# Министерство науки и высшего образования Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»

NSTU_Logo_blue

## Кафедра теоретической и прикладной информатики

### Лабораторная работа № 4 по дисциплине «Базы данных»

Вариант 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| сигма градиент синий1 | Факультет: | ПМИ |  |  |
| Группа: | ПМ-63 |  |  |
| Студенты: | Кожекин М.В.  Утюганов Д.С. |  |  |
|  |  |  |  |
| Преподаватели: | Стасышина Т.Л.  Хайленко Е.А. |  |  |

Новосибирск

2019

Разработать программу на ESQL/C, цель которой – решение следующих задач:

1. Выдать число деталей, которые поставлялись поставщиками, имеющими поставки с объёмом от 600 до 700 деталей.

|  |
| --- |
| -- число деталей от этих поставщиков  select count(distinct n\_det)  from spj  where n\_post in (      -- поставщики с объёмом поставок от 600 до 700 деталей      select n\_post      from spj      group by n\_post      having 600 <= sum(kol) and sum(kol) <= 700  ); |

2. Поменять местами цвета самой тяжёлой и самой лёгкой детали, т. е. деталям с наибольшим весом установить цвет детали с минимальным весом, а деталям с минимальным весом установить цвет детали с наибольшим весом. Если цветов несколько, брать первый по алфавиту из этих цветов.

|  |
| --- |
| update p set cvet = (          case when p.ves = (select max(ves) from p p5)              then (select p3.cvet cvet1              from p p3              order by p3.ves, p3.cvet              limit 1)          else (select p4.cvet cvet2              from p p4              order by p4.ves DESC, p4.cvet              limit 1)          end)  where p.ves = (select min(ves) from p p1)  or p.ves = (select max(ves) from p p2); |

3. Найти поставщиков, имеющих поставки, вес которых составляет менее четверти наибольшего веса поставки этого поставщика. Вывести номер поставщика, вес поставки, четверть наибольшего веса поставки поставщика.

|  |
| --- |
| -- вес поставок, которые легче самой тяжёлой поставки поставщика в 4 раза  select distinct spj.n\_post, kol\*ves as p\_weight, temp.quater  from spj  join p on p.n\_det = spj.n\_det  join (      -- четверть максимальной поставки каждого поставщика      select n\_post, max(kol\*ves) / 4 as quater      from spj      join p on p.n\_det = spj.n\_det      group by n\_post  ) temp  on spj.n\_post = temp.n\_post  where kol\*ves < temp.quater; |

4. Выбрать изделия, для которых не поставлялось ни одной из деталей, поставляемых поставщиком S4.

|  |
| --- |
| -- изделия, для которых не поставлялось ни одной из деталей,  -- поставляемых поставщиком S4  (      -- все изделия      select n\_izd      from j  )  except  (      -- изделия с деталями от поставщика S4      select distinct n\_izd      from spj      where n\_det in(          select n\_det          from spj          where n\_post = 'S4'      )  )  order by 1; |

5. Выдать полную информацию о деталях, которые поставлялись ТОЛЬКО поставщиками с максимальным рейтингом.

|  |
| --- |
| -- полная информация о деталях от поставщиков только с максимальным рейтингом  select \*  from p  where n\_det in(      -- детали от поставщиков только с максимальным рейтингом      (          -- детали от поставщиков с максимальным рейтингом          select distinct n\_det          from spj          where n\_post in (              select n\_post              from s              where reiting = (                  select max(reiting)                  from s)          )          order by 1      )      except      (          -- детали от остальных поставщиков          select distinct n\_det          from spj          where n\_post not in (              select n\_post              from s              where reiting = (                  select max(reiting)                  from s)          )          order by 1      )  ); |

