

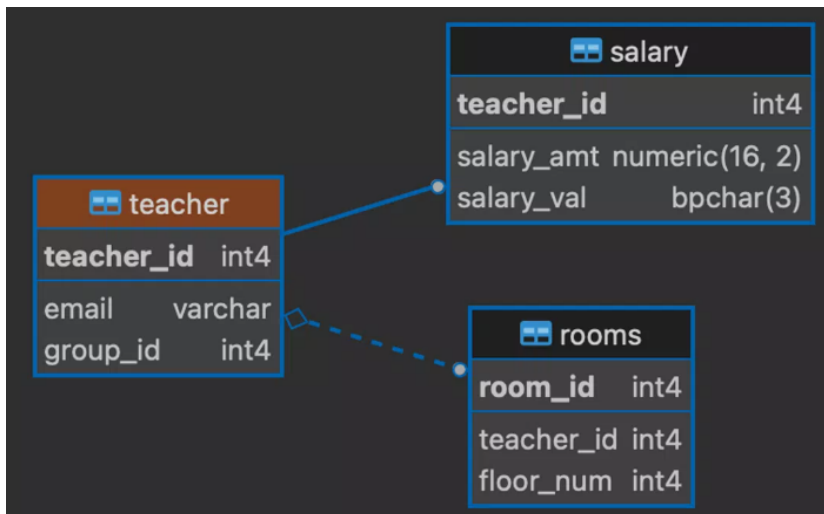
Создание таблиц, SELECT- запросы.

1. Создайте таблицу salary (teacher_id, salary_amt, salary_val)

Решение:

```
CREATE TABLE salary (  
    teacher_id INTEGER,  
    salary_amt NUMERIC(16,2),  
    salary_val CHAR(3)  
);
```

2. Создайте таблицу rooms (кабинеты). Поля таблицы и их тип можно найти на схеме БД.



Решение:

```
CREATE TABLE rooms (  
    room_id INTEGER,  
    teacher_id INTEGER,  
    floor_num INTEGER  
);
```

3. Заполните таблицу salary данными.

Решение:

```
INSERT INTO salary (teacher_id, salary_amt, salary_val)  
VALUES  
(965, 70000, 'RUR'),  
(29091, 100000, 'RUR'),  
(29410, 1000, 'EUR'),  
(33576, 1800, 'USD'),  
(33902, 1200, 'EUR');
```

4. Заполните таблицу rooms следующими данными:

Номер комнаты	id учителя	Номер этажа
1	965	1
2	29091	1
3	29410	1
10	33576	2
11	33902	2

Решение:

```
INSERT INTO rooms (room_id, teacher_id, floor_num)
VALUES
(1, 965, 1),
(2, 29091, 1),
(3, 29410, 1),
(10, 33576, 2),
(11, 33902, 2);
```

5. Выведите id учителей с кабинетами на втором этаже.

Решение:

```
SELECT teacher_id, floor_num FROM rooms WHERE floor_num = 2;
```

6. Выведите id учителей, которые получают зарплату в евро.

Решение:

```
SELECT teacher_id, salary_val FROM salary WHERE salary_val = 'EUR';
```

7. Даны две таблицы:

Посты (posts)

post_id	text_len	post_date
1	6799	09.03.2023
2	7800	12.03.2023
3	6907	19.03.2023
4	8890	22.03.2023
5	9802	30.03.2023

Теги (tags)

post_id	tag
1	nofilter
2	happy
3	joy
4	music
5	dark

Создайте таблицы в БД и заполните их данными.

Решение:

```
CREATE TABLE posts (
```

```
    post_id INTEGER,
```

```
    text_len INTEGER,
```

```
    post_date DATE);
```

```
INSERT INTO posts (post_id, text_len, post_date)
```

```
VALUES
```

```
(1, 6799, '2023.03.09'),
```

```
(2, 7800, '2023.03.12' ),
```

```
(3, 6907, '2023.03.19'),
```

```
(4, 8890, '2023.03.22'),
```

```
(5, 9802, '2023.03.30');
```

```
CREATE TABLE tags (
```

```
    post_id INTEGER,
```

```
    tag TEXT);
```

```
INSERT INTO tags (post_id, tag)
```

```
VALUES
```

```
(1, 'nofilter'),
```

```
(2, 'happy' ),
```

```
(3, 'joy'),
```

```
(4, 'music'),
```

```
(5, 'dark');
```

8. Выведите данные по постам, которые длиннее 8000 символов.

Решение:

```
SELECT * FROM posts WHERE text_len > 8000;
```

9. Выведите id постов, которые отмечены тегами happy или joy.

Решение:

```
SELECT * FROM tags WHERE tag = 'happy' OR tag = 'joy';
```

10. Выведите id постов, которые были размещены до 10 марта 2023.

Решение:

```
SELECT post_id, post_date FROM posts WHERE post_date < '2023.03.10';
```

11. Добавьте пост с id 6, сделанный 31 марта 2023, содержащий 10782 символ и отмеченный тегом love (два INSERT в обе таблицы).

Решение:

```
INSERT INTO posts (post_id, text_len, post_date) VALUES (6, 10782, '2023.03.31');
```

```
INSERT INTO tags (post_id, tag) VALUES (6, 'love');
```

12. Измените тег для поста с id =5 с dark на live.

Решение:

```
UPDATE tags SET tag = 'live' WHERE post_id = 5;
```