

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
"НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"
КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ**

Лабораторная работа №3
по дисциплине Технологии баз данных

вариант задания № 1

**Факультет: ФПМИ
Группа: ПМ-13
Бригада №3
Студенты: Вострецова Екатерина
Исакин Даниил**

**Преподаватели: Стасышина Т.Л.
Сивак М.А.**

Новосибирск 2024

Цель работы:

Приобрести практические навыки работы со встроенным SQL средствами ESQL/C, включая использование курсоров и средств динамического SQL.

Ход работы:

Часть 1:

- 1. Выдать число поставок, выполненных для изделий с деталями зеленого цвета.**

```
SELECT COUNT(n_izd)
FROM spj
WHERE spj.n_izd IN(SELECT n_izd
                    FROM spj
                    JOIN p ON p.n_det=spj.n_det
                    WHERE p.cvet='Зелёный'
)
```

count
13
1 запис(ь/и/ей)

2. Поменять местами города, где размещены изделия с самым коротким и самым длинным названием, т. е. изделия с самым коротким названием перевести в город, где размещено изделие с самым длинным названием, и наоборот, изделия с самым длинным названием перевести в город, где размещено изделие с самым коротким названием. Если городов несколько, брать первый по алфавиту из этих городов.

Данные ДО изменений:

Действия	n_idz	name	town
Редактировать	Удалить	32 Перфоратор	Рим
Редактировать	Удалить	33 Считыватель	Афины
Редактировать	Удалить	35 Флоппи-диск	Лондон
Редактировать	Удалить	36 Терминал	Осло
Редактировать	Удалить	37 Лента	Лондон
Редактировать	Удалить	31 Жесткий диск	Париж
Редактировать	Удалить	34 Принтер	Афины

7 запис(ь/и/ей)

[Назад](#) | [Расширить](#) | [Вставить](#) | [Обновить](#)

```
UPDATE j SET town = (CASE WHEN length(j.name) =  
                        (SELECT MAX(length(name))  
                         FROM j)  
    THEN ( SELECT j1.town  
            FROM j j1  
            ORDER BY length(j1.name), j1.town  
            LIMIT 1)  
    ELSE ( SELECT j2.town  
            FROM j j2  
            ORDER BY length(j2.name) DESC, j2.town  
            LIMIT 1)  
END)
```

WHERE length(j.name) = (SELECT MIN(length(name))
FROM j j5)

OR

length(j.name) = (SELECT MAX(length(name))
FROM j j6)

Данные ПОСЛЕ изменений:

Действия		n_izd	name	town
Редактировать	Удалить	12	Перфоратор	Рим
Редактировать	Удалить	13	Считыватель	Афины
Редактировать	Удалить	15	Флоппи-диск	Лондон
Редактировать	Удалить	16	Терминал	Осло
Редактировать	Удалить	14	Принтер	Афины
Редактировать	Удалить	17	Лента	Париж
Редактировать	Удалить	11	Жесткий диск	Лондон
7 запис(ь/и/ей)				

После данного запроса таблицы были возвращены в исходное состояние для написания дальнейших запросов

3. Найти детали, имеющие поставки, вес которых меньше среднего веса поставок этой детали для изделий из Лондона.

```
SELECT spj.n_det
FROM spj
JOIN p ON p.n_det=spj.n_det
JOIN (SELECT spj.n_det,avg(spj.kol*p.ves) mves
      FROM spj
      JOIN p ON p.n_det=spj.n_det
      JOIN j ON j.n_izd=spj.n_izd
      WHERE j.town='Лондон'
      GROUP BY spj.n_det
) zap ON zap.n_det=spj.n_det
WHERE spj.kol*p.ves<mves
```

n_det
P3
P3
P3
P3
P3
P3
P5
P3
P5
P6
P3

11 запис(ь/и/ей)

4. **Выбрать поставщиков, не поставляющих ни одной из деталей, поставляемых поставщиками, находящимися в Лондоне.**

```
SELECT n_post
FROM spj
EXCEPT
SELECT DISTINCT n_post
FROM spj
WHERE n_det IN (SELECT n_det
                FROM spj
                JOIN s ON s.n_post=spj.n_post
                WHERE town='Лондон'
            )
```

n_post
S2
S3

2 запис(ь/и/ей)