**Текст программы**

|  |
| --- |
| #include <sqlca.h>  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  // 1. Выдать число деталей, которые поставлялись поставщиками, имеющими поставки  // с объемом от 600 до 700 деталей.  int problem1()  {      printf("\n\nЗадание 1\n");      printf("Вывести число деталей, которые поставлялись поставщиками, имеющими поставки с объемом от 600 до 700 деталей\n");      exec sql begin declare section;      int details\_count;      exec sql end declare section;      /////////////////////////////////////////////////////////////////////////SQL      exec sql begin work; // начало транзакции      exec sql      -- число деталей от этих поставщиков      select count(distinct n\_det) into :details\_count      from spj      where n\_post in (          -- поставщики с объёмом поставок от 600 до 700 деталей          select n\_post          from spj          group by n\_post          having 600 <= sum(kol) and sum(kol) <= 700      );      /////////////////////////////////////////////////////////////////////////SQL      if (sqlca.sqlcode < 0)      {          printf("Возникла ошибка. sqlca.sqlcode = %d\n", sqlca.sqlcode);          exec sql rollback work; // отмена транзакции          return -1;      }      else if (sqlca.sqlcode == 0)      {          printf("%d\n", details\_count);          exec sql commit work; // конец транзакции          return 0;      }  }  char \*my\_copy(char \*str)  {      char new\_str[64];      int i = 0;      while (i<strlen(str) && str[i] != ' ')      {          new\_str[i] = str[i];          i++;      }      return new\_str;  }    // 2. Поменять местами цвета самой тяжелой и самой легкой детали, т. е. деталям  // с наибольшим весом установить цвет детали с минимальным весом, а деталям с  // минимальным весом установить цвет детали с наибольшим весом. Если цветов несколько,  // брать первый по алфавиту из этих цветов.  int problem2()  {      printf("\n\nЗадание 2\n");      printf("Поменять местами цвета самой тяжелой и самой легкой детали:\n");      int status;      exec sql begin declare section;      char cvet[20];      exec sql end declare section;      /////////////////////////////////////////////////////////////////////////SQL      exec sql begin work; // начало транзакции      exec sql      update p set cvet = (              case when p.ves = (select max(ves) from p p5)                  then (select p3.cvet cvet1                  from p p3                  order by p3.ves, p3.cvet                  limit 1)              else (select p4.cvet cvet2                  from p p4                  order by p4.ves DESC, p4.cvet                  limit 1)              end)      where p.ves = (select min(ves) from p p1)      or p.ves = (select max(ves) from p p2);      /////////////////////////////////////////////////////////////////////////SQL      // P6 не должно быть красным      if (sqlca.sqlcode < 0)      {          char errmsg[400];          printf("Возникла ошибка. sqlca.sqlcode = %d\n", sqlca.sqlcode);          rgetmsg(sqlca.sqlerrd[1], errmsg, sizeof(errmsg));          printf("Сообщение ISAM: %s\n", errmsg);          exec sql rollback work; // отмена транзакции          return -1;      }      else if (sqlca.sqlcode == 0)      {          printf("Замена выполнена успешно.\n");          printf("Заменено %d строк\n", sqlca.sqlerrd[2]);          exec sql commit work;      }  }    // 3. Найти поставщиков, имеющих поставки, вес которых составляет менее четверти  // наибольшего веса поставки этого поставщика. Вывести номер поставщика, вес поставки,  // четверть наибольшего веса поставки поставщика.  int problem3()  {      printf("\n\nЗадание 3\n");      printf("Поставщики, имеющие поставки, вес которых составляет менее четверти наибольшего веса поставки этого поставщика:\n");      int status;      exec sql begin declare section;      int ves, quater\_of\_weight;      char n\_post[6];      exec sql end declare section;      /////////////////////////////////////////////////////////////////////////SQL      exec sql begin work; // начало транзакции      exec sql declare my\_cursor3 cursor for      -- вес поставок, которые легче самой тяжёлой поставки поставщика в 4 раза      select distinct spj.n\_post, kol\*ves as p\_weight, temp.quater      from spj      join p on p.n\_det = spj.n\_det      join (          -- четверть максимальной поставки каждого поставщика          select n\_post, max(kol\*ves) / 4 as quater          from spj          join p on p.n\_det = spj.n\_det          group by n\_post      ) temp      on spj.n\_post = temp.n\_post      where kol\*ves < temp.quater;      /////////////////////////////////////////////////////////////////////////SQL      exec sql open my\_cursor3;      exec sql fetch from my\_cursor3 into :n\_post, :ves, :quater\_of\_weight;        if (sqlca.sqlcode < 0)      {          printf("Возникла ошибка. sqlca.sqlcode = %d\n", sqlca.sqlcode);          exec sql rollback work; // отмена транзакции          return -1;      }      else if (sqlca.sqlcode == 100)      {          printf("Данных нет.\n");          exec sql commit work; // конец транзакции          return 0;      }      else if (sqlca.sqlcode == 0)      {          printf("n\_post\tves\tquater\n");          printf("%s\t%d\t%d\n", n\_post, ves, quater\_of\_weight);          while (sqlca.sqlcode == 0) // основной цикл          {              exec sql fetch from my\_cursor3 into :n\_post, :ves, :quater\_of\_weight;              if (sqlca.sqlcode != 100)                  printf("%s\t%d\t%d\n", n\_post, ves, quater\_of\_weight);          }          exec sql commit work; // конец транзакции          return 0;      }  }  // 4. Выбрать изделия, для которых не поставлялось ни одной из деталей,  // поставляемых поставщиком S4.  int problem4()  {      printf("\n\nЗадание 4\n");      printf("Изделия, для которых не поставлялось ни одной из деталей, поставляемых поставщиком S4:\n");      int status;      exec sql begin declare section;      char n\_izd[6];      exec sql end declare section;      /////////////////////////////////////////////////////////////////////////SQL      exec sql begin work; // начало транзакции      exec sql declare my\_cursor4 cursor for      -- изделия, для которых не поставлялось ни одной из деталей,      -- поставляемых поставщиком S4      (          -- все изделия          select n\_izd          from j      )      except      (          -- изделия с деталями от поставщика S4          select distinct n\_izd          from spj          where n\_det in(              select n\_det              from spj              where n\_post = 'S4'          )      )      order by 1;      /////////////////////////////////////////////////////////////////////////SQL      exec sql open my\_cursor4;      exec sql fetch from my\_cursor4 into :n\_izd;        if (sqlca.sqlcode < 0)      {          printf("Возникла ошибка. sqlca.sqlcode = %d\n", sqlca.sqlcode);          exec sql rollback work; // отмена транзакции          return -1;      }      else if (sqlca.sqlcode == 100)      {          printf("Данных нет.\n");          exec sql commit work; // конец транзакции          return 0;      }      else if (sqlca.sqlcode == 0)      {          printf("n\_izd\n");          printf("%s\n", n\_izd);          while (sqlca.sqlcode == 0) // основной цикл          {              exec sql fetch from my\_cursor4 into :n\_izd;              if (sqlca.sqlcode != 100)                  printf("%s\n", n\_izd);          }          exec sql commit work; // конец транзакции          return 0;      }  }  // 5. Выдать полную информацию о деталях, которые поставлялись  // ТОЛЬКО поставщиками с максимальным рейтингом.  int problem5()  {      printf("\n\nЗадание 5:\n");      printf("Детали, которые поставлялись только поставщиками с максимальным рейтингом:\n");      int status;      exec sql begin declare section;      int ves;      char n\_det[6], name[20], cvet[20], town[20];      exec sql end declare section;        /////////////////////////////////////////////////////////////////////////SQL      exec sql begin work; // начало транзакции      exec sql declare my\_cursor5 cursor for      -- полная информация о деталях от поставщиков только с максимальным рейтингом      select \*      from p      where n\_det in(          -- детали от поставщиков только с максимальным рейтингом          (              -- детали от поставщиков с максимальным рейтингом              select distinct n\_det              from spj              where n\_post in (                  select n\_post                  from s                  where reiting = (                      select max(reiting)                      from s)              )              order by 1          )          except          (              -- детали от остальных поставщиков              select distinct n\_det              from spj              where n\_post not in (                  select n\_post                  from s                  where reiting = (                      select max(reiting)                      from s)              )              order by 1          )      );      /////////////////////////////////////////////////////////////////////////SQL      exec sql open my\_cursor5;      exec sql fetch from my\_cursor5 into :n\_det, :name, :cvet, :ves, :town;      if (sqlca.sqlcode < 0)      {          printf("Возникла ошибка. sqlca.sqlcode = %d\n", sqlca.sqlcode);          exec sql rollback work; // отмена транзакции          return -1;      }      else if (sqlca.sqlcode == 100)      {          printf("Данных нет.\n");          exec sql commit work; // конец транзакции          return 0;      }      else if (sqlca.sqlcode == 0)      {          printf("n\_det   name\t\t  cvet         ves\t\ttown\n");          printf("%s\t%s%s%d\t\t%s\n", n\_det, name, cvet, ves, town);          while (sqlca.sqlcode == 0) // основной цикл          {              exec sql fetch from my\_cursor5 into :n\_det, :name, :cvet, :ves, :town;              if (sqlca.sqlcode != 100)                  printf("%s\t%s%s%d\t\t%s\n", n\_det, name, cvet, ves, town);          }          exec sql commit work; // конец транзакции          return 0;      }  }    int main()  {      // присоединяемся к БД students на fpm2.ami.nstu.ru      exec sql connect to students @fpm2.ami.nstu.ru user "pmi-b6306" using "Jemlift5";      if (sqlca.sqlcode < 0)      {          printf("Error: can't sign in\n");          return -1;      }      // устанавливаем путь к схеме      exec sql set search\_path to pmib6306;      if (problem1() < 0)      {          printf("Ошибка в решении задания 1\n");          return -1;      }      if (problem2() < 0)      {          printf("Ошибка в решении задания 2\n");          return -1;      }      if (problem3() < 0)      {          printf("Ошибка в решении задания 3\n");          return -1;      }      if (problem4() < 0)      {          printf("Ошибка в решении задания 4\n");          return -1;      }      if (problem5() < 0)      {          printf("Ошибка в решении задания 5\n");          return -1;      }      exec sql disconnect;      return 0;  } |

