минобрнауки россии

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт информационных технологий |
| наименование института (факультета) |
| Кафедра математического и программного обеспечения ЭВМ |
| наименование кафедры |
| Базы данных |
| наименование дисциплины в соответствии с учебным планом |

ЛАБОРАТОРНАЯ РАБОТА №6

SQL. Язык манипулирования данными (DML)

|  |  |
| --- | --- |
| Исполнитель |  |
| студент | 1ПИб-02-2оп-22 |
|  | группа |
|  | Зернов В. А. |
|  | Фамилия, имя, отчество |
| Руководитель | Селяничев О.Л. |
|  | Ф.И.О. преподавателя |
| Оценка |  |
| Подпись |  |

2024 год

1. Создайте БД ПАНСИОН (см. электронный учебник Book1.sql) .
2. Создайте в ней таблицы БЛЮДА и ПОСТАВЩИКИ. Задание выполните средствами SQL.

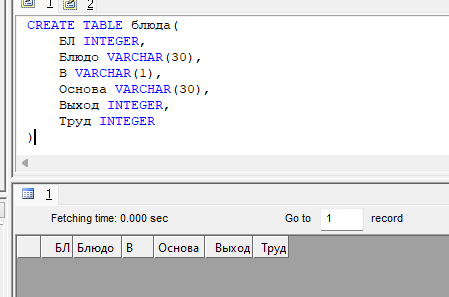


Рис.1. Создание таблицы «блюда».

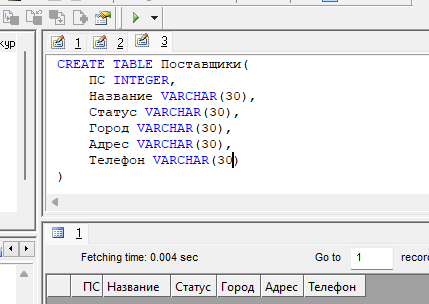


Рис.2. Создание таблицы «Поставщики».

1. Занесите в них записи. (Для ускорения ввода данных можно скопировать их непосредственно в Access’овскую таблицу через clipboard.)

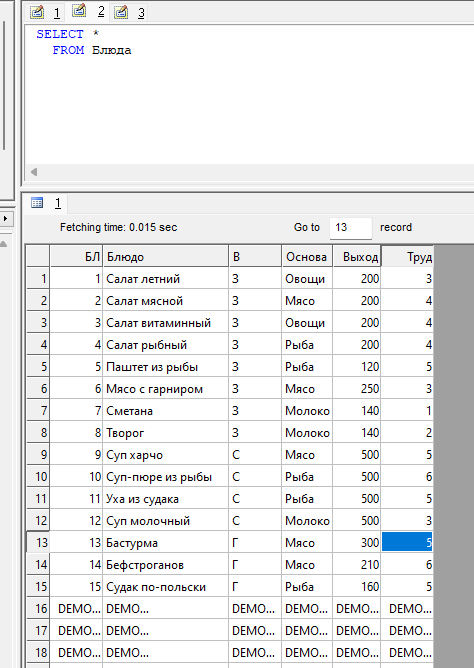


Рис.3. Заполнение таблицы «блюда».

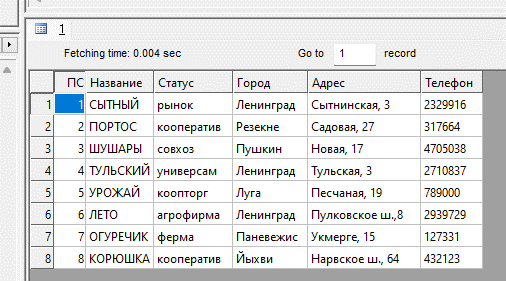


Рис.4. Заполнение таблицы «Поставщики».

