

Створюєте в себе локальну БД і виконуєте задачі.

Завдання 1

CREATE таблиці в БД

CREATE TABLE profile (

id int not null. – унікальний ідентифікатор профіля

name varchar(255) null. – ім'я

lastname varchar(255) null. – прізвище

profession_id int null. – ідентифікатор вказаної професії пошукачем при реєстрації

phone varchar(255) null. – телефон

email varchar(255) null. – електронна пошта

create_date datetime null – дата створення

);

```
INSERT into profile VALUES (1, 'Ivan', 'Ivanov', 1, '0979999999', 'ivanov@email.ua', '2021-07-16 13:35:14.483'),
(2, 'Mykhailo', 'Mykhailov', 2, '0978888888', 'mykhailo@email.ua', '2021-07-12 10:35:15.413'),
(3, 'Andrew', 'Kryvohliad', 3, '0978788488', 'kryvohliad@email.ua', '2021-07-11 11:33:11.182'),
(4, 'Volodymyr', 'Miniailo', 2, '0978485848', 'miniailo@email.ua', '2021-05-11 09:22:01.383'),
(5, 'Sergii', 'Triasylo', 4, '0978998088', 'triasylo@email.ua', '2021-03-16 15:18:05.593'),
(6, 'Dmytro', 'Priadun', 5, '09784888228', 'priadun@email.ua', '2021-02-16 19:15:59.583'),
(7, 'Valentina', 'Melnik', 3, '0978778888', 'melnik@email.ua', '2021-03-16 14:49:11.432'),
(8, 'Olga', 'Palii', 4, '0978088088', 'palii@email.ua', '2021-05-16 21:32:18.113'),
(9, 'Catherine', 'Maliar', 1, '0978888888', 'maliar@email.ua', '2021-06-16 22:55:19.411'),
(10, 'Elena', 'Ruban', 5, '0978248858', 'ruban@email.ua', '2021-03-16 00:31:45.123'),
(11, 'Julia', 'Tkach', 5, '0978568788', 'tkach@email.ua', '2021-05-16 03:05:03.427');
```

CREATE table profession (

id int not null. – унікальний ідентифікатор професії

profession_name varchar(255) null. – назва професії

description varchar(255) null – опис

);

```
INSERT into profession VALUES (1, 'Sql developer', null), (2, 'Driver category D', null),
(3, 'Manager', null), (4, 'Junior ASP.NET developer', null), (5, 'Bukhhalter', null);
```

CREATE table vacancy (

id int not null. – унікальний ідентифікатор вакансії

vacancy_name varchar(255) null. – назва розміщеної вакансії

description varchar(255) null. – опис вакансії

profession_id int null. – ідентифікатор професії вказаної при розміщенні вакансії

employer_id int not null – ідентифікатор роботодавця який розмістив вакансію на сайті

);

INSERT into vacancy (id, vacancy_name, profession_id, employer_id) VALUES (1, 'Developer', 1, 1), (2, 'Developer', 4, 7), (3, 'Manager', 3, 2), (4, 'Driver', 2, 3), (5, 'Driver', 2, 6), (6, 'Manager', 3, 4), (7, 'Developer', 4, 7), (7, 'Developer', 4, 1), (9, 'Developer', 1, 5), (10, 'Manager', 3, 6), (11, 'Bukhhalter', 5, 1), (12, 'Developer', 1, 7), (13, 'Bukhhalter', 5, 1), (14, 'Bukhhalter', 5, 7), (15, 'Driver', 2, 1);

