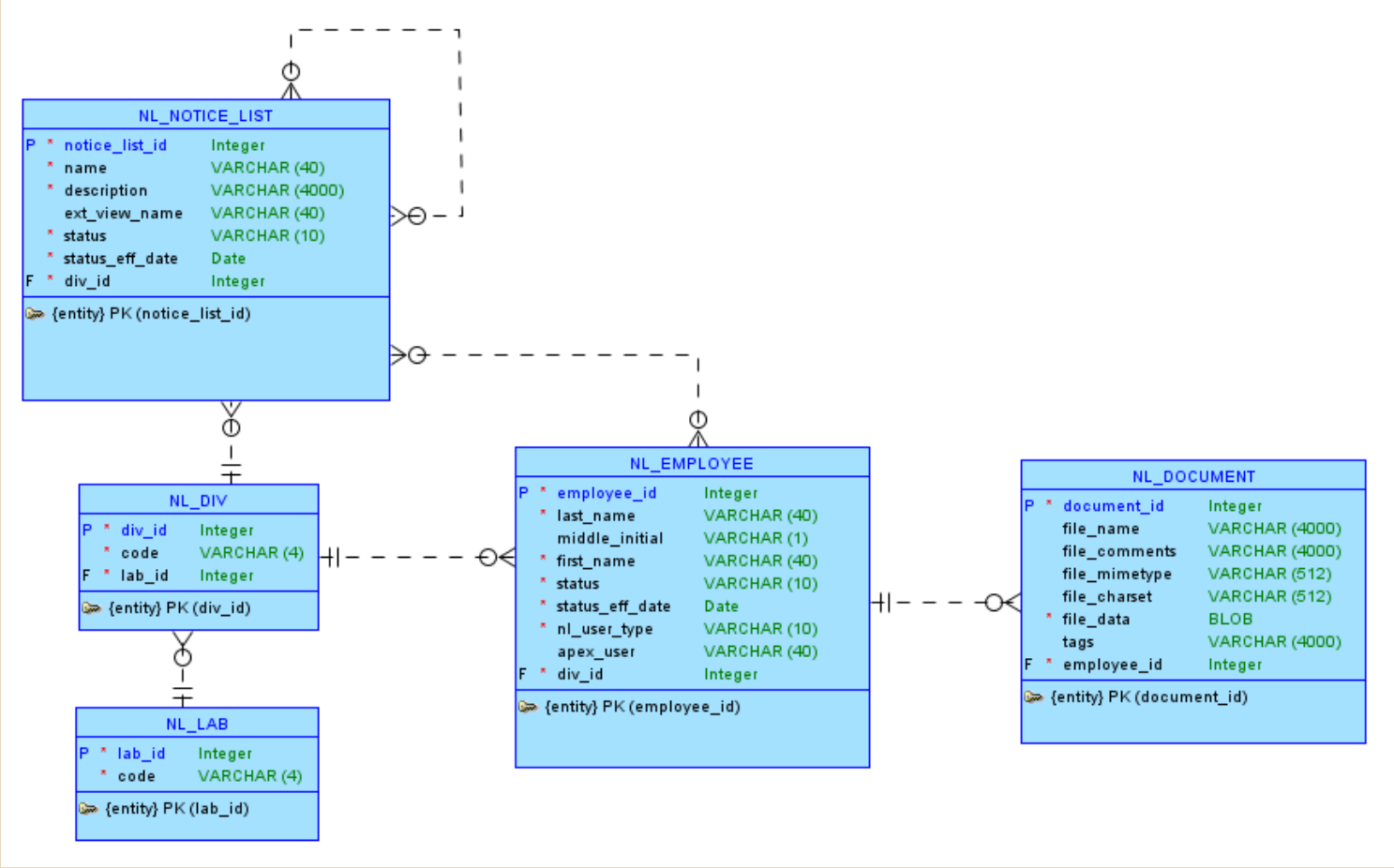
ARL Notice List Project

CS347 Final Presentation

Steven Ritchie, Andrew Nguyen

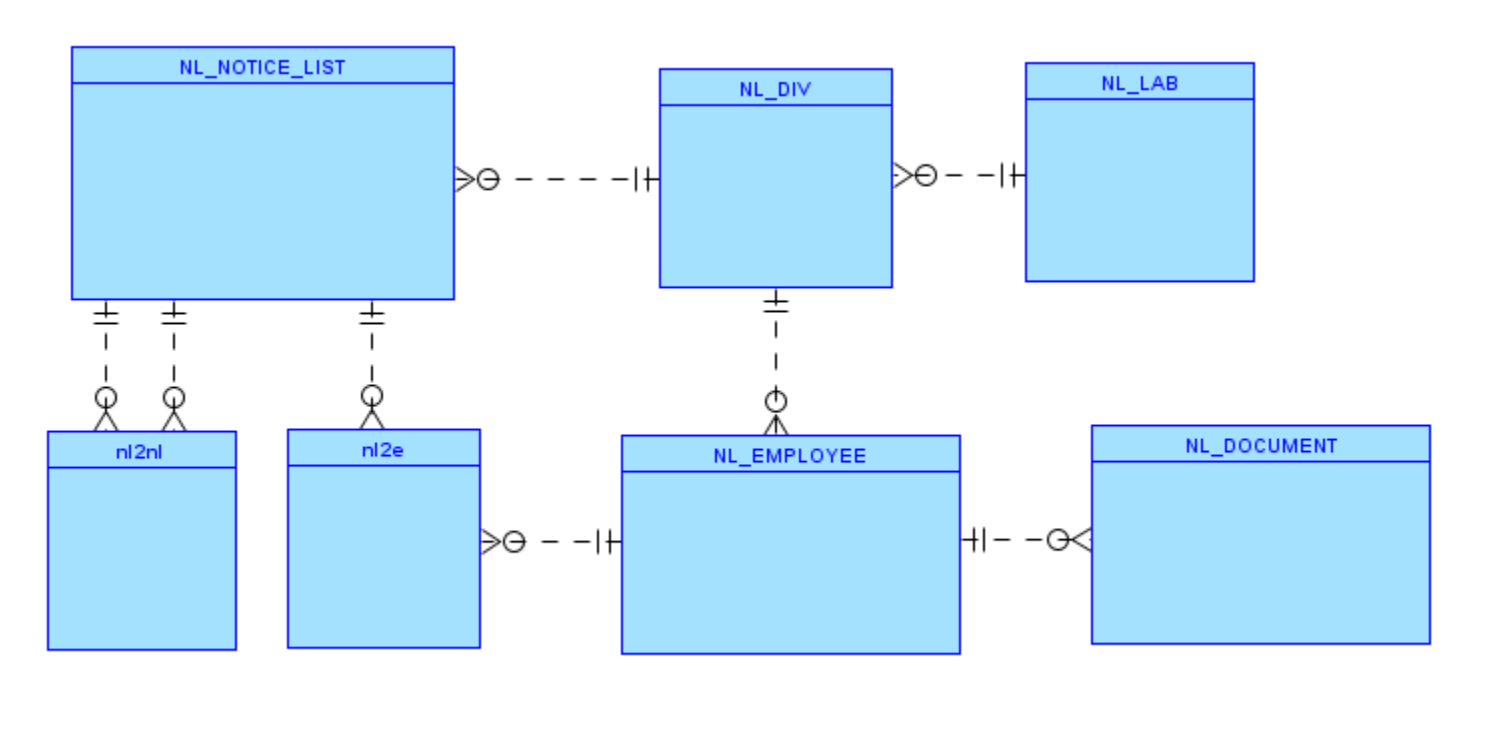
