Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського» Інститут прикладного системного аналізу

[Кафедра математичних методів системного аналізу](http://e-u.in.ua/ukr/?page=35)

# ЗВІТ

про виконання комп’ютерного практикуму № 3

з дисципліни «Системи баз даних»

Виконав:

Студент ІІІ курсу Групи КА-65

Іванов Д.С.

Перевірив:

Караюз І. В.

Київ – 2018

# Порядок виконання роботи

Відповідно до обраної теми курсового проекту, реалізувати скрипт, у якому передбачити: демонстрацію роботи кожного з заданих пунктів:

# Робота з кількома таблицями:

1. Представлення, яке розміщує один набір рядків під іншим [1, стор.62].

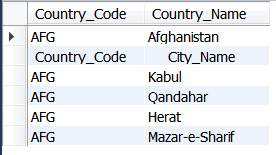
SELECT Code as Country\_Code, Name as Country\_Name FROM country

WHERE Code="AFG" UNION ALL

SELECT ' Country\_Code ', ' City\_Name ' UNION ALL

SELECT CountryCode, Name FROM city

WHERE CountryCode="AFG"



1. Представлення, яке поєднує пов’язані рядки [1, стор.64]

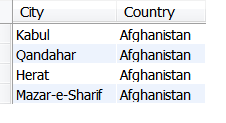
JOIN ~ EQUI-JOIN ~ INNER JOIN

SELECT ct.Name as City, cnt.Name as Country

FROM city ct, country cnt

WHERE ct.CountryCode = cnt.Code

AND cnt.Code = 'AFG';



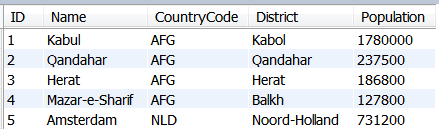
1. Представлення, яке знаходить однакові рядки в двох таблицях[1, стор.66].

SELECT c.ID, c.Name, c.CountryCode, c.District, c.Population FROM city c,VW

WHERE c.Name = VW.Name

AND c.CountryCode = VW.CountryCode

AND c.District = VW.District;



1. Представлення, яке повертає записи із значеннями, яких немає в іншій таблиці [1, стор.68]

ANTI-JOIN ~ LEFT (RIGHT) OUTER JOIN ~ LEFT (RIGHT) JOIN

SELECT CountryCode FROM city WHERE CountryCode NOT IN (SELECT Code FROM country);

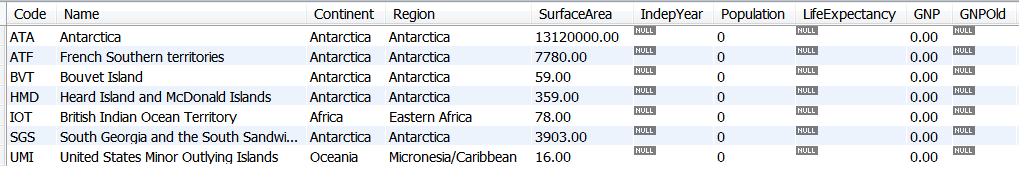
Empty set (0.01 sec)

1. Представлення, яке повертає записи із значеннями, для яких немає відповідності в іншій таблиці [1, стор.72].

SELECT c.\*

FROM country c LEFT JOIN city

ON (c.Code = city.CountryCode) WHERE city.CountryCode IS NULL;



1. Представлення, в якому реалізовано незалежне додавання об’єднань у запит [1, стор.74].

SELECT c.name, city.CountryCode, cl.Language

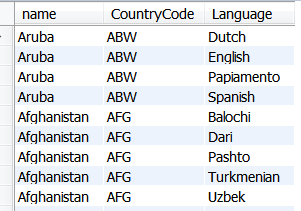
FROM city JOIN country c

ON (city.CountryCode = c.Code)

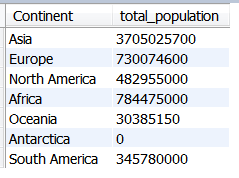
LEFT JOIN countrylanguage cl

ON (cl.CountryCode=city.CountryCode)

order by 2;



1. Представлення, в якому реалізовано об’єднання з використанням агрегованих функцій [1, стор.85].



