PROJET

Concevez la solution technique d’une application de restauration en ligne

Express Food

Projet réalisé par Zina Amararene

1. Les tables SQL

```
-- MySQL Script generated by MySQL Workbench
-- Tue Apr 10 19:56:09 2018
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';


-- -----------------------------------------------------
-- Schema expressfood
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `expressfood` DEFAULT CHARACTER SET utf8
;
USE `expressfood` ;

-- -----------------------------------------------------
-- Table `expressfood`.`User`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`User` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `last_name` VARCHAR(150) NOT NULL,
  `first_name` VARCHAR(150) NOT NULL,
  `login` VARCHAR(70) NOT NULL,
  `password` VARCHAR(250) NOT NULL,
  `email` VARCHAR(250) NOT NULL,
  `phone` INT NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`Administrator`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Administrator` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `super_admin` TINYINT NOT NULL,
  `date_connection` DATE NOT NULL,
  `User_id` INT NOT NULL,
  PRIMARY KEY (`id`, `User_id`),
  INDEX `fk_Administrator_User1_idx` (`User_id` ASC),
  CONSTRAINT `fk_Administrator_User1`
    FOREIGN KEY (`User_id`)
    REFERENCES `expressfood`.`User` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`Customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Customer` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `civility` VARCHAR(100) NOT NULL,
  `date_birth` DATE NULL,
  `User_id` INT NOT NULL,
  PRIMARY KEY (`id`, `User_id`),
  INDEX `fk_Customer_User1_idx` (`User_id` ASC),
  CONSTRAINT `fk_Customer_User1`
    FOREIGN KEY (`User_id`)
    REFERENCES `expressfood`.`User` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`DelivryMan`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`DelivryMan` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `status` VARCHAR(200) NOT NULL,
  `latitude` DECIMAL(8,8) NOT NULL,
  `longitude` DECIMAL(8,8) NOT NULL,
  `User_id` INT NOT NULL,
  PRIMARY KEY (`id`, `User_id`),
  INDEX `fk_DelivryMan_User_idx` (`User_id` ASC),
  CONSTRAINT `fk_DelivryMan_User`
    FOREIGN KEY (`User_id`)
    REFERENCES `expressfood`.`User` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```sql
-- -----------------------------------------------------
-- Table `expressfood`.`Address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Address` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `track_number` INT NOT NULL,
  `street` VARCHAR(250) NOT NULL,
  `city` VARCHAR(250) NOT NULL,
  `postal_code` INT NOT NULL,
  `building` CHAR(50) NULL,
  `floor` INT NULL,
  `latitude` DECIMAL(8,8) NOT NULL,
  `longitude` DECIMAL(8,8) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`Speciality`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Speciality` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `category` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`Dish`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Dish` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(150) NOT NULL,
  `description` TEXT(600) NOT NULL,
  `amount` INT NOT NULL,
  `unit_price` DECIMAL(5,2) NOT NULL,
  `date` DATE NOT NULL,
  `Administrator_id` INT NOT NULL,
  `Administrator_User_id` INT NOT NULL,
  `Speciality_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Dish_Administrator1_idx` (`Administrator_id` ASC,
`Administrator_User_id` ASC),
  INDEX `fk_Dish_Speciality1_idx` (`Speciality_id` ASC),
  CONSTRAINT `fk_Dish_Administrator1`
```

```
    FOREIGN KEY (`Administrator_id` , `Administrator_User_id`)
    REFERENCES `expressfood`.`Administrator` (`id` , `User_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Dish_Speciality1`
    FOREIGN KEY (`Speciality_id`)
    REFERENCES `expressfood`.`Speciality` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`Order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Order` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `date` DATE NOT NULL,
  `state` VARCHAR(255) NOT NULL,
  `total_price` DECIMAL(7,2) NOT NULL,
  `Customer_id` INT NOT NULL,
  `Customer_User_id` INT NOT NULL,
  `Address_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Order_Customer1_idx` (`Customer_id` ASC, `Customer_User_id` ASC),
  INDEX `fk_Order_Address1_idx` (`Address_id` ASC),
  CONSTRAINT `fk_Order_Customer1`
    FOREIGN KEY (`Customer_id` , `Customer_User_id`)
    REFERENCES `expressfood`.`Customer` (`id` , `User_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Order_Address1`
    FOREIGN KEY (`Address_id`)
    REFERENCES `expressfood`.`Address` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `expressfood`.`CustomerAddress`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`CustomerAddress` (
  `Customer_id` INT NOT NULL,
  `Customer_User_id` INT NOT NULL,
  `Address_id` INT NOT NULL,
  PRIMARY KEY (`Customer_id`, `Customer_User_id`, `Address_id`),
  INDEX `fk_Customer_Address_Address1_idx` (`Address_id` ASC),
  CONSTRAINT `fk_Customer_Address_Customer1`
    FOREIGN KEY (`Customer_id` , `Customer_User_id`)
    REFERENCES `expressfood`.`Customer` (`id` , `User_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Customer_Address_Address1`
    FOREIGN KEY (`Address_id`)
    REFERENCES `expressfood`.`Address` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `expressfood`.`Bag`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`Bag` (
  `amount` INT NOT NULL,
  `Dish_id` INT NOT NULL,
  `DelivryMan_id` INT NOT NULL,
  `DelivryMan_User_id` INT NOT NULL,
  PRIMARY KEY (`Dish_id`, `DelivryMan_id`, `DelivryMan_User_id`),
  INDEX `fk_Bag_DelivryMan1_idx` (`DelivryMan_id` ASC, `DelivryMan_User_id` ASC),
  CONSTRAINT `fk_Bag_Dish1`
    FOREIGN KEY (`Dish_id`)
    REFERENCES `expressfood`.`Dish` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Bag_DelivryMan1`
    FOREIGN KEY (`DelivryMan_id` , `DelivryMan_User_id`)
    REFERENCES `expressfood`.`DelivryMan` (`id` , `User_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `expressfood`.`CommandLine`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `expressfood`.`CommandLine` (
  `amount` INT NOT NULL,
  `Dish_id` INT NOT NULL,
  `Order_id` INT NOT NULL,
  `DelivryMan_id` INT NOT NULL,
  `DelivryMan_User_id` INT NOT NULL,
  PRIMARY KEY (`Dish_id`, `Order_id`, `DelivryMan_id`, `DelivryMan_User_id`),
  INDEX `fk_CommandLine_Order1_idx` (`Order_id` ASC),
  INDEX `fk_CommandLine_DelivryMan1_idx` (`DelivryMan_id` ASC,
`DelivryMan_User_id` ASC),
  CONSTRAINT `fk_CommandLine_Dish1`
    FOREIGN KEY (`Dish_id`)
    REFERENCES `expressfood`.`Dish` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_CommandLine_Order1`
    FOREIGN KEY (`Order_id`)
    REFERENCES `expressfood`.`Order` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_CommandLine_DelivryMan1`
    FOREIGN KEY (`DelivryMan_id` , `DelivryMan_User_id`)
    REFERENCES `expressfood`.`DelivryMan` (`id` , `User_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

2. Les requêtes SQL:

- **Afficher tous les utilisateurs, classés par ordre alphabétique**