1. Остальные таблицы создавайте по мере необходимости – тоже средствами SQL. И их наполняйте данными из файлов учебника.
2. Ознакомьтесь с теорией в главах 2 и 3 (список необходимых для изучения пунктов – ниже). Каждый встретившийся запрос проверьте практически -воспроизведите запрос в программе DTM SQL Editor. Изменяйте всякий запрос по-своему так, чтобы и он давал правильное решение (естественно, другое).
3. В отчет по ЛР должны попасть задания и ответы на них – из учебника и свое. В отчет занесите это в форме, например:

2.2.3. Выборка вычисляемых значений

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Получить значение калорийности всех продуктов (учесть, что при окислении 1 г углеводов или белков в организме освобождается в среднем 4.1 ккал, а при окислении 1 г жиров - 9.3 ккал). |  |
| Решение | SELECT Продукт, ((Белки+Углев)\*4.1+Жиры\*9.3) FROM Продукты; |  |

**Пункты учебника Book1.sql, которые должны быть рассмотрены.**

Глава 2. Запросы с использованием единственной таблицы

2.1. О предложении SELECT

2.2. Выборка без использования фразы WHERE

2.2.1. Простая выборка

2.2.2. Исключение дубликатов

2.2.3. Выборка вычисляемых значений

2.3. Выборка c использованием фразы WHERE

2.3.1. Использование операторов сравнения

2.3.2. Использование BETWEEN

2.3.3. Использование IN

2.3.4. Использование LIKE

2.3.5. Вовлечение неопределенного значения

2.4. Выборка с упорядочением

2.5. Агрегирование данных

2.5.1. SQL-функции

2.5.2. Функции без использования фразы GROUP BY

2.5.3. Фраза GROUP BY

2.5.4. Использование фразы HAVING

Глава 3. Запросы с использованием нескольких таблиц

3.1. О средствах одновременной работы с множеством таблиц

3.2. Запросы, использующие соединения

3.2.1. Декартово произведение таблиц

3.2.2. Эквисоединение таблиц

3.2.3. Естественное соединение таблиц

3.2.4. Композиция таблиц

3.2.5. Тета-соединение таблиц

3.2.6. Соединение таблиц с дополнительным условием

3.2.7. Соединение таблицы со своей копией

**Составление SQL-запросов**

2.2.1. Простая выборка

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Запрос выдать название, статус и адрес поставщиков | Создать запрос на получение основы и выхода блюд |
| Решение | SELECT Название, Статус, Адрес  FROM Поставщики; | SELECT Основа, Выход  FROM Блюда |

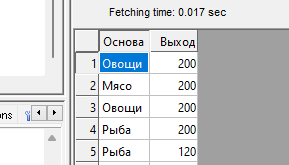


Рис.5. Результат SQL-запроса.

2.2.2. Исключение дубликатов

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Запрос на получение уникальных основ блюд | Запрос на получение уникальных статусов поставщиков |
| Решение | SELECT DISTINCT Основа  FROM Блюда; | SELECT Distinct Статус  FROM Поставщики |

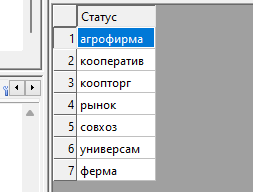


Рис.6. Результат SQL-запроса.

2.2.3. Выборка вычисляемых значений

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Получить значение калорийности всех продуктов (учесть, что при окислении 1 г углеводов или белков в организме освобождается в среднем 4.1 ккал, а при окислении 1 г жиров - 9.3 ккал). | Получить с суммарную стоимость продуктов в наличии |
| Решение | SELECT Продукт, ((Белки+Углев)\*4.1+Жиры\*9.3) FROM Продукты; | SELECT ПР, (К\_во\*Стоим) AS Сумма  FROM Наличие |

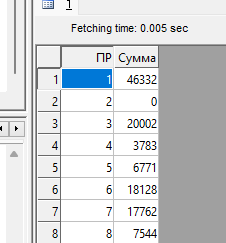


Рис.7. Результат SQL-запроса.

