Создаем таблицы в БД:

create table Блюда(

[БЛ] Long not null,

[Блюдо] VarChar(255) null,

[В] VarChar(1) null,

[Основа] VarChar(255) null,

[Выход] Double null,

[Труд] Long null

)

create table Поставщики(

[ПС] Long not null,

[Название] VarChar(255) null,

[Статус] VarChar(255) null,

[Город] VarChar(255) null,

[Адрес] VarChar(255) null,

[Телефон] Long null

)

create table Рецепты(

[БЛ] Long not null,

[Рецепт] VarChar(255) null

)

create table Поставки(

[ПС] Long not null,

[ПР] Long not null,

[Цена] float,

[К\_во] Long null

)

create table Наличие(

[ПР] Long not null,

[К\_во] Long null,

[Стоим] float

)

create table Вид\_блюд(

[В] VarChar(1) null,

[Вид] VarChar(255) null

)

create table Трапезы(

[Т] Long not null,

[Трапеза] VarChar(255) null

)

create table Меню(

[Т] Long not null,

[В] VarChar(1) null,

[БЛ] Long not null

)

create table Выбор(

[СМ] Long default 2 not null,

[Т] Long not null,

[В] VarChar(1) null,

[БЛ] Long not null

)

create table Выбрано(

[СМ] Long default 1 not null,

[Т] Long not null,

[БЛ] Long not null

)

create table Состав(

[БЛ] Long null,

[ПР] Long null,

[Вес] Long null

)

create table Продукты(

[ПР] Long null,

[Продукт] VarChar(255) null,

[Белки] float,

[Жиры] float,

[Углев] float,

[K] Long null,

[Ca] Long null,

[Na] Long null,

[B2] float,

[PP] float,

[C] Long null

)

|  |  |
| --- | --- |
| Из учебника | Свое |
| выдать название, статус и адрес поставщиков |  |
| SELECT Название, Статус, Адрес  FROM Поставщики; | SELECT Название, Статус, Адрес  FROM Поставщики; |
| 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). |  |
| SELECT Продукт, ((Белки+Углев)\*4.1+Жиры\*9.3) FROM Продукты; | SELECT Продукт, ((Белки+Углев)\*4.1+Жиры\*9.3) FROM Продукты; |
| получение перечня продуктов, практически не содержащих углеводов и натрия |  |
| SELECT Продукт, Белки, Жиры, Углев, K, Ca, Na, B2, PP, C  FROM Продукты  WHERE Углев = 0 AND Na = 0; | SELECT Продукт, Белки, Жиры, Углев, K, Ca, Na, B2, PP, C  FROM Продукты  WHERE Углев = 0 AND Na = 0; |
| выдать перечень продуктов, в которых значение содержания белка находится в диапазоне от 10 до 50 |  |
| SELECT Продукт, Белки  FROM Продукты  WHERE Белки BETWEEN 10 AND 50; | SELECT Продукт, Белки  FROM Продукты  WHERE Белки BETWEEN 10 AND 50; |
| узнать, какие изменения минимальных окладов производились в 1993/94 учебном году, то можно выдать запрос |  |
| SELECT Начало, Миноклад  FROM Миноклады  WHERE Начало BETWEEN '1-9-1993' AND '31-8-1994' | SELECT Начало, Миноклад  FROM Миноклады  WHERE Начало BETWEEN '1-9-1993' AND '31-8-1994' |
| Выдать сведения о блюдах на основе яиц, крупы и овощей |  |
| SELECT \*  FROM Блюда  WHERE Основа IN (Яйца Крупа Овощи); | SELECT \*  FROM Блюда  WHERE Основа IN(Яйца Крупа Овощи); |
| Выдать перечень салатов |  |
| SELECT Блюдо  FROM Блюда  WHERE Блюдо LIKE 'Салат%'; | SELECT Блюдо  FROM Блюда  WHERE Блюдо LIKE 'Салат%'; |
| выявление названий продуктов, отсутствующих в кладовой |  |
| SELECT DISTINCT ПР  FROM Наличие  WHERE К\_во IS NULL; | SELECT DISTINCT ПР  FROM Наличие  WHERE К\_во IS NULL; |
| выдать перечень продуктов и содержание в них основных веществ в порядке убывания содержания белка |  |
| SELECT Продукт, Белки,  Жиры, Углев  FROM Продукты  ORDER BY Белки DESC; | SELECT Продукт, Белки,  Жиры, Углев  FROM Продукты  ORDER BY Белки DESC; |
| выдать данные о массе лука (ПР=10), проданного поставщиками, и указать количество этих поставщиков |  |
| SELECT SUM(К\_во),COUNT(К\_во)  FROM Поставки  WHERE ПР = 10; | SELECT SUM(К\_во),COUNT(К\_во)  FROM Поставки  WHERE ПР = 10; |
| вычислить общую массу каждого из продуктов, поставляемых в настоящее время поставщиками |  |
| SELECT ПР, SUM(К\_во)  FROM Поставки  GROUP BY ПР; | SELECT ПР, SUM(К\_во)  FROM Поставки  GROUP BY ПР; |
| выдать коды продуктов, поставляемых более чем двумя поставщиками |  |
| |  | | --- | | SELECT  FROM Поставки  GROUP BY ПС  HAVING COUNT(\*) > 2; | | | |  | | --- | | SELECT  FROM Поставки  GROUP BY ПС  HAVING COUNT(\*) > 2; | | |
| получения декартова произведения Вид\_блюд и Трапезы |  |
| SELECT Вид\_блюд.\*, Трапезы.\*  FROM Вид\_блюд, Трапезы; | SELECT Вид\_блюд.\*, Трапезы.\*  FROM Вид\_блюд, Трапезы; |
| из декартова произведения убрать ненужные строки и столбцы |  |
| 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 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; |
| исключение дубликатов столбцов |  |
| 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 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; |
| исключение всех столбцов, по которым проводится соединение таблиц |  |
| 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 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; |
| выбрать из декартова произведения таблиц Вид\_блюд и Трапезы лишь те строки, в которых значение трапезы "меньше" (по алфавиту) значения вида блюда |  |
| SELECT Вид\_блюд.\*, Трапезы.\*  FROM Вид\_блюд, Трапезы  WHERE Вид > Трапеза; | SELECT Вид\_блюд.\*, Трапезы.\*  FROM Вид\_блюд, Трапезы  WHERE Вид > Трапеза; |
| вывод таких пар блюд таблицы Блюда, в которых совпадает основа, а название первого блюда пары меньше (по алфавиту), чем номер второго блюда пары |  |
| SELECT Блюдо, Копия.Блюдо, Основа  FROM Блюда, Блюда Копия  WHERE Основа = Копия.Основа  AND Блюдо < Копия.Блюдо; | SELECT Блюдо, Копия.Блюдо, Основа  FROM Блюда, Блюда Копия  WHERE Основа = Копия.Основа  AND Блюдо < Копия.Блюдо; |