```
SELECT DISTINCT id, last_name AS "Nom", first_name AS "Prénom",
            login AS "Identifiant", password AS "Mot de passe", email AS "Email", phone
            AS "Téléphone"
FROM User
ORDER BY last_name;
```

id	Nom	Prénom	Identifiant	Mot de passe	Email	téléphone
1	Dubois	Paul	dubois00001	Pldb0987	paul.dubois@gmail.com	612325678
2	Durand	Marine	marine00002	duma6574	marine.durand@gmail.com	652304578
10	Duval	Axelle	axelle00010	duax1133	axelle.duval@gmail.com	634009567
7	Fournier	Louise	louise00007	folo3000	louise.fournier@gmail.com	634765893
3	Gras	Camille	camille00003	cmgr0000	camille.gras@gmail.com	610305608
6	Lambert	Michel	michel00006	mila3456	michel.lambert@gmail.com	644784509
8	Le roux	Amelie	amelie00008	amle2345	amelie.leroux@gmail.com	656983356
5	Lemaire	Sophie	sophie00005	lese0234	sophie.lemaire@gmail.com	637264523
9	Mercier	Alfred	alfred00009	meal7800	alfred.mercier@gmail.com	634119076
4	Robert	Daniel	daniel00004	daro0022	daniel.robert@gmail.com	634768921

- **Afficher nom, prénom des administrateurs**

