел	45000	1
8	Дырко Артем Александрович	Технический отдел	45000	2
7	Тихомирова Екатерина Максимовна	Технический отдел	49000	3
12	Меркулов Максим Евгеньевич	Технический отдел	52000	4
1	Беседин Матвей Сергеевич	Технический отдел	55000	5
10	Маркина Мария Олеговна	Технический отдел	55000	6

Функция DENSE_RANK() вычисляет ранг записи в рамках указанного окна. Первой записи окна присваивается ранг, равный 1.

Увеличение ранга происходит только если очередная запись в поле (или полях) содержит значение, отличное от значения предыдущей записи.

Пример 3. Проведем ранжирование сотрудников с помощью функции DENSE_RANK(), используя сортировку по возрастанию заработной платы.

Для этого используется следующий запрос:

```
SELECT id, full_name, department, salary,
    DENSE_RANK() OVER (ORDER BY salary) AS dense_ran
FROM Staff;
```

Результат выполнения запроса приведен в таблице 5.

Таблица 5 – Результат применения функции DENSE_RANK()

id	full_name	department	salary	dense_ ran
3	Даниленко Максим Андреевич	Технический отдел	45000	1
8	Дырко Артем Александрович	Технический отдел	45000	1
4	Ковалев Никита Андреевич	Отдел снабжения	49000	2
7	Тихомирова Екатерина Максимовна	Технический отдел	49000	2
11	Цылюрик Артем Александрович	Отдел продаж	49000	2
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000	3
9	Хан Виолетта Владимировна	Отдел снабжения	52000	3
12	Меркулов Максим Евгеньевич	Технический отдел	52000	3
1	Беседин Матвей Сергеевич	Технический отдел	55000	4
2	Бородин Ярослав Романович	Отдел продаж	55000	4
10	Маркина Мария Олеговна	Технический отдел	55000	4
5	Назлуханов Владимир Олегович	Отдел продаж	60000	5

Функция DENSE_RANK() выполняет ранжирование без пропусков, в то время как функция RANK() при повторении рангов следующий ранг отбрасывает.

Пример 4. Проведем ранжирование сотрудников с помощью функции RANK(), используя сортировку по возрастанию заработной платы.

Составим следующий запрос:

```
SELECT id, full_name, department, salary,
    DENSE_RANK() OVER salary AS dense_rank_function,
    RANK() OVER salary AS rank_function
FROM Staff
WINDOW salary AS (ORDER BY salary);
```

Результат выполнения запроса приведен в таблице 6.

Таблица 6 – Результат сравнения функций DENSE_RANK() и RANK()

id	full_name	department	salary	dense_ rank_ function	rank_ function
3	Даниленко Максим Андреевич	Технический отдел	45000	1	1
8	Дырко Артем Александрович	Технический отдел	45000	1	1
4	Ковалев Никита Андреевич	Отдел снабжения	49000	2	3
7	Тихомирова Екатерина Максимовна	Технический отдел	49000	2	3
11	Цылюрик Артем Александрович	Отдел продаж	49000	2	3
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000	3	6
9	Хан Виолетта Владимировна	Отдел снабжения	52000	3	6
12	Меркулов Максим Евгеньевич	Технический отдел	52000	3	6
1	Беседин Матвей Сергеевич	Технический отдел	55000	4	9
2	Бородин Ярослав Романович	Отдел продаж	55000	4	9
10	Маркина Мария Олеговна	Технический отдел	55000	4	9
5	Назлуханов Владимир Олегович	Отдел продаж	60000	5	12

Функция NTILE() используется для разбиения окна на заданное количество групп. Она принимает в качестве аргумента целое число n, разбивает окно на n групп и определяет номер группы, к которой относится запись.

Пример 5. Распределим всех сотрудников по четырем группам.

Составим следующий запрос:

```
SELECT id, full_name, department,
    NTILE(4) OVER (ORDER BY id) AS group_num
FROM Staff;
```

Результат выполнения запроса приведен в таблице 7.

Таблица 7 – Результат применения функции NTILE()

id	full_name	department	group_num
1	Беседин Матвей Сергеевич	Технический отдел	1
2	Бородин Ярослав Романович	Отдел продаж	1
3	Даниленко Максим Андреевич	Технический отдел	1
4	Ковалев Никита Андреевич	Отдел снабжения	2
5	Назлуханов Владимир Олегович	Отдел продаж	2
6	Крикунова Анна Дмитриевна	Отдел снабжения	2
7	Тихомирова Екатерина Максимовна	Технический отдел	3
8	Дырко Артем Александрович	Технический отдел	3
9	Хан Виолетта Владимировна	Отдел снабжения	3
10	Маркина Мария Олеговна	Технический отдел	4
11	Цылюрик Артем Александрович	Отдел продаж	4
12	Меркулов Максим Евгеньевич	Технический отдел	4

Пример 6. Распределим всех сотрудников по пяти группам.