SELECT Continent, SUM(distinct population) as total\_population FROM (

SELECT c.\* FROM country c

)x GROUP BY Continent;

1. Представлення, в якому реалізовано **зовнішнє** об’єднання з використанням агрегованих функцій [1, стор.90].

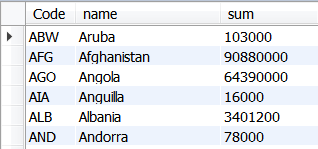
SELECT Code ,Name, SUM(Population) as sum

FROM (

SELECT c.name,c.Code,c.Population

FROM country c LEFT OUTER JOIN city

ON (c.Code=city.CountryCode))x GROUP BY Code;



1. Представлення, в якому відсутні дані в кількох таблицях [1, стор.93]

FULL OUTER JOIN

\*FULL OUTER JOIN не підтримується MySQL 8\*

select c.code, c.name as country, city2.name as city

from country2 c right outer join city2

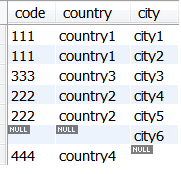
on (c.code = city2.countrycode)

union

select c.code, c.name as country, city2.name as city

from country2 c left outer join city2

on (c.code = city2.countrycode);



1. Представлення, в якому містяться NULL-значення в операціях та порівняннях [1, стор.97].

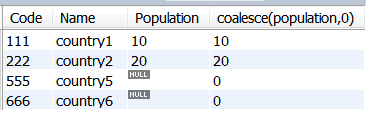
select \*, coalesce(population,0)

from country2

where coalesce(population,0)<(select population

from country2

where name='country3');



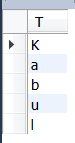
# Робота з масивами символів – рядками:

1. Представлення з проходом рядка [1, стор.135].

SELECT SUBSTR(c.name,iter.pos,1) as T FROM (SELECT name FROM city where

name='Kabul') c,

(SELECT id as pos from t10 ) iter WHERE iter.pos <= LENGTH(c.name);



1. Представлення, в якому виводяться одиночні лапки [1, стор.137].

SELECT 'View with ''kavychki'' ' AS K;



1. Представлення, в якому видаляються всі непотрібні символи [1, стор.139].

SELECT name,

REPLACE( REPLACE(name,'u','a'),'l','k') as result FROM city;



14.(\*) Представлення, в якому розділяються числові та символьні дані [1, стор.141].

SELECT

REPLACE(

REPLACE( REPLACE(f,'t',''),'e',''),'x','')

AS numbers, REPLACE( REPLACE(

f,'1',''),'2','')

AS words FROM txtnum;



15.(\*) Представлення, в якому вибираються ініціали з імені [1, стор.150].



SELECT REPLACE(

REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(

REPLACE(REPLACE(REPLACE('Daniil Sergeevich','D','D.'),'a',''),'n',''),'i',''),'l',''),'S','S

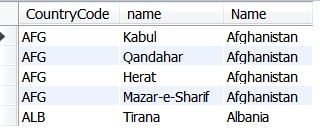
.'),'e',''),'r',''),'g',''),'v',''),'c',''),'h','') AS initials;

16.(\*\*) Представлення, в якому перетворити рядок у список оператора IN із множиною значень [1, стор.168].

SELECT city.CountryCode, city.name, c.Name

FROM city, country c

WHERE city.CountryCode IN ('AFG','ALB') AND c.Code=city.CountryCode;



# Робота з числами:

1. Представлення, в якому вивести мінімальне/максимальне значення у стовпці [1, стор.199].

SELECT MIN(id) AS smallest\_id FROM users;



1. Представлення, яке повертає кількість записів у таблиці [1, стор.203].

SELECT COUNT(\*) AS number\_of\_cities FROM city;



1. Представлення, яке повертає кількість визначених (не-NULL) значень [1,

стор.205].

SELECT COUNT(id) AS number\_of\_cities FROM city;



1. Представлення, в якому обчислити поточну суму [1, стор.205].



SELECT SUM(Population) FROM country;

# Робота з датами:

1. Представлення, в якому обчислити кількість днів між двома датами [1, стор.234].

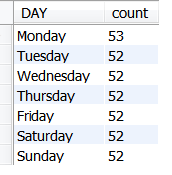
SELECT DATEDIFF(DAY2,DAY1) FROM(

SELECT create\_date as DAY1 FROM customer WHERE customer\_id=1)x,

(SELECT last\_update as DAY2 FROM customer WHERE customer\_id=3)y;



22.(\*) Представлення, в якому обчислити кількість повторів днів тижня протягом року [1, стор.246].



