Каркушевський Владислав Леонідович

№ 16 Облік телефонних переговорів

Лабораторна робота №5

Використання виразу CASE у вибірках даних. Оператор UNION.

|  |  |
| --- | --- |
| Завдання 1. А) написати пошуковий запит SELECT з використанням першої форми виразу CASE | |
| Скрипт | Скрін |
| # 1 форма case  SELECT  ID\_call,  ID\_abonent,  Duration\_call,  CASE  WHEN Duration\_call >= 20 THEN 'large call'  WHEN Duration\_call >= 10 THEN 'middle call'  ELSE 'small call'  END Rating\_call  FROM  Calls  LIMIT 15; |  |
| Завдання 1. Б) написати пошуковий запит SELECT з використанням другої форми виразу CASE | |
| Скрипт | Скрін |
| # 2 форма case  SELECT  ID\_call,  ID\_abonent,  Duration\_call,  CASE Day\_or\_night  WHEN 'D' THEN 'Day'  ELSE 'Night'  END Day\_or\_night  FROM  calls  LIMIT 10; |  |
| Завдання 2 А) написати запит з використанням інструкції IF | |
| Скрипт | Результат |
| SELECT IF(2 < 1, 'YES', 'NO'); |  |
| Завдання 2 Б) написати запит з використанням кількох вкладених інструкцій IF | |
| Скрипт | Результат |
| SELECT IF(2 = 3, '1111', IF(4 = 3, '2222', IF(3 = 3, '3333', '4444'))); |  |
| Завдання 3. А) написати запит SELECT з використанням CASE або IIF в різних розділах пошукового запиту (IN) | |
| Скрипт | Результат |
| #in  SELECT  ID\_abonent, Name\_abonent, Card\_number  FROM  abonents  WHERE  ID\_abonent IN (IF(Card\_number LIKE '%17%',  ID\_abonent,  NULL)); |  |
| Завдання 3. Б) написати запит SELECT з використанням CASE або IIF в різних розділах пошукового запиту (WHERE) | |
| Скрипт | Результат |
| # where  SELECT  ID\_call, Cost  FROM  calls  WHERE  CASE  WHEN cost >= 10 THEN 1  ELSE 0  END = 1; |  |
| Завдання 3. В) написати запит SELECT з використанням CASE або IIF в різних розділах пошукового запиту (ORDER BY) | |
| Скрипт | Результат |
| # ORDER BY  SELECT  ID\_call, ID\_abonent, Duration\_call, Cost  FROM  calls  ORDER BY CASE  WHEN Duration\_call >= 15 THEN 1  ELSE 0  END , ID\_call  ; |  |
| Завдання 3. Г) написати запит SELECT з використанням CASE або IIF в різних розділах пошукового запиту (HAVING) | |
| Скрипт | Результат |
| SELECT  ID\_abonent,  COUNT(\*) AS 'count',  AVG(Duration\_call) AS 'average\_number\_of\_calls',  ROUND(SUM(Cost), 4) AS 'total\_cost'  FROM  calls  GROUP BY 1  HAVING count >= 2  AND IF(total\_cost > 20, 1, 0) = 1  ORDER BY 2 DESC; |  |
| Завдання 4. А) навести приклади використання CASE або IF в інших запитах (UPDATE) | |
| Скрипт | Пояснення |
| #UPDATE  SELECT \*  FROM calls;  UPDATE calls  SET Cost = IF(Day\_or\_night = 'D', Cost + 1, Cost); | Ціна усіх дзвінків, які були здійснені у світлу пору доби, збільшиться на 1 |
| Завдання 4. Б) навести приклади використання CASE або IF в інших запитах (DELETE) | |
| Скрипт | Пояснення |
| DELETE FROM calls  WHERE IF(Cost < 2, 1, 0) = 1; | Усі дзвінки, у яких ціна менша ніж 2 будуть видалені |
| Завдання 5. написати запит з використанням функції CHOOSE / COALESCE (MS SQL /MySQL); | |
| Скрипт | Результат |
| #COALESCE  SELECT  ID\_abonent, COALESCE(Name\_abonent, 'anonim') AS Result  FROM  abonents; |  |
| Завдання 6. Написати запит з використанням оператора UNION. Пояснити відмінність UNION від JOIN. | |
| Скрипт | Результат |
| #UNION  SELECT  Name\_city  FROM  cities  UNION SELECT  Name\_city  FROM  PolandCities; |  |
| UNION та JOIN - це дві різні операції у SQL, які використовуються для з'єднання даних з різних таблиць або запитів.  JOIN використовується для об'єднання даних з двох або більше таблиць на основі спільного стовпця, щоб повернути результат з обох таблиць. Існують різні типи JOIN, такі як INNER JOIN, LEFT JOIN, RIGHT JOIN та FULL OUTER JOIN, які дозволяють виконувати різні операції з даними в таблицях.  UNION використовується для об'єднання даних з двох або більше запитів, що повертають різні набори стовпців, але мають однакову структуру даних. UNION дозволяє об'єднати результати запитів у один результат.  Обидві операції дозволяють збирати дані з різних джерел та робити складні запити до бази даних. UNION використовується, коли потрібно об'єднати результати двох або більше запитів, щоб отримати один результат з усіма унікальними рядками, тоді як JOIN використовується, коли потрібно отримати інформацію з двох або більше таблиць, пов'язаних за спільними стовпцями. | |