
Documentation technique BDD

Diapazen

Sommaire

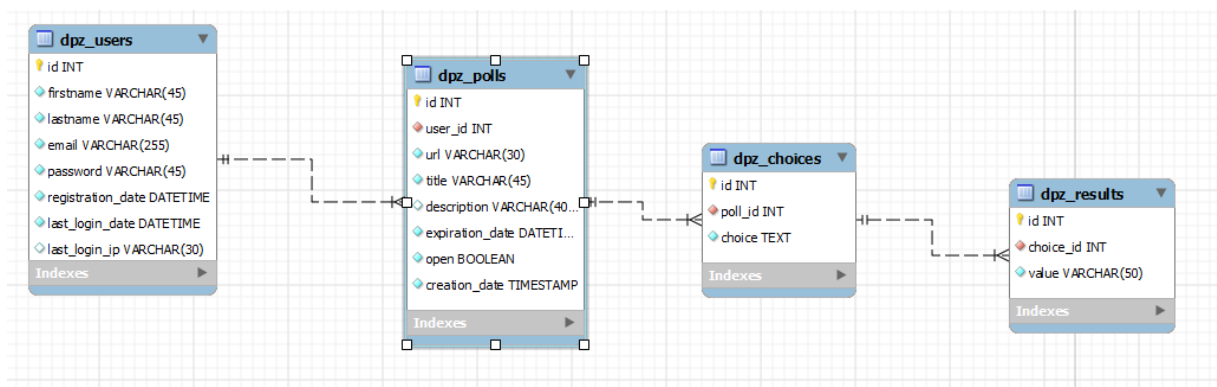
Introduction.....	2
Création de la base et du schéma	1
Table : Utilisateurs.....	1
Table : Sondage	1
Table : Choix	2
Table : Résultats	2
Vue : Connexion	3
Vue : Utilisateurs (sans mot de passe)	3
Vue : Jointure entre les utilisateurs et les sondages.....	4
Vue : Jointure entre les sondages et les choix possible	5
Vue : Jointure entre les choix et les votes.....	6

Introduction

Le site utilisera le SGBDR **MySQL 5.5.X** d'Oracle avec le moteur **InnoDB**. La base sera modélisée grâce au logiciel de conception MySQL WorkBench par Oracle. Les images suivantes sont tirées du script extrait de ce logiciel.

Il y a un total de 4 tables et 5 vues.

Notre système de table est basé sur une cascade des clefs primaires :



Chaque table (sauf user) référence la clef primaire de la table précédente en tant que clef étrangère. C'est-à-dire :

- Un sondage ne peut exister s'il ne référence pas un utilisateur valide (via son identifiant)
- Un choix ne peut exister s'il ne référence pas un sondage valide (via son identifiant)
- Un résultat ne peut exister s'il ne référence pas un choix valide (via son identifiant)

La validité des clefs étrangères est essentielle.

Création de la base et du schéma

```
1 DROP DATABASE IF EXISTS diapazen;
2
3 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
6
7 CREATE SCHEMA IF NOT EXISTS `diapazen` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci ;
8 USE `diapazen` ;
9
10
```

Table : Utilisateurs

```
11  -- -----
12  -- Table `diapazen`.`dpz_users`
13  -- -----
14  CREATE TABLE IF NOT EXISTS `diapazen`.`dpz_users` (
15    `id` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
16    `firstname` VARCHAR(45) NOT NULL ,
17    `lastname` VARCHAR(45) NOT NULL ,
18    `email` VARCHAR(255) NOT NULL ,
19    `password` VARCHAR(60) NOT NULL ,
20    `registration_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
21    `last_login_date` timestamp NULL ,
22    `last_login_ip` VARCHAR(30) NULL ,
23    PRIMARY KEY (`id`) ,
24    UNIQUE INDEX `id_UNIQUE` (`id` ASC) )
25  ENGINE = InnoDB;
```

Table : Sondage

```
28  -- -----
29  -- Table `diapazen`.`dpz_polls`
30  -- -----
31  CREATE TABLE IF NOT EXISTS `diapazen`.`dpz_polls` (
32    `id` INT NOT NULL AUTO_INCREMENT ,
33    `user_id` INT UNSIGNED NOT NULL ,
34    `url` VARCHAR(30) NOT NULL ,
35    `title` VARCHAR(45) NOT NULL ,
36    `description` VARCHAR(400) NULL ,
37    `creation_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
38    `expiration_date` timestamp NULL ,
39    `open` TINYINT(1) NOT NULL ,
40    PRIMARY KEY (`id`) ,
41    UNIQUE INDEX `id_UNIQUE` (`id` ASC) ,
42    INDEX `fk_dp_polls_dp_users_idx` (`user_id` ASC) ,
43    CONSTRAINT `fk_dp_polls_dp_users`
44      FOREIGN KEY (`user_id`)
45      REFERENCES `diapazen`.`dpz_users` (`id` )
46      ON DELETE NO ACTION
47      ON UPDATE NO ACTION)
48  ENGINE = InnoDB;
```

Table : Choix

```
51  -- -----
52  -- Table `diapazen`.`dpz_choices`
53  -- -----
54  CREATE TABLE IF NOT EXISTS `diapazen`.`dpz_choices` (
55    `id` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
56    `poll_id` INT NOT NULL ,
57    `choice` TEXT NOT NULL ,
58    PRIMARY KEY (`id`) ,
59    UNIQUE INDEX `id_UNIQUE` (`id` ASC) ,
60    INDEX `fk_dp_choices_dp_polls1_idx` (`poll_id` ASC) ,
61    CONSTRAINT `fk_dp_choices_dp_polls1`
62      FOREIGN KEY (`poll_id` )
63      REFERENCES `diapazen`.`dpz_polls` (`id` )
64      ON DELETE NO ACTION
65      ON UPDATE NO ACTION)
66  ENGINE = InnoDB;
```

