МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»

Факультет информатики и
вычислительной техники

Кафедра информационной
безопасности

Отчёт по курсовому проекту

по дисциплине «Безопасность систем баз данных»

Разработка базы данных для ресторана (кафе)

Выполнили: студенты группы БИ-31 и БИ-32
Жаркова М.В.
Сметанина М.А.
Манчук А.С.

Проверил: доцент кафедры
ИБ Сучков Д.С.

Йошкар-Ола
2019 г.

СОДЕРЖАНИЕ

Задачи к выполнению

В данной работе были реализованы такие задачи, как: создание базы данных раменной, автоматизация оформления онлайн заказа. Реализованная база данных позволяет потенциальному клиенту изучить ассортимент и состав рамена. Вследствие чего, пользователь может самостоятельно сформировать онлайн заказ.

1. Техническое задание

1.1 Требования к курсовой работе:

- Получить структуру данных из файла, согласно варианта. Привести к 3й нормальной форме. Добавить недостающие таблицы;
- Составить ER-диаграмму;
- Разработать API для базы данных на любом языке, выполняющемся на стороне сервера (php, ASP.NET, Java, python, node.js, etc);
- Взаимодействие должно осуществляться по клиент-серверной архитектуре, подключение с клиентской программы недопустимо;
- Провести настройку пользователей базы данных для разграничения прав доступа, привести пример конфигурации;
- Все документы и исходные коды для курсовой работы должны храниться под контролем системы контроля версий — git или mercurial (https://github.com/, https://bitbucket.org/);
- Во время сдачи курсового проекта необходимо предоставить отчет о проделанной работе в печатном виде (отчет).

1.2 Требования к базе данных

- Наличие не менее 7 таблиц, в том числе таблицы сессий и пользователей
- Структура таблицы должна содержать не менее 3-х полей, одно из которых ключевое
- Правомерное использование типов данных
- Обязательно использование триггеров и/или хранимых процедур
- Форма нормализации не менее 3NF
- Индексирование по полям поиска

1.3 Требование к API (минимальное количество реализованных методов)

- аутентификация пользователя (создание сессии);
- добавление/удаление/изменение данных в таблицах;
- выборка данных их ключевых таблиц по запросам;
- выборка данных из таблиц с объединением результатов.

2. Порядок выполнения работы

2.1 Этапы разработки базы данных

Разработана база данных, содержащая 9 таблиц. В том числе таблицы сессий (*restaurant_session*) и пользователей (*restaurant_users*). Для авторизации пользователя используется таблица - *restaurant_accounts*, содержащая информацию об аккаунте пользователя. Для хранения информации об ассортименте и составе используются таблицы:

restaurant_dish – название и цена рамена;
restaurant_ingredients – состав рамена (ингредиенты);

Для автоматизации оформления заказа, были созданы следующие таблицы:

restaurant_availability – баланс и платежные данные покупателя;
restaurant_cart – информация о раменах, добавленных в корзину;
restaurant_orders – оформленные заказы на доставку рамена;
Структуры, реализованных таблиц:

- Таблица restaurant_session:

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	**session_id**	varchar(36)	utf8mb4_general_ci		Нет	*Нет*
2	**session_date**	datetime			Нет	*Нет*
3	**acc_id**	varchar(36)	utf8mb4_general_ci		Нет	*Нет*

- Индексы таблицы restaurant_session:

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	**PRIMARY**	BTREE	Да	Нет	acc_id
Изменить Удалить	**ind_user_id**	BTREE	Да	Нет	user_id

- Таблица restaurant_users.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	**u_id**	varchar(36)	utf8mb4_general_ci		Нет	*Нет*
2	**u_name**	varchar(255)	utf8mb4_general_ci		Нет	*Нет*
3	**u_address**	varchar(320)	utf8mb4_general_ci		Нет	*Нет*
4	**u_phone**	varchar(255)	utf8mb4_general_ci		Нет	*Нет*
5	**u_birthday**	datetime			Нет	*Нет*
6	**u_company**	varchar(255)	utf8mb4_general_ci		Нет	*Нет*
7	**u_about**	varchar(320)	utf8mb4_general_ci		Нет	*Нет*
8	**u_role**	varchar(255)	utf8mb4_general_ci		Нет	user

- Индексы таблицы restaurant_users.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	PRIMARY	BTREE	Да	Нет	u_id
Изменить Удалить	ind_name_address	BTREE	Нет	Нет	u_name (100) u_address (100)

