**Міністерство освіти і науки України**

**Національний університет харчових технологій**

Кафедра інформаційних технологій, штучного

інтелекту та кібербезпеки

**Звіт**

із лабораторної роботи № **4**

з дисципліни «Менеджмент баз та сховищ даних, їх аналіз»

на тему: «DML. JSON-функції, віконні функції та регулярні вирази.»

Варіант **13**

Виконав:

Студент групи КН-1-3М

Кучерявий М. В.

Перевірив:

асистент Струзік В.А.

**Київ — 2024**

**Завдання:**

1. Використовуйте репозиторій з навчальними базами даних.

2. Створити мінімум три запити, з використанням функції REGEXP,

що містить різні регулярні вирази.

3. Створити мінімум три запити, з використанням функції

REGEXP\_REPLACE, що містить різні регулярні вирази.

4. Створити мінімум три запити, з використанням функції

REGEXP\_SUBSTR, що містить різні регулярні вирази.

5. Створити запити, з використанням усіх функцій з переліку:

JSON\_OBJECT,

JSON\_ARRAY,

JSON\_ARRAYAGG,

JSON\_ARRAY\_APPEND, JSON\_CONTAINS, JSON\_MERGE,

JSON\_OBJECT,

JSON\_OBJECTAGG,

JSON\_DETAILED,

JSON\_REMOVE, JSON\_SET, JSON\_VALUE, JSON\_QUERY.

6. Створити запити, з використанням усіх віконних функцій з переліку:

AVG, COUNT, MAX, MIN, SUM, STD, MEDIAN, FIRST\_VALUE,

LAST\_VALUE, LAG, LEAD, ROW\_NUMBER, NTH\_VALUE,

DENSE\_RANK, RANK.

7. Завантажити звіт з виконання лабораторної роботи до системи

дистанційного навчання.

**Хід роботи:**

2. **SELECT** \*

**FROM** customer *c*

**WHERE** *c*.customer\_short\_name **REGEXP** ('^a')

**SELECT** \*

**FROM** customer *c*

**WHERE** *c*.customer\_short\_name **REGEXP** ('b$')

**SELECT** \*

**FROM** customer *c*

**WHERE** c.customer\_short\_name **NOT** **REGEXP** ('b$')

Запити повертають імена клієнтів з першою літерою «А», останньою літерою «В», та всі крім рядків з останньою літерою «В».

3. **SELECT** *c*.customer\_short\_name,

**REGEXP\_REPLACE**(*c*.customer\_short\_name, 'B$', **"b"**)

**FROM** customer *c*

**SELECT** *c*.customer\_short\_name,

**REGEXP\_REPLACE**(*c*.customer\_short\_name, '^A', **"a"**)

**FROM** customer *c*

**SELECT** *c*.customer\_short\_name,

**REGEXP\_REPLACE**(*c*.customer\_short\_name, 'L', **"LBIG"**)

**FROM** customer *c*

Запити повертають змінені імена клієнтів.

4. **SELECT** *c*.customer\_short\_name,

**REGEXP\_SUBSTR** (*c*.customer\_short\_name, '[A-Z]')

**FROM** customer *c*

**SELECT** *c*.customer\_short\_name,

**REGEXP\_SUBSTR** (*c*.customer\_short\_name, '[0-9]')

**FROM** customer *c*

**SELECT** *c*.customer\_short\_name,

**REGEXP\_SUBSTR** (*c*.customer\_short\_name, '^AB')

**FROM** customer *c*

**WHERE** **REGEXP\_SUBSTR** (*c*.customer\_short\_name, '^AB') != **""**

Запити повертають частини рядків та певну та обробку по ним.