CREATE table employer (

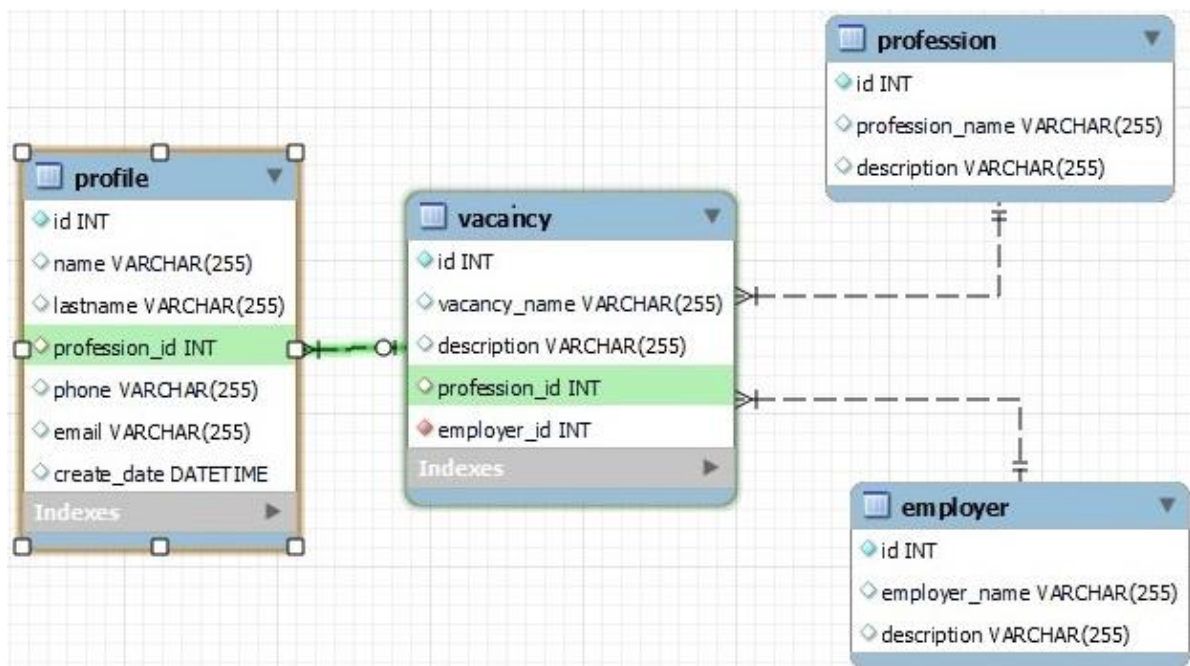
id int not null. – унікальний ідентифікатор роботодавця

employer_name varchar(255) null. – назва компанії роботодавця

description varchar(255) null – опис

);

INSERT into employer VALUES (1, 'SoftDev', null), (2, 'InmicroMin', null), (3, 'BuhOblik', null), (4, 'TOVup', null), (5, 'DreamsWork', null), (6, 'CatDog', null), (7, 'SoftBest', null), (8, 'PaperCompany', null);



1. Отримати всі профіля з назвою професії.

```
SELECT p.id, CONCAT(p.name, ' ', p.lastname) AS user,  
p.profession_id, pr.profession_name  
FROM profile p  
LEFT JOIN profession pr ON p.profession_id = pr.id;
```

2. Написати запит який відобразить назви всіх вакансії кожної з компаній.

```
SELECT e.employer_name as JobGiver,  
GROUP_CONCAT(v.vacancy_name  
ORDER BY e.employer_name SEPARATOR '; ') AS Vacancies  
FROM vacancy v  
LEFT JOIN employer e ON v.employer_id = e.id  
GROUP BY e.employer_name  
ORDER BY e.employer_name;
```

```
SELECT e.employer_name as Employer, v.vacancy_name as Vacancy  
FROM vacancy v  
LEFT JOIN employer e ON v.employer_id = e.id  
ORDER BY e.employer_name;
```

3. Визначити кількість профілів по кожній професії.

```
SELECT pr.profession_name as Vacancy, COUNT(p.id) AS Applicants  
FROM profile p  
LEFT JOIN profession pr ON p.profession_id = pr.id  
GROUP BY pr.profession_name  
ORDER BY pr.profession_name;
```

4. Визначити кількість вакансій опублікованих кожною компанією.

```
SELECT e.employer_name AS Company, COUNT(v.id) AS Total_vacancies  
FROM employer e  
LEFT JOIN vacancy v ON e.id = v.employer_id  
GROUP BY e.employer_name  
ORDER BY Total_vacancies DESC;
```

PaperCompany має null! Перевірів:

```
SELECT e.employer_name, v.vacancy_name  
FROM employer e  
LEFT JOIN vacancy v on v.employer_id = e.id  
WHERE e.employer_name = 'PaperCompany';
```

5. Визначити компанію, яка не опублікувала жодної вакансії.

PaperCompany! Їм не треба люди. В них все роботизовано, цехи.

```
SELECT e.employer_name AS Employer  
FROM employer e  
LEFT JOIN vacancy v ON e.id = v.employer_id  
WHERE v.id IS NULL;
```

6. Відобразити усі унікальні вакансії (бажано 2-ма способами).

```
SELECT DISTINCT vacancy_name as Unique_Vac  
FROM vacancy  
ORDER BY Unique_Vac;
```

```
SELECT vacancy_name as Vacancy  
FROM vacancy  
GROUP BY Vacancy  
ORDER BY Vacancy;
```

7. Вивести з таблиці профілів максимальний id та дату його створення.

```
SELECT id AS MaxID, create_date AS Date  
FROM profile  
ORDER BY id DESC  
LIMIT 1;
```

8. Вивести всі компанії в яких більше 3 розміщених вакансій.

```
SELECT e.employer_name as Company,  
GROUP_CONCAT(v.vacancy_name  
ORDER BY e.employer_name SEPARATOR ', ') as Vacancies,  
COUNT(v.id) AS VacancyCount
```

```

FROM employer e
INNER JOIN vacancy v ON e.id = v.employer_id
GROUP BY e.employer_name
HAVING VacancyCount > 3
ORDER BY VacancyCount DESC;

```

*# До речі, повторюються деякі вакансії кілька разів. Так треба, така задумка.
Якщо COUNT(DISTINCT v.vacancy_name) беремо, то нічого не поверне*

9. Визначити якій компанії теоретично підходить профіль пошукача

```

SELECT e.employer_name as Company,
GROUP_CONCAT(
DISTINCT CONCAT(pr.name, ' ', pr.lastname)
ORDER BY pr.name ASC SEPARATOR ' ' ) as SuitableProfiles,
GROUP_CONCAT(DISTINCT pr.id ORDER BY
pr.name ASC SEPARATOR ' ' ) as ProfileIDs
FROM employer e
INNER JOIN vacancy v ON e.id = v.employer_id
INNER JOIN profession p ON v.profession_id = p.id
INNER JOIN profile pr ON p.id = pr.profession_id
GROUP BY e.employer_name
HAVING COUNT(v.id) > 0
ORDER BY Company;

```

(option) Підставляємо id користувача, і вибиває компанії, які підходять йому/їй

```

SELECT e.employer_name as Company,
p.profession_name as MatchingProfession,
COUNT(v.id) as MatchingVacancies
FROM employer e
INNER JOIN vacancy v ON e.id = v.employer_id
INNER JOIN profession p ON v.profession_id = p.id
INNER JOIN profile pr ON pr.profession_id = p.id
WHERE pr.id = 11 # ВСТАВЛЯЄМО id користувача

```

```
GROUP BY e.employer_name, p.profession_name
HAVING COUNT(v.id) > 0
ORDER BY Company, MatchingVacancies DESC;
```

Завдання 2

Аналіз статистики інтернет магазину

На кожній сторінці товару є можливість зберегти товар до закладок за доп. відповідної кнопки.

До відома: конверсія сайту - це виконання відвідувачем цільової дії, під якою найчастіше ми маємо на увазі покупку товару/послуги. Чим вище конверсія, тим більше продажів.

Формула: $(Q \text{ відвідувачів, які виконали якусь дію} / \text{загальна } Q \text{ відвідувачів} * 100\%)$

Приклад результату:

Bookmarks_Count	PageViews_Count	Conversion_Rate
10	100	0.10

Середнім показником хорошої конверсії є хоча б 1% покупців від загальних відвідувачів сайту. Якщо розрахунок йде за кількістю дзвінків - приблизно 3%. Але це усереднені показники. Конверсія - індивід. значення. Не варто порівнювати свої показники з іншими сайтами і засмучуватися.

CREATE таблиці в БД

CREATE table product_page_view (-- подія «показ сторінки товару» починаючи з 01 грудня 2019

id bigint primary key,

id_product bigint. – ід товару

dt datetime.

dti int.

id_user_account int – ід акаунта користувача, якщо він залогінився

);

CREATE table product_page_action (-- дії ... починаючи з 01 грудня 2019

id bigint primary key,

id_page_view bigint, -- ід показу сторінки товару, на якому було виконано дію

type int, -- тип події (значення 2 – це те значення, яке нас цікавить - «додавання товару до закладок»)

dt datetime,

dti int

);

CREATE table transaction (-- історія покупок користувачів починаючи з 01 грудня 2019

id bigint primary key.