- Таблица restaurant_accounts.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	acc_id	varchar(36)	utf8mb4_general_ci		Нет	*Нет*
2	acc_email	varchar(255)	utf8mb4_general_ci		Нет	*Нет*
3	acc_password	varchar(32)	utf8mb4_general_ci		Нет	*Нет*
4	acc_registration	datetime			Нет	*Нет*
5	user_id	varchar(36)	utf8mb4_general_ci		Нет	*Нет*

- Индексы таблицы restaurant_accounts.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	PRIMARY	BTREE	Да	Нет	acc_id
Изменить Удалить	ind_user_id	BTREE	Да	Нет	user_id

- Таблица restaurant_dish.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	dish_tittle	varchar(128)	utf8mb4_general_ci		Нет	*Нет*
2	dish_price	decimal(6,2)			Нет	*Нет*
3	dish_img	varchar(128)	utf8mb4_general_ci		Нет	*Нет*

- Индексы таблицы restaurant_dish.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	PRIMARY	BTREE	Да	Нет	dish_tittle

- Таблица restaurant_ingredients.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	dish_tittle	varchar(128)	utf8mb4_general_ci		Нет	*Нет*
2	ingredient	varchar(255)	utf8mb4_general_ci		Нет	*Нет*
3	count	int(128)			Нет	*Нет*

- Индексы таблицы restaurant_ingredients.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	ind_ingredients_dish_tittle	BTREE	Нет	Нет	dish_tittle

- Таблица restaurant_availability.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	acc_id	varchar(36)	utf8mb4_general_ci		Нет	*Нет*
2	amount	decimal(6,2)			Нет	*Нет*
3	card_number	varchar(255)	utf8mb4_general_ci		Нет	*Нет*

- Индексы таблицы restaurant_availability.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	ind_availability_acc_id	BTREE	Да	Нет	acc_id

- Таблица restaurant_cart.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	session_id	varchar(36)	utf8mb4_general_ci		Да	*NULL*
2	order_id	varchar(36)	utf8mb4_general_ci		Да	*NULL*
3	price	decimal(6,2)			Нет	*Нет*
4	count	int(11)			Нет	*Нет*
5	dish_tittle	varchar(128)	utf8mb4_general_ci		Нет	*Нет*
6	acc_id	varchar(36)	utf8mb4_general_ci		Нет	*Нет*

- Индексы таблицы restaurant_cart.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	ind_cart_dish_tittle	BTREE	Нет	Нет	dish_tittle
Изменить Удалить	ind_cart_order_id	BTREE	Нет	Нет	order_id
Изменить Удалить	ind_cart_acc_id	BTREE	Нет	Нет	acc_id

- Таблица restaurant_orders.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	order_id	varchar(36)	utf8mb4_general_ci		Да	*NULL*
2	order_status	varchar(255)	utf8mb4_general_ci		Да	*NULL*
3	user_id	varchar(36)	utf8mb4_general_ci		Да	*NULL*

- Индексы таблицы restaurant_orders.

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
Изменить Удалить	ind_orders_order_id	BTREE	Да	Нет	order_id
Изменить Удалить	ind_orders_user_id	BTREE	Нет	Нет	user_id

Используемые триггеры:

- *drop_order* – триггер отвечающий за удаление товаров из корзины;

```
DELIMITER $$
CREATE TRIGGER `drop_order` BEFORE DELETE ON `restaurant_orders` FOR EACH ROW BEGIN
        DELETE FROM `restaurant_cart`
    WHERE `restaurant_cart`.order_id = OLD.order_id;
END
$$
DELIMITER ;
```

- *generate_id* – триггер отвечающий за генерацию id аккаунта;

```
DELIMITER $$
CREATE TRIGGER `generate_id` BEFORE INSERT ON `restaurant_accounts` FOR EACH ROW BEGIN
        IF ISNULL(NEW.acc_id) THEN
                SET NEW.acc_id = UUID();
        END IF;
END
$$
DELIMITER ;
```

Проведена настройка пользователей базы данных для разграничения прав доступа:

Имя пользователя	Имя хоста	Тип	Привилегии	Grant
admin	localhost	глобальный	ALL PRIVILEGES	Да
editor	localhost	глобальный	SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,ALTER,CREATE TEMPORARY TABLES,CREATE VIEW,EVENT,TRIGGER,SHOW VIEW,CREATE ROUTINE,ALTER ROUTINE,EXECUTE	Нет
moderator	localhost	глобальный	SELECT,INSERT,UPDATE,DELETE	Нет

