Міністерство освіти і науки України Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Інститут прикладного системного аналізу Кафедра математичних методів системного аналізу

ЗВІТ про виконання комп'ютерного практикуму № 3 з дисципліни «Системи баз даних»

Виконав:

Студент III 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].

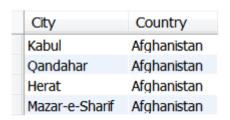
SELECT Code as Country_Code, Name as Country_Name FROM country WHERE Code="AFG" UNION ALL SELECT ' Country_Code ', ' City_Name 'UNION ALL SELECT CountryCode, Name FROM city WHERE CountryCode="AFG"

	Country_Code	Country_Name
•	AFG	Afghanistan
	Country_Code	City_Name
	AFG	Kabul
	AFG	Qandahar
	AFG	Herat
	AFG	Mazar-e-Sharif

2. 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';



3. 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;

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabol	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200

4. 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)

5. 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;

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld
ATA	Antarctica	Antarctica	Antarctica	13120000.00	NULL	0	NULL	0.00	NULL
ATF	French Southern territories	Antarctica	Antarctica	7780.00	HULL	0	NULL	0.00	NULL
BVT	Bouvet Island	Antarctica	Antarctica	59.00	HULL	0	NULL	0.00	NULL
HMD	Heard Island and McDonald Islands	Antarctica	Antarctica	359.00	HULL	0	HULL	0.00	NULL
IOT	British Indian Ocean Territory	Africa	Eastern Africa	78.00	HULL	0	HULL	0.00	NULL
SGS	South Georgia and the South Sandwi	Antarctica	Antarctica	3903.00	HULL	0	HULL	0.00	NULL
UMI	United States Minor Outlying Islands	Oceania	Micronesia/Caribbean	16.00	HULL	0	NULL	0.00	NULL

6. 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;

name	CountryCode	Language
Aruba	ABW	Dutch
Aruba	ABW	English
Aruba	ABW	Papiamento
Aruba	ABW	Spanish
Afghanistan	AFG	Balochi
Afghanistan	AFG	Dari
Afghanistan	AFG	Pashto
Afghanistan	AFG	Turkmenian
Afghanistan	AFG	Uzbek

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

Continent	total_population
Asia	3705025700
Europe	730074600
North America	482955000
Africa	784475000
Oceania	30385150
Antarctica	0
South America	345780000

```
SELECT Continent, SUM(distinct population) as
total_population FROM (
SELECT c.* FROM country c
)x GROUP BY Continent;
```

8. 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;

	Code	name	sum
•	ABW	Aruba	103000
	AFG	Afghanistan	90880000
	AGO	Angola	64390000
	AIA	Anguilla	16000
	ALB	Albania	3401200
	AND	Andorra	78000

9. 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);

code	country	city
111	country1	city1
111	country1	city2
333	country3	city3
222	country2	city4
222	country2	city5
NULL	HULL	city6
444	country4	NULL

10. 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');</pre>

Code	Name	Population	coalesce(population,0)
111	country1	10	10
222	country2		20
555	country5	NULL	0
666	country6	NULL	0

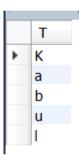
Робота з масивами символів – рядками:

11. 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);



12. Представлення, в якому виводяться одиночні лапки [1, стор.137]. SELECT 'View with "kavychki" ' AS K;



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

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

	name	result
١	Kabul	Kabak

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

SELECT

REPLACE(

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

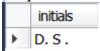
AS numbers, REPLACE(REPLACE(

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

AS words FROM txtnum;



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



SELECT REPLACE(

REPLACE(REP

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

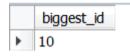
WHERE city.CountryCode IN ('AFG','ALB') AND c.Code=city.CountryCode;

	CountryCode	name	Name
•	AFG	Kabul	Afghanistan
	AFG	Qandahar	Afghanistan
	AFG	Herat	Afghanistan
	AFG	Mazar-e-Sharif	Afghanistan
	ALB	Tirana	Albania

Робота з числами:

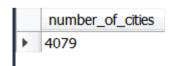
17. Представлення, в якому вивести мінімальне/максимальне значення у стовпці [1, стор.199].

SELECT MIN(id) AS smallest_id FROM users;



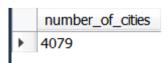
18. Представлення, яке повертає кількість записів у таблиці [1, стор.203].

SELECT COUNT(*) AS number of cities FROM city;



19. Представлення, яке повертає кількість визначених (не-NULL) значень [1, стор.205].

SELECT COUNT(id) AS number_of_cities FROM city;



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

```
SUM(Population)

• 6078749450
```

SELECT SUM (Population) FROM country;

Робота з датами:

21. 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;
```

	DATEDIFF(DAY2,DAY1)
•	1

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

DAY	count
Monday	53
Tuesday	52
Wednesday	52
Thursday	52
Friday	52
Saturday	52
Sunday	52

```
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');</pre>
```

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

```
days_in_february

≥ 28.0000
```

```
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;
```

```
first_day last_day

2018-11-01 16:01:41 2018-11-30
```

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;
```

	Мо	Tu	We	Th	Fr	Sa	Su	
•	NULL	NULL	NULL	01	02	03	04	
	05	06	07	08	09	10	11	
	12	13	14	15	16	17	18	
	19	20	21	22	23	24	25	
	26	27	28	29	30	NULL	NULL	

```
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