id_user_account int. -- id аккаунта користувача,

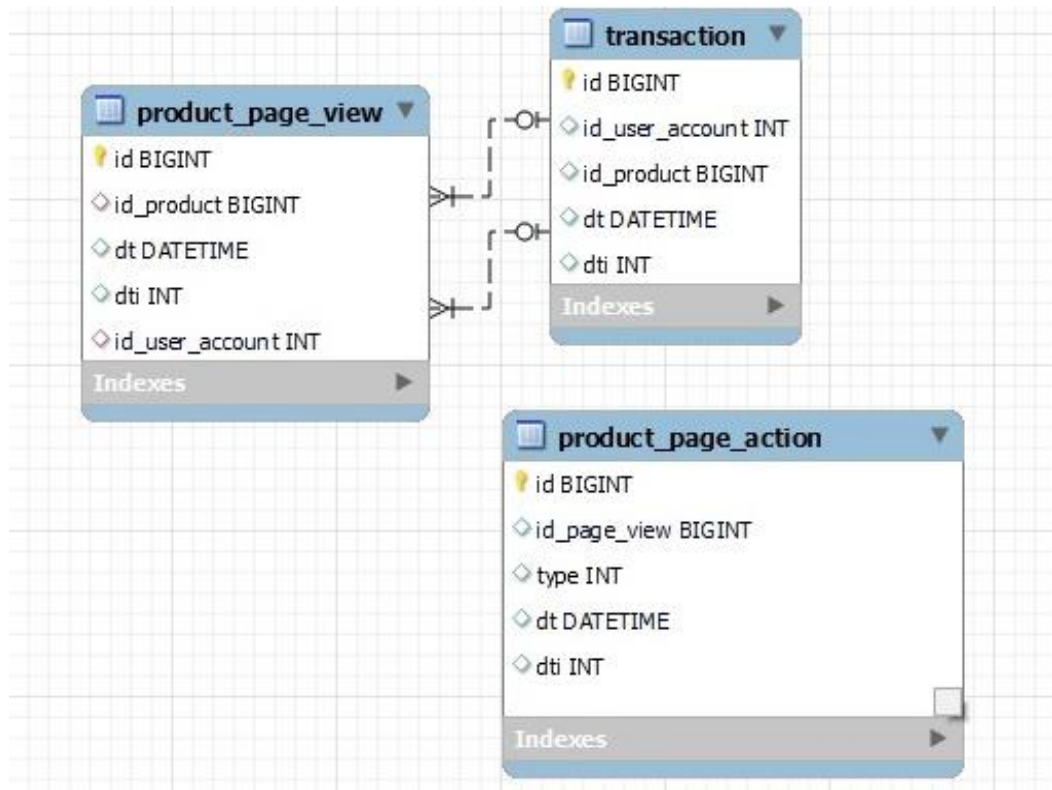
-- not null: купівлю у нас може виконувати тільки зареєстрований користувач

id_product bigint.

dt datetime.

dti int

);



ERD_db_task2

Завдання! Напишіть SQL запити, які розраховують:

1. конверсію цієї функції - кількість додавань в закладки по відношенню до кількості переглядів сторінки товару

SELECT COUNT(DISTINCT act.id) AS Bookmarks_Count, -- кількість додавань в закладки

COUNT(DISTINCT viw.id) AS PageViews_Count, -- кількість переглядів сторінок товару

COUNT(DISTINCT act.id) * 1.0 / COUNT(DISTINCT viw.id) AS CONVERSION_R

FROM product_page_action act

LEFT JOIN product_page_view viw ON act.id_page_view = viw.id

WHERE act.type = 2 -- 2 = дія "додавання товару до закладок"

AND act.dti >= 20191201 -- Фільтр за датою з таблиці p_r_action

AND viw.dti >= 20191201; -- Фільтр за датою перегляду сторінки товару

середньострокова конверсія – змінити дати, наприклад 3-6 місяців:

... AND act.dti BETWEEN 20191201 AND 20200501; як раз моє день народження.

короткострокова - 20191201 AND 20200101, відповідно

2. ту саму конверсію, але тільки для тих показів сторінки товару, коли користувач - зареєстрований, і є інформація про те, що користувач раніше виконував хоча б одну купівлю

SELECT COUNT(DISTINCT act.id) AS Bookmarks_Count, -- кількість додавань в закладки

COUNT(DISTINCT viw.id) AS PageViews_Count, -- кількість переглядів сторінок товару

FORMAT((COUNT(DISTINCT act.id) * 1.0 / COUNT(DISTINCT viw.id)) * 100, 2) AS

Conversion_Rate -- конверсія у відсотках з 2 decimalami після крапки

FROM product_page_action act

LEFT JOIN product_page_view viw ON act.id_page_view = viw.id LEFT JOIN

transaction trans ON viw.id_user_account = trans.id_user_account

WHERE act.type = 2 -- 2 equals "додавання товару до закладок"

AND act.dti >= 20191201 -- Фільтр за датою з таблиці product_page_action

AND viw.dti >= 20191201 -- Фільтр за датою перегляду сторінки товару

AND trans.id IS NOT NULL; -- Фільтр для наявності історії покупок

The screenshot shows a database management tool interface. On the left, a tree view displays the database structure with tables like 'imdb_data_final', 'product_page_action', 'product_page_view', and 'transaction'. The main area shows a SQL query with line numbers 10 to 15. The query is a SELECT statement with three columns: Bookmarks_Count, PageViews_Count, and Conversion_Rate. The results are displayed in a table at the bottom with one row of data: 6, 6, and 100.00. The Conversion_Rate is highlighted with a red underline.

Bookmarks_Count	PageViews_Count	Conversion_Rate
6	6	100.00

Powerful_Conversion_Rate (100%)

My business is absolutely gorgeous