SELECT DATE\_FORMAT( DATE\_ADD( CAST(

CONCAT(YEAR(NOW()),'-01-01') AS DATE),

INTERVAL customer.customer\_id - 1 DAY), '%W') DAY,

COUNT(customer\_id) as count FROM customer

WHERE customer.customer\_id<= DATEDIFF( CAST(

CONCAT(YEAR(NOW())+1,'-01-01') AS date), CAST(

CONCAT(YEAR(NOW()),'-01-01') AS date)) GROUP BY DATE\_FORMAT(

DATE\_ADD(

CAST(

CONCAT(YEAR(NOW()),'-01-01') AS date),

INTERVAL customer.customer\_id - 1 DAY), '%W');

23.(\*) Представлення, в якому визначити, чи рік високосний. [1, стор.265].



SELECT (SELECT AVG(DAY( LAST\_DAY( DATE\_ADD( DATE\_ADD( DATE\_ADD(

NOW(), INTERVAL - DAYOFYEAR(NOW()) DAY), INTERVAL

1 DAY),

INTERVAL 1 MONTH)))

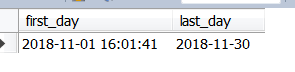
) FROM customer) days\_in\_february;

24.(\*) Представлення, в якому визначити перший та останній день місяця[1,

стор. 277].

SELECT DATE\_ADD(NOW(), INTERVAL -DAY(NOW())+1 day)

first\_day, LAST\_DAY(NOW()) last\_day;



25.(\*\*) Представлення, в якому створити календар [1, стор.295].

select max(case dw when 2 then dm end) as Mo,

max(case dw when 3 then dm end) as Tu,

max(case dw when 4 then dm end) as We,

max(case dw when 5 then dm end) as Th,

max(case dw when 6 then dm end) as Fr,

max(case dw when 7 then dm end) as Sa,

max(case dw when 1 then dm end) as Su

from (

select date\_format(dy,'%u') wk,

date\_format(dy,'%d') dm,

date\_format(dy,'%w')+1 dw

from (

select adddate(x.dy,customer.customer\_id-1) dy,

x.mth

from (

select adddate(current\_date, - dayofmonth(current\_date)+1) dy,

date\_format(

adddate(current\_date,

-dayofmonth(current\_date)+1),

'%m') mth

from customer ) x,

customer

where customer.customer\_id <= 31

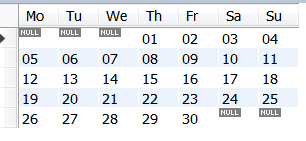
and date\_format(adddate(x.dy,customer.customer\_id-1),'%m') = x.mth

) y

) z

group by wk

order by wk;



26.(\*\*) Представлення, в якому доповнити дати, що відсутні [1, стор.327].

select z.mth, count(e.hiredate) num\_hired from (

select date\_add(min\_hd,interval t500.id1 month) mth from (

select min\_hd, date\_add(max\_hd,interval 11 month) max\_hd

from (

select adddate(min(hiredate),- dayofyear(min(hiredate))+1) min\_hd,

adddate(max(hiredate),-dayofyear(max(hiredate))+1) max\_hd

from emp

) x

) y, customer

where date\_add(min\_hd,interval t500.id-1 month) <= max\_hd

z left join emp e on (z.mth = adddate( date\_add(

last\_day(e.hiredate),interval -1 month),1)) group by z.mth

order by 1;

27.(\*\*) Представлення, в якому виявити накладення діапазонів дат [1, стор.342].

select a.empno,a.ename,

concat('project ',b.proj\_id,' overlaps project ',a.proj\_id) as msg from emp\_project a,

emp\_project b

where a.empno = b.empno

and b.proj\_start >= a.proj\_start and b.proj\_start <= a.proj\_end and a.proj\_id != b.proj\_id;

# Висновки:

Виконавши даний КП я засвоїв те, як в MySQL працює інструмент вибірки а також вбудовані функції для роботи з чисельними, строковими даними та

датами.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Молинаро Э. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ- Плюс, 2009. – 672 с., ил. ISBN-13: 978-5-93286-125-7 ISBN-10: 5-93286- 125-8