Составим следующий запрос:

```
SELECT id, full_name, department,
    NTILE(5) OVER (ORDER BY id) AS group_num
FROM Staff;
```

Результат выполнения запроса приведен в таблице 8.

Таблица 8 – Результат применения функции NTILE() при разбиении по 5 группам

id	full_name	department	group_num
1	Беседин Матвей Сергеевич	Технический отдел	1
2	Бородин Ярослав Романович	Отдел продаж	1
3	Даниленко Максим Андреевич	Технический отдел	1
4	Ковалев Никита Андреевич	Отдел снабжения	2
5	Назлуханов Владимир Олегович	Отдел продаж	2
6	Крикунова Анна Дмитриевна	Отдел снабжения	2
7	Тихомирова Екатерина Максимовна	Технический отдел	3
8	Дырко Артем Александрович	Технический отдел	3
9	Хан Виолетта Владимировна	Отдел снабжения	4
10	Маркина Мария Олеговна	Технический отдел	4
11	Цылюрик Артем Александрович	Отдел продаж	5

12	Меркулов Максим Евгеньевич	Технический отдел	5

Если число записей (m) невозможно нацело разбить на число групп (n), то первые m % n (остаток от деления m на n) групп будут содержать ровно на 1 запись больше, чем остальные группы. В нашем примере 12 %5 = 2, то есть первые две группы будут содержать на одну запись больше, чем остальные (т.е. по три записи, а не по две, как 3, 4 и 5 группы).

Агрегатные оконные функции – это агрегатные функции группировки, применяемые к окнам. Использование агрегатных функций позволяет выполнять агрегатные вычисления на основе набора записей.

Функция AVG() вычисляет среднее арифметическое числовых значений, хранящихся в определенном поле окна.

Пример 7. Необходимо написать запрос, в результате которого рядом с заработной платой сотрудника указывалась средняя заработная плата по компании.

Составим следующий запрос:

```
SELECT id, full_name, department, salary,
    AVG(salary) OVER () AS avg_salary
FROM Staff;
```

Результатом выполнения запроса является таблица 9.

Таблица 9 – Результат выполнения запроса по определению средней заработной платы

id	full_name	department	salary	avg_salary
1	Беседин Матвей Сергеевич	Технический отдел	55000	51500,0
2	Бородин Ярослав Романович	Отдел продаж	55000	51500,0
3	Даниленко Максим Андреевич	Технический отдел	45000	51500,0
4	Ковалев Никита Андреевич	Отдел снабжения	49000	51500,0
5	Назлуханов Владимир Олегович	Отдел продаж	60000	51500,0
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000	51500,0
7	Тихомирова Екатерина Максимовна	Технический отдел	49000	51500,0
8	Дырко Артем Александрович	Технический отдел	45000	51500,0
9	Хан Виолетта Владимировна	Отдел снабжения	52000	51500,0
10	Маркина Мария Олеговна	Технический отдел	55000	51500,0
11	Цылюрик Артем Александрович	Отдел продаж	49000	51500,0

12	Меркулов Максим Евгеньевич	Технический отдел	52000	51500,0

Пример 8. Рассчитаем разницу между заработной платой сотрудника и средней заработной платой.

Составим следующий запрос:

```
SELECT id, full_name, salary,
    AVG(salary) OVER () AS avg_salary,
    ABS(salary - AVG(salary) OVER ()) AS salary_diff
FROM Staff;
```

Результатом выполнения запроса является таблица 10.

Таблица 10 – Результат выполнения запроса по определению разницы в величине заработной платы

id	full_name	salary	avg_salary	salary_diff
1	Беседин Матвей Сергеевич	55000	51500,0	3500,0
2	Бородин Ярослав Романович	55000	51500,0	3500,0
3	Даниленко Максим Андреевич	45000	51500,0	6500,0
4	Ковалев Никита Андреевич	49000	51500,0	2500,0
5	Назлуханов Владимир Олегович	60000	51500,0	8500,0
6	Крикунова Анна Дмитриевна	52000	51500,0	500,0
7	Тихомирова Екатерина Максимовна	49000	51500,0	2500,0
8	Дырко Артем Александрович	45000	51500,0	6500,0
9	Хан Виолетта Владимировна	52000	51500,0	500,0
10	Маркина Мария Олеговна	55000	51500,0	3500,0
11	Цылюрик Артем Александрович	49000	51500,0	2500,0
12	Меркулов Максим Евгеньевич	52000	51500,0	500,0

Пример 9. Определим среднюю зарплату в рамках определенного отдела.