**Результат работы программы**

|  |
| --- |
| Задание 1  Вывести число деталей, которые поставлялись поставщиками, имеющими поставки с объемом от 600 до 700 деталей  3  Задание 2  Поменять местами цвета самой тяжелой и самой легкой детали:  Замена выполнена успешно.  Заменено 3 строк  Задание 3  Поставщики, имеющие поставки, вес которых составляет менее четверти наибольшего веса поставки этого поставщика:  n\_post ves quater  S5 1200 2800  S2 1200 3400  S5 1700 2800  Задание 4  Изделия, для которых не поставлялось ни одной из деталей, поставляемых поставщиком S4:  n\_izd  J1  J5  J6  Задание 5:  Детали, которые поставлялись только поставщиками с максимальным рейтингом:  n\_det name cvet ves town  P2 Болт Зеленый 17 Париж  P4 Винт Красный 14 Лондон |

**Дополнительно:**

Проверим выполнение второго задания, при ограничении в таблице p:

n\_det = 'P6' and cvet <> 'Голубой' or n\_det <>'P6'

|  |
| --- |
| Задание 2  Поменять местами цвета самой тяжелой и самой легкой детали:  Возникла ошибка. sqlca.sqlcode = -400  Сообщение ISAM: h  Ошибка в решении задания 2 |