

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут”
Кафедра АСОІУ

ЗВІТ
про виконання лабораторної роботи №1
з дисципліни
“Програмне застосування баз даних”
Тема: Аналітичні запити

Прийняв:
Клименко О.М.

Виконав:
студент 3-го курсу
гр. ПІ-52 ФІОТ
Набоков Е.М

Київ 2017

1 ПОСТАНОВКА ЗАДАЧІ

9.	База трейлерів 1. У якому гаражі найменша кількість ремонтів. 2. Надати список назв ремонтів по маркам машин. В кінці надати кількість ремонтів за кожною маркою загалом. 3. Чи є гаражі не укомплектовані механіками. 4. Надати номер гаража, марку машин та табельний номер для машин, що не були у ремонті жодного разу. 5. У яких гаражах ремонтувалися машини з «Дата» по «Дата» за ремонтом „Назва”	Парк трейлерів	Номенклатурний н-р машини Марка машини Дата вироблення Номер гаража
		Види	Код виду ремонтів Назва виду
		Персонал	Табельний номер механіка Прізвище ІПБ Номер гаража Стаж роботи
		Ремонтні	Дата Номенклатурний номер машини Табельний номер механіка Код вигляду ремонту.

Було використано дві таблиці як два статичних довідника — Ремонтні та Персонал, що використовуються в таблиці оперативних даних.

2 СХЕМА БАЗИ ДАНИХ

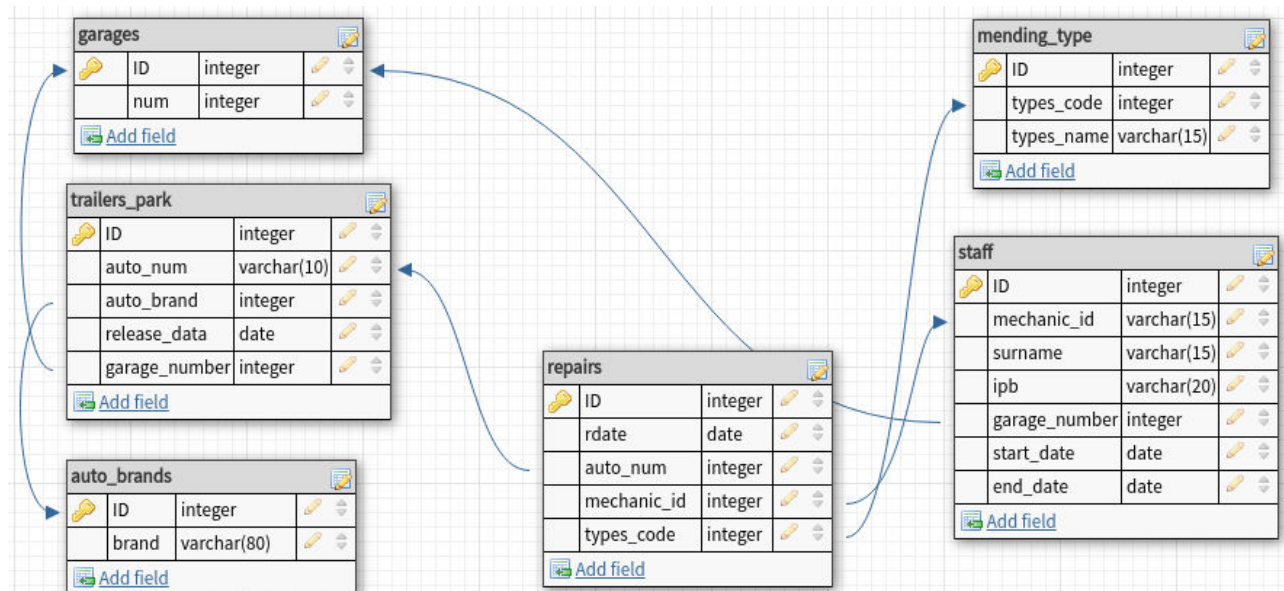


Рисунок 2.1 — схема.

3 РЕЗУЛЬТАТ ВИКОНАННЯ ПЕРШОГО ЗАВДАННЯ

3.1 Використовуючи count() (або будь-яку іншу агрегатну функцію), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Знайти як часто певний тип ремонту (за кодом) найрозповсюдженіший

-- with analytical function

```
SELECT DISTINCT types_code, COUNT(*) OVER(PARTITION by types_code) as count
FROM repairs
ORDER by types_code;
```

```
myTrailers=# SELECT DISTINCT types_code, COUNT(*) OVER(PARTITION by types_code) as count
FROM repairs
ORDER by types_code;
 types_code | count
-----+-----
          1 |    13
          2 |    23
          3 |    23
          4 |    19
          5 |    19
          6 |    16
          7 |    23
          8 |    18
          9 |    25
         10 |    21
(10 rows)
```

-- without analytical function

```
WITH c AS (SELECT types_code, COUNT(types_code) AS counter FROM repairs GROUP BY
types_code)
SELECT DISTINCT types_code, c.counter as Count FROM repairs JOIN c USING (types_code)
ORDER BY types_code;
```

```

myTrailers=# WITH c AS (
  SELECT types_code, COUNT(types_code) AS counter
  FROM repairs
  GROUP BY types_code
)
SELECT DISTINCT types_code, c.counter as Count
FROM repairs
JOIN c
  USING (types_code)
ORDER BY
  types_code;
types_code | count
-----+-----
          1 |    13
          2 |    23
          3 |    23
          4 |    19
          5 |    19
          6 |    16
          7 |    23
          8 |    18
          9 |    25
         10 |    21
(10 rows)

```

3.2 Використовуючи rank() або dense_rank(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Знайти прізвище кожного механіка, який здійснив своє останнє ремонтування авто

-- with analytical function

```

WITH c AS (
  SELECT *,
    DENSE_RANK() OVER(PARTITION by mechanic_id ORDER BY rdate)
    as rank_
  FROM repairs
), cc AS (
  SELECT mechanic_id, MAX(rank_) as maxRank FROM c
  GROUP BY c.mechanic_id
  ORDER BY c.mechanic_id
)
SELECT staff.surname, c.rdate FROM cc
JOIN c ON cc.mechanic_id = c.mechanic_id and cc.maxRank = c.rank_
JOIN staff ON cc.mechanic_id = staff.id;

```

surname	rdate
Mccormick	2001-05-17
Walter	1983-10-07
Browning	1994-10-08
Stokes	2017-03-20
Shelton	2005-08-14
Rosales	2003-05-29
Stephens	2005-11-07
Torres	2015-11-01
Nixon	2015-12-02
Hodges	2008-07-15
Payne	2007-12-05
Kaiser	2010-02-08
Dillon	2013-07-18
Schneider	2000-03-20
Bush	2015-09-05
Webb	2017-06-13
Farmer	2008-08-30
Allison	2004-07-02
Forbes	1994-02-06
Aguirre	1997-09-12
Pratt	2004-07-25

-- without analytical function

```
SELECT staff.surname, f.date FROM (  
    SELECT mechanic_id, MAX(rdate) as date FROM repairs  
    GROUP BY mechanic_id  
    ORDER BY mechanic_id  
) as f JOIN staff ON staff.id = f.mechanic_id;
```

surname	date
Mccormick	2001-05-17
Walter	1983-10-07
Browning	1994-10-08
Stokes	2017-03-20
Shelton	2005-08-14
Rosales	2003-05-29
Stephens	2005-11-07
Torres	2015-11-01
Nixon	2015-12-02
Hodges	2008-07-15
Payne	2007-12-05
Kaiser	2010-02-08
Dillon	2013-07-18
Schneider	2000-03-20
Bush	2015-09-05
Webb	2017-06-13
Farmer	2008-08-30
Allison	2004-07-02
Forbes	1994-02-06
Aguirre	1997-09-12
Pratt	2004-07-25

3.3 Використовуючи sliding window (rows), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Знайти середню заробітню плату за кожний гараж