Рис 10. Пользователи базы данных

Имя пользователя	Имя хоста	Тип	Привилегии	Grant
admin	localhost	глобальный	ALL PRIVILEGES	Да

Рис 11. Привилегии пользователя admin

Имя пользователя	Имя хоста	Тип	Привилегии	Grant
editor	localhost	глобальный	SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,ALTER,CREATE TEMPORARY TABLES,CREATE VIEW,EVENT,TRIGGER,SHOW VIEW,CREATE ROUTINE,ALTER ROUTINE,EXECUTE	Нет

Рис 12. Привилегии пользователя editor

Имя пользователя	Имя хоста	Тип	Привилегии	Grant
moderator	localhost	глобальный	SELECT,INSERT,UPDATE,DELETE	Нет

Рис 13. Привилегии пользователя moderator

2.2 Этапы разработки API

Было разработано API для аутентификации пользователя, состоящий из 3 функций, написанных на языке PHP.

```
function mySession_start()
{
        if (isset($_COOKIE['SESSID']))
        {

                global $db;
                $sess_id = $_COOKIE['SESSID'];

                $sql = 'SELECT acc_id FROM restaurant_session WHERE session_id = :sess_id';
                $stmt = $db->prepare($sql);
                $stmt->execute([':sess_id' => $sess_id]);
                $acc = $stmt->fetch(PDO::FETCH_OBJ);

                if ($acc)
                {

                        $sess_update = 'UPDATE restaurant_session SET session_date = :s_date, session_id = :s_id WHERE acc_id = :acc_id';
                        $params_update = [ ':s_date' => date("Y-m-d H:i:s"), ':s_id' => $sess_id, ':acc_id' => $acc->acc_id];

                        $stmt = $db->prepare($sess_update);
                        $stmt->execute($params_update);

                        return true;
                }

                return false;

        }
}
function mySession_write($acc_id)
{
        $SESSID = uniqid();

        global $db;

        setcookie('SESSID', $SESSID, time()+60*60*24*30);
        $_COOKIE['SESSID'] = $SESSID;


        $sess_check = 'SELECT acc_id FROM restaurant_session WHERE acc_id = :account_id';
        $stmt = $db->prepare($sess_check);
        $stmt->execute([':account_id' => $acc_id]);
        $acc = $stmt->fetch(PDO::FETCH_OBJ);

        if ($acc)
        {
                $sess_update = 'UPDATE restaurant_session SET session_date = :s_date, session_id = :s_id';
                $params_update = [ ':s_date' => date("Y-m-d H:i:s"), ':s_id' => $SESSID];
                $stmt = $db->prepare($sess_update);
                $stmt->execute($params_update);


                return;
        }

        $sql = 'INSERT INTO restaurant_session(session_id, session_date, acc_id) VALUES (:session_id, :session_date, :acc_id)';
        $params = [ ':session_id' => $SESSID, ':session_date' => date("Y-m-d H:i:s"), ':acc_id' => $acc_id ];
        $stmt = $db->prepare($sql);
        $stmt->execute($params);

}


function mySession_stop()
{
        global $db;
        $SESSID = $_COOKIE['SESSID'];

        $sql = 'DELETE FROM restaurant_session WHERE session_id = :sess_id';
        $stmt = $db->prepare($sql);
        $stmt->execute([':sess_id' => $SESSID]);

        setcookie('ACCID', '', time());
        setcookie('SESSID', '', time());
}
```

Рис. 14 Аутентификация пользователя

Разработана функция добавления, удаления и изменения данных.

```
$add_order = 'INSERT INTO restaurant_orders (order_id, order_status, user_id)
                            SELECT
                                        :order_id AS order_id,
                                        "Обработка заказа" AS order_status,
                                        restaurant_accounts.user_id AS user_id
                            FROM restaurant_cart
                            INNER JOIN restaurant_accounts
                            ON restaurant_accounts.acc_id = restaurant_cart.acc_id
                            LIMIT 1
                           ';
$stmt = $db->prepare($add_order);
$stmt->execute([':order_id' => $order_id]);
```

Рис.15 Добавление и изменение данных

```
if (isset($_GET['action']) && $_GET['action']=="drop")
{
        $sql = 'DELETE FROM restaurant_orders WHERE order_id = :order_id';
        $params = [ ':order_id' => strval(trim($_GET['id'])) ];
        $stmt = $db->prepare($sql);
        $stmt->execute($params);
}
```

