SQL????

??????

1. 学生のデータベースを作ろうとし、以下のデータを管理したい:

- 学生 ID (数値)
- お名前 (テキスト)
- 生年月日 (日付)

SQL文

```
CREATE TABLE students (id INT, name TEXT, birthday DATE);
```

---

2. プレーヤーのログイン情報を管理するテーブルを作成し、以下のデータを管理したい:

- プレーヤー ID (数値)
- ユーザー名 (テキスト - 255文字まで)
- パスワード (テキスト - 255文字まで)
- 最後のログイン (日時)

SQL文

```
CREATE TABLE players (id INT, name VARCHAR(255), password VARCHAR(255), last_login
DATETIME);
```

??????

3. 学生のテーブルに2つの学生を追加してください:

- 大原太郎
  - 生年月日: 2006年 5月 17日
  - ID: 1
- 盛岡一郎

- 生年月日: 2006年 3月 29日
- ID: 2

- 石割さくら
  - 生年月日: 2004年 6月 4日
  - ID: 3

SQL文

```
INSERT INTO students SET id=1, name="大原太郎", birthday="2006-5-17";
INSERT INTO students SET id=2, name="盛岡一郎", birthday="2006-3-29";
INSERT INTO students SET id=3, name="石割さくら", birthday="2004-6-4";
```

または

```
INSERT INTO students(id, name, birthday) VALUES
    (1,'大原太郎','2006-05-15'),
    (2,'盛岡一郎','2006-03-29'),
    (3,'石割さくら','2004-6-4');
```

4.

プレーヤーテーブルに1つのプレーヤーを追加してください。ユーザー名とパスワードは適当で、最後のログインは今日の朝 10時にしてください。

SQL文

```
INSERT INTO players SET id="1", name="test_player", password="qwerty", last_login='2026-09-02 10:00:00';
```

または

```
INSERT INTO players (id, name, password, last_login) VALUES
    ('1', 'test_player', 'qwerty', '2026-09-02 10:00:00');
```

??????

5. 大原太郎（ID 1）の生年月日を訂正し、2006年 5月 25日に直してください。

SQL文

```
UPDATE students SET birthday='2006-5-25' WHERE id=1
```

---

6. プレーヤーのパスワードを「 pass1234」に更新し、最後のログインは、今日の午後 3時 30 分にしてください。

SQL文

```
UPDATE players SET password="pass1234", last_login="2026-09-02 15:40:00" WHERE id=1;
```

??????

7. 盛岡一郎（ ID2）が学校を辞めたので、削除してください

SQL文

```
DELETE FROM students WHERE id=2;
```

??????

8. 2006年生まれの学生を求めてください（ 2006年 1月 1日～ 2026年 12月 31日）

SQL文

```
SELECT * FROM students WHERE birthday >= "2006-01-01" AND birthday <= "2026-12-31";
```

9. プレーヤーの名前だけを表示してください。

SQL文

```
SELECT name FROM players;
```

??????

10. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

SQL□

```
DROP TABLE players, students;
```

??????

□□□□□□□□□□□□□□ SQL□□□□□□□□□□□□□□

□□□□□□□□□□

```
CREATE TABLE `items` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `name` TINYTEXT NOT NULL ,
  `cost` SMALLINT NOT NULL , PRIMARY KEY (`id`));

INSERT INTO `items` (`id`, `name`, `cost`) VALUES
  (1, '□□□□□□', '100'),
  (2, '□□□□□□□', '500'),
  (3, '□□□□', '600');
```

□□□□□□□□□□□

```
CREATE TABLE `players` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `name` TEXT NOT NULL , PRIMARY KEY (`id`));

INSERT INTO `players` (`id`, `name`) VALUES
  (1, '□□□□□□□'),
  (2, '□□□□□□□');
```

各プレーヤーが持っているアイテムの関係テーブル

```
CREATE TABLE `player_items` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `player_id` INT NOT NULL ,
  `item_id` INT NOT NULL ,
  `count` INT NOT NULL , PRIMARY KEY (`id`));

INSERT INTO `player_items` (`id`, `player_id`, `item_id`, `count`) VALUES
  (1, 1, 1, 2),
  (2, 1, 3, 1),
  (3, 2, 1, 5),
  (4, 2, 2, 1);
```

この場合は:

11. player_itemsと playersのテーブルを id で連携し、すべてを表示してください

SQL文

```
SELECT *
FROM player_items, players
WHERE player_id = players.id;
```

12. 上記の SQL文を編集し、 items の連携を追加してください（つまり、各プレーヤーのアイテムを表示）

SQL文

```
SELECT *
FROM player_items, players, items
WHERE player_id = players.id AND item_id = items.id;
```

13. 上記の SQL文を編集し、 id を隠し、この順番でカラムを表示してください:

プレーヤー名 | アイテム名 | 数

SQL文

```
SELECT players.name, items.name, player_items.count
FROM player_items, players, items
WHERE player_id = players.id AND item_id = items.id;
```

Revision #1
Created 2026-09-02 03:17:31 UTC by Menendez Francisco
Updated 2026-09-02 03:18:03 UTC by Menendez Francisco