Составим следующий запрос:

```
SELECT id, full_name, department, salary,
    round(AVG(salary) OVER (PARTITION BY department),2) AS dep_avg_salary
FROM Staff
```

При округлении оставим два знака после запятой. Результатом выполнения запроса является таблица 11. При использовании агрегатных оконных функций порядок записей в окне не имеет значения.

Функция SUM() вычисляет сумму числовых значений, хранящихся в определенном поле окна.

Таблица 11 – Результат выполнения запроса по определению средней заработной платы по отделу

id	full_name	department	salary	dep_avg_ salary
2	Бородин Ярослав Романович	Отдел продаж	55000	54666,67
5	Назлуханов Владимир Олегович	Отдел продаж	60000	54666,67
11	Цылюрик Артем Александрович	Отдел продаж	49000	54666,67
4	Ковалев Никита Андреевич	Отдел снабжения	49000	51000,0
6	Крикунова Анна Дмитриевна	Отдел снабжения	52000	51000,0
9	Хан Виолетта Владимировна	Отдел снабжения	52000	51000,0
1	Беседин Матвей Сергеевич	Технический отдел	55000	50166,67
3	Даниленко Максим Андреевич	Технический отдел	45000	50166,67
7	Тихомирова Екатерина Максимовна	Технический отдел	49000	50166,67
8	Дырко Артем Александрович	Технический отдел	45000	50166,67
10	Маркина Мария Олеговна	Технический отдел	55000	50166,67
12	Меркулов Максим Евгеньевич	Технический отдел	52000	50166,67

Пример 10. Необходимо написать запрос, позволяющий получить информацию о заработной плате сотрудника, суммарной заработной плате по компании, а также суммарной заработной плате по отделу.

Используем следующий запрос:

```
SELECT full_name, department, salary,
    SUM(salary) OVER () AS total_salary,
    SUM(salary) OVER (PARTITION BY department) AS dep_salary
FROM Staff;
```

Результатом выполнения запроса является таблица 12.

Таблица 12 – Результат выполнения запроса по определению суммарной заработной платы по компании и отделу

full_name	department	salary	total_ salary	dep_ salary
Бородин Ярослав Романович	Отдел продаж	55000	618000	164000
Назлуханов Владимир Олегович	Отдел продаж	60000	618000	164000
Цылюрик Артем Александрович	Отдел продаж	49000	618000	164000
Ковалев Никита Андреевич	Отдел снабжения	49000	618000	153000
Крикунова Анна Дмитриевна	Отдел снабжения	52000	618000	153000

Окончание таблицы 12

full_name	department	salary	total_ salary	dep_ salary
Хан Виолетта Владимировна	Отдел снабжения	52000	618000	153000
Беседин Матвей Сергеевич	Технический отдел	55000	618000	301000
Даниленко Максим Андреевич	Технический отдел	45000	618000	301000
Тихомирова Екатерина Максимовна	Технический отдел	49000	618000	301000
Дырко Артем Александрович	Технический отдел	45000	618000	301000
Маркина Мария Олеговна	Технический отдел	55000	618000	301000
Меркулов Максим Евгеньевич	Технический отдел	52000	618000	301000

Функция COUNT() используется для подсчета записей в окне или непустых значений в поле окна.

Пример 11. Необходимо написать запрос, который для каждого сотрудника указывает количество людей, работающих в его отделе.

Используем следующий запрос:

```
SELECT full_name, department,
    COUNT(*) OVER (PARTITION BY department) AS number
FROM Staff;
```

Результатом выполнения запроса является таблица 13.

Таблица 13 – Результат выполнения запроса по определению количества работников в отделе

full_name	department	number
Бородин Ярослав Романович	Отдел продаж	3
Назлуханов Владимир Олегович	Отдел продаж	3
Цылюрик Артем Александрович	Отдел продаж	3
Ковалев Никита Андреевич	Отдел снабжения	3
Крикунова Анна Дмитриевна	Отдел снабжения	3
Хан Виолетта Владимировна	Отдел снабжения	3
Беседин Матвей Сергеевич	Технический отдел	6
Даниленко Максим Андреевич	Технический отдел	6
Тихомирова Екатерина Максимовна	Технический отдел	6
Дырко Артем Александрович	Технический отдел	6
Маркина Мария Олеговна	Технический отдел	6
Меркулов Максим Евгеньевич	Технический отдел	6

Функция MIN() вычисляет минимальное значение, хранящиеся в определенном поле окна, функция MAX() – максимальное.

Пример 12. Необходимо написать запрос, который зарплате каждого сотрудника ставит в соответствие наименьшую и наибольшую зарплату среди сотрудников его отдела.

Результатом приведенного ниже запроса:

```
SELECT full_name, department,
    MIN(salary) OVER (PARTITION BY department) AS min_salary,
    MAX(salary) OVER (PARTITION BY department) AS max_salary
FROM Staff;
```

Результатом выполнения запроса является таблица 14.