-- with analytical function

```
SELECT DISTINCT garage_number, AVG(salary)
      OVER (PARTITION by garage_number ORDER BY salary ROWS BETWEEN 2 preceding AND 2
following)
FROM staff
ORDER BY garage_number;
```

garage_number	avg
4	415
6	192
9	571.666666666667
12	988
15	601
17	852
23	467
24	972.5
26	671
29	152
30	588
31	829
33	624
34	666
35	678
36	275
41	1028.5
48	901
49	682
52	682
58	503
60	688.5
61	492
62	769
65	110
66	1234
67	805
71	364
72	615
73	561
78	1500
84	582
85	582
86	433
87	635
88	473
89	827
91	532
95	123
96	908
100	1320

(41 rows)

-- without analytical function

```
SELECT garage_number, AVG(salary)
```

```
FROM staff
```

```
GROUP BY garage_number
```

```
ORDER BY garage_number;
```

garage_number	avg
4	415
6	192
9	571.6666666666667
12	988
15	601
17	852
23	467
24	972.5
26	671
29	152
30	588
31	829
33	624
34	666
35	678
36	275
41	1028.5
48	901
49	682
52	682
58	503
60	688.5
61	492
62	769
65	110
66	1234
67	805
71	364
72	615
73	561
78	1500
84	582
85	582
86	433
87	635
88	473
89	827
91	532
95	123
96	908
100	1320

(41 rows)

3.4 Використовуючи sliding window (range) , partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Знайти загальні витрати на кожен гараж окремо, які власник повинен покривати

-- with analytical function

```
SELECT DISTINCT garage_number, SUM(salary)
      OVER (PARTITION by garage_number ORDER BY salary RANGE BETWEEN UNBOUNDED
preceding AND UNBOUNDED FOLLOWING)
FROM staff
ORDER BY garage_number;
```

garage_number	sum
4	415
6	384
9	1715
12	988
15	601
17	852
23	934
24	1945
26	671
29	152
30	588
31	829
33	624
34	666
35	678
36	275
41	2057
48	901
49	682
52	682
58	503
60	1377
61	984
62	769
65	110
66	1234
67	805
71	728
72	615
73	561
78	1500
84	582
85	582
86	433
87	635
88	473
89	827
91	532
95	123
96	908
100	1320

(41 rows)

-- without analytical function

```
SELECT DISTINCT garage_number, SUM(salary)
```

```
FROM staff
```

```
GROUP BY garage_number
```

```
ORDER BY garage_number;
```

garage_number	sum
4	415
6	384
9	1715
12	988
15	601
17	852
23	934
24	1945
26	671
29	152
30	588
31	829
33	624
34	666
35	678
36	275
41	2057
48	901
49	682
52	682
58	503
60	1377
61	984
62	769
65	110
66	1234
67	805
71	728
72	615
73	561
78	1500
84	582
85	582
86	433
87	635
88	473
89	827
91	532
95	123
96	908
100	1320

(41 rows)

3.5 Самостійно розібратися, як застосовується функція lag(). Використовуючи lag(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Вирахувати збалансовані (усереднені) кошти за минулий рік:

-- with analytical function

```
SELECT DISTINCT garage_number,  
               lag(CAST(salary AS real), 1, CAST(salary AS real)) OVER (PARTITION by garage_number ORDER BY  
end_date) as balanced_salary  
FROM staff  
ORDER BY garage_number;
```

garage_number	balanced_salary
1	111
2	421
4	415
6	150
9	642
9	753
12	988
15	601
17	852
23	351
24	1024
26	671
29	152
30	588
31	829
33	624
34	666
35	678
36	275
41	533

-- without analytical function

WITH shifted AS (

```

SELECT id + 1 as incr_id FROM staff
), new_salary AS (
SELECT id, garage_number, salary as balanced_salary FROM staff
JOIN shifted ON shifted.incr_id = staff.id
), addition AS(
SELECT staff.id, staff.garage_number, salary, balanced_salary FROM staff
JOIN new_salary ON new_salary.id = staff.id+1
)
SELECT staff.garage_number, balanced_salary FROM staff
JOIN addition ON addition.id = staff.id
ORDER BY staff.garage_number;

```

garage_number	balanced_salary
1	111
2	421
4	415
6	150
9	642
9	753
12	988
15	601
17	852
23	351
24	1024
26	671
29	152
30	588
31	829
33	624
34	666
35	678
36	275
41	533
48	901
49	682
52	682

3.6 Самостійно розібратися, як застосовується функція lead(). Використовуючи lead(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Вирахувати збалансовані (усереднені) кошти на наступний рік:

-- with analytical function

```
SELECT DISTINCT garage_number, salary,  
               lead(CAST(salary AS real), 1, CAST(salary AS real)) OVER (PARTITION by garage_number ORDER  
BY end_date) as balanced_salary  
FROM staff  
ORDER BY garage_number;
```

garage_number	salary	balanced_salary
1	111	111
2	421	421
4	415	415
6	150	234
6	234	234
9	320	320
9	642	320
9	753	642
12	988	988
15	601	601
17	852	852
23	351	583
23	583	583
24	921	921
24	1024	921
26	671	671

-- without analytical function

```
WITH shifted AS (  
  SELECT id - 1 as decr_id FROM staff  
) , new_salary AS (  
  SELECT id, garage_number, salary as balanced_salary FROM staff  
  LEFT OUTER JOIN shifted ON shifted.decr_id = staff.id  
) , addition AS(  
  SELECT staff.id, staff.garage_number, salary, balanced_salary FROM staff  
  JOIN new_salary ON new_salary.id = staff.id
```

)

```
SELECT staff.garage_number, staff.salary, balanced_salary FROM staff
```

```
JOIN addition ON addition.id = staff.id
```

```
ORDER BY staff.garage_number;
```

garage_number	salary	balanced_salary
1	111	111
2	421	421
4	415	415
6	150	234
6	234	234
9	320	320
9	642	320
9	753	642
12	988	988
15	601	601
17	852	852
23	351	583
23	583	583
24	921	921
24	1024	921
26	671	671
29	152	152
30	588	588
31	829	829
33	624	624
34	666	666
35	678	678
36	275	275

4 РЕЗУЛЬТАТ ВИКОНАННЯ ДРУГОГО ЗАВДАННЯ

4.1 Змінити таблицю так, щоб вона не була у першій нормальній формі, навести цю таблицю у наповненому вигляді. Потім описати, які зміни треба зробити, щоб вона відповідала першій нормальній формі, навести цю таблицю у наповненому вигляді.

Неправильно:

50	72299	Duran	G.B	30	1986-06-28	2008-12-25	588
1111	10101	Matt, Aniston	A.X	1	1888-08-08	1889-09-09	111

Зі скріншота можна побачити фрагмент таблиці «Персонал», де в одну комірку записано два прізвища — Matt, Aniston. Це вже нам повідомляє, що таблиця не знаходиться у першій нормальній формі. Потрібно розділити ці прізвища та записати їх окремими рядками у таблицю.

Правильно:

50	72299	Duran	G.B	30	1986-06-28	2008-12-25	588
1111	10101	Matt	A.X	1	1888-08-08	1889-09-09	111
1112	10102	Aniston	B.T	2	1998-12-12	2009-09-09	421

Таблиця знаходиться в першій нормальній формі, тільки тоді коли усі її атрибути є простими. Не повинно бути повторних рядків у таблиці.

4.2 Описати, які зміни треба зробити, щоб таблиця відповідала другій нормальній формі, навести цю таблицю у наповненому вигляді.

Неправильно:

id	auto_num	release_date	brand	garage_number
1	1001	1998-09-09	Audi	10
2	8888	1982-10-02	BMW	50

Зі скріншота можна побачити фрагмент таблиці «Парк трейлерів», де знаходиться 4 характеристики одного трейлеру. Дана таблиця не знаходиться у другій нормальній формі, так як модель(brand) та номер гаражу (garage_number) не залежать від реєстраційного номеру автомобіля (auto_num). Окрім цього марки автомобілів обмежені, тому є сенс винести до окремої таблиці на яку буде посилення (зовнішній ключ). Так само і з гаражами.

