

## SQL Query Assignment 2: ToDo App

---

### 1. List users who have at least one completed todo.

```
SELECT * FROM users
WHERE id IN (
  SELECT user_id FROM todos WHERE status = 'completed'
);
```

---

### 2. Find todos that belong to the same category as the todo titled 'Submit Report'.

```
SELECT * FROM todos
WHERE category_id = (
  SELECT category_id FROM todos WHERE title = 'Submit Report'
);
```

---

### 3. Show all users whose email is not used in any todo.

```
SELECT * FROM users
WHERE id NOT IN (
  SELECT DISTINCT user_id FROM todos
);
```

---

### 4. List users whose total number of todos is equal to the maximum todos of any user.

```
SELECT user_id, COUNT(*) AS todo_count
FROM todos
GROUP BY user_id
HAVING COUNT(*) = (
  SELECT MAX(todo_count) FROM (
    SELECT COUNT(*) AS todo_count
    FROM todos
    GROUP BY user_id
  ) AS counts
);
```

---

**5. List users along with the number of completed todos they have.**

```
SELECT
  u.id,
  u.name,
  (
    SELECT COUNT(*) FROM todos t
    WHERE t.user_id = u.id AND t.status = 'completed'
  ) AS completed_todos
FROM users u;
```

---

**6. Find todos that are due after the average due date of that user's own todos.**

```
SELECT * FROM todos t1
WHERE due_date > (
  SELECT AVG(EXTRACT(EPOCH FROM due_date))::BIGINT::DATE
  FROM todos t2
  WHERE t2.user_id = t1.user_id
);
```

---

**7. Display each user and their earliest todo title.**

```
SELECT u.name, t.title
FROM todos t
JOIN users u ON t.user_id = u.id
WHERE t.due_date = (
  SELECT MIN(due_date)
  FROM todos t2
  WHERE t2.user_id = t.user_id
);
```

---

**8. Add a column priority (values: 'low', 'medium', 'high') to the todos table.**

```
ALTER TABLE todos
ADD COLUMN priority VARCHAR(10);
```

---

**9. Make sure priority only allows those 3 values using a CHECK constraint.**

```
ALTER TABLE todos
ADD CONSTRAINT chk_priority
CHECK (priority IN ('low', 'medium', 'high'));
```

---

**10. Rename the description column in todos to details.**

```
ALTER TABLE todos
RENAME COLUMN description TO details;
```

---

**11. Drop the email column from the users table.**

```
ALTER TABLE users
DROP COLUMN email;
```

---

**12. Add a foreign key to todos(user\_id) referencing users(id) with ON DELETE CASCADE.**

```
ALTER TABLE todos
ADD CONSTRAINT fk_user
FOREIGN KEY (user_id) REFERENCES users(id)
ON DELETE CASCADE;
```

---

**13. List todos that belong to categories with more than 2 total todos.**

```
SELECT * FROM todos
WHERE category_id IN (
  SELECT category_id
  FROM todos
  GROUP BY category_id
  HAVING COUNT(*) > 2
);
```

---

**14. Show users who have todos in every category.**

```
SELECT u.id, u.name
FROM users u
JOIN todos t ON u.id = t.user_id
GROUP BY u.id, u.name
HAVING COUNT(DISTINCT t.category_id) = (
    SELECT COUNT(*) FROM categories
);
```