Таблица 14 – Результат выполнения запроса по определению минимальной и максимальной заработной платы в отделе

full_name	department	min_salary	max_salary
Бородин Ярослав Романович	Отдел продаж	49000	60000
Назлуханов Владимир Олегович	Отдел продаж	49000	60000
Цылюрик Артем Александрович	Отдел продаж	49000	60000
Ковалев Никита Андреевич	Отдел снабжения	49000	52000
Крикунова Анна Дмитриевна	Отдел снабжения	49000	52000
Хан Виолетта Владимировна	Отдел снабжения	49000	52000
Беседин Матвей Сергеевич	Технический отдел	45000	55000
Даниленко Максим Андреевич	Технический отдел	45000	55000
Тихомирова Екатерина Максимовна	Технический отдел	45000	55000
Дырко Артем Александрович	Технический отдел	45000	55000
Маркина Мария Олеговна	Технический отдел	45000	55000
Меркулов Максим Евгеньевич	Технический отдел	45000	55000

Окно может обладать еще одной спецификацией – определением границ (позволяет обозначить, с какими именно записями окна должна взаимодействовать оконная функция).

Определение границ окна выполняется путем указания двух граничных точек: начальной и конечной.

Фрейм – часть окна, получаемая в результате применения установленных границ.

Определить границы окна можно с помощью оператора ROWS:

ROWS BETWEEN <начальная граничная точка> AND <конечная граничная точка>

Значения начальной и конечной граничных точек приведены в таблице 15.

Таблица 15 – Значения начальной и конечной граничных точек

Граничная точка	Значение	Расшифровка
Начальная граничная точка	CURRENT ROW	текущая запись
	n PRECEDING	n-запись перед текущей
	n FOLLOWING	n-запись после текущей
	UNBOUNDED PRECEDING	самая первая запись окна (или секции окна, если было применено секционирование)
Конечная граничная точка	CURRENT ROW	текущая запись
	n PRECEDING	n-запись перед текущей
	n FOLLOWING	n-запись после текущей
	UNBOUNDED FOLLOWING	самая последняя запись окна (или секции окна, если было применено секционирование)

Оператор ROWS в определении окна должен располагаться после операторов PARTITION BY и ORDER BY.

Определить границы окна также можно с помощью оператора RANGE, синтаксис использования которого имеет следующий вид:

RANGE BETWEEN <начальная граничная точка> AND <конечная граничная точка>

Сущность функций смещения отражена в таблице 16.

Таблица 16 – Функции смещения

Функция	Назначение
FIRST_VALUE ()	используется для получения значения, которое содержится в определенном поле первой записи окна
LAST_VALUE()	используется для получения значения, которое содержится в определенном поле последней записи окна
NTH_VALUE()	используется для получения значения, которое содержится в определенном поле n-ой записи окна (начиная с 1)

Окончание таблицы 16

Функция	Назначение
LAG()	используется для получения значения, которое содержится в определенном поле записи окна, отстающей от текущей на n
LAG()	возвращает значение NULL, если пытается получить значение записи, выходящей за рамки окна
LEAD()	используется для получения значения, которое содержится в определенном поле записи окна, опережающей текущую на n

Функции смещения можно использовать и при работе с информацией о сотрудниках компании, но, наиболее эффективно их применение при оценке динамики и структуры финансовых результатов, а также их прогнозировании.

Список использованной литературы:

1. Применение аналитических возможностей оконных функций при анализе финансовых результатов /Оксанич Е. А., Рыбянцева М. С.// Естественно-гуманитарные исследования. 2024. № 2(52). - С. 203-207

2. Использование многотабличных запросов при принятии управленческих решений / Авакимян Н. Н., Рыбянцева М. С. // Вестник академии знаний. 2024. № 3(62). - С.666-670

3. Базы и банки данных : учеб. пособие / М. И. Попова, М. С. Рыбянцева. – Краснодар : КубГАУ, 2024. – 106 с.

References:

1. Primenenie analiticheskih vozmozhnostej okonnyh funkcij pri analize finansovyh rezul'tatov /Oksanich E. A., Rybjanceva M. S.// Estestvenno-gumanitarnye issledovanija. 2024. № 2(52). - S. 203-207

2. Ispol'zovanie mnogotablichnyh zaprosov pri prinjatii upravlencheskih reshenij / Avakimjan N. N., Rybjanceva M. S. // Vestnik akdemii znanij. 2024. № 3(62). - S.666-670

3. Bazy i banki dannyh : ucheb. posobie / M. I. Popova, M. S. Rybjanceva. – Krasnodar : KubGAU, 2024. – 106 s.