5. **Выдать полную информацию о поставщиках, выполнивших поставки ТОЛЬКО с объемом от 200 до 500 деталей.**

```
SELECT *
FROM s
WHERE n_post IN(SELECT n_post
                FROM spj
                EXCEPT
                SELECT n_post
                FROM spj
                WHERE not (kol<=500 and kol>=200)
            )
```

n_post	name	reiting	town
S3	Блейк	30	Париж
S4	Кларк	20	Лондон

2 запис(ь/и/ей)

Часть 2:

Текст программы:

```
#include <sqlca.h>
#include <stdio.h>
#include <stdlib.h>
// сообщение об ошибке
int error_msg(char *desc)
{
    printf("\n%s\nКод: %d\nОписание ошибки: %s\n", desc, sqlca.sqlcode,
    sqlca.sqlerrm.sqlerrmc);
    return -1;
}
```

```

void connectBD()
{
    // подключение к бд
    exec sql connect to students@students.ami.nstu.ru user "pmi-b1303" using "Shlasow0";
    if(sqlca.sqlcode < 0)
    {
        printf("\nНеверный логин или пароль\nОписание ошибки: %s\n*Выход из
программы*\n\n",
sqlca.sqlerrm.sqlerrmc);
        exit(1);
    }
}
void closeBD()
{
    // закрываем сессию
    exec sql disconnect current;
}
int useScheme()
{
    // путь к схеме
    exec sql set search_path to "pmi-b1303", public;
    if (sqlca.sqlcode < 0)
    {
        printf("Ошибка выбора схемы");
        return 1;
    }
    else
        printf("Схема выбрана успешно!\n");
    return 0;
}
int main()
{
    // секция объявления переменных
    exec sql begin declare section;//начало
    int reiting, count_post, weight, min_weight, pves, ves,kol;
    float mves;
    char n_post[7], name[21], town[21], n_izd[7], n_det[7],cvet[21] task_num;
    exec sql end declare section;// конец

    connectBD();

    if(!useScheme())
    {
        // менюшечка
        printf("\n1. Выдать число поставок, выполненных для изделий с деталями
зеленого цвета. ");
        printf("\n2. Поменять местами города, с самым коротким и самым длинным
названием изделия.");
        printf("\n3. Найти детали, имеющие поставки, вес которых меньше среднего
веса поставок этой детали для изделий из Лондона.");
        printf("\n4. Выбрать поставщиков, не поставляющих ни одной из деталей,
поставляемых поставщиками, находящимися в Лондоне.");
        printf("\n5. Выдать полную информацию о поставщиках, выполнивших поставки
ТОЛЬКО с объемом от 200 до 500 деталей. ");
        printf("\nВыберите задание (1-5): ");
        while(scanf(" %c", &task_num) && task_num > '0' && task_num < '6')
        {
            switch(task_num)
            {

```

```

case '1':
    // начало транзакции
    exec sql begin work;

    // первый запрос
    exec sql select count(n_izd)
    into :count_post//сюда сохраняем результат
    from spj
    where spj.n_izd in(select n_izd
                        from spj
                        join p on p.n_det=spj.n_det
                        where p.cvet='Зелёный');

    // проверяем результат
    if (sqlca.sqlcode < 0)
    {
        error_msg("В запросе есть ошибка.");
        exec sql rollback work;//отменяем все сделанные изменения в рамках
транзакции
        break;
    }
    else
        if (sqlca.sqlcode == 0)
            printf("\nЧисло поставок = %d\n", count_post);
    //подтверждение транзакции
    exec sql commit work;
break;
case '2':
    exec sql begin work;
    // запрос намбер ту
    exec sql update j set town = (case when length(j.name) =
                                (select max(length(name))
                                 from j)
                                then (select j1.town
                                     from j j1
                                     order by length(j1.name), j1.town
                                     limit 1)
                                else (select j2.town
                                     from j j2
                                     order by length(j2.name) desc, j2.town
                                     limit 1)
                                end)
                                where length(j.name) = (select min(length(name))
                                                         from j j5)
                                                         or length(j.name) = (select max(length(name))
                                                         from j j6);

    if (sqlca.sqlcode < 0)
    {
        error_msg("Ошибка при изменении (UPDATE).");
        exec sql rollback work;
        break;
    }
    else
        if (sqlca.sqlcode == 0)
        {
            printf("\nКоличество обработанных записей: %d\n", sqlca.sqlerrd[2]);
        };
    exec sql commit work;
break;