Рис.16 Удаление данных

Добавлена функция вывода и объединения данных из ключевых таблиц.

```
<?php

        $sql = 'SELECT * FROM restaurant_users
                        INNER JOIN restaurant_accounts ON restaurant_accounts.user_id = restaurant_users.u_id
                        INNER JOIN restaurant_session ON restaurant_session.acc_id = restaurant_accounts.acc_id
                        INNER JOIN restaurant_availability ON restaurant_availability.acc_id = restaurant_accounts.acc_id
                        WHERE restaurant_session.session_id = :sess_id';
        $stmt = $db->prepare($sql);
        $stmt->execute([':sess_id' => $_COOKIE['SESSID']]);
        $user = $stmt->fetch(PDO::FETCH_OBJ);

        echo '<br>Здравсвуйте,<u>'.$user->u_name.'</u>Вы вошли на сайт Ramen laboratory.
                  <br>На вашем счету '.$user->amount.' ₽
                 ';

        if ($user->u_role == 'admin')
        {

                echo "<br><br>Меню:
                  <br><a href='orders.php'> Заказы</a>
                ";

        }

?>
```

Рис.17 Вывод данных по заказам

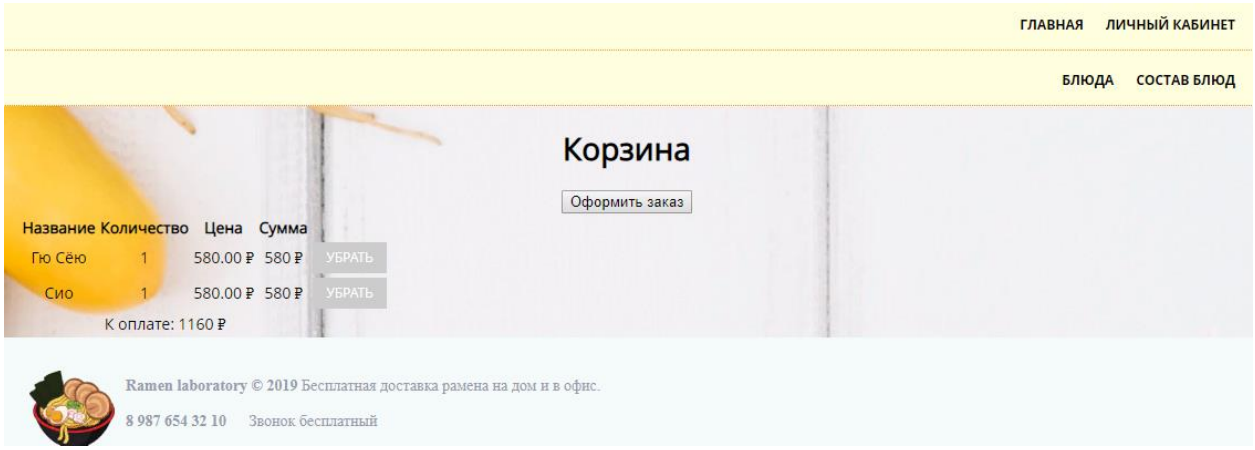


Рис.18 Корзина покупателя

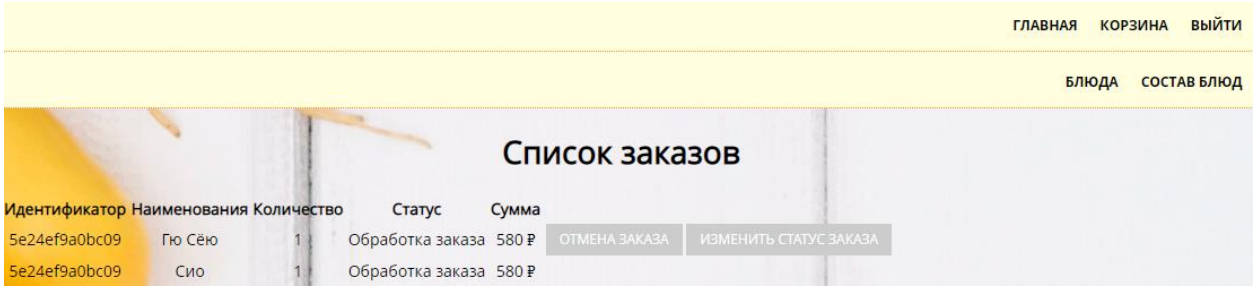


Рис.19 Список заказов (доступен только администратору)

