**При виконанні запитів з кількома таблицями використовуйте операції JOIN, а не підзапити select.**

|  |  |
| --- | --- |
| 1 | Чому цей оператор є помилковим?  USE sample; SELECT project\_name  FROM project  WHERE project\_no =  (SELECT project\_no FROM works\_on WHERE Job = 'Clerk');  Напишіть коректну синтаксичну форму для цього оператора.  #1  SELECT project\_name  FROM project  JOIN works\_on ON project.project\_no = works\_on.project\_no  WHERE job = "clerk"; |
| 2 | Напишіть запит для отримання повних відомостей про працівників (включаючи місце розташування їхніх відділень), які живуть і працюють в одному і тому ж місті.  #2  SELECT DISTINCT emp\_no, emp\_fname, emp\_lname, emp\_live, employee.dept\_no  FROM employee  JOIN department WHERE location = emp\_live; |
| 3 | Напишіть запит для отримання прізвищ всіх працівників, які працюють у відділеннях, що не розміщені у Сіетлі.x  #3  SELECT emp\_lname  FROM employee  JOIN department ON employee.dept\_no = department.dept\_no  WHERE location <> "Seatle"; |
| 4 | Напишіть запит для отримання найбільшого номера працівника.  #4  SELECT MAX(emp\_no)  FROM employee |
| 5 | Напишіть запит для отримання прізвища, імені, місця проживання та місця знаходження відділення для працівників, у яких місце проживання у алфавітному порядку знаходиться перед місцем роботи.  #5  SELECT DISTINCT emp\_lname, emp\_fname, emp\_live, department.location  FROM employee  JOIN department ON department.dept\_no = employee.dept\_no  WHERE emp\_live = '\_%' > location = '\_%'; |
| 6 | Якщо ви збираєтеся з’єднати кілька таблиць у запиті (наприклад, n таблиць), скільки умов з’єднання вам потрібно використати?  кожен join потребує вказання умови,окрім cross join |
| 7 | Напишіть запит для отримання номерів всіх працівників чиї відділення знаходяться у Сіетлі  #7  SELECT emp\_no  FROM employee  JOIN department ON department.dept\_no = employee.dept\_no  WHERE department.location LIKE ('Seattle'); |
| 8 | Яке практичне застосування тимчасових таблиць?  Для збереження результатів виклику процедури, зменшення числа рядків при з'єднаннях |
| 9 | Напишіть запит для отримання номерів та прізвищ всіх працівників, чиї імена містять дві букви ‘t’  #9  SELECT emp\_no, emp\_lname  FROM employee  WHERE emp\_fname LIKE ('\_\_t%'); |
| 10 | Напишіть запит для отримання імен і прізвищ всіх працівників, які приступили до роботи над їхніми проектами 04.01.2007  #10  SELECT emp\_fname, emp\_lname  FROM employee  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  WHERE enter\_date = '2007-01-04'; |
| 11 | Напишіть запит для отримання імен і прізвищ всіх аналітиків, чиї відділення знаходяться у Сіетлі.  #11  SELECT emp\_fname, emp\_lname  FROM works\_on  JOIN employee ON works\_on.emp\_no = employee.emp\_no  JOIN department ON employee.dept\_no = department.dept\_no  AND locotion = ‘Seattle’ AND job = ‘analyst’; |
| 12 | Напишіть запит для отримання номеру відділення для всіх працівників, які приступили до роботи над проектами 15 жовтня 2007 року.  #12  SELECT department.dept\_no  FROM department  JOIN employee ON department.dept\_no = employee.dept\_no  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  WHERE enter\_date = "2007-10-15"; |
| 13 | Напишіть запит для отримання імен і прізвищ всіх працівників, які приступили до роботи над проектом в той же час, що і як мінімум ще один працівник.  #13  SELECT emp\_fname, emp\_lname  FROM employee  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  GROUP BY employee.emp\_no  HAVING COUNT(enter\_date) > 1; |
| 14 | Напишіть запит для отримання номерів працівників, які або працюють у відділі d1, або приступили до роботи над проектом до 1 січня 2007 року. Впорядкуйте результат за зростанням номерів працівників.  #14  SELECT DISTINCT emp.emp\_no  FROM employee emp  JOIN works\_on wor ON emp.emp\_no = wor.emp\_no  WHERE emp.dept\_no LIKE ('d1')  OR wor.enter\_date < '2007-02-1'; |
| 15 | Напишіть запит для отримання номерів працівників, які працюють у відділі d3 і не приступали до роботи над проектом після 1 січня 2008 року.  #15  SELECT DISTINCT emp.emp\_no  FROM employee emp  JOIN works\_on wor ON emp.emp\_no = wor.emp\_no  WHERE emp.dept\_no LIKE ('d3')  AND wor.enter\_date <= '2008-02-1'  GROUP BY emp\_no ASC; |
| 16 | Напишіть запит для отримання імен і прізвищ всіх працівників, які є менеджерами і працюють над проектом Mercury.  #16  SELECT emp\_fname, emp\_lname  FROM employee  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  JOIN project ON project.project\_no = works\_on.project\_no  WHERE works\_on.job LIKE ('manager')  AND project.proj\_name LIKE ('Mercury'); |
| 17 | Напишіть запит для отримання номерів та прізвищ всіх працівників, другий символ прізвища яких містять букву «о» або «а», а закінчується прізвище буквами «es».  #17  SELECT emp\_no, emp\_lname  FROM employee  WHERE (emp\_lname LIKE ('\_a%')  OR emp\_lname LIKE ('\_o%')) AND emp\_lname LIKE ('%es'); |
| 18 | Напишіть запит для отримання номерів працівників і назв їхніх посад (job) для всіх працівників, що працюють над проектом Gemini.  #18  SELECT emp\_no, job  FROM works\_on  JOIN project ON project.project\_no = works\_on.project\_no  WHERE project.proj\_name LIKE ('Gemini'); |
| 19 | Напишіть запит для отримання номерів працівників, які приступили до роботи над проектом не в 2007 р.  #19  #SELECT employee.emp\_no  #FROM employee  #JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  #WHERE enter\_date BETWEEN "2007-01-01" AND "2007-12-31"; |
| 20 | Напишіть запит для отримання імен і прізвищ всіх працівників, які працюють у відділі Research або Accounting.  #20  SELECT emp\_fname, emp\_lname  FROM employee  JOIN department ON department.dept\_no = employee.dept\_no  WHERE department.dept\_name LIKE ('research')  OR department.dept\_name LIKE ('accounting'); |
| 21 | Напишіть запит для отримання назв проектів (з видаленням надлишкових дублікатів), з якими працюють працівники з відділу бухгалтерського обліку.  #21  SELECT DISTINCT project\_name  FROM project  JOIN works\_on ON project.project\_no = works\_on.project\_no  JOIN employee ON works\_on.emp\_no = employee.emp\_no  JOIN department ON employee.dept\_no = department.dept\_no  WHERE dept\_name = "accounting"; |
| 22 | Як оператор GROUP BY опрацьовує значення NULL? Чи відповідає це загальному трактуванню таких значень?  якщо ми групуємо по полю яке містить нуль,то всі такі кортежі попадуть в одну групу,це не зовсім відповідає трактуванню, бо null не дорівнює іншому null, null не дорівнює нічому |
| 23 | В чому різниця між COUNT (\*) і COUNT (стовпець)?  count(\*) поверне кількість всіх записів включно з null,а count(стовпець) не врахує null значення |
| 24 | Переробіть завдання з попереднього заняття, у яких використовувалися підзапити, за допомогою оператора JOIN. (якщо це можливо)  #24  SELECT emp\_fname, emp\_lname #24  FROM employee  JOIN department ON department.dept\_no = employee.dept\_no  WHERE dept\_name LIKE ('research');  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  SELECT \*  FROM employee  JOIN department ON department.dept\_no = employee.dept\_no  WHERE location LIKE ('Dallas'); |
| 25 | Напишіть запит для отримання всіх рядків таблиці works\_on.  #25  SELECT \*  FROM works\_on; |
| 26 | Напишіть запит для отримання повних даних про всі відділення, що розміщені у тому ж місті, що і хоча б одне інше відділення  #26  SELECT d1.\*  FROM department d1  JOIN department d2 ON d1.location = d2.location  GROUP BY d1.dept\_no  HAVING COUNT(\*) > 1; |
| 27 | Напишіть запит для отримання номерів всіх працівників, що працюють у відділенні Marketing. Напишіть два еквівалентні запити використовуючи оператор JOIN та корельований підзапит.  #27(1)  SELECT emp\_no  FROM employee  WHERE 'marketing' IN (SELECT dept\_name  FROM department  WHERE department.dept\_no = employee.dept\_no);  #27(2)  SELECT emp\_no  FROM employee  JOIN department ON department.dept\_no = employee.dept\_no  WHERE department.dept\_name LIKE ('marketing'); |
| 28 | Напишіть запит для отримання детальних даних про всі відділення, а також про місце проживання їхніх працівників для всіх міст, які розташовані там же, де і відділення.  #28  SELECT department.\*, employee.domicile  FROM department  JOIN employee ON employee.dept\_no = department.dept\_no  WHERE department.location = employee.domicile; |
| 29 | Напишіть запит для отримання дати роботи над проектом p2 для всіх  працівників, у яких поки що не визначена посада (job)  #29  SELECT enter\_date  FROM works\_on  WHERE project\_no LIKE ('p2') AND job IS NULL; |
| 30 | Напишіть запит для отримання номерів працівників для всіх клерків.  #30  SELECT employee.emp\_no  FROM employee  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  WHERE job = "clerk"; |
| 31 | Напишіть запит для отримання назв проектів, у яких працюють два або більше клерків.  #31  SELECT proj\_name  FROM project  JOIN works\_on ON project.project\_no = works\_on.project\_no  WHERE works\_on.job LIKE ('clerk')  GROUP BY project.budget  HAVING COUNT(job) > 1; |
| 32 | Знайдіть номери всіх працівників, які є клерками, або працюють у відділенні d3.  #32  SELECT distinct employee.emp\_no  FROM employee  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  WHERE job LIKE ('clerk')  OR dept\_no LIKE ('d3'); |
| 33 | Напишіть запит для отримання номерів працівників, які працюють у відділі d1 і приступили до роботи над проектом до 1 січня 2008 року.  #33  SELECT distinct employee.emp\_no  FROM employee  JOIN works\_on ON employee.emp\_no = works\_on.emp\_no  WHERE dept\_no LIKE ('d1')  AND enter\_date < '2008-02-01'; |
| 34 | Напишіть запит для отримання номерів тих працівників, які живуть у тому ж місті і працюють у тому ж відділенні, що і ще хтось з інших працівників.  #34  SELECT employee1.emp\_no  FROM employee employee1  JOIN employee employee2 ON employee1.dept\_no = employee2.dept\_no  WHERE employee1.domicile= employee2.domicile  AND employee1.emp\_no!= employee2.emp\_no; |
| 35 | У чому різниця між операторами DISTINCT і GROUP BY?  GROUP BY дозволяє використовувати агрегатні функції,а DISTINCT просто видаляэ повторюваны значення. |
| 36 | Напишіть запит для отримання номерів працівників, які обіймають посаду аналітика або менеджера в проекті p1.  #36  SELECT emp\_no  FROM works\_on  WHERE project\_no LIKE ('p1')  AND (job LIKE ('analyst')  OR job LIKE ('manager')); |
| 37 | Згрупуйте всі відділення за їхнім місцерозташуванням  #37  SELECT \*  FROM department  GROUP BY location; |
| 38 | Напишіть запит для отримання для кожного працівника всіх даних про нього, а також дані про відділ, у якому він працює (назву та місце розташування)  #38  SELECT emp.\*, dept.dept\_name, dept.location  FROM employee emp  JOIN department dept ON dept.dept\_no = emp.dept\_no; |
| 39 | Напишіть запит для отримання номерів працівників, що працюють над проектом p2 і мають номер працівника менше 10 000. Напишіть запит за допомогою двох різних, але еквівалентних операторів SELECT.  #39  SELECT emp\_no  FROM works\_on  WHERE project\_no LIKE ('p2')  AND emp\_no < 10000;  Запит нічого не поверне |
| 40 | Напишіть запит для отримання посад (job), які були призначені більше ніж двом працівникам  #40  SELECT job  FROM works\_on  WHERE job IS NOT NULL  GROUP BY job  HAVING COUNT(emp\_no) > 2; |
| 41 | Напишіть запит для отримання дати початку роботи для всіх клерків, які працюють у відділенні d1.  #41  SELECT enter\_date  FROM works\_on  JOIN employee ON employee.emp\_no = works\_on.emp\_no  WHERE job LIKE ('clerk')  AND dept\_no LIKE ('d1'); |