```

```

case '3':
    exec sql begin work;
    // запрос 3
    //курсор помогает просматривать результаты построчно
    exec sql declare cursor_3 cursor for
    select spj.n_det
    into :n_det
    from spj
    join p on p.n_det=spj.n_det
    join (select spj.n_det,avg(spj.kol*p.ves) mves
          from spj
          join p on p.n_det=spj.n_det
          join j on j.n_izd=spj.n_izd
          where j.town='Лондон'
          group by spj.n_det
        ) zap on zap.n_det=spj.n_det
    where spj.kol*p.ves<mves;

    exec sql open cursor_3;
    if (sqlca.sqlcode < 0)
    {
        error_msg("Ошибка при открытии курсора (OPEN).");
        exec sql close cursor_3;
        exec sql rollback work;
        break;
    }
    //команда, запрашивающая следующую строку курсора
    exec sql fetch cursor_3;
    if (sqlca.sqlcode < 0)
    {
        error_msg("Ошибка при чтении курсора(FETCH).");
        exec sql close cursor_3;
        exec sql rollback work;
        break;
    }
    else
    if (sqlca.sqlcode == 100)
        printf("\nДанные отсутствуют.\n");
    else
    {
        printf("\nНомер детали\n%s\n", n_det);
        while (sqlca.sqlcode == 0)
        {
            exec sql fetch cursor_3;
            if (sqlca.sqlcode < 0)
            {
                error_msg("Ошибка при чтении курсора(FETCH).");
                exec sql close cursor_3;
                exec sql rollback work;
                break;
            }
            else
            if (sqlca.sqlcode == 0)
                printf("%s\n", n_det);
        }
    }
    exec sql close cursor_3;
    exec sql commit work;
break;

```

```

case '4':
    exec sql begin work;
    //запрос 4
    exec sql declare cursor_4 cursor for
    select n_post
    into :n_post
    from spj
    except
    select distinct n_post
    from spj
    where n_det in (select n_det
                    from spj
                    join s on s.n_post=spj.n_post
                    where town='Лондон');

    exec sql open cursor_4;
    if (sqlca.sqlcode < 0)
    {
        error_msg("Ошибка при открытии курсора(OPEN).");
        exec sql close cursor_4;
        exec sql rollback work;
        break;
    }
    //читаем курсор
    exec sql fetch cursor_4;
    if (sqlca.sqlcode < 0)
    {
        error_msg("Ошибка при чтении курсора (FETCH).");
        exec sql close cursor_4;
        exec sql rollback work;
        break;
    }
    else
    if (sqlca.sqlcode == 100)
        printf("\nДанные отсутствуют.\n");
    else
    {
        printf("\nСписок поставщиков:\n%s\n", n_post);
        while (sqlca.sqlcode == 0)
        {
            exec sql fetch cursor_4;
            if (sqlca.sqlcode < 0)
            {
                error_msg("Ошибка при чтении курсора(FETCH).");
                exec sql close cursor_4;
                exec sql rollback work;
                break;
            }
            else
            if (sqlca.sqlcode == 0)
                printf("%s\n", n_post);
        }
    }
    exec sql close cursor_4;
    exec sql commit work;
break;
case '5':
    exec sql begin work;
    //запрос номер 5