Правильно:

- Створимо таблицю для збереження номеру гаражів (первинного ключа та номер гаража)

id	garage_number
1	10
2	50

(2 rows)

id	brand
1	Audi
2	BMW

(2 rows)

- Створимо таблицю для марок автомобілів

id	auto_num	auto_brand	release_date	garage_number
1	1001	1	1998-09-09	1
2	8888	2	1982-10-02	2

- Використаємо зовнішній ключ на ці дві бази від нашої головної

Таблиця знаходиться у другій нормальній формі, тільки тоді коли знаходиться у першій НФ та кожен її атрибут залежить від первинного ключа.

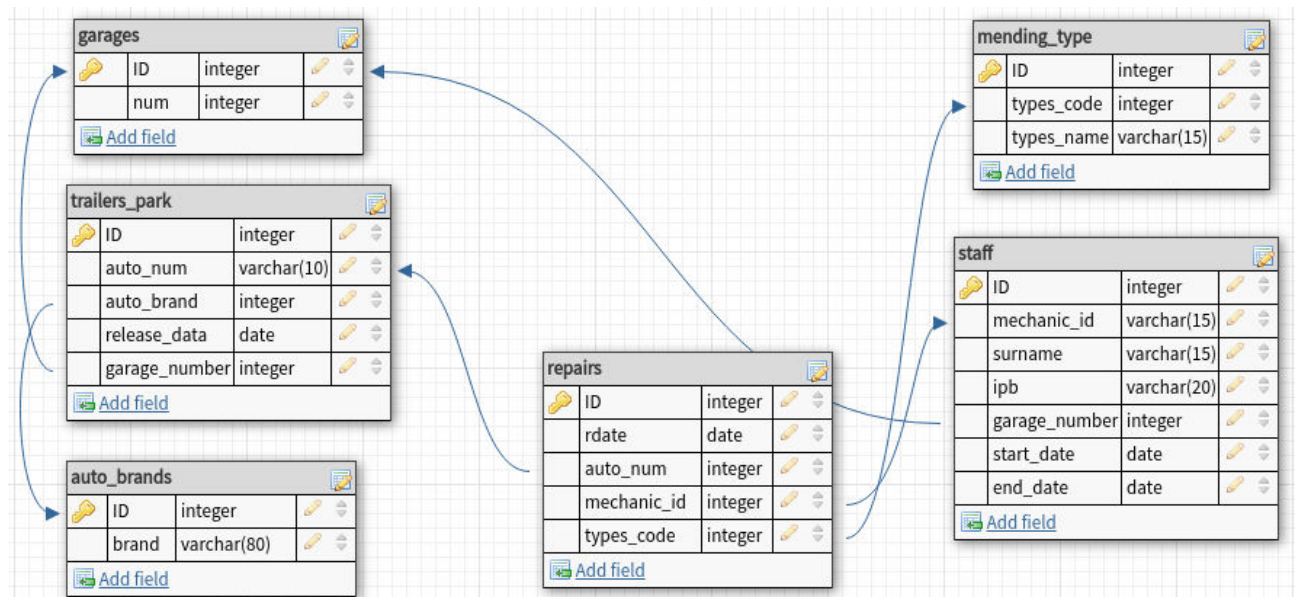
4.3 Описати, які зміни треба зробити, щоб таблиця відповідала третій нормальній формі, навести цю таблицю у наповненому вигляді.

Візьмемо попередній приклад про другу НФ.

Третя нормальна форма каже нам про те, щоб усі неключові дані були винесені з таблиці. Перевіряє дані на транзитивність.

4.4 Описати, які зміни треба зробити, щоб таблиця відповідала нормальній формі Бойса-Кода, навести цю таблицю у наповненому вигляді.

Наприклад, візьмемо до уваги мою схему бази даних:



На схемі видно, що таблиця парку трейлерів та персоналу мають спільний атрибут — номер гаражу.

Уявімо, що парк трейлерів та персонал — об'єднанні у єдину таблицю, вона і буде суперечити цій формі.

НФ Бойса-Кода заявляє про те, що визначення третьої нормальної форми не зовсім підходить для таких відношень як : відношення має два або більше потенційних ключа, два та більше потенційних ключа є складовими, вони перетинаються, тобто мають спільний атрибут.