5. **SELECT** **JSON\_OBJECT**("customer\_id", customer\_id, "customer\_short\_name", customer\_short\_name)

**FROM** customer *c*

Повертає новий JSON об’єкт з полями «customer\_id» «customer\_short\_name»

**SELECT** **JSON\_ARRAY**(customer\_id, customer\_short\_name)

**FROM** customer *c*

Повертає новий JSON масив з значеннями

**SELECT** **JSON\_ARRAYAGG**(customer\_region\_code)

**FROM** customer *c*

Повертає JSON масив з елементів «customer\_id» «customer\_short\_name»

**SELECT** **JSON\_ARRAY\_APPEND**(**JSON\_OBJECT**(**"arr"**, **JSON\_ARRAY**(1, 2)), **"$.arr"**, **"item"**) **AS** *"arr"*

Повертає створює JSON об’єкт та додає в його масив arr значення item

**SELECT** **JSON\_OBJECT**(**"a"**, customer\_short\_name), **JSON\_CONTAINS**(**JSON\_OBJECT**(**"a"**, customer\_short\_name), '"AAI"', **"$.a"**)

**FROM** customer *c*

Перевіряє чи є в JSON об’єкті значення «AAI»

**SELECT** **JSON\_MERGE**(**JSON\_OBJECT**(**"name"**, *c*.customer\_short\_name), **JSON\_OBJECT**(**"region"**, *c*.customer\_region\_code))

**FROM** customer *c*

Створює та поєднує 2 JSON об’єкта

**SELECT** **JSON\_OBJECTAGG**(*c*.customer\_short\_name, *o*.order\_date)

**FROM** customer *c*

**RIGHT** **JOIN** orderreg *o*

**ON** *c*.customer\_id = *o*.customer\_id

Автоматично створює об’єкт з полями customer\_short\_name та order\_date

**SELECT** **JSON\_REMOVE**(*ij*.invoice\_data, **"$.order\_id"**)

**FROM** invoice\_json *ij*

Видаляє з результату поле order\_id

**SELECT** **JSON\_SET**(*ij*.invoice\_data, **"$.order\_id"**, 12)

**FROM** invoice\_json *ij*

Встановлює в результаті поле order\_id

**SELECT** **JSON\_VALUE**(*ij*.invoice\_data, **"$.order\_id"**)

**FROM** invoice\_json *ij*

Отримує значення поля order\_id

**SELECT** JSON\_QUERY(**JSON\_OBJECT**(**"item"**, **JSON\_OBJECT**(**"name"**, *c*.customer\_short\_name, **"region"**, *c*.customer\_region\_code)), **"$.item"**)

**FROM** customer *c*

Отримує JSON об’єкт item з створеного об’єкту

6. **SELECT**

**AVG**(*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"avg"**,

**COUNT**(*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"count"**,

**MAX**(*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"max"**,

**MIN**(*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"min"**,

MEDIAN(*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"median"**,

STD(*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"std"**,

**FIRST\_VALUE** (*i*.invoice\_total) **OVER** (**PARTITION** **BY** c.customer\_short\_name) **AS** **"first"**,

**LAST\_VALUE** (*i*.invoice\_total) **OVER** customer\_name **as** **"LAST\_VALUE"**,

**ROW\_NUMBER** () **OVER** customer\_name **AS** **"ROW\_NUMBER"**,

NTH\_VALUE(*i*.invoice\_total, 2) **OVER** customer\_name **AS** **"NTH\_VALUE"**,

**DENSE\_RANK** () **OVER** customer\_name **AS** **"dense rank"**,

**RANK** () **OVER** customer\_name **AS** **"rank"**,

*c*.customer\_short\_name,

*c*.customer\_region\_code

**FROM** invoice *i*

**INNER** **JOIN** customer *c*

**ON** *i*.customer\_id = *c*.customer\_id

**WINDOW** customer\_name **AS** (**PARTITION** **BY** c.customer\_short\_name **ORDER** **BY** *c*.customer\_short\_name **DESC**)

Результати дії віконних функцій над ціною рахунку. З використанням WINOW скорочення та без нього