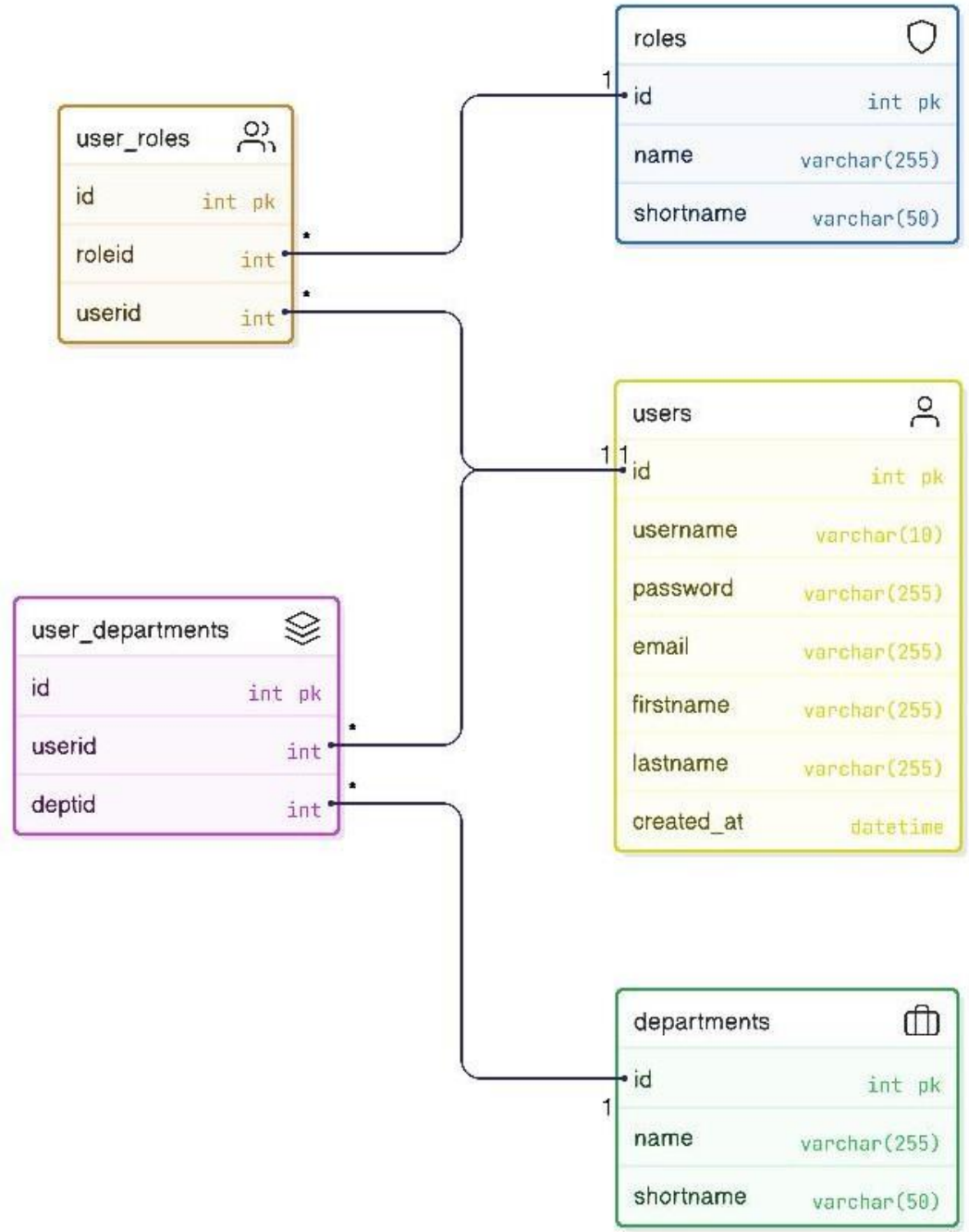

users: Stores information about the people using the system (e.g., their ID, username, password, name)
roles: Stores the different types of permissions or titles a user can have (e.g., Admin, Editor, Guest).
departments: Stores the different organizational groups or departments (e.g., Marketing, Sales, HR).
user_roles: Links users to the roles they have.
user_departments: Links users to the departments they belong to.

User-to-Role Relationship

Tables **users**, **roles**, joined as **user_roles**.
A single user can have many roles. A single role can be assigned to many users. The **user_roles** table acts as a bridge. It contains a userid and a roleid, allowing it to record every combination.

User-to-Department Relationship

Tables **users**, **departments**, joined table as **user_departments**.
Single user can belong to many departments. Conversely, a single department will have many users. The **user_departments** table acts as a connection between the said tables. It contains a userid and a deptid, allowing it to record which users belong to which departments.

Now, generate an SQL query for the following requests. You may use any SQL dialect (PostgreSQL, MySQL, or others).
- Return all entries from the users table where the first name is “Lloyd”.

```
SELECT * FROM users WHERE firstname = 'Lloyd';
```

- Return all the usernames and dates of creation of the ten (10) most recently registered users. Sort them by creation date in ascending order.

```
SELECT username, created_at FROM (
SELECT username, created_at FROM users
ORDER BY created_at DESC LIMIT 10
 ) AS recent_users ORDER BY created_at ASC;
```

- Return the usernames, first names, last names, roles, and department names of all users who:
  - Belong to the "Admin" role;
  - And work in the "IT" department.

```
SELECT
   u.username,
   u.firstname,
   u.lastname,
   r.name AS role_name,
   d.name AS department_name
FROM users AS u
JOIN user_roles AS ur ON u.id = ur.userid
JOIN roles AS r ON ur.roleid = r.id
JOIN user_departments AS ud ON u.id = ud.userid
JOIN departments AS d ON ud.deptid = d.id
WHERE
   r.name = 'Admin' AND d.shortname = 'IT';
```

- Create a query that displays the user's first name, last name, email, role shortname, and department name.

```
SELECT
    u.firstname,
    u.lastname,
    u.email,
    r.shortname AS role_shortname,
    d.name AS department_name
  FROM users AS u
  JOIN user_roles AS ur ON u.id = ur.userid
  JOIN roles AS r ON ur.roleid = r.id
  JOIN user_departments AS ud ON u.id = ud.userid
  JOIN departments AS d ON ud.deptid = d.id;
```

- Create a new column in the users table called updated_at with data type datetime.

```
ALTER TABLE users
  ADD COLUMN updated_at DATETIME;
```

- Delete the updated_at column from the users table.

```
ALTER TABLE users
  DROP COLUMN updated_at;
```

- Update the departments table to set the name value of all rows to "moodLearning Employee" where the short name is "ML".

```
UPDATE departments
  SET name = 'moodLearning Employee'
  WHERE shortname = 'ML';
```

-- Create a new table named department_time_schedules with the following fields:

- id - integer with max 11 digits, primary key, not null, serial
- deptid - integer with max 11 digits, references departments.id
- hours - integer with max 2 digits, not null

```
CREATE TABLE department_time_schedules (
   id INT(11) NOT NULL AUTO_INCREMENT,
   deptid INT(11) NOT NULL,
   hours INT(2) NOT NULL,
   PRIMARY KEY (id),
   FOREIGN KEY (deptid) REFERENCES departments(id)
);
```