Conceptual Model

- NL_EMPLOYEE, NL_DIV, NL_LAB are read-only tables



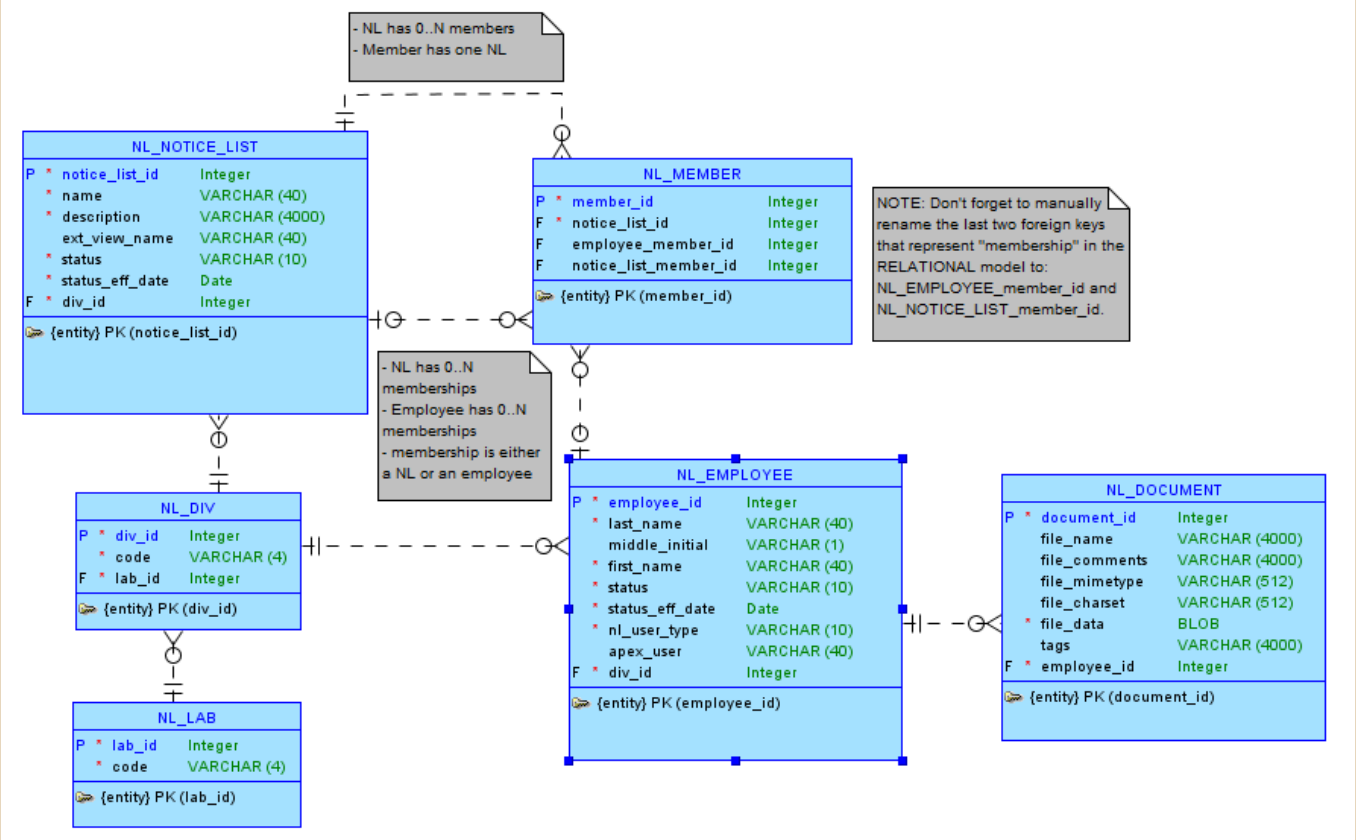
Logical Model (Intermediate)

- Convert the two M:N relationships into two junction tables

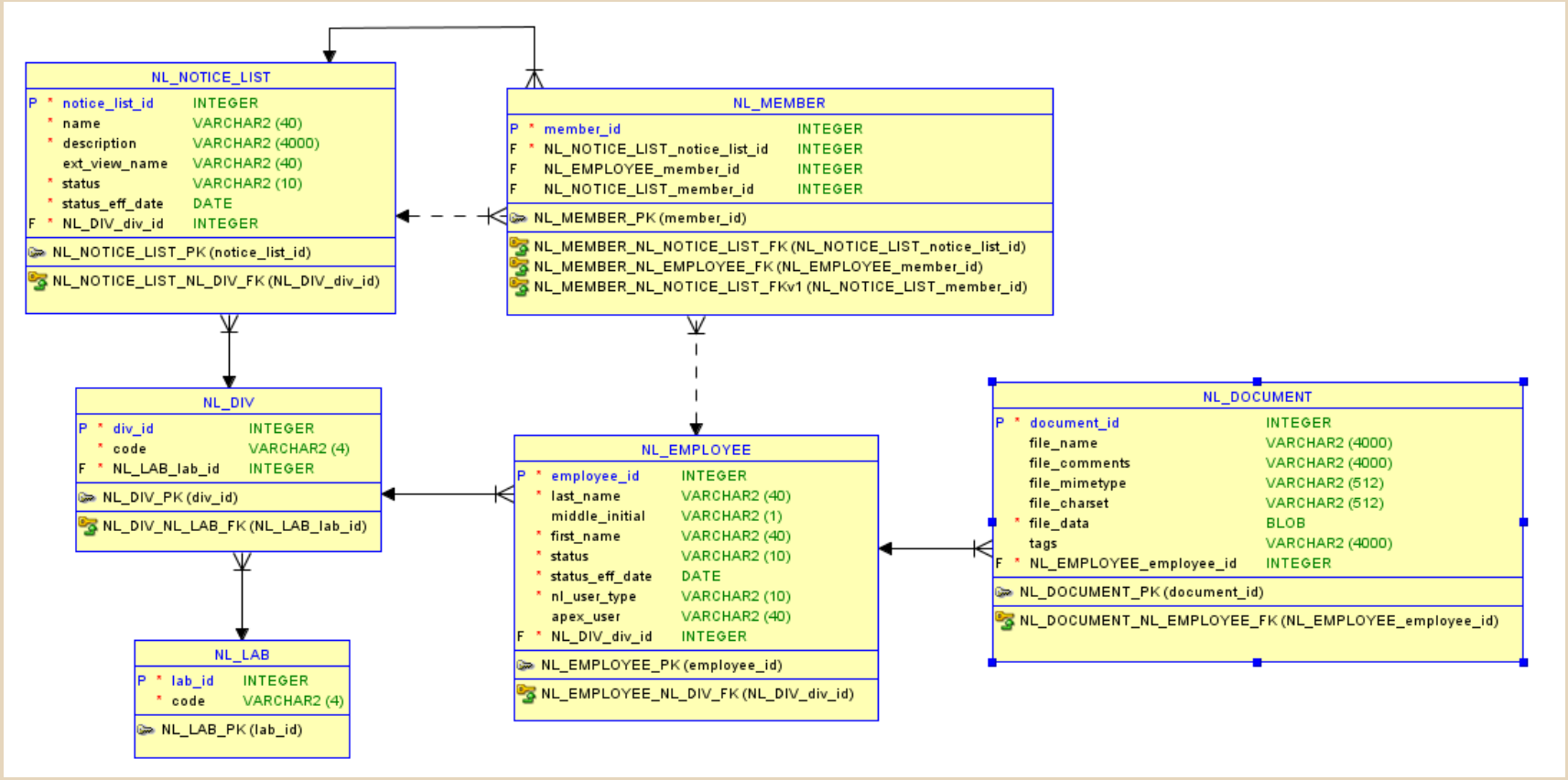


Logical Model (Final)

- To match the schema given by ARL, the **NL_MEMBER** junction table combines what was two junction tables (previous slide) into one table where EITHER **employee_member_id** OR **notice_list_member_id** is used.



Logical Model Relational

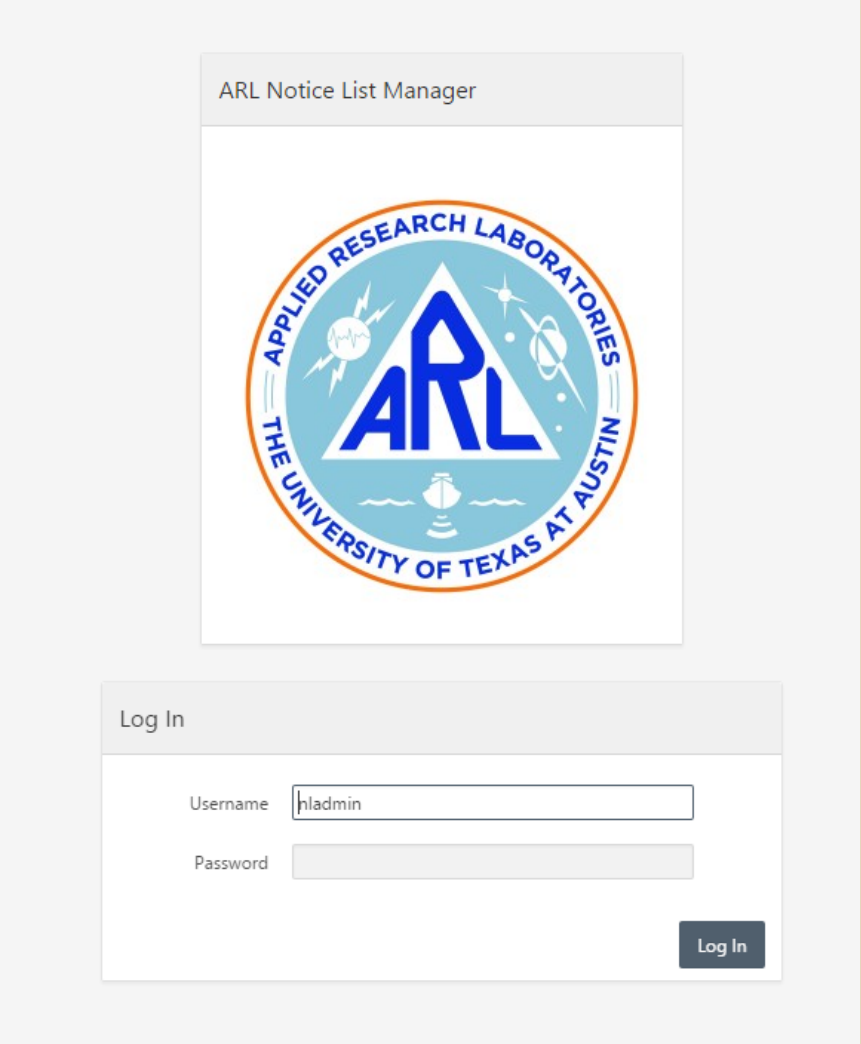


APEX Application

- URL
  https://apex.oracle.com/pls/apex/f?p=90854:LOGIN_DESKTOP:106205508786492:::::
- ACCOUNTS
  - ADMIN
    - user: nladmin, pwd: nladmin
  - STANDARD
    - user: nlstandard, pwd: nlstandard
  - BASIC
    - user: nlbasic, pwd: nlbasic
  - NONE
    - user: nlnone, pwd: nlnone

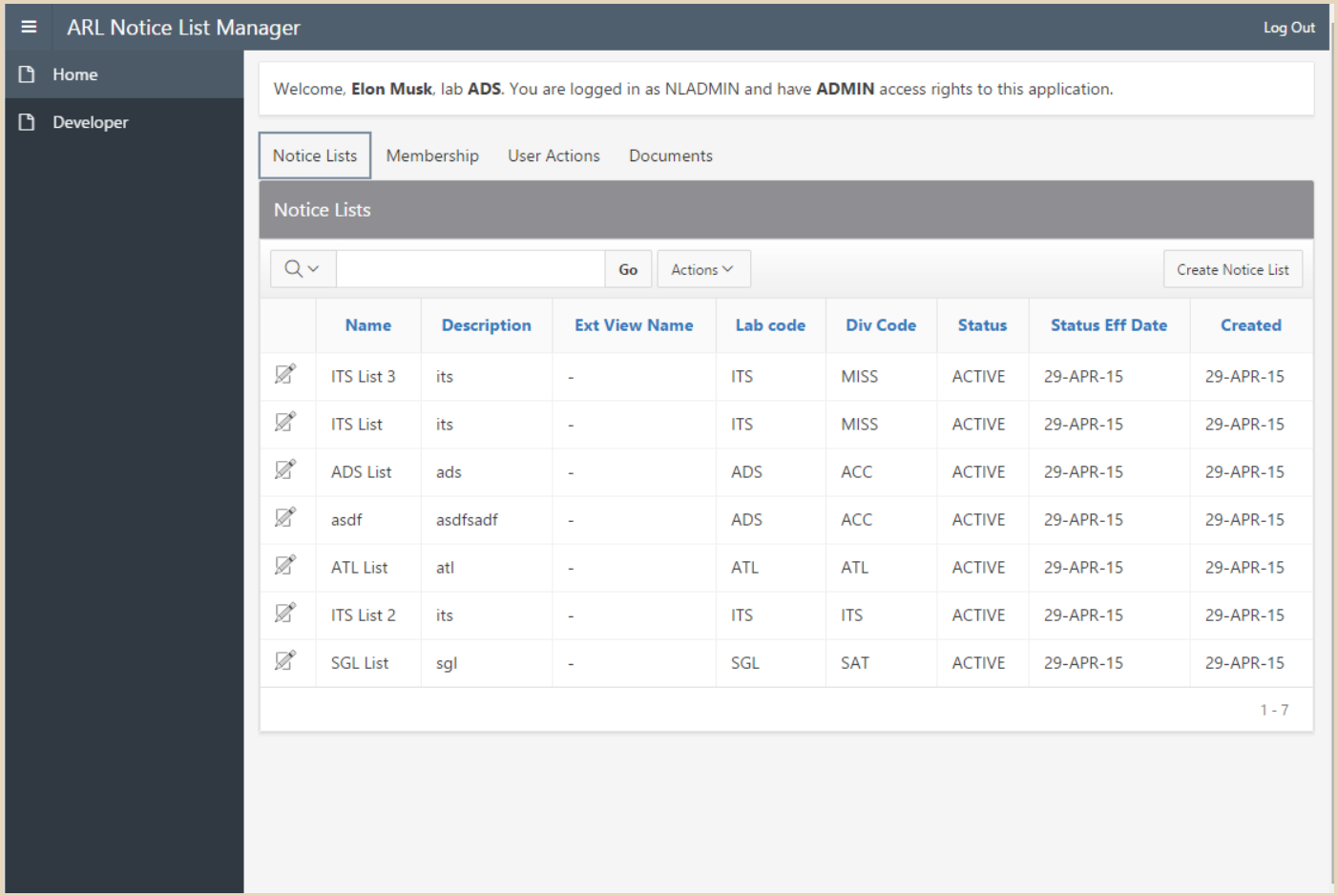
Login Page

- Uses APEX authentication and links APEX users to NL_EMPLOYEES



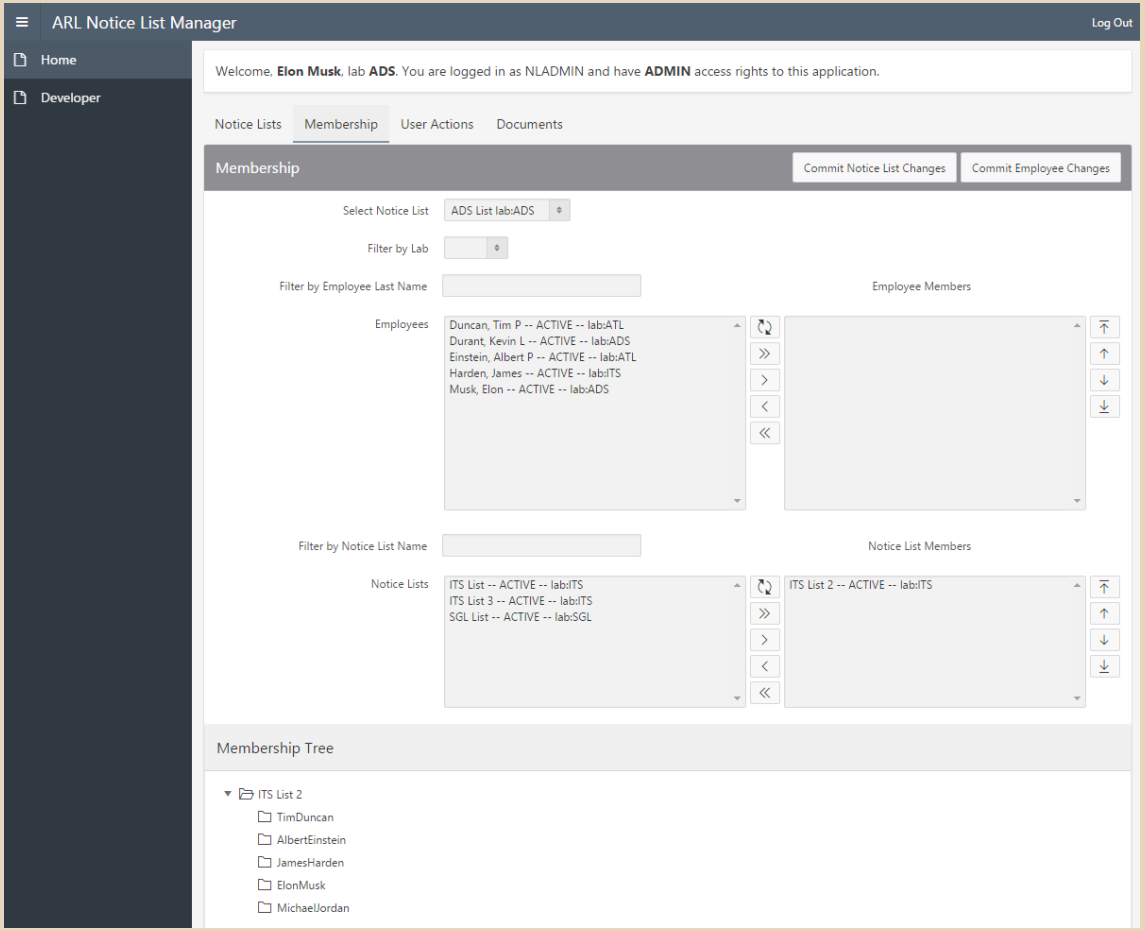
Notice List Tab

- ADMIN only, manage notice lists



