

Ковальов Олександр Олексійович

№22, **Зайнятість акторів театру**

Лабораторна робота №4

Вибірка даних з використанням розділів GROUP BY і HAVING

Використати агрегатні функції MAX(), MIN(), AVG() та COUNT() без GROUP BY та із застосуванням GROUP BY:

Максимальна кількість років досвіду серед акторів:

```
SELECT MAX(years_of_experience) AS 'Max years of experience'
FROM actors;
```

	Max years of experience`
1	9

Аналіз: який максимальний бонус серед відповідних діапазонів зарплат:

```
SELECT salary, MAX(bonus) FROM actors
INNER JOIN contracts c on actors.id = c.actors_id
GROUP BY salary;
```

	salary`	MAX(bonus)`
1	6000	3000
2	5500	2000
3	8000	2000
4	12000	2000
5	2000	0
6	0	0
7	5000	0
8	4000	0
9	3000	0
10	1000	0

Мінімальна кількість років досвіду серед акторів:

```
SELECT MIN(years_of_experience)
FROM actors;
```

	MIN(years_of_experience)`
1	0

Аналіз: який мінімальний бонус серед відповідних діапазонів зарплат:

```
SELECT salary, MIN(bonus) FROM actors
INNER JOIN contracts c on actors.id = c.actors_id
GROUP BY salary;
```

	salary`	MIN(bonus)`
1	6000	0
2	5500	2000
3	8000	0
4	12000	2000
5	2000	0
6	0	0
7	5000	0
8	4000	0
9	3000	0
10	1000	0

Середня заробітна плата в театрі:

```
SELECT AVG(salary)
FROM contracts;
```

	AVG(salary)
1	4340.0000

Середній бонус відповідно діапазонів заробітної плати:

```
SELECT salary, AVG(bonus) AS average FROM actors
INNER JOIN contracts c on actors.id = c.actors_id
GROUP BY salary
ORDER BY average DESC, salary DESC;
```

	salary	average
1	12000	2000.0000
2	5500	2000.0000
3	8000	1000.0000
4	6000	1000.0000
5	5000	0.0000
6	4000	0.0000
7	3000	0.0000
8	2000	0.0000
9	1000	0.0000
10	0	0.0000

Кількість акторів, які грають якусь роль:

```
SELECT COUNT(DISTINCT actors_id) AS Actors
FROM additionaltable;
```

	Actors
1	5

Кількість шоу за місяцями:

```
SELECT COUNT(title) AS 'shows', MONTH(date) AS month
FROM shows
GROUP BY month
ORDER BY month;
```

	shows	month
1	2	1
2	1	2
3	2	3
4	7	4
5	2	8
6	2	10
7	2	11
8	2	12

Пояснити дію розділу GROUP BY на прикладах із використанням та без використання агрегатних функцій.

Оператор GROUP BY групує значення за певними полями.

Приклад: розподіл по заробітним платам:

```
SELECT salary, COUNT(Actors_id)
FROM contracts
GROUP BY salary
ORDER BY salary DESC;
```

	salary	COUNT(Actors_id)
1	12000	1
2	8000	2
3	6000	5
4	5500	1
5	5000	4
6	4000	3
7	3000	2
8	2000	2
9	1000	3
10	0	2

При цьому, є деякі обмеження. Наприклад: в операторі SELECT не можна використовувати колонки, які не вказані в GROUP BY (якщо можливе більше ніж одне значення) або над якими не проводяться обрахунки за допомогою агрегатних функцій:

```
3 SELECT salary, Actors_id, COUNT(Actors_id)
4 FROM contracts
5 GROUP BY salary
6 ORDER BY salary DESC;
```

[42000][1055] Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'theateractors.contracts.actors_id' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by

Створити запит з використанням розділу HAVING для задання умов, що містять агрегатні функції та застосовуються після групування даних.

Знайти середню кількість зіграних спектаклів відповідно досвіду. Вивести лише дані де отримане число більше 0:

```
SELECT years_of_experience AS years,
       COUNT(last_name) AS people,
       AVG(spectacle_counter) as average
FROM actors
GROUP BY years
HAVING average > 0
ORDER BY years DESC;
```

	years	people	average
1	0	7	1.5714

Варто зазначити, що в HAVING можуть використовуватись лише агрегатні функції, або ті колонки, за якими йде групування.

Пояснити відмінність між розділами HAVING та WHERE у випадках коли проводимо та коли не проводимо групування даних.

Оператор HAVING може використовуватись лише разом з GROUP BY, і має обмеження, які зазначені вище. Також, він використовується для того, щоб відібрати значення за умовою – але вже після групування (тобто, застосовується до «групових» значень).

Where – навпаки, застосовується лише до «одиничних» значень. В нього є обмеження – не можна застосовувати агрегатні функції:

```
1  ✓ USE theateractors;
2
3  ! SELECT last_name, years_of_experience
4     FROM actors
5     WHERE years_of_experience < AVG(years_of_experience);
```

[HY000][1111] Invalid use of group function

Але, можна використовувати підзапити:

```
1  USE theateractors;
2
3  ✓ SELECT last_name, years_of_experience
4     FROM actors
5     WHERE years_of_experience < (SELECT AVG(years_of_experience) FROM actors)
6     ORDER BY years_of_experience DESC;
```

Output theateractors.actors

	last_name	years_of_experience
1	CHASE	3
2	SWANK	3
3	WOOD	3
4	GUINNESS	2
5	WAHLBERG	2

Порядок виконання команд:

ORDER	CLAUSE	FUNCTION
1	from	Choose and join tables to get base data.
2	where	Filters the base data.
3	group by	Aggregates the base data.
4	having	Filters the aggregated data.
5	select	Returns the final data.
6	order by	Sorts the final data.
7	limit	Limits the returned data to a row count.

Пояснити відмінність між розділами GROUP BY та ORDER BY, GROUP BY та DISTINCT.

GROUP BY – групує дані, а ORDER BY – їх сортує.
ORDER BY:

```
SELECT date, budget
FROM shows
ORDER BY date;
```

	date	budget
1	2023-04-14 00:00:00	300000
2	2023-04-14 00:00:00	300000
3	2023-04-14 00:00:00	300000
4	2023-04-14 00:00:00	300000
5	2023-04-14 00:00:00	300000
6	2023-04-20 00:00:00	241150
7	2023-08-17 00:00:00	197410

GROUP BY:

```
SELECT YEAR(date), SUM(budget)
FROM shows
GROUP BY YEAR(date);
```

	`YEAR(date)`	`SUM(budget)`
1	2023	283230845000000
2	2024	121441510000000

DISTINCT – оператор, за допомогою якого обираються унікальні значення (видаляються дублікати):

```
SELECT DISTINCT Actors_id
FROM additionaltable
ORDER BY 1;
```

	Actors_id
1	1
2	2
3	3
4	4
5	5

Предметна область (22 варіант):

Зайнятість акторів театру. Комерційний директор театру організує залучення акторів та укладання контрактів. Щороку театр здійснює постановку різних вистав. Кожна вистава має певний бюджет. Для участі у конкретних постановках у певних ролях залучаються актори. З кожним із акторів укладається персональний контракт на певну суму. Кожен актор має стаж роботи, деякі з них удостоєні різних звань. В рамках одного спектаклю на ту саму роль залучається кілька акторів. Договір визначає базову зарплату актора, а за підсумками реально відіграних вистав актору призначається премія. У базі даних слід зберігати інформацію протягом кількох років.

Діаграма прецедентів:

