Таблица 1 – Актуализация данных в БД

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Бронирование(booking)** | | | | | | | | | |
| **Код брони** | **Номер брони** | | **Дата создания** | **Время создания** | **Дата посещения** | **Время посещения** | **Кол-во гостей** | **Код пользователя** | **Код стола** |
| **Booking\_kod** | **Booking\_number** | | **Book\_making\_date** | **Book\_making\_time** | **Book\_plan\_date** | **Book\_plan\_time** | **Book\_guests\_count** | **V\_kod** | **Wood\_table\_kod** |
| 1 | № БР/24/0000000001 | | 01.01.2024 | 16:01:10 | 02.02.2024 | 12:00 | 2 | зп3 | ч2 |
| 2 | № БР/24/0000000002 | | 20.01.2024 | 09:50:21 | 19.02.2024 | 16:30 | 8 | зп1 | ОБ1 |
| 3 | № БР/24/0000000002 | | 20.01.2024 | 09:50:21 | 19.02.2024 | 16:30 | 8 | зп1 | ОБ2 |
| 4 | № БР/24/0000000003 | | 20.01.2024 | 17:09:25 | 10.02.2024 | 12:45 | 1 | зп3 | ОБ3 |
| 5 | № БР/24/0000000004 | | 02.02.2024 | 18:00:35 | 06.02.2024 | 17:20 | 2 | зп1 | ОБ3 |
| 6 | № БР/24/0000000005 | | 10.02.2024 | 15:43:59 | 11.02.2024 | 13:00 | 4 | зп1 | Ч1 |
| Запрос на добавление данных | | | | | | | | | |
| call booking\_insert('01.01.2024','16:01:10','02.02.2024','12:00',2,9,'зп3');  call booking\_insert('20.01.2024','09:50:21','19.02.2024','16:30',8,1,'зп1');  call booking\_insert('20.01.2024','09:50:21','19.02.2024','16:30',8,2,'зп1');  call booking\_insert('20.01.2024','17:09:25','10.02.2024','12:45',1,3,'зп3');  call booking\_insert('02.02.2024','18:00:35','06.02.2024','17:20',2,3,'зп1');  call booking\_insert('10.02.2024','15:43:59','11.02.2024','13:00',4,8,'зп1'); | | | | | | | | | |
|  | | | | | | | | | |
| Результат | | | | | | | | | |
|  | | | | | | | | | |
| **Состав блюд бронирования(planning\_dishes)** | | | | | | | | | |
| **Код состава брони** | | **Номер брони** | | **Код блюда** | **Время подачи** | **Кол-во порций** | | | |
| **Planning\_dishes\_kod** | | **Book\_kod** | | **Dish\_kod** | **Planning\_dishes\_time** | **Dishes\_count** | | | |
| 1 | | №БР/24/0000000001 | | 1 | 12:00 | 4 | | | |
| 2 | | №БР/24/0000000001 | | 4 | 13:00 | 2 | | | |
| 3 | | №БР/24/0000000003 | | 3 | 12:45 | 1 | | | |
| 4 | | №БР/24/0000000003 | | 5 | 12:45 | 2 | | | |
| 5 | | №БР/24/0000000004 | | 2 | 17:40 | 2 | | | |
| Запрос на добавление данных | | | | | | | | | |
| call planning\_dishes\_insert ('12:00', 1, '№БР/24/0000000001', 4);  call planning\_dishes\_insert ('13:00', 4, '№БР/24/0000000001', 2);  call planning\_dishes\_insert ('12:45', 3, '№БР/24/0000000003', 1);  call planning\_dishes\_insert ('12:45', 5, '№БР/24/0000000003', 2);  call planning\_dishes\_insert ('17:40', 2, '№БР/24/0000000004', 2); | | | | | | | | | |
| Результат | | | | | | | | | |
| ‘ | | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Блюда (dishes) | | | | |
| **Код блюда** | **Имя блюда** | **Стоимость блюда** | **Вес блюда** | **Изображение блюда** |
| **Dish\_kod** | **Dish\_name** | **Dish\_cost** | **Dish\_weight** | **Dish\_picture** |
| **Serial** | **Varchar(50** | **Int** | **Decimal(6,2)** | **Varchar(50)** |
| 1 | Филе порося | 950 | 350 | 1.png |
| 2 | Суп мечты | 750 | 200 | 2.png |
| 3 | Картофель по своему | 120 | 120 | 3.png |
| 4 | Мясная тарелка | 1670 | 1200 | 4.png |
| 5 | Гарнир офощной | 950 | 350 | 5.png |
| Запрос на изменение данных | | | | |
| call dishes\_update('Филе порося', 950, 350.00, '1.png',1);  call dishes\_update('Суп мечты', 750, 200.00, '2.png',2);  call dishes\_update('Картофель по своему', 120, 120.00, '3.png',3);  call dishes\_update('Мясная тарелка', 1670, 1200.00, '4.png',4);  call dishes\_update('Гарнир офощной', 950, 350.00, '5.png',5); | | | | |
| результат | | | | |
|  | | | | |

1. Использование фильтров;

Таблица 2 – Отработка фильтров

| Название фильтра | Решение |
| --- | --- |
| Равно | Запрос |
| Вывести все номер заказа, дата создания заказа, номер стола и ФИО сотрудника, где логин of\_SemenovKN. |
| select  order\_name as "Номер заказа",  order\_date as "Дата заказа",  concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic) as "ФИО сотрудника"  from orders  inner join public.employee  on employee.employee\_login = 'of\_SemenovKN'; |
| Результат |
|  |
| Меньше | Запрос |
| Вывести номера заказов, названия блюд, количество блюд в заказе, где количество строго меньше 3. |
| select  order\_name as "Номер заказа",  order\_dishcount as "Количество блюд",  dish\_name as "Название блюда"  from orders  inner join public.dishes  on dishes.dish\_kod = orders.dish\_kod  where order\_dishcount < 3; |
| Результат |
|  |
| Похоже на | Запрос |
| Вывести ФИО клиента, логин и пароль, где серия паспорта имеет значение 45 |
| select  concat(rv\_surname,' ', rv\_name,' ', rv\_patronymic) as "ФИО",  rv\_login as "Логин",  rv\_password as "пароль"  from public.registered\_visitors  where rv\_pasports similar to '%45%'; |
| Результат |
|  |
| Один из коллекции | Запрос |
| Вывести название блюда, цену блюда, ингредиенты, где есть ингредиенты: лук, перец, морковь |
| select  dish\_name as "Имя блюда",  dish\_weight as "Вес блюда",  dish\_cost as "Стоимость",  ingredient\_name as "Название ингредиента"  from dishes  inner join sostav  on sostav.dish\_kod = dishes.dish\_kod  inner join public.ingredientes  on ingredientes.ingredient\_kod = sostav.ingredient\_kod  where ingredient\_name in ('Лук','Морковь','Перец'); |
| Результат |
|  |
| В диапазоне | Запрос |
| Вывести название блюда, вес и цену блюда, где вес в диапазоне от 100 до 500 гр |
| select  dish\_name as "Имя блюда",  dish\_weight as "Вес блюда",  dish\_cost as "Стоимость"  from dishes  where dish\_weight > 100 and dish\_weight < 500; |
| Результат |
|  |
| Два и более фильтра | Запрос |
| Вывести номер заказа, ФИО сотрудника, название и количество позиции заказа, где сдача не ровна 0, количество позиций в заказе строго больше 1, сотрудник of\_SemenovKN.. |
| select  checkues.order\_name as "Номер заказа",  order\_date as "Дата заказа",  order\_dishcount as "Количество порций",  dish\_name as "Название ингредиента",  concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic) as "ФИО сотрудника"  from orders  inner join public.dishes  on dishes.dish\_kod = orders.dish\_kod  inner join public.employee  on employee.employee\_kod = orders.employee\_kod  inner join checkues  on checkues.order\_name = orders.order\_name  where checkue\_changesum !=0 and order\_dishcount > 0 and employee\_login = 'of\_SemenovKN'; |
| Результат |
|  |
| Не равно | Запрос |
| Вывести названия блюд, названия ингредиентов, вес и цену блюда, где нет ингредиента морковь. |
| select  dish\_name as "Имя блюда",  dish\_weight as "Вес блюда",  dish\_cost as "Стоимость",  ingredient\_name as "Название ингредиента"  from dishes  inner join sostav  on sostav.dish\_kod = dishes.dish\_kod  inner join public.ingredientes  on ingredientes.ingredient\_kod = sostav.ingredient\_kod  where ingredient\_name != 'Морковь'; |
| Результат |
|  |
| Больше | Запрос |
| Вывести номер сметы на поставку, название поставщика, название поставляемых ингредиентов, где вес больше или равен 50 кг. |
| select  smeta\_number as "Номер сметы",  provider\_name as "Поставщик",  ingredient\_name as "Название Ингредиента"  from smeta  inner join public.shipments  on shipments.smeta\_kod = smeta.smeta\_kod  inner join providers  on providers.provider\_kod = shipments.provider\_kod  inner join public.purchaseingredient  on purchaseingredient.shipment\_kod = shipments.shipment\_kod  inner join public.ingredientes  on ingredientes.ingredient\_kod = purchaseingredient.ingredient\_kod  where purchaseingredient.ingredient\_weight > 50; |
| Результат |
|  |
| Не похоже на | Запрос |
| Вывести ФИО клиента, номер банковской карты, где номер карты не содержит 77. |
| select  concat(rv\_surname,' ', rv\_name,' ', rv\_patronymic) as "ФИО",  rv\_card as "Карта"  from public.registered\_visitors  where rv\_card not similar to '%77%'; |
| Результат |
|  |
| Не один из коллекции | Запрос |
| Вывести номер заказа, ФИО клиента, ФИО сотрудника, номер стола, за исключением столов: ОБ3, ОБ4, ОБ5, Ч1. |
| select  order\_name as "Номер заказа",  wood\_table\_name as "Название стола",  concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic) as "ФИО сотрудника",  concat(rv\_surname,' ', rv\_name,' ', rv\_patronymic) as "ФИО"  from orders  inner join public.employee  on employee.employee\_kod = orders.employee\_kod  inner join public.individualorders  on individualorders.order\_kod = orders.order\_kod  inner join public.registered\_visitors  on registered\_visitors.rv\_kod = individualorders.v\_kod  inner join public.wood\_tables  on wood\_tables.table\_kod = orders.table\_kod  where wood\_table\_name not in ('ОБ3', 'ОБ4', 'ОБ5', 'ч1'); |
| Результат |
|  |
| Вне диапазона | Запрос |
| Вывести номер бронирования, ФИО клиента, дату посещения, где количество гостей вне диапазона от 2 до 5. |
| select  book\_number as "Номер бронирования",  book\_plan\_date as "Дата посещения",  concat(rv\_surname,' ', rv\_name,' ', rv\_patronymic) as "ФИО"  from booking  inner join public.registered\_visitors  on registered\_visitors.rv\_kod = booking.v\_kod  where book\_guests\_count > 5 or book\_guests\_count < 2; |
| Результат |
|  |
| Переключатель | Запрос |
| Вывести название блюда, ингредиенты блюда, вес блюда, если вес от 0 до 250, вывести «Лёгкое блюдо», от 251 до 750 «Средней тяжести», выше 751 «Тяжёлое» |
| select  dish\_name as "Имя блюда",  dish\_weight as "Вес блюда",  ingredient\_name as "Название ингредиента",  case  when dish\_weight > 0 and dish\_weight < 250 then 'Легкое блюдо'  when dish\_weight > 251 and dish\_weight < 759 then 'Средней тяжести'  when dish\_weight > 751 then 'Тяжелое'  end as "Тяжесть блюда"  from dishes  inner join sostav  on sostav.dish\_kod = dishes.dish\_kod  inner join public.ingredientes  on ingredientes.ingredient\_kod = sostav.ingredient\_kod; |
| Результат |
|  |

1. Использование встроенных функций;

Таблица 3 – Отработка функций

| Название функции | Решение |
| --- | --- |
| Количество записей | Запрос |
| Вывести название блюда, цену и вес блюда, с указанием количества ингредиентов. |
| select  dish\_name as "Название блюда",  dish\_weight as "Вес блюда",  dish\_cost as "Цена блюда",  count(ingredient\_kod) as "Количестов ингредиентов"  from dishes  inner join sostav  on sostav.dish\_kod = dishes.dish\_kod  group by dish\_name,dish\_weight,dish\_cost; |
| Результат |
|  |
| Суммарный результат | Запрос |
| Задача: Вывести ФИО сотрудников и сумму стоимости заказов на каждого сотрудника |
| select  concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic) as "ФИО сотрудника",  sum(checkue\_sumfinal) as "Сумма заказа"  from orders  inner join employee  on employee.employee\_kod = orders.employee\_kod  inner join checkues  on checkues.order\_name = orders.order\_name  group by concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic); |
| Результат |
|  |
| Среднеарифметический результат | Запрос |
| Вывести ФИО клиентов и среднеарифметическую стоимость всех заказов на каждого клиента. |
| select  concat(rv\_surname,' ', rv\_name, ' ', rv\_patronymic) as "FIO",  avg(person\_cost) as "AVG cost"  from public.registered\_visitors  inner join public.individualorders  on individualorders.v\_kod = registered\_visitors.rv\_kod  group by concat(rv\_surname,' ', rv\_name, ' ', rv\_patronymic); |
| Результат |
|  |
| Округление значения | Запрос |
| Вывести название блюда, стоимость блюда умноженное на ставку НДС, с округлением до двух знаков после запятой. |
| select  dish\_name as "Название блюда",  round(avg(dish\_cost)\*0.2, 2) as "Налог 20%"  from public.dishes  group by dish\_name; |
| Результат |
|  |
| Объединение столбцов | Запрос |
| Вывести ФИО сотрудников – ФИО клиентов и номера заказов. |
| select  concat(rv\_surname,' ', rv\_name, ' ', rv\_patronymic) as "Фио клиента",  concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic) as "Фио официанта",  orders.order\_kod as "Код заказа"  from public.registered\_visitors  inner join public.individualorders  on individualorders.v\_kod = registered\_visitors.rv\_kod  inner join orders  on orders.order\_kod = individualorders.order\_kod  inner join employee  on employee.employee\_kod = orders.employee\_kod; |
| Результат |
|  |
| Объединение строк | Запрос |
| Вывести номер заказа, дату создания заказа, и перечислить на против каждого заказа позиции блюд в заказе с количеством позиций. |
| select  order\_name as "Номер заказа",  order\_date as "Дата заказа",  string\_agg('(Название: '||dish\_name||'; Количество: '||order\_dishcount||')',' , ') as "Состав"  from orders  inner join dishes  on dishes.dish\_kod = orders.dish\_kod  group by order\_name, order\_date; |
| Результат |
|  |
| Длина строки | Запрос |
| Вывести название блюда, вес и цену блюда, вывести столбец с количеством символов в названии блюда |
| select  dish\_name as "Название блюда",  dish\_cost as "Стоимость блюда",  dish\_weight as "Вес блюда",  length(dish\_name) as "Длинна названия"  from public.dishes; |
| Результат |
|  |
| Подстрока | Запрос |
| Вывести номер кассового чека, исключив КЧ- и все наборы нулей. |
| select  substring(checkue\_number,10,1) as "Номер чека"  from checkues; |
| Результат |
|  |
| Верхний регистр | Запрос |
| Вывести названия блюда в верхнем регистре. |
| select  upper(dish\_name) as "Название блюда"  from public.dishes; |
| Результат |
|  |
| Нижний регистр | Запрос |
| Вывести название блюда, названия ингредиентов в нижнем регистре |
| select  lower(dish\_name) as "Название блюда",  string\_agg(lower(ingredient\_name),',') as "Название ингредиента"  from dishes  inner join sostav  on sostav.dish\_kod = dishes.dish\_kod  inner join ingredientes  on ingredientes.ingredient\_kod = sostav.ingredient\_kod  group by lower(dish\_name); |
| Результат |
|  |
| Форматированный вывод даты и/или времени | Запрос |
| Вывести номер бронирования, дату формирования брони в формате Год: ГГ, Месяц: ММ, День: ДД. |
| select  book\_number as "Book name",  to\_char(book\_making\_date, 'YYYY.MM.DD') as "Book date"  from booking; |
| Результат |
|  |
| Вывод части даты | Запрос |
| Вывести номер заказа, в отдельные столбцы: день, месяц, час открытия заказа. |
| select  order\_name as "Название заказа",  'Месяц: '||date\_part('month', order\_date) as "Месяц",  'День: '||date\_part('day', order\_date) as "День",  'Час: '||date\_part('hour', order\_opentime) as "Час"  from orders; |
| Результат |
|  |
| Сложение дат и/или времени | Запрос |
| Вывести номер бронирования, планируемую дату посещения, в отдельный столбец вывести сумму даты посещения и 21 день, до момента отмены брони |
| select  book\_number as "Book name",  book\_plan\_date + interval'21 day' as "Book ban date"  from booking; |
| Результат |
|  |
| Вычитание дат и/или времени | Запрос |
| Вывести номер заказа клиента, название позиции в заказе и разницу в минутах, сколько прошло времени с момента открытия заказа до момента выдачи блюд. |
| select  order\_name as "Заказ",  order\_opentime as "Время заказа",  order\_dishtime as "Время заказа порции",  order\_dishtime - order\_opentime as "Прошло времени"  from orders; |
| Результат |
|  |
| Возвращение возраста | Запрос |
| Вывести ФИО клиента, номер кассового чека ,дату создания кассового чека, номер бронирования, дату создания бронирования, и вернуть разницу, сколько прошло времени с момента закрытия чека до момента создания бронирования. |