```



```

exec sql declare cursor_5 cursor for
select *
into :n_post, :name, :reiting, :town
from s
where n_post in(select n_post
                 from spj
                 except
                 select n_post
                 from spj
                 where not (kol<=500 and kol>=200)
);
exec sql open cursor_5;
if (sqlca.sqlcode < 0)
{
    error_msg("Ошибка при открытии курсора(OPEN).");
    exec sql close cursor_5;
    exec sql rollback work;
    break;
}
exec sql fetch cursor_5;
if (sqlca.sqlcode < 0)
{
    error_msg("Ошибка при чтении курсора (FETCH).");
    exec sql close cursor_5;
    exec sql rollback work;
    break;
}
else
    if (sqlca.sqlcode == 100)
        printf("\nДанные отсутствуют.\n");
    else
    {
        printf("\nНомер\tИмя\t\tРейтинг\tГород\n%s\t%s\t%d\t%s\n", n_post,
name, reiting,
town);

        while (sqlca.sqlcode == 0)
        {
            exec sql fetch cursor_5;
            if (sqlca.sqlcode < 0)
            {
                error_msg("Ошибка при чтении курсора(FETCH).");
                exec sql close cursor_5;
                exec sql rollback work;
                break;
            }
            else
                if (sqlca.sqlcode == 0)
                    printf("%s\t%s\t%d\t%s\n", n_post, name, reiting, town);
        }
    }
exec sql close cursor_5;
exec sql commit work;
break;
}
printf("\n1. Выдать число поставок, выполненных для изделий с деталями
зеленого цвета. ");
printf("\n2. Поменять местами города, с самым коротким и самым длинным
названием изделия.");
printf("\n3. Найти детали, имеющие поставки, вес которых меньше среднего

```

```

веса поставок этой детали для изделий из Лондона.");
printf("\n4. Выбрать поставщиков, не поставляющих ни одной из деталей,
поставляемых поставщиками, находящимися в Лондоне.");
printf("\n5. Выдать полную информацию о поставщиках, выполнивших поставки
ТОЛЬКО с объемом от 200 до 500 деталей. ");
printf("\nВыберите задание (1-5): ");
}
}
closeBD();
return 0;
}

```

Проверка работы:

```

[pmi-b1303@students ~]$ pgccci main
[pmi-b1303@students ~]$ ./main.exe
Схема выбрана успешно!

```

1. Выдать число поставок, выполненных для изделий с деталями зеленого цвета.
2. Поменять местами города, с самым коротким и самым длинным названием изделия.
3. Найти детали, имеющие поставки, вес которых меньше среднего веса поставок этой детали для изделий из Лондона.
4. Выбрать поставщиков, не поставляющих ни одной из деталей, поставляемых поставщиками, находящимися в Лондоне.
5. Выдать полную информацию о поставщиках, выполнивших поставки ТОЛЬКО с объемом от 200 до 500 деталей.

Выберите задание (1-5): 1

Число поставок = 13

Выберите задание (1-5): 3

Номер детали

P6
P3
P3
P3
P3
P3
P3
P3
P3
P5
P5

Список поставщиков:

S2
S3

Выберите задание (1-5): 5

Номер	Имя	Рейтинг	Город
S3	Блейк	30	Париж
S4	Кларк	20	Лондон

Выберите задание (1-5): 2

Количество обработанных записей: 2

Проверка на различные ошибки

- 1) Допустим, одной из таблицы не существует(sprj, она самая частая)

Таблица удалена.

Таблица	Пользователь	Tablespace	Estimated row count	Действия								
<input type="checkbox"/> j	pmi-b1303		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс
<input type="checkbox"/> p	pmi-b1303		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс
<input type="checkbox"/> s	pmi-b1303		7	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс

Результат:

Выберите задание (1-5): 3

Ошибка при открытии курсора (OPEN).

Код: -400

Описание ошибки: permission denied for relation spj on line 138

- 2) Ошибка изменения данных, добавим ограничение:

alter table j

add constraint new_constraint check(case when name = 'Лента' then town != 'Париж' end)

Выберите задание (1-5): 2

Ошибка при изменении (UPDATE).

Код: -400

Описание ошибки: new row for relation "j" violates check constraint "new_constraint" on line 91