

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №3
із дисципліни
Бази даних

Виконав:
Ст. групи КН-207
Гуменчук Б.Р.
Прийняв:
Мельникова Н.І.

Львів – 2018 р.

Мета роботи: Розробити SQL запити для вибору записів з однієї чи кількох таблиць із застосуванням агрегатних функцій для отримання підсумкових значень полів.

Короткі теоретичні відомості.

Для побудови аналітичних та підсумкових запитів на SQL використовують директиву GROUP BY, а також агрегатні функції. Основні агрегатні функції подані в таблиці. Аргументами функцій можуть бути як задані множини значень, так і результати підзапиту.

Функція (оператор)	Опис
MAX(), MIN()	Знаходить максимальне, або мінімальне значення для заданих аргументів.
AVG()	Знаходить середнє значення для заданих аргументів.
AVG(DISTINCT ...)	Знаходить середнє значення не враховуючи повтори.
SUM()	Обчислює суму значень.
SUM(DISTINCT ...)	Обчислює суму різних значень.
COUNT()	Рахує кількість рядків, які повертає запит.
COUNT(DISTINCT ...)	Рахує кількість різних значень.
BIT_AND(), BIT_OR()	Повертає побітове "і", "або" для аргументів.
STD(), STDDEV_POP()	Обчислює значення стандартного відхилення для аргументів.
VAR_POP()	Обчислює значення дисперсії для аргументів.

Для застосування агрегатних функцій SUM або AVG з часовими типами даних потрібно проводити двосторонню конвертацію типів за допомогою спеціальних функцій, наведених нижче.

TO_DAYS () – перевести дату у число, що означає кількість днів починаючи з 0-го року.

FROM_DAYS () – перевести кількість днів у дату.

TIME_TO_SEC () – перевести значення часу у кількість секунд.

SEC_TO_TIME () – перевести кількість секунд у час.

Наприклад,

```
SELECT FROM_DAYS (SUM (TO_DAYS (дата) )) FROM таблиця;
```

Хід роботи.

1. Визначити кількість замовлень у кожного робітника:

```
171 • select performer.name as Pname, count(book.performer_idperformer) as Performers
172 from performer inner join book on book.performer_idperformer = performer.idperformer
173 group by name;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Pname	Performers		
▶	Vasiliy	3		
	Kolya	2		




2. Визначимо сумарну кількість копій замовлення.

```
173 • select performer.name as Pname, sum(book.amount) as Numbook
174 from performer inner join book
175 on book.performer_idperformer = performer.idperformer
176 group by performer.name with rollup;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Pname	Numbook		
▶	Kolya	20		
	Vasiliy	30		
	NULL	50		

3. Визначити середню кількість копій яку має зробити виконавець не враховуючи повторів.

```
3 • select office.Address as Office, avg(distinct Appointment.Salary) as Salaries
4   from (office inner join worker) inner join Appointment
5   on worker.office_idoffice = office.idOffice
6   and worker.appointment_idappointment = Appointment.idAppointment
7   group by Address;
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	Office	Salaries
►	Kniazia Romana 3	125.0000
	Puluj 21	183.3333

Висновок: на цій лабораторній роботі було розглянуто методи застосування агрегатних функцій до результатів вибору даних з таблиць БД.