2.3.1. Использование операторов сравнения

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Получение перечня продуктов, практически не содержащих углеводов | Получить продукты, запасы которых превышают 80 ед. |
| Решение | SELECT Продукт, Белки, Жиры, Углев, K, Ca, Na, B2, PP, C  FROM Продукты  WHERE Углев = 0; | SELECT ПР, К\_во  FROM Наличие  Where К\_во > 80 |

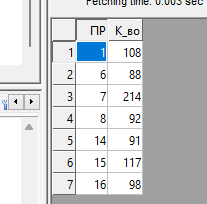


Рис.8. Результат SQL-запроса.

2.3.2. Использование BETWEEN

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | выдать перечень продуктов, в которых значение содержания белка находится в диапазоне от 10 до 50 | Выдать перечень блюд, чей выход от 200 до 550 |
| Решение | SELECT Продукт, Белки  FROM Продукты  WHERE Белки BETWEEN 10 AND 50; | SELECT Блюдо, Выход  FROM Блюда Where Выход BETWEEN 200 AND 550 |

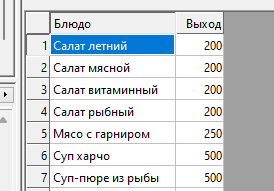


Рис.9. Результат SQL-запроса.

2.3. 3. Использование IN

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выдать сведения о блюдах на основе яиц, крупы и овощей | Выдать поставщиков, чей город – Ленинград, Пушкин или Луга |
| Решение | SELECT \*  FROM Блюда  WHERE Основа IN (Яйца Крупа Овощи); | SELECT \*  FROM Поставщики  WHERE Город IN ('Луга', 'Пушкин', 'Ленинград') |

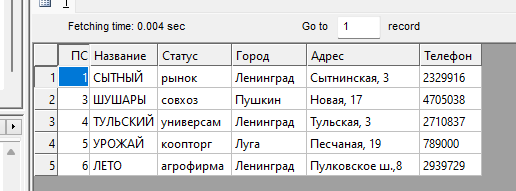


Рис.10. Результат SQL-запроса.

2.3.4. Использование LIKE

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выдать перечень салатов | Выдать перечень супов |
| Решение | SELECT Блюдо  FROM Блюда  WHERE Блюдо LIKE 'Салат%'; | SELECT Блюдо  FROM Блюда  WHERE Блюдо LIKE 'Суп%'; |

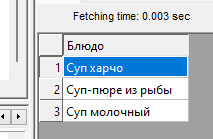


Рис.11. Результат SQL-запроса.

2. 3.5. Вовлечение неопределенного значения

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выявление названий продуктов, отсутствующих в кладовой | Выявление продуктов, цена поставки которых не указана |
| Решение | SELECT DISTINCT ПР  FROM Наличие  WHERE К\_во IS NULL; | SELECT ПС, ПР  FROM Поставки  WHERE Цена IS NULL; |

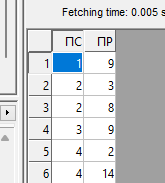


Рис.12. Результат SQL-запроса.

2.4. Выборка с упорядочением

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выдать перечень продуктов и содержание в них основных веществ в порядке убывания содержания белка | Выдать перечень блюд в порядке возрастания их выхода |
| Решение | SELECT Продукт, Белки,  Жиры, Углев  FROM Продукты  ORDER BY Белки DESC; | SELECT \*  FROM Блюда  ORDER BY Выход ASC; |

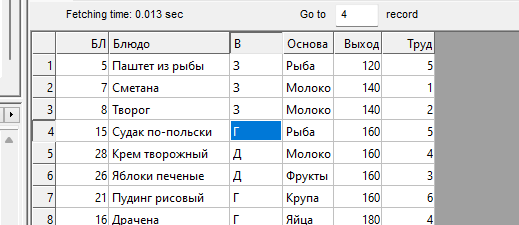


Рис.13. Результат SQL-запроса.

2.5.2. Функции без использования фразы GROUP BY

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выдать данные о массе лука (ПР=10), проданного поставщиками, и указать количество этих поставщиков | Выдать среднюю цену одной поставки и общее число поставленных продуктов. |
| Решение | SELECT SUM(К\_во),COUNT(К\_во)  FROM Поставки  WHERE ПР = 10; | SELECT AVG(Цена) AS Ср\_цена, SUM(К\_во) AS Всего\_поставлено  FROM Поставки |

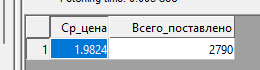


Рис.14. Результат SQL-запроса.