```
SELECT id, last_name AS "Nom", first_name AS "Prénom" FROM User
WHERE EXISTS(
SELECT 1 FROM Administrator
WHERE User.id = User_id);
```

id	Nom	Prénom
2	Durand	Marine
6	Lambert	Michel

- **Afficher nom, prénom des clients**

```
SELECT DISTINCT id, last_name  AS "Nom", first_name AS "Prénom"
FROM User
WHERE id IN
        (SELECT User_id FROM Customer);
```

id	nom	prenom
1	Dubois	Paul
4	Robert	Daniel
8	Le roux	Amelie
9	Mercier	Alfred
10	Duval	Axelle

- **Afficher la liste des Livreurs**

```
SELECT id, last_name AS "Nom", first_name AS "Prénom"
FROM User
WHERE EXISTS
        ( SELECT 1 FROM DelivryMan
        WHERE User.id = User_id);
```

id	Nom	Prénom
3	Gras	Camille
5	Lemaire	Sophie
7	Fournier	Louise

- ## Afficher le statut de tous les livreurs

```
SELECT u.id, u.last_name AS "Nom", u.first_name AS "Prénom", d.status AS "Statut",
            d.latitude AS "Latitude" , d.longitude AS "Longitude"
FROM User u
JOIN  DelivryMan d ON u.id = d.User_id;
```

id	nom	prenom	statut	Latitude	Longitude
3	Gras	Camille	Disponible	48.84273510000000000	2.32834120000006800
5	Lemaire	Sophie	En cours de livraison	48.86272500000000000	2.28759200000000000
7	Fournier	Luoise	Indisponible	48.83202980000001000	2.32743060000007060

- ## Sélectionner tous les livreurs qui sont disponibles

```
SELECT id, last_name AS "Nom", first_name AS "Prénom", login AS "Identifiant",
            password AS "Mot de passe", email AS "Email", phone AS "Téléphone"
FROM User
WHERE id IN
(SELECT User_id FROM DelivryMan WHERE
        DelivryMan.status = 'disponible');
```

id	Nom	Prénom	Identifiant	Mot de passe	Email	Téléphone
3	Gras	Camille	camille00003	cmgr0000	camille.gras@gmail.com	610305608

- ## Sélectionner tous les livreurs qui sont indisponibles

```
SELECT id, last_name AS "Nom", first_name AS "Prénom", login AS "Identifiant",
            password AS "Mot de passe", email AS "Email", phone AS "Téléphone"
FROM User WHERE
id IN
(SELECT User_id FROM DelivryMan WHERE
        DelivryMan.status != 'disponible');
```

id	Nom	Prénom	Identifiant	Mot de passe	Email	Téléphone
5	Lemaire	Sophie	sophie00005	lese0234	sophie.lemaire@gmail.com	637264523
7	Fournier	louise	louise00007	folo3000	louise.fournier@gmail.com	634765893

- **Afficher les commandes passés par les client, ils sont classés par ordre croissant**

```
SELECT u.last_name AS "Nom", u.first_name AS "Prénom", o.date AS "Date", o.state
            AS "Etat de la commande", o.total_price AS "Prix total"
FROM User u
JOIN Customer c ON u.id = c.User_id
JOIN `Order` o ON o.Customer_User_id = c.User_id
ORDER BY u.last_name ASC;
```

Nom	Prénom	Date	Etat de la commande	Prix total
Dubois	Paul	2018-02-08	En cours de livraison	35.50
Dubois	Paul	2018-02-11	Livrée	13.00
Duval	Axelle	2018-02-11	Livrée	20.00
Le roux	Amelie	2018-02-02	Livrée	40.00
Mercier	Alfred	2018-02-13	En cours de livraison	90.00
Robert	Louise	2018-02-02	Livrée	20.00

- **Afficher la spécialité de chaque plat, sa date d'élaboration, son prix et sa quantité**