Table : Résultats

```
69  -- -----
70  -- Table `diapazen`.`dpz_results`
71  -- -----
72  CREATE TABLE IF NOT EXISTS `diapazen`.`dpz_results` (
73    `id` INT NOT NULL AUTO_INCREMENT ,
74    `choice_id` INT UNSIGNED NOT NULL ,
75    `value` VARCHAR(50) NOT NULL ,
76    PRIMARY KEY (`id`) ,
77    INDEX `fk_dp_results_dp_choices1_idx` (`choice_id` ASC) ,
78    CONSTRAINT `fk_dp_results_dp_choices1`
79      FOREIGN KEY (`choice_id` )
80      REFERENCES `diapazen`.`dpz_choices` (`id` )
81      ON DELETE NO ACTION
82      ON UPDATE NO ACTION)
83  ENGINE = InnoDB;
84
85  USE `diapazen` ;
86
87
88  SET SQL_MODE=@OLD_SQL_MODE;
89  SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
90  SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Vue : Connexion

```
101 CREATE OR REPLACE VIEW diapazen.dpz_view_connexion AS
102 SELECT
103     dpz_users.id,
104     dpz_users.firstname,
105     dpz_users.lastname,
106     dpz_users.email,
107     dpz_users.password
108 FROM diapazen.dpz_users;
```

Cette vue nous permet de récupérer les informations essentielles à la connexion.

Le mot de passe crypté n'est accessible que par cette vue et en lecture seule, pour un souci de sécurité.

Vue : Utilisateurs (sans mot de passe)

```
110 CREATE OR REPLACE VIEW diapazen.dpz_view_users AS
111 SELECT
112     dpz_users.id,
113     dpz_users.firstname,
114     dpz_users.lastname,
115     dpz_users.email,
116     dpz_users.registration_date,
117     dpz_users.last_login_date,
118     dpz_users.last_login_ip
119 FROM diapazen.dpz_users;
```

Cette vue contient tous les champs de la table users sauf le mot de passe. Elle est utilisée pour toutes les requêtes concernant les utilisateurs (sauf la connexion).

Vue : Jointure entre les utilisateurs et les sondages

```
121 CREATE OR REPLACE VIEW diapazen.dpz_view_users_join_polls AS
122 SELECT
123     dpz_users.id AS USER_ID,
124     dpz_users.firstname,
125     dpz_users.lastname,
126     dpz_users.email,
127     dpz_users.registration_date,
128     dpz_users.last_login_date,
129     dpz_users.last_login_ip,
130     dpz_polls.creation_date,
131
132     dpz_polls.id AS POLL_ID,
133     dpz_polls.url,
134     dpz_polls.title,
135     dpz_polls.description,
136     dpz_polls.expiration_date,
137     dpz_polls.open
138 FROM diapazen.dpz_users
139 INNER JOIN diapazen.dpz_polls
140 ON dpz_users.id=dpz_polls.user_id
141 ORDER BY USER_ID ASC;
```

Cette vue est une jointure interne entre les utilisateurs et les sondages. Ainsi, on se sert de cette vue pour :

- Récupérer tous les sondages associés à un utilisateur :
On dispose de l'identifiant d'un utilisateur. On veut savoir quels sont ses sondages.
- Savoir à quel utilisateur appartient un sondage
On dispose de l'identifiant d'un sondage. On veut savoir à quel utilisateur il appartient.

Vue : Jointure entre les sondages et les choix possible

```
151 CREATE OR REPLACE VIEW diapazen.dpz_view_poll AS
152 SELECT
153     dpz_polls.id AS POLL_ID,
154     dpz_polls.url,
155     dpz_polls.title,
156     dpz_polls.description,
157     dpz_polls.expiration_date,
158
159     dpz_choices.id AS CHOICE_ID,
160     dpz_choices.choice
161
162 FROM diapazen.dpz_polls
163 INNER JOIN diapazen.dpz_choices
164 ON dpz_polls.id=dpz_choices.poll_id
165 ORDER BY POLL_ID ASC;
```

Cette vue est une jointure entre les sondages et les choix. Ainsi, on se sert de cette vue pour :

- Récupérer tous les choix associés à un sondage :

On dispose de l'identifiant d'un sondage. On veut savoir quels sont ses choix.

- Savoir à quel sondage appartient un choix

On dispose de l'identifiant d'un choix. On veut savoir à quel sondage il appartient.

Vue : Jointure entre les choix et les votes

```
167 CREATE OR REPLACE VIEW diapazen.dpz_view_choice AS
168 SELECT
169     dpz_choices.id AS CHOICE_ID,
170     dpz_choices.poll_id AS POLL_ID,
171     dpz_choices.choice,
172
173     dpz_results.value
174 FROM diapazen.dpz_choices
175 INNER JOIN diapazen.dpz_results
176 ON dpz_choices.id=dpz_results.choice_id
177 ORDER BY CHOICE_ID ASC;
```

Cette vue est une jointure entre les choix et les votes. Ainsi, on se sert de cette vue pour :

- Récupérer tous les votes associés à un choix :

On dispose de l'identifiant d'un choix. On veut savoir quels sont ses votes.

- Savoir à quel choix appartient un vote

On dispose de l'identifiant d'un vote. On veut savoir à quel choix il appartient.