

Тема: модифікація структури таблиць бази даних засобами sql.

Мета: розробити sql запити для зміни структури таблиць бази даних (включення нових полів, видалення полів таблиць, зміни опису полів, зміни обмежень).

Хід роботи

1. Видалення зв'язку між таблицями шляхом видалення обмеження для таблиці:

```
ALTER TABLE Shop.Storage  
DROP FOREIGN KEY Storage_Shop,  
DROP INDEX Storage_Shop;
```

Для перевірки результату можна виконати команди

```
SHOW INDEX FROM Storage  
SHOW CREATE TABLE Storage
```

```
mysql> ALTER TABLE Shop.Storage  
-> DROP FOREIGN KEY Storage_Shop,  
-> DROP INDEX Storage_Shop;  
Query OK, 0 rows affected (0.29 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> SHOW INDEX FROM Storage;  
+-----+-----+-----+-----+-----+-----+  
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Col |  
| Index_type | Comment | Index_comment | Visible | Expression |  
+-----+-----+-----+-----+-----+-----+  
| storage | 0 | PRIMARY | 1 | idStorage | A  
| BTREE | | | YES | NULL |  
+-----+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> SHOW CREATE TABLE Storage;  
+-----+-----+  
| Table | Create Table  
+-----+-----+  
| Storage | CREATE TABLE `storage` (  
  `idStorage` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,  
  `phone` char(10) NOT NULL,  
  `idShop` mediumint(8) unsigned NOT NULL,  
  PRIMARY KEY (`idStorage`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

2. Видалення поля і зміна розмірності поля в таблиці Storage:

```

ALTER TABLE Shop.Storage
DROP COLUMN idShop,
MODIFY COLUMN phone char(13) NOT NULL;

```

```

mysql> ALTER TABLE Shop.Storage
      -> DROP COLUMN idShop,
      -> MODIFY COLUMN phone char(13) NOT NULL;
Query OK, 0 rows affected (0.99 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW CREATE TABLE Storage;
+-----+-----+
| Table | Create Table
+-----+-----+
| Storage | CREATE TABLE `storage` (
  `idStorage` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
  `phone` char(13) NOT NULL,
  PRIMARY KEY (`idStorage`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

3. Зробимо поле email в таблиці Producer необов'язковим, і змінимо його розмірність:

```

ALTER TABLE Shop.Producer
MODIFY COLUMN email CHAR(60) NULL;

```

```

mysql> ALTER TABLE Shop.Producer
      -> MODIFY COLUMN email CHAR(60) NULL;
Query OK, 0 rows affected (0.79 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Producer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProducer | mediumint(8) unsigned | NO | PRI | NULL | auto_increment |
| name | varchar(50) | NO | | NULL | |
| address | varchar(99) | NO | | NULL | |
| email | char(60) | YES | | NULL | |
| phone | char(10) | NO | | NULL | |
| contact_person | varchar(50) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

4. Додамо до таблиці Producer поле idBrand, щоб знати з якими брендами працює виробник. Спочатку змінимо поле idProducer, забравши лічильник (властивість auto_increment). Після додавання поля idBrand, створимо нове обмеження унікальності для полів (idProducer, idBrand).

```

ALTER TABLE Shop.Producer
MODIFY idProducer MEDIUMINT(8) UNSIGNED NOT NULL,
ADD COLUMN `idBrand` MEDIUMINT(8) UNSIGNED NOT NULL
AFTER idProducer,

```

```
ADD CONSTRAINT `Producer_Brand_UNIQUE` UNIQUE
(`idProducer`, `idBrand`);
```

```
mysql> ALTER TABLE Shop.Producer
-> MODIFY idProducer MEDIUMINT(8) UNSIGNED NOT NULL,
-> ADD COLUMN `idBrand` MEDIUMINT(8) UNSIGNED NOT NULL AFTER idProducer,
-> ADD CONSTRAINT `Producer_Brand_UNIQUE` UNIQUE (`idProducer`, `idBrand`);
Query OK, 0 rows affected (1.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW CREATE TABLE Producer;
+-----+
| Producer | CREATE TABLE `producer` (
  `idProducer` mediumint(8) unsigned NOT NULL,
  `idBrand` mediumint(8) unsigned NOT NULL,
  `name` varchar(50) NOT NULL,
  `address` varchar(99) NOT NULL,
  `email` char(60) DEFAULT NULL,
  `phone` char(10) NOT NULL,
  `contact_person` varchar(50) NOT NULL,
  PRIMARY KEY (`idProducer`),
  UNIQUE KEY `Producer_Brand_UNIQUE` (`idProducer`, `idBrand`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

- Змінимо обмеження цілісності Thing_Type так, щоб при видаленні категорії з бази видалявся запис про приєднаний до нього вид продукту. Для цього потрібно видалити обмеження і створити нове, оскільки MySQL не підтримує директиви ALTER CONSTRAINT.

```
ALTER TABLE Shop.Thing_Type
DROP FOREIGN KEY `Thing_Type_Category1`,
ADD CONSTRAINT `Thing_Type_Category` FOREIGN KEY
(`idCategory`)
REFERENCES Shop.Category (`idCategory`) ON DELETE
CASCADE ON UPDATE NO ACTION;
```

```
mysql> ALTER TABLE Shop.Thing_Type
-> DROP FOREIGN KEY `Thing_Type_Category1`,
-> ADD CONSTRAINT `Thing_Type_Category` FOREIGN KEY (`idCategory`)
-> REFERENCES Shop.Category (`idCategory`)
-> ON DELETE CASCADE ON UPDATE NO ACTION;
Query OK, 0 rows affected (1.18 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW CREATE TABLE Thing_Type;
+-----+
| Thing_Type | CREATE TABLE `thing_type` (
  `idThingType` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `number` int(10) unsigned NOT NULL,
  `idCategory` mediumint(8) unsigned NOT NULL,
  PRIMARY KEY (`idThingType`),
  UNIQUE KEY `name_UNIQUE` (`name`),
  KEY `Thing_Type_Category` (`idCategory`),
  CONSTRAINT `Thing_Type_Category` FOREIGN KEY (`idCategory`) REFERENCES `cat
egory` (`idCategory`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

Висновки

Було розглянуто методи модифікації структури таблиць БД засобами SQL та проведено модифікацію полів, зав'язків і обмежень, зокрема

розроблено SQL-запити для включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень.