
Site web d'hôtellerie (Hôtel Neptune) avec gestion de bdd

DESCRIPTION DU PROJET :

Il s'agit d'un projet d'école réalisé en groupe de 4 élèves. Du 15/11/22 au 15/12/22.

L'hôtel 3 étoiles Neptune situé à Carnon souhaite informatiser la réservation de ses chambres. Le gérant de l'hôtel a décidé de faire un site Web en PHP en lien avec une base de données MySQL.

ENVIRONNEMENT PROFESSIONNEL

L'hôtel gère 12 chambres et à de nombreux clients.

Un premier MCD a été produit mais il est incomplet. Il est fortement conseillé de l'améliorer.

Vous n'avez pas d'adresse mail et de mot de passe dans la table client !

Le site doit avoir :

- Une partie administrative
- Une partie client

Le site doit permettre de gérer :

- Les chambres
 - Administration : Ajout, modification, suppression, tri, recherche
 - Administration : Gestion de photos associées
 - Client : visualisation, recherche
- Les clients
 - Administration : Ajout, modification, suppression, tri, recherche
 - Client : Inscription, modification et suppression de son profil
- *Les réservations*
 - *Administration : visualisation, filtrage d'après des dates, suppression de toutes les réservations*
 - *Client : visualisation, suppression, filtrage d'après des dates, réservation avec envoi de mail, simulation de paiement des réservations, édition des factures au format PDF*

CONTRAINTES TECHNIQUES

Le site Web doit être développé en PHP (objet) en lien avec une base de données MySQL

COMPETENCES DEVELOPPEES

Répondre aux incidents et aux demandes d'assistance et d'évolution

- " ▶ Collecter, suivre et orienter des demandes
- ▶ Traiter des demandes concernant les services réseau et système, applicatifs
- ▶ Traiter des demandes concernant les applications"

Ce projet s'inscrit dans un cadre où il nous a été demandé (via une consigne) de répondre aux demandes d'évolution de la structure Hotel Neptune via plusieurs étapes de réalisation du site web. De plus, une revue a été faite par notre prof nous demandant de traiter ses demandes, corrections, etc... Il nous a donc fallu « collecter, suivre et orienter des demandes » en équipe via une gestion de projet précise ainsi que « traiter des demandes concernant les applications ».

Travailler en mode projet

- " ▶ Analyser les objectifs et les modalités d'organisation d'un projet
- ▶ Planifier les activités
- ▶ Évaluer les indicateurs de suivi d'un projet et analyser les écarts"

Etant un projet en groupe, il nous a fallu nous organiser et définir un cahier des charges précis pour s'attribuer les tâches et les classer, c'est-à-dire « Analyser les objectifs et les modalités d'organisation d'un projet ». Nous avons donc « planifier les activités » avec des deadlines définies dans notre groupe de discussion. Et « Évaluer les indicateurs de suivi d'un projet et analyser les écarts » pour pouvoir finaliser ce site à temps. Nous organisons des réunions pour faire le point sur l'avancée de chacun et regardions si nous avons du retard afin de gérer les éventuels problèmes pour que le projet soit rendu dans les temps.

Mettre à disposition des utilisateurs un service informatique

- " ▶ Réaliser les tests d'intégration et d'acceptation d'un service
- ▶ Déployer un service

- Accompagner les utilisateurs dans la mise en place d'un service"

Bien sur, nous avons du tester notre projet pour s'assurer de sa bonne conformité et voir s'il répondait bien à toutes les contraintes et demandes du cahier des charges (compétence « Réaliser les tests d'intégration et d'acceptation d'un service »). De plus, pendant la dernière semaine de sa finalisation, il était disponible sur internet via une URL fournie par notre école (« Déployer un service »). Nous avons également dû expliquer comment notre site fonctionnait par le biais d'une démo et d'un échange avec notre professeur, nous avons donc « Accompagné les utilisateurs dans la mise en place d'un service », notamment pour ce qui concerne l'interface d'administration.

Organiser son développement professionnel

- " ▸ Mettre en place son environnement d'apprentissage personnel
- Mettre en œuvre des outils et stratégies de veille informationnelle
- Gérer son identité professionnelle
- Développer son projet professionnel"

Pour la réalisation de ce site web en html, css, php et sql, nous avons du tout d'abord nous former par le biais de documentation ou de nombreux tutos en ligne (dont ceux d'OpenClassrooms). Il nous a fallu « Mettre en place son environnement d'apprentissage personnel ». On a aussi dû s'informer sur les dernières bonnes pratiques du langage PHP et de tout ce qui concerne l'accès aux bases de données. Pour avoir un site sécurisé et à jour. Cela nous a aussi permis de « Développer son projet professionnel » puisqu'il s'inscrit directement dans notre cadre d'étude et le métier que nous voulons réaliser plus tard (développeuse full stack pour ma part).

DESCRIPTION DE CHAQUE PAGE

Connexion.php :

- Bouton retour pour retourner à la page précédente ou à la page d'accueil
- Lien vers la page d'inscription
- Formulaire permettant au client de se connecter
- Doit comporter les champs "Adresse mail/nom" et "mot de passe" (l'utilisateur peut se connecter avec son adresse mail ou son nom)
- Vérifier les données rentrées par l'utilisateur et les comparer avec ceux de la bdd pour voir si l'utilisateur peut se connecter.
- Mettre des "placeholder" et des "required" sur le formulaire html
- Optimiser le style (avec la bonne palette de couleur, des animations ?)
- Sécuriser les données insérées dans la bdd et se protéger des injections SQL en utilisant des fonctions php dédiées

Inscription.php :

- Bouton retour pour retourner à la page précédente ou à la page d'accueil
- Lien vers la page d'inscription
- Formulaire permettant au client de s'inscrire
- Doit comporter les champs "Nom", "Prénom", "mail", "téléphone" (non requis), "adresse" (non requis), "mot de passe", "confirmez votre mot de passe".
- Vérifier que cet utilisateur n'est pas déjà présent dans la bdd (nom ET prénom unique par compte), si tous les champs requis sont remplis et que les cases "mot de passe" et "confirmez votre mdp" sont identiques, insérer les données dans la bdd sinon afficher des messages ex: "Nom obligatoire", "mdp non identiques". (les valeurs de "date_inscription" et "admin" se mettent automatiquement)
- Mettre des "placeholder" et des "required" là où il le faut sur le form html et mettre type=email pour le mail, type=password pour le mdp...
- Vérifier en php la conformité des données (longueur, présence d'un @ pour le mail, présence d'une majuscule et d'un symbole pour le mdp, pas de chiffres pour le nom/prénom, etc...) avant de les insérer dans la bdd
- Sécuriser les données insérées dans la bdd et se protéger des injections SQL en utilisant des fonctions php dédiées
- Hasher le password de l'utilisateur pour qu'il ne puisse pas être lu / vu même dans la bdd (c'est une ou deux fonctions php, je sais plus)

Header.php :

- On mettra (include) un fichier php dans le nav de chaque fichier html pour pouvoir gérer l'affichage du **menu (et uniquement du menu)** correctement.
- Si l'utilisateur client est co : espace membre
- Si l'utilisateur est co et que c'est un admin : Espace administration
- Si l'utilisateur n'est pas connecté : bouton se connecter.

Index.html : (la page d'accueil)

- Include du fichier Header.php dans la partie Nav
- Faire une belle page d'accueil avec le nom de l'hôtel et sa photo (en background) en haut de la page
- En défilant sur cette page, faire des sections :
 - une partie avec : Lien/bouton vers "réserver une chambre" (avec des photos de chambres, descriptif...)
 - une partie avec : Présentation (rapide) de l'hôtel (mots clés (ex : piscine, 12 chambres, restaurant, garage/parking...) + photo) avec lien/bouton qui redirige vers "en savoir plus"
 - une partie avec : photo de bonne bouffe, restaurant avec lien/bouton "Découvrez notre menu" et qui redirige vers la page "restaurant"
 - d'autres parties si vous avez des idées
- Footer
- Améliorer le style en faisant des animations (en js c'est mieux) pour faire une belle page d'accueil voire même de faire défiler des images automatiquement dans une case pour la présentation de l'hotel, les photos de nourriture... (
- Optimisation SEO : contenu, balises strong, title, meta description,

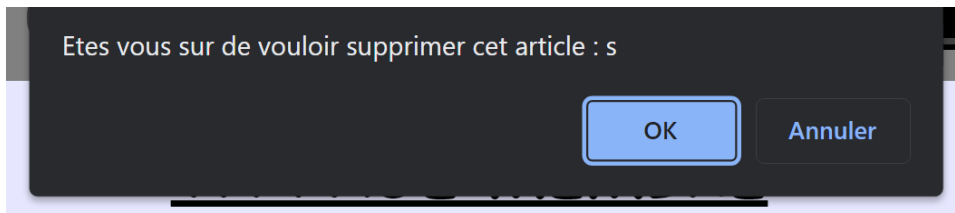
Chambres.html :

- Include du fichier Header.php dans la partie Nav
- Lister toutes les chambres LIBRES (avec leurs nom ex: "chambre n°1", "chambre n°6" sous forme de lien <a> avec leurs caractéristiques en dessous (nombre de lits, thème...) (je rappelle que c'est la version minimale (v1))
- Si on clique sur une chambre (vu que le nom des chambres sont des liens), renvoyer vers la page Reservation.php (il peut aussi s'appeler Rervation.html vu qu'il va y avoir du php et du html dans ce fichier) ATTENTION méthode GET donc on revoie vers Reservation.php?id=[id de la chambre]. Voir exemple :

```
<a href="P_blog.php?theme='.$theme.'&page='.$i.'">'.$i.' </a>'
```
- Footer

Reservation.php (ou .html) :

- Include du fichier Header.php dans la partie Nav
- Affiche le détail de la chambre sur laquelle l'utilisateur à cliqué (grâce à la méthode GET qui recueille l'ID de la chambre dans l'URL)
- Formulaire avec 2 champs de type "date" pour la date de début et de fin de réservation
- Gros bouton "Réserver la chambre"
- Affiche un message "chambre déjà réservée" si la date / période sélectionnée a déjà été sélectionnée par une autre personne (on a ces infos dans la table réservations de la bdd)
- Si tout est ok, recueille l'id de l'utilisateur (\$_SESSION) et les dates du form pour les INSERT INTO réservations
- Vérifier en php la fiabilité des données (les dates ne doivent pas être dans le passé, DateD_reservation dans être antérieure ou égale à DateF_reservation, etc....)
- Faire une pop-up en js (sur le bouton "réserver la chambre" en utilisant On click ()) pour confirmer la réservation : "Etes-vous sur de vouloir réserver cette chambre [nom de la chambre] ?". Voir l'exemple :



Restaurant.html :

- Include du fichier Header.php dans la partie Nav
- Décrire le menu de l'hôtel avec du texte et des photos en faisant une belle mise en forme
- Footer

Newsletter.html :

- Include du fichier Header.php dans la partie Nav
- Décrire les actus / événements de l'hôtel
- Faire des div Flexbox pour chaque actu en lui donnant un titre, une photo, une date et sa description
- Footer

EnSavoirPlus.html :

- Include du fichier Header.php dans la partie Nav
- Décrire les infos supplémentaires relatives à l'hôtel Neptune (histoire, environs, labels, témoignages, photos...)
- Footer

Contact.html :

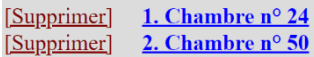
- Include du fichier Header.php dans la partie Nav
- Page de contact
- Footer

PageMembre.html :

- Include du fichier Header.php dans la partie Nav
- Changer les fonctionnalités en fonction de si on est client ou administrateur (voir si admin = True ou admin = False)
- Footer

Pour un non-admin (Admin = False dans la table client) :


- Affichage des informations personnelles du client avec un bouton modifier

- Affichage de toutes les reservations (passées, en cours et à venir) du client avec un bouton supprimer comme dans cet exemple 
 - 1. Chambre n° 24
 - 2. Chambre n° 50
- Cliquer sur le nom de la chambre permet au client d'avoir toutes les details de la reservation correspondant à cette chambre (au pire renvoie à la page Reservation.php?id=[id de la chambre cliquée])

Pour un admin (Admin = True dans la table Client)

- Mettre un texte "vous êtes connecté en tant qu'administrateur"
- "Vos outils" : Lien vers les pages infoChambres.php, infoClient.php et infoReservation.php

infoChambres.php :

- Include du fichier Header.php dans la partie Nav
- Rajouter un sous-menu ou des boutons pour naviguer entre les 3 outils de l'admin (pour infoChambres.php, infoClient.php et infoReservation.php)
- ajout, modification, suppression, tri, recherche de chambres en affichant toutes leurs caractéristiques
- Liste de toutes les chambres et des boutons "supprimer" et "modifier" à côté de chacune d'elles un peu comme dans cet exemple 
 - 1. Chambre n° 24
 - 2. Chambre n° 50
- Faire une option de tri (form html type select pour le nombre de lit et le thème, un de type c et un de type range pour le prix). Et afficher les chambres correspondantes en utilisant le Select ... Where + condition (en php et SQL). Si aucune chambre ne correspond à ces critères : afficher "aucune chambre correspondante"
- Bouton ajouter qui affiche un form (sans changer de page) pour ajouter une chambre
- Insérer la chambre ajoutée dans la bdd en faisant les vérifications nécessaires (sécurisation des données, tous les champs sont bien remplis ?...)

infoClient.php :

- Include du fichier Header.php dans la partie Nav
- Rajouter un sous-menu ou des boutons pour naviguer entre les 3 outils de l'admin (pour infoChambres.php, infoClient.php et infoReservation.php)
- ajout, modification, suppression, tri, recherche des clients en affichant toutes leurs caractéristiques
- Donc la même chose que infoChambre.php mais pour les clients
- Pour chaque client, pouvoir connaître ses reservations passées, en cours et à venir

infoReservation.php :

- Include du fichier Header.php dans la partie Nav
- Rajouter un sous-menu ou des boutons pour naviguer entre les 3 outils de l'admin (pour infoChambres.php, infoClient.php et infoReservation.php)

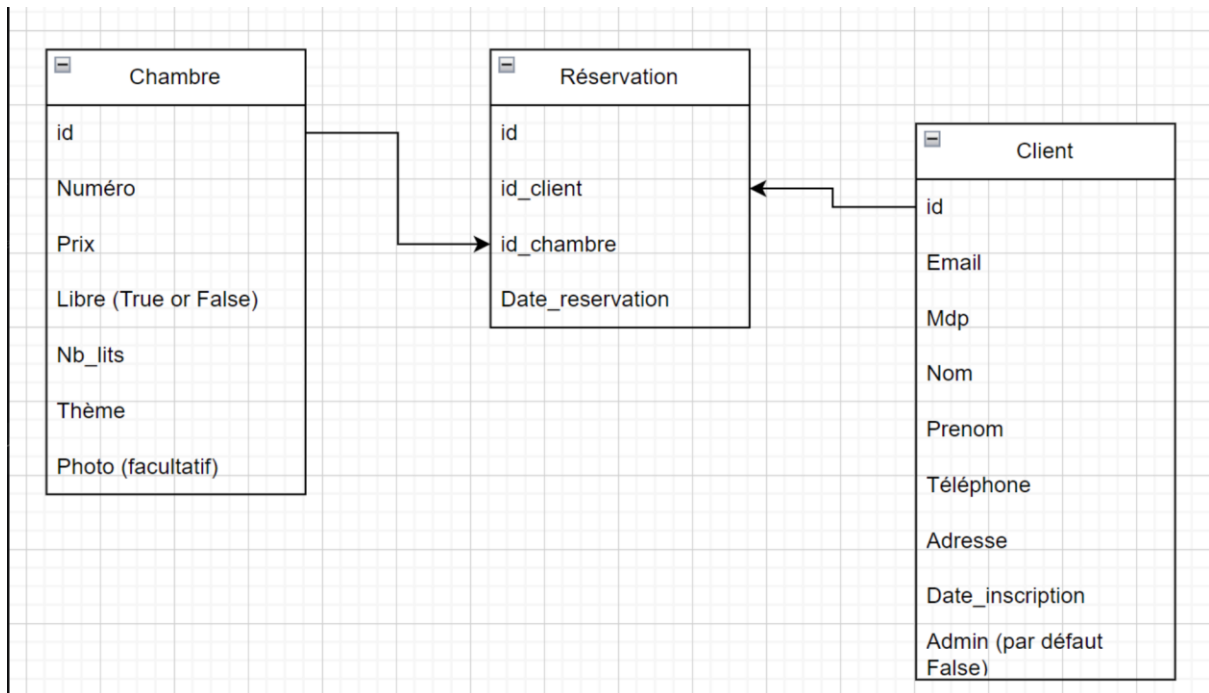
- visualisation, filtrage d'après les dates, suppression des réservations en affichant toutes leurs caractéristiques
- Donc la même chose que infoChambre.php mais pour les reservations

ANNEXE

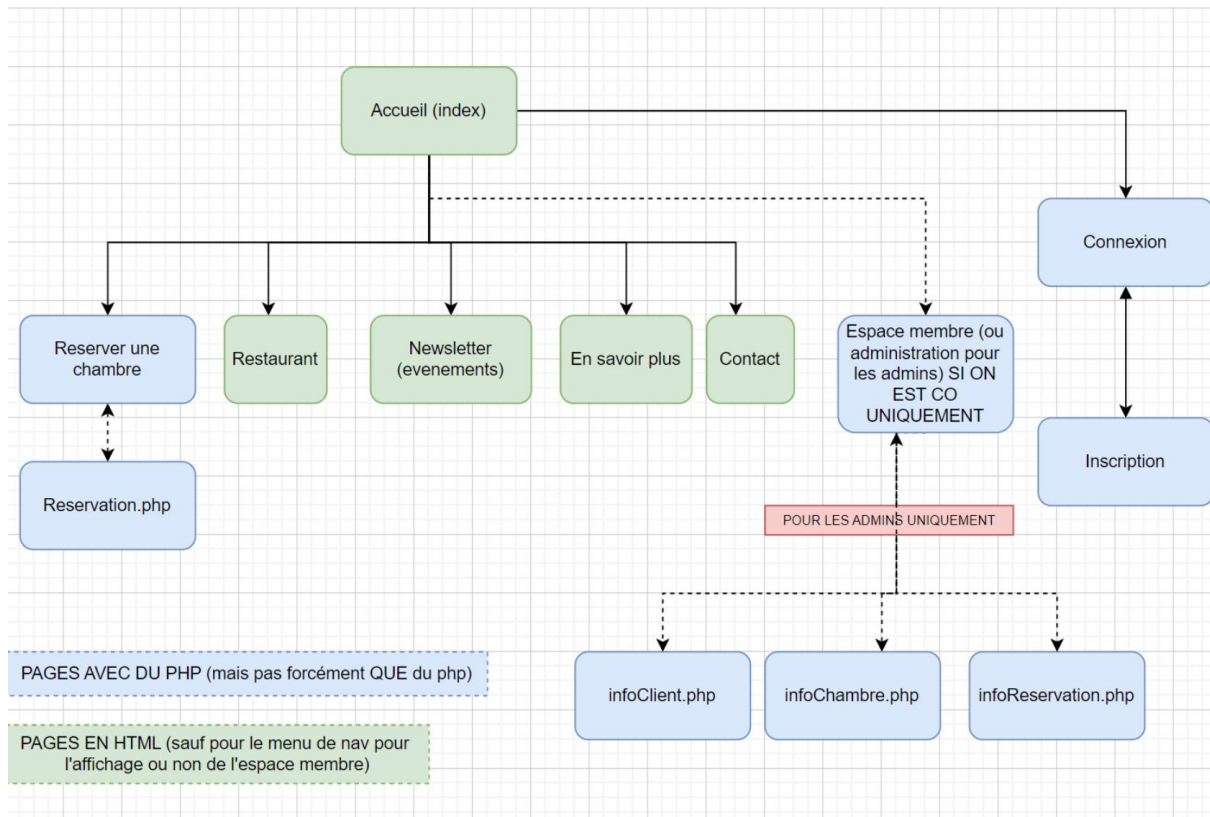
POUR LE SYSTEME DE PAGINATION ET LA BARRE DE RECHERCHE

```
} else { // on a fait une recherche
    if(isset($_GET['search']) AND !empty($_GET['search'])) {
        // Système de pagination
        if (isset($_GET['page']) AND !empty($_GET['page']) AND $_GET['page'] > 0) {
            $pageCourante = intval($_GET['page']); // pour transformer $_GET['page'] en int (nb entier)
        } else {
            $pageCourante = 1;
        }
        $billetsParPage = 5;
        $depart = ($pageCourante-1)*$billetsParPage;
        $search = htmlspecialchars($_GET['search']);
        $req = $bdd->query('SELECT id FROM Articles WHERE titre LIKE "%'.$search.'" '); // LIKE va permettre de rechercher un titre approximatif (les % signifie qu'il peut y
        avoir des caractères avant ou après ce qu'on à recherché: $search)
        $count = $req->rowCount();
        $req = $bdd->query('SELECT id,auteur,titre,theme,resume,DATE_FORMAT(date_post, \"%d/%m/%Y à %Hh%i\") AS date_postFR FROM Articles WHERE titre LIKE "%'.$search.'"
        ORDER BY date_post DESC LIMIT ' . $depart . ', ' . $billetsParPage);
        if($count == 0) { // si aucun résultat, on regarde le titre + le résumé
            $req = $bdd->query('SELECT id FROM Articles WHERE CONCAT(titre,resume) LIKE "%'.$search.'" ');
            $count = $req->rowCount();
            $req = $bdd->query('SELECT id,auteur,titre,theme,resume,DATE_FORMAT(date_post, \"%d/%m/%Y à %Hh%i\") AS date_postFR FROM Articles WHERE CONCAT(titre,resume) LIKE "%
            %'.$search.'" ORDER BY date_post DESC LIMIT ' . $depart . ', ' . $billetsParPage);
        }
        if($count != 0) {
            // Système de pagination
            $billetsTotaux = $count;
            $pagesTotales = ceil($billetsTotaux/$billetsParPage);
        }
    }
}
```

Schéma de la base de donnée



Arborescence du site



Code de la base de données

```

-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Hôte : 127.0.0.1:3306
-- Généré le : ven. 11 nov. 2022 à 10:10
-- Version du serveur : 5.7.36
-- Version de PHP : 7.4.26

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données : `hotel`
--
  
```

```

-----

--
-- Structure de la table `chambre`
--

DROP TABLE IF EXISTS `chambre`;
CREATE TABLE IF NOT EXISTS `chambre` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `Numero` int(11) NOT NULL,
  `Prix` int(11) NOT NULL,
  `Libre` tinyint(1) NOT NULL,
  `Nb_lits` int(11) NOT NULL,
  `Theme` varchar(100) NOT NULL,
  `Photo` blob,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-----

--
-- Structure de la table `client`
--

DROP TABLE IF EXISTS `client`;
CREATE TABLE IF NOT EXISTS `client` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(100) NOT NULL,
  `mdp` varchar(255) NOT NULL,
  `nom` varchar(50) NOT NULL,
  `prenon` varchar(50) NOT NULL,
  `telephone` int(10) DEFAULT NULL,
  `adresse` varchar(255) DEFAULT NULL,
  `date_inscription` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `admin` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-----

--
-- Structure de la table `reservation`
--

DROP TABLE IF EXISTS `reservation`;
CREATE TABLE IF NOT EXISTS `reservation` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```
`id_client` int(11) NOT NULL,  
`id_chambre` int(11) NOT NULL,  
`dateD_reservation` date NOT NULL,  
`dateF_reservation` date NOT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;  
COMMIT;
```