| select  concat(rv\_surname,' ', rv\_name, ' ', rv\_patronymic) as "FIO",  checkue\_number as "Номер чека",  checkue\_date as " Дата чека",  book\_number as "Номер брони",  book\_making\_date as "Дата брони",  age(book\_making\_date,checkue\_date) as "Возраст"  from public.registered\_visitors  inner join public.individualorders  on individualorders.v\_kod = registered\_visitors.rv\_kod  inner join orders  on orders.order\_kod = individualorders.order\_kod  inner join checkues  on checkues.order\_name = orders.order\_name  inner join booking  on booking.v\_kod = registered\_visitors.rv\_kod; |
| Результат |
|  |
| Максимальное значение столбца | Запрос |
| Вывести самое дорогое блюдо, с выводом цены и названия. |
| select  dish\_name as "Название блюда",  dish\_cost as "Цена блюда"  from dishes  where dish\_cost = (select max(dish\_cost) from dishes); |
| Результат |
|  |
| Минимальное значение столбца | Запрос |
| Вывести самый дешёвый заказ, с выводом номера заказа, ФИО клиента |
| select  concat(rv\_surname,' ', rv\_name, ' ', rv\_patronymic) as "FIO",  checkue\_sumfinal as "Сумма заказа"  from public.registered\_visitors  inner join public.individualorders  on individualorders.v\_kod = registered\_visitors.rv\_kod  inner join orders  on orders.order\_kod = individualorders.order\_kod  inner join checkues  on checkues.order\_name = orders.order\_name  where checkue\_sumfinal = (select min(checkue\_sumfinal) from checkues); |
| Результат |
|  |
| Нумерация выводимых строк | Запрос |
| Задача: Вывести с порядковым номером, названия групп и ФИО студентов, отсортировав данные по фамилии студента по возрастанию. |
| select  order\_name as "Номер заказа",  order\_date as "Дата заказа",  dish\_name as "Состав",  row\_number() over (partition by order\_name) as "Нумерация"  from orders  inner join dishes  on dishes.dish\_kod = orders.dish\_kod  group by order\_name, order\_date, dish\_name; |
| Результат |
|  |
| Уникальные строки | Запрос |
| Вывести уникальные названия ингредиентов. |
| select distinct ingredient\_name from public.ingredientes; |
| Результат |
|  |
| Лимит выводимых строк | Запрос |
| Вывести название блюд, количество раз сколько блюдо было заказано, вывести 3- менее заказываемых блюд. |
| select  dish\_name as "Название блюда",  sum(order\_dishcount) as "Количество заказов"  from dishes  inner join orders  on orders.dish\_kod = dishes.dish\_kod  group by dish\_name  order by sum(order\_dishcount) ASC  limit 3; |
| Результат |
|  |
| Проверка наличия файла | Запрос |
| Вывести название, вес, цену блюда, название ингредиентов, иллюстрацию, если файл существует вывести адрес файла, если файла нет вывести информацию – «Файл отсутствует» |
| select  dish\_name as "Название блюда",  dish\_weight as "Вес",  dish\_cost as "Стоимсоть",  string\_agg(ingredient\_name,',') as "Состав",  case  when pg\_stat\_file(dish\_picture, true) = null then 'Файл отсутствует'  else dish\_picture  end as "Файл фото"  from dishes  inner join sostav  on sostav.dish\_kod = dishes.dish\_kod  inner join ingredientes  on ingredientes.ingredient\_kod = sostav.ingredient\_kod  group by dish\_cost,dish\_weight,dish\_name,dish\_picture; |
| Результат |
|  |
| Арифметические действия | Запрос |
| Вывести название, вес, цену блюда, название ингредиентов, иллюстрацию, если файл существует вывести адрес файла, если файла нет вывести информацию – «Файл отсутствует». |
| select  concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic) as "Фио официанта",  count(checkue\_number) as "Количество заказов",  sum(checkue\_sumfinal) \* 0.8 \* 0.15 as "Доход"  from employee  inner join orders  on orders.employee\_kod = employee.employee\_kod  inner join checkues  on checkues.order\_name = orders.order\_name  group by concat(employee\_surname,' ', employee\_name,' ', employee\_patronymic); |
| Результат |
|  |

1. Создание резервной копии БД.

