# **Тестове завдання: SQL (MySQL 8+)**

Використовуйте надану тестову базу даних (структура й сид у файлі **recruiting\_crm\_schema.sql** [**https://drive.google.com/file/d/1CeUrJhRBICVt-zNWLkPpOPaDVVbL9Dbm/view?usp=sharing**](https://drive.google.com/file/d/1CeUrJhRBICVt-zNWLkPpOPaDVVbL9Dbm/view?usp=sharing)).  
 Мета — перевірити вміння працювати з реалістичною схемою рекрутинг-системи: відбори з умовами, анти-джойни, агрегати, оновлення через підзапити.

**Фіксація поточного користувача:** у всіх місцях, де згадується «поточний HR», вважати, що це **Alice Recruiter** з hr\_id = 1 (див. aspnetusers).

**Формат здачі:** кожне завдання — окремий .sql-файл (task1.sql … task5.sql). Угорі кожного файлу короткий коментар, що робить запит.

## **Task 1. Ліди без подальших дій**

Повернути список кандидатів з early\_statuses, для яких одночасно виконується:

1. Є хоча б один статус із type\_id = 1 (“Lead”) у вказаному діапазоні дат (creation\_date).
2. **Немає** жодного пізнішого статусу по **тому ж кандидату і тій самій вакансії**.
3. **Немає** жодного резюме з відправкою (sent\_at IS NOT NULL) по тій самій парі кандидат–вакансія.
4. І кандидат, і вакансія **доступні поточному HR** (Alice, hr\_id = 1) згідно з таблицею access (entity\_type IN ('candidate','vacancy'), right\_code='Read').

Вивести колонки:  
 candidate\_id, full\_name, linkedin\_url, vacancy\_id, vacancy\_title, creation\_date, comment\_text, is\_friend, is\_pro.  
 Сортувати за creation\_date ↑.

## 

## **Task 2. Місячна статистика по вакансіях**

Сформувати зведення за один календарний місяць (наприклад, березень 2025; інтервал [2025-03-01 00:00:00, 2025-04-01 00:00:00)):

* total\_candidates — кількість **унікальних** кандидатів з типами подій у (1, 3, 4, 10, 11, 12, 14, 19).
* resumes\_sent — кількість резюме з sent\_at IS NOT NULL.
* contracts — кількість статусів із type\_id = 10.
* rejections — кількість статусів із type\_id = 11.
* calls — кількість статусів із type\_id = 2.
* interviews — кількість статусів із type\_id IN (12, 14).

Вивести: vacancy\_id, vacancy\_title, month, total\_candidates, resumes\_sent, contracts, rejections, calls, interviews.  
 Сортувати за vacancy\_id.

## 

## **Task 3. Щоденний KPI по HR**

Створити у себе тимчасову таблицю kpi\_table зі схемою:

day\_date DATE, hr\_id INT,

leads\_created INT, statuses\_added INT, resumes\_prepared INT,

resumes\_sent INT, calls\_made INT, contracts\_signed INT

Вставити один рядок на кожного HR із aspnetusers, підрахувавши метрики **за вчорашній день** (інтервал [учора 00:00:00, сьогодні 00:00:00), UTC):

* leads\_created — кількість подій із type\_id = 1.
* statuses\_added — кількість подій з **усіма типами, крім** type\_id = 1.
* resumes\_prepared — кількість подій із type\_id = 3.
* resumes\_sent — кількість записів у resumes з sent\_at IS NOT NULL.
* calls\_made — type\_id = 2.
* contracts\_signed — type\_id = 10.

## **Task 4. Оновлення лічильників навичок**

Оновити skill\_variants.cnt, щоб значення дорівнювало кількості **унікальних** кандидатів із цією навичкою (зв’язки — у candidate\_skills).  
 Форма рішення — через UPDATE ... JOIN (SELECT ... GROUP BY).

## 

## **Task 5. Нагадування на сьогодні**

Повернути всі нагадування з reminders **на сьогоднішню дату** (інтервал [сьогодні 00:00:00, завтра 00:00:00)) для **поточного HR** (Alice, hr\_id = 1), причому сама подія нагадування також повинна бути доступна Alice у access (entity\_type='reminder', right\_code='Read').

Вивести: reminder\_id, remdate, candidate\_id, full\_name, note.  
 Сортувати за remdate.

## **Вимоги до оформлення**

1. Усі фільтри по датах задавайте як інтервал **[start, end)** — верхня межа виключається.
2. Обробляйте NULL через COALESCE/IFNULL, щоб уникнути пропусків у підрахунках.
3. Не дублюйте однакову логіку у кількох підзапитах — за можливості використовуйте JOIN або CTE.
4. Використовуйте точні назви таблиць і колонок із наданої recruiting\_crm\_schema-схеми.