Рис. 20 Главная страница

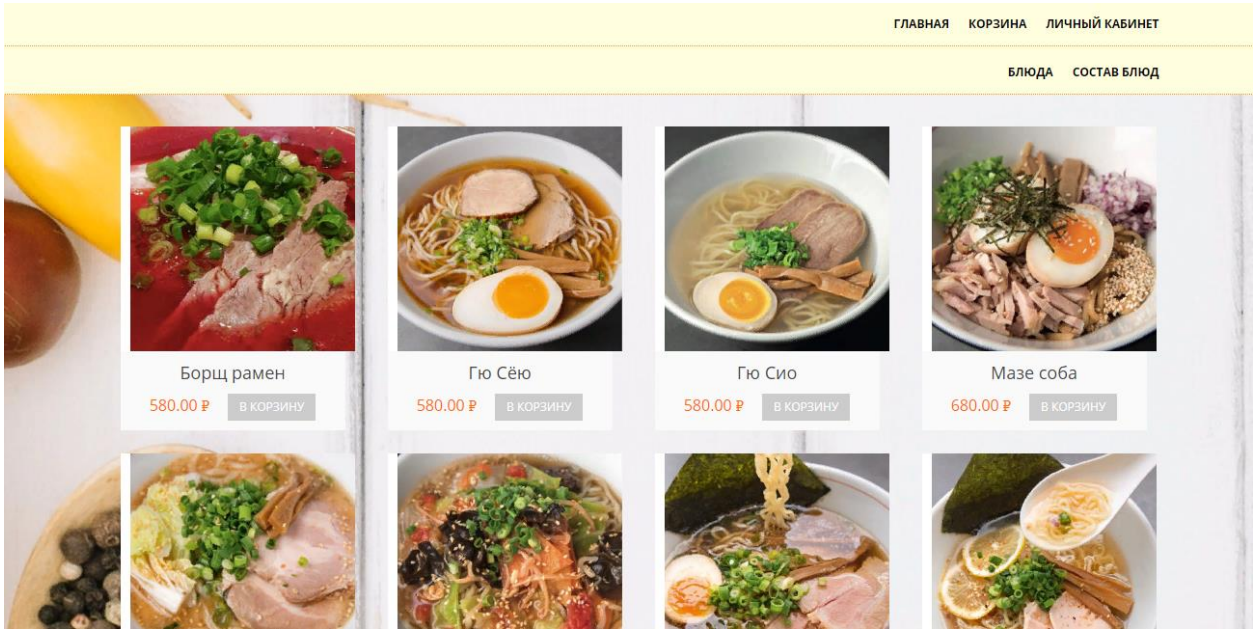


Рис.21 Страница выбора рамена

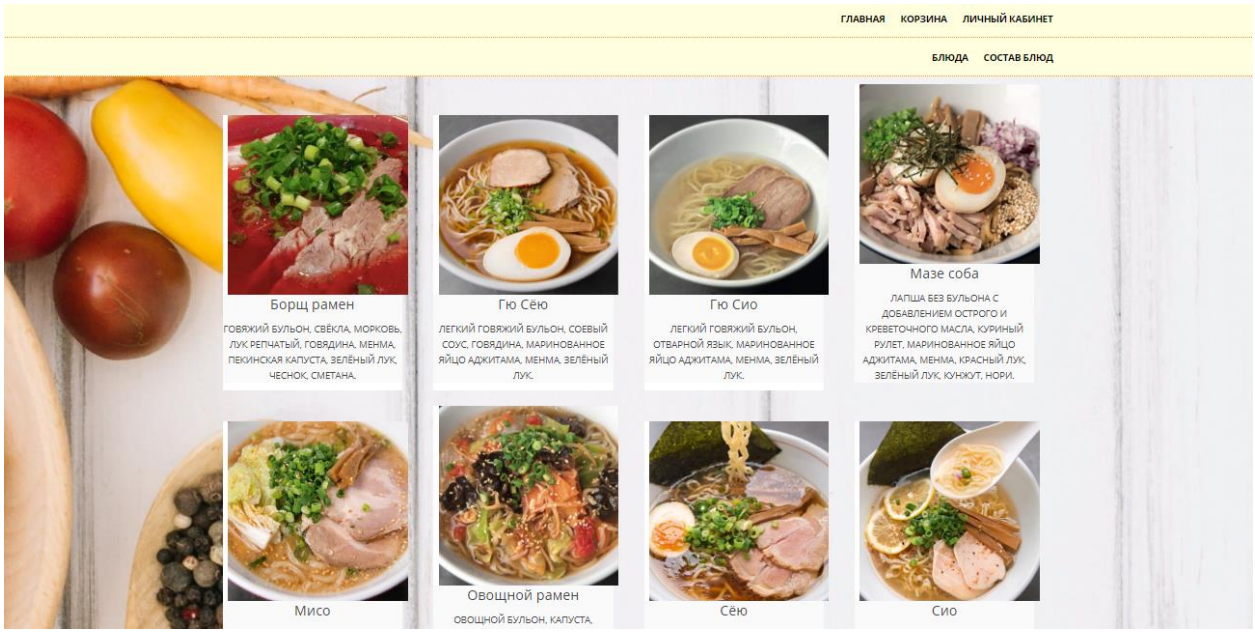
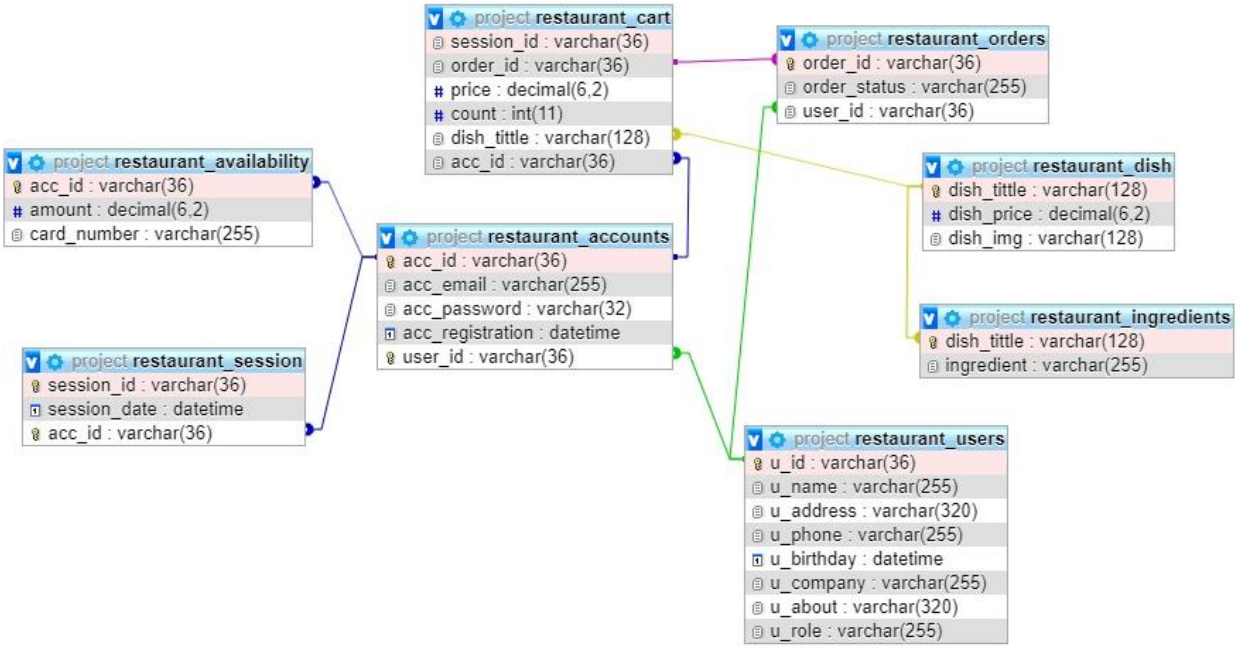


Рис. 21 Состав рамена

3. Приложения

3.1 ER-диаграмма



База данных находится в первой нормальной форме, т.к. в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов. Т.к. база данных находится в 1НФ, то она также находится во второй нормальной форме, потому что каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа. Наличие 2НФ и отсутствие зависимости неключевых атрибутов от ключевых доказывает, что база данных находится в третьей нормальной форме.

3.2 Исходные коды и документы:

https://github.com/faletskaya/kursovaya_bd

4. Вывод

Во время выполнения курсового проекта были изучены методы работы с базами данных, способы управления. Таким образом, в ходе работы была разработана база данных, а также API, для автоматизации работы раменной.