```
SELECT d.name AS "Plat", s.category AS "Spécialité", d.date AS "Date",
            d.unit_price AS "Prix unitaire", d.amount AS "Quantité"
FROM Dish d
JOIN Speciality s on d.Speciality_id = s.id;
```

Plat	Spécialité	Date	Prix unitaire	Quantité
Boeuf bourguignon	Plat	2018-02-08	15.00	10
Tartare de boeuf aux herbes	Plat	2018-02-08	8.00	12
Tortilla	Plat	2018-02-02	10.00	12
Le Fish and chips	Plat	2018-02-02	10.00	18
Cannellonis	Plat	2018-02-11	25.00	20
Poulet tandoori	Plat	2018-02-11	10.00	13
Canard laqué	Plat	2018-02-13	15.00	12
Moussaka	Plat	2018-02-13	18.00	18

Strudel	Dessert	2018-02-08	6.00	20
Charlotte aux pommes et mousse au chocolat	Dessert	2018-02-08	8.00	20
Gâteau au chocolat	Dessert	2018-02-02	6.00	15
Tarte aux figues	Dessert	2018-02-02	8.00	15
Muffins aux figues parfumées à l'orange	Dessert	2018-02-11	6.00	25
Crumble de mirabelles aux spéculoos	Dessert	2018-02-11	9.00	22
Tarte aux mirabelles	Dessert	2018-02-13	5.00	8
Moelleux au chocolat	Dessert	2018-02-13	7.00	30

- **Afficher le contenu du sac, pour chaque livreur en indiquant les plats, la date de livraison et le nombre de plat**

```
SELECT u.last_name AS "Livreur", h.name AS "Plat", h.date AS "Date",
            b.amount As "Nombre plat"
FROM User u
JOIN DelivryMan d ON u.id =  d.User_id
JOIN Bag b ON b.DelivryMan_User_id = d.User_id
JOIN Dish h ON b.Dish_id = h.id;
```

Livreur	Plat	Date	Nombre plat
Gras	Boeuf bourguignon	2018-02-08	3
Gras	Tartare de boeuf aux herbes	2018-02-08	2
Gras	Strudel	2018-02-08	2
Gras	Charlotte aux pommes et mousse au chocolat	2018-02-08	2
Lemaire	Tortilla	2018-02-02	2
Lemaire	Le Fish and chips	2018-02-02	2
Lemaire	Gâteau au chocolat	2018-02-02	2
Lemaire	Tarte aux figues	2018-02-02	2

Afficher toutes les commandes en attente de livraison

```
SELECT o.id AS "Commande", o.state AS "Etat", u.last_name As "Client",
                a.track_number AS "N°", a.street AS "Rue", a.postal_code AS "Code postal",
               a.city AS "Ville"
From `Order` o
JOIN Customer c ON o.Customer_User_id = c.User_id
JOIN User u ON u.id= c.User_id
JOIN Address a ON o.Address_id = a.id
WHERE o.state != "livrée";
```

Commande	Etat	Client	N°	Rue	Code Postal	Ville
1	En cours de livraison	Dubois	112	Rue Vavin	75006	Paris
6	En cours de livraison	Mercier	112	Rue Vavin	75006	Paris

Afficher tous les plats

