

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Кафедра СШІ

**ЗВІТ**

До лабораторної роботи № 9

**З дисципліни: “Бази даних ”**

*“Аналітичні та підсумкові запити”*

**Виконала:**

ст. гр. КН-207

Данків Анастасія

**Прийняв:**

викладач

Мельникова Н.І.

Львів – 2019

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

```
mysql> SELECT FROM_DAYS(SUM(TO_DAYS(date))) FROM offer;
+-----+
| FROM_DAYS(SUM(TO_DAYS(date))) |
+-----+
| 6058-10-11                     |
+-----+
1 row in set (0.17 sec)

mysql>
```

```
mysql> select TIME_TO_SEC(cooking_time) from Drinks_and_dishes;
+-----+
| TIME_TO_SEC(cooking_time) |
+-----+
| 300                        |
| 300                        |
| 180                        |
| 900                        |
+-----+
4 rows in set (0.03 sec)
```

```
mysql> SELECT cooking_time, (MIN(DRINKS_AND_DISHES.COOKING_TIME)) AS BEST_TIME
-> FROM DRINKS_AND_DISHES;
+-----+
| cooking_time | BEST_TIME |
+-----+
| 00:05:00     | 00:03:00  |
+-----+
1 row in set (0.12 sec)

mysql>
```

Визначимо найкращого працівника, який має найбільше замовлення, замовлення рахуються за вересень

```
mysql> SELECT STAFF.NAME AS BEST WORKER, (COUNT(DISTINCT OFFER_DISHES.COUNT)) AS RATING
-> FROM (STAFF INNER JOIN OFFER) INNER JOIN OFFER_DISHES
-> ON STAFF.ID_STAFF = OFFER.STAFF_ID_STAFF
-> WHERE MONTH(OFFER.DATE) = (MONTH(CURRENT_DATE)+4);
+-----+
| BEST_WORKER | RATING |
+-----+
| Edik        | 3      |
+-----+
1 row in set (0.04 sec)
```

Визначимо середній розмір страви в меню

```
mysql> select drinks_and_dishes.name as name_words , AVG (drinks_and_dishes.size) as middlesize
-> from offer_dishes inner join drinks_and_dishes
-> ON offer_dishes.drinks_and_dishes_id = drinks_and_dishes.id
-> group by drinks_and_dishes.name;
+-----+
| name_words      | middlesize |
+-----+
| Coffee          | 250.0000   |
| Coffee_with_milk | 250.0000   |
| Milk            | 200.0000   |
+-----+
3 rows in set (0.08 sec)
```

```

from offer group by date, time WITH ROLLUP at line 2
mysql> select DATE (date) ,TIME(time),
-> COUNT(DISTINCT(id_offer)) AS count_offers
-> from offer group by date, time WITH ROLLUP;
+-----+-----+-----+
| DATE (date) | TIME(time) | count_offers |
+-----+-----+-----+
| 2019-04-25 | 10:30:03 | 1 |
| 2019-04-25 | NULL | 1 |
| 2019-09-20 | 13:50:00 | 1 |
| 2019-09-20 | NULL | 1 |
| 2019-09-30 | 13:40:00 | 1 |
| 2019-09-30 | NULL | 1 |
| NULL | NULL | 3 |
+-----+-----+-----+
7 rows in set (0.06 sec)

```

```

mysql> select staff.name, (sum(drinks_and_dishes.price)* offer_dishes.count) as salary, MONTHNAME(offer
.date) as month
-> FROM ((staff INNER JOIN offer)INNER JOIN offer_dishes)INNER JOIN drinks_and_dishes
-> ON staff.id_staff = offer.staff_id_staff
-> AND offer.id_offer = offer_dishes.Offer_id_offer
-> AND offer_dishes.Drinks_and_dishes_id = drinks_and_dishes.id
-> GROUP BY staff.name;
+-----+-----+-----+
| name | salary | month |
+-----+-----+-----+
| Anastasia | 62 | April |
| Edik | 105 | September |
| Sara | 500 | September |
| Mark | 60 | May |
| Vitalik | 45 | May |
| Eva | 300 | April |
| Den | 30 | April |
| Kate | 60 | April |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>

```

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