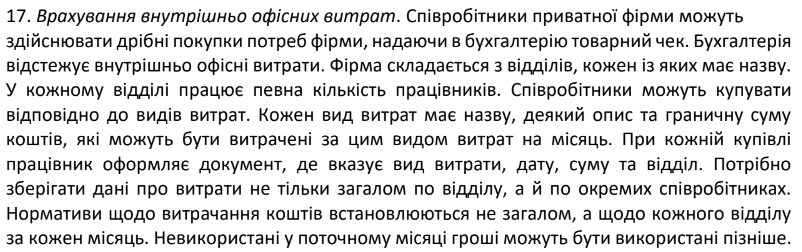
Ровний Григорій Олександрович

№ 17, врахування внутрішньо-офісних витрат

Лабораторна робота №4

Агрегатні функції. Сортування та групування. Обмеження та виведення даних.



|  |  |
| --- | --- |
| Завдання 1. Використати агрегатні функції, не менше чотирьох **БЕЗ GROUP BY, ORDER BY** Спробувати **застосувати функції MAX (), MIN (), AVG ()**. | |
| Скрипт | Скрін |
| -- Count (без Group/Order by)  SELECT COUNT(\*) AS total\_employees FROM Employees;  -- AVG (без Group/Order by)  SELECT AVG(age) AS average\_age FROM Employees;  -- MAX (без Group/Order by)  SELECT MAX(age) AS max\_age FROM Employees;  -- MIN (без Group/Order by)  SELECT MIN(age) AS min\_age FROM Employees; |  |
| Використати агрегатні функції, не менше чотирьох **ІЗ застосуванням GROUP BY, ORDER BY**. | |
| Скрін | Скрін |
| SELECT department\_id, MAX(age) AS MaxAge, MIN(age) AS MinAge, AVG(age) AS AvgAge, COUNT(\*) AS TotalEmployees  FROM Employees  GROUP BY department\_id  ORDER BY MinAge DESC; |  |
| Завдання 2. Створити запит з використанням розділу **HAVING** для задання умов, які містять агрегатні функції та застосовуються після групування даних; | |
| Скрипт | Скрін |
| -- Використання Having (Аналог Where для агрег. ф-цій)  SELECT department\_id, Employees.last\_name, AVG(age) AS AvgAge, COUNT(\*) AS TotalEmployees  FROM Employees  GROUP BY department\_id, Employees.last\_name  HAVING AvgAge > 30; |  |
| Завдання 3. Зробити запити, які демонструють відмінність між розділами HAVING та WHERE у випадках коли проводимо та коли не проводимо групування даних; | |
| Скрипт | Скрін |
| SELECT \* FROM Employees WHERE age > 28;  **HAVING може використовуватись лише разом з GROUP BY, і щоб відібрати значення за умовою але після групування. Where – навпаки, застосовується лише до одиничних значень. В нього є обмеження – не можна застосовувати агрегатні функції:**  SELECT department\_id, AVG(age) AS AvgAge  FROM Employees  GROUP BY department\_id  HAVING AvgAge > 28; |  |
| Завдання 4. Створити запит **використовуючи оператори DISTINCT та LIMIT** та ознайомитись з його діалектом **TOP** | |
| Скрипт | Скрін |
| -- Вибір унікальних значень (**DISTINCT**)  SELECT DISTINCT category\_name FROM expensecategories;  -- Вибір перших 3 працівників за алфавітом  SELECT \* FROM Employees ORDER BY last\_name LIMIT 3;  **TOP не підтримується в MySQL** |  |