2.5.3. Фраза GROUP BY

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Вычислить общую массу каждого из продуктов, поставляемых в настоящее время поставщиками. | Вычислить Среднюю средний выход блюд из каждого вида основы |
| Решение | SELECT ПР, SUM(К\_во)  FROM Поставки  GROUP BY ПР; | SELECT Основа, AVG(Выход) AS Ср\_выход  FROM Блюда  GROUP BY Основа |

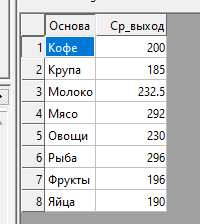


Рис.15. Результат SQL-запроса.

2.5. 4. Использование фразы HAVING

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выдать коды продуктов, поставляемых более чем двумя поставщиками | Выдать основы, средний выход которых превышает 200 |
| Решение | SELECT  FROM Поставки  GROUP BY ПС  HAVING COUNT(\*) > 2; | SELECT Основа, AVG(Выход) AS Ср\_выход  FROM блюда  GROUP BY Основа  HAVING AVG(Выход) > 200 |

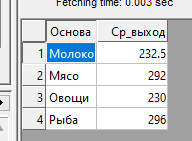


Рис.16. Результат SQL-запроса.

3.2.1. Декартово произведение таблиц

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Получить декартово произведение Вид\_блюд и Трапезы | Получить декартово произведение Поставки и Поставщики |
| Решение | SELECT Вид\_блюд.\*, Трапезы.\*  FROM Вид\_блюд, Трапезы; | SELECT Поставки.\*, Поставщики.\*  FROM Поставки, Поставщики |

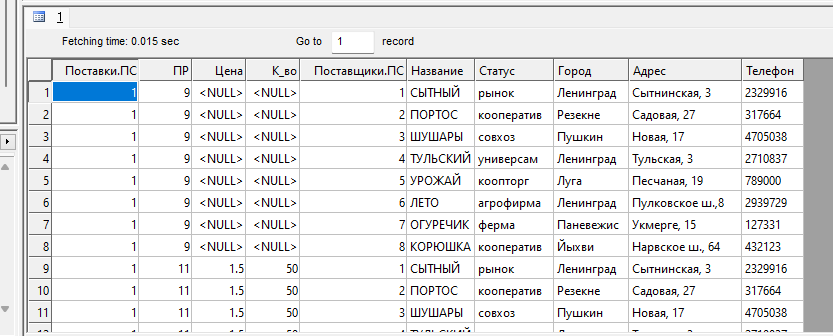


Рис.17. Результат SQL-запроса.

Эквисоединение таблиц - набор, содержащий пары строк двух таблиц, для которых выполняется условие соединения, представляющее собой равенство значений пар соответствующих столбцов.

3.2.2. Эквисоединение таблиц

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Получить эквисоединение таблиц Меню, Трапезы, Вид\_блюд и Блюда | Получить эквисоединение таблиц Поставки, Поставщики, Продукты |
| Решение | SELECT Меню.\*, Трапезы.\*, Вид\_блюд.\*, Блюда.\*  FROM Меню, Трапезы, Вид\_блюд, Блюда  WHERE Меню.Т = Трапезы.Т  AND Меню.В = Вид\_блюд.В  AND Меню.БЛ = Блюда.БЛ; | SELECT Продукты.\*, Поставщики.\*, Поставки.\*  FROM Продукты, Поставщики, Поставки  WHERE Продукты.ПР = Поставки.ПР  AND Поставки.ПС = Поставщики.ПС |

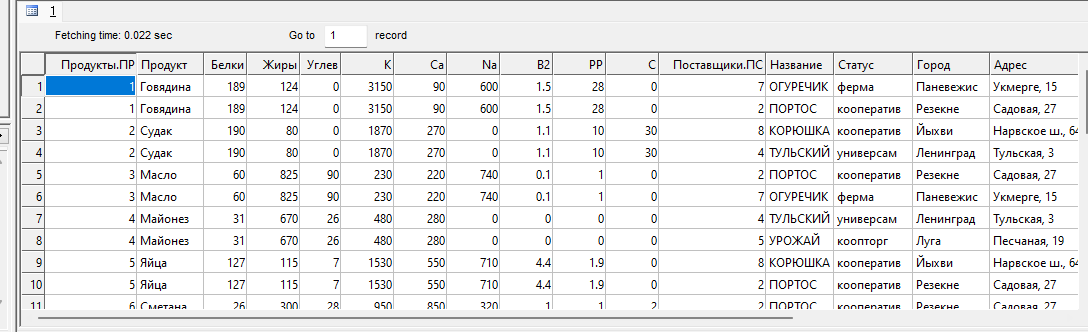


Рис.18. Результат SQL-запроса.

3.2.3. Естественное соединение таблиц

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Исключенить из предыдущего запроса все дубликаты столбцов, по которым проводится соединение таблиц | Исключенить из предыдущего запроса все дубликаты столбцов, по которым проводится соединение таблиц |
| Решение | SELECT Т, В, БЛ, Трапеза, Вид, Блюдо, Основа,  Выход, Труд  FROM Меню, Трапезы, Вид\_блюд, Блюда  WHERE Меню.Т = Трапезы.Т  AND Меню.В = Вид\_блюд.В  AND Меню.БЛ = Блюда.БЛ; | . SELECT Продукты.ПР, Поставщики.ПС, Продукты.Продукт, Продукты.Белки, Продукты.Жиры, Поставщики.Название, Поставщики.Статус, Поставщики.Город, Поставщики.Адрес, Поставки.Цена, Поставки.К\_во  FROM Продукты, Поставщики, Поставки  WHERE Продукты.ПР = Поставки.ПР  AND Поставки.ПС = Поставщики.ПС |

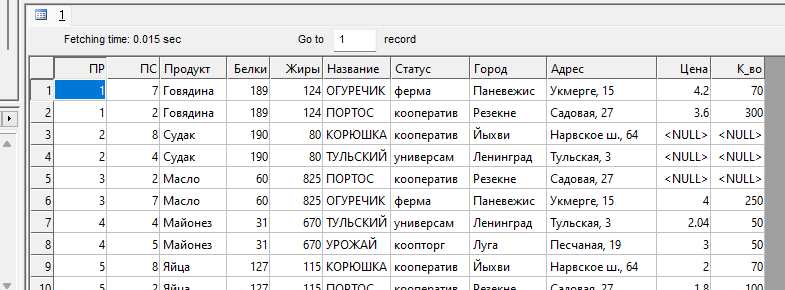


Рис.19. Результат SQL-запроса.

3.2.4. Композиция таблиц

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Исключенить из предыдущего запроса все столбцы, по которым проводится соединение таблиц | Получить Продукты, Белки в них, Название поставщика, Статус поставщика, Цену поставки и кол-во единиц за поставку. |
| Решение | SELECT Трапеза, Вид, Блюдо, Основа, Выход, Труд  FROM Меню, Трапезы, Вид\_блюд, Блюда  WHERE Меню.Т = Трапезы.Т  AND Меню.В = Вид\_блюд.В  AND Меню.БЛ = Блюда.БЛ; | SELECT Продукты.Продукт, Продукты.Белки, Поставщики.Название, Поставщики.Статус, Поставки.Цена, Поставки.К\_во  FROM Продукты, Поставщики, Поставки  WHERE Продукты.ПР = Поставки.ПР  AND Поставки.ПС = Поставщики.ПС |

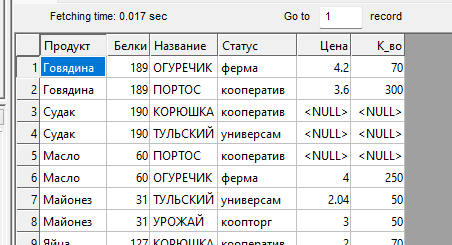


Рис.20. Результат SQL-запроса.

Тета-соединение — это соединение, которое связывает таблицы на основе отношений, отличных от равенства между двумя столбцами.

3.2.5. Тета-соединение таблиц

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Выбрать из полученного в п.3.2.1 декартова произведения таблиц Вид\_блюд и Трапезы лишь те строки, в которых значение трапезы "меньше" (по алфавиту) значения вида блюда | Выбрать из декартова произведения таблиц Поставки и Наличие строки, в которых значение количества наличия > зачения цены поставок |
| Решение | SELECT Вид\_блюд.\*, Трапезы.\*  FROM Вид\_блюд, Трапезы  WHERE Вид > Трапеза; | SELECT Поставки.\*, Наличие.\*  FROM Поставки, Наличие  WHERE Наличие.К\_во > Поставки.Цена |

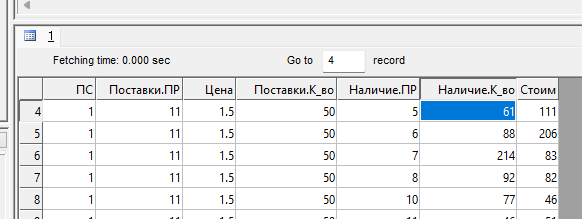


Рис.21. Результат SQL-запроса.

3.2.6. Соединение таблиц с дополнительным условием

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | Получение перечня блюд, предлагаемых в меню на завтрак | Получение названий поставщиков, средняя цена закупки за шт. у которых > 2 |
| Решение | SELECT Вид, Блюдо, Основа, Выход, 'Номер -', БЛ  FROM Меню, Трапезы, Вид\_блюд, Блюда  WHERE Меню.Т = Трапезы.Т  AND Меню.В = Вид\_блюд.В  AND Меню.БЛ = Блюда.БЛ  AND Трапеза = ’Завтрак’; | SELECT Поставщики.Название, AVG(Поставки.Цена) AS Ср\_цена  FROM Поставщики, Поставки  WHERE Поставщики.ПС = Поставки.ПС  GROUP BY Поставщики.Название  HAVING AVG(Поставки.Цена) > 2 |

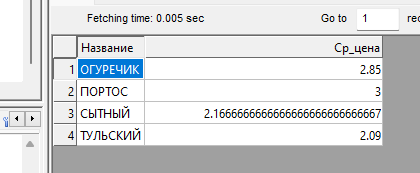


Рис.22. Результат SQL-запроса.

3.2. 7. Соединение таблицы со своей копией

|  |  |  |
| --- | --- | --- |
|  | Из учебника | Свое |
| Задание | вывод таких пар блюд таблицы Блюда, в которых совпадает основа, а название первого блюда пары меньше (по алфавиту), чем номер второго блюда пары. | Вывод пар, где название первого блюда больше названия второго по алфавиту, а название основы - меньше |
| Решение | SELECT Блюдо, Копия.Блюдо, Основа  FROM Блюда, Блюда Копия  WHERE Основа = Копия.Основа  AND Блюдо < Копия.Блюдо; | SELECT Оригинал.Основа, Копия.Основа, Оригинал.Выход  FROM Блюда Оригинал, Блюда Копия  WHERE Оригинал.Блюдо > Копия.Блюдо  AND Оригинал.Основа < Копия.Основа; |

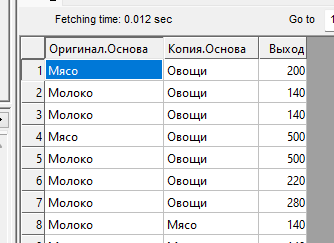


Рис.23. Результат SQL-запроса.

Заключение.

В ходе выполнения лабораторной работы были протестированы написанные в учебнике SQL-запросы (выяснилось, что далеко не все из них могут пройти проверку на работоспособность), были составлены собственные задания, созданы SQL-запросы для их решения. По мере работы, с помощью команды CREATE TABLE создавались новые таблицы, а данные в них заносились из приложенного к ЛР учебника.