```
SELECT d.name AS "Plat", d.description AS "Description", d.amount AS "Quantité",
               d.unit_price AS "Prix unitaire", d.date AS "Date"
FROM Dish d
WHERE EXISTS (
SELECT 1 FROM Speciality s
WHERE s.id = d.Speciality_id
AND s.category= "plat");
```

Plat	Description	Quantité	Prix unitaire	Date
Boeuf bourguignon	Un plat de viande cuit longuement avec un bouillon...	10	15.00	2018-02-08
Tartare de boeuf aux herbes	Le vrai steak tartare est coupé au couteau puis on...	12	8.00	2018-02-08
Tortilla	une omelette épaisse à base de pommes de terre. Un...	12	10.00	2018-02-02
Le Fish and chips	Un menu simple composé de poisson enrobé de chapel...	18	10.00	2018-02-02
Cannellonis	Composé de pâtes avec une sauce bolognaise et parm...	20	25.00	2018-02-11
Poulet tandoori	Préparé avec de filets de poulet, une sauce tandoo...	13	10.00	2018-02-11
Canard laqué	Accompagné de légumes crus.	12	15.00	2018-02-13
Moussaka	Un plat riche et complet cuit au four à base de vi...	18	18.00	2018-02-13

Afficher tous les desserts

```
SELECT d.name AS "Dessert", d.description AS "Description",
            d.amount AS "Quantité", d.unit_price AS "Prix unitaire", d.date AS "Date"
FROM Dish d
WHERE EXISTS (
SELECT 1 FROM Speciality s
WHERE s.id = d.Speciality_id
AND s.category= "dessert");
```

Dessert	description	quantite	prix_unitaire	date
Strudel	Feuilleté aux pommes, strudel, feuilleté aux fruit...	20	6.00	2018-02-08
Charlotte aux pommes et mousse au chocolat	L'alliance des biscuits à la cuillère, trempés ...	20	8.00	2018-02-08
Gâteau au chocolat	Gâteau au chocolat parfumé au fruit de la passion,...	15	6.00	2018-02-02
Tarte aux figues	Tarte aux figues à la crème d'amandes	15	8.00	2018-02-02
Muffins aux figues parfumées à l'orange	Les saveurs des figues violettes, du zeste d'orang...	25	6.00	2018-02-11
Crumble de mirabelles aux spéculoos	Crumble de mirabelles aux spéculoos et sucre vanil...	22	9.00	2018-02-11
Tarte aux mirabelles	Tarte aux mirabelles façon alsacienne	8	5.00	2018-02-13
Moelleux au chocolat	Moelleux au chocolat sans farine, crème à la vanil...	30	7.00	2018-02-13

Liste des plats du jour élaborés le 2018-02-13

```
SELECT d.name AS "Plat", d.description AS "Description", d.amount AS "Quantité",
            d.unit_price AS "Prix unitaire", d.date AS "Date"
FROM Dish d
WHERE EXISTS (
SELECT 1 FROM Speciality s
WHERE s.id = d.Speciality_id
AND s.category= "plat"
AND date = "2018-02-13");
```

Plat	description	quantite	prix_unitaire	date
Canard laqué	Accompagné de légumes crus.	12	15.00	2018-02-13
Moussaka	Un plat riche et complet cuit au four à base de vi...	18	18.00	2018-02-13

- **Modifier le prix et la quantité de plat “boeuf bourguignon”**

```
UPDATE Dish
SET amount= 30,
        unit_price= 17.00
WHERE id = 1;
```

id	Nom	Description	Quantité	Prix unitaire	Date	Administrateur_ Utilisateur_id	specialite_id
1	Boeuf bourguignon	Un plat de viande cuit longuement avec un bouillon...	30	17.00	2018-02-08	2	1

- **Mettre les noms de tous les utilisateurs en majuscule**

```
UPDATE User
SET last_name = upper(last_name);
```

On affiche les utilisateurs, nom et prénom dans la même colonne

```
SELECT  id, CONCAT(last_name,' ',first_name) AS "Utilisateurs"
FROM User;
```

id	Utilisateurs
1	DUBOIS Paul
2	DURAND Marine
3	GRAS Laurent
4	ROBERT Louise
5	LEMAIRE David
6	LAMBERT Michel
7	FOURNIER Richard
8	LE ROUX Amelie
9	MERCIER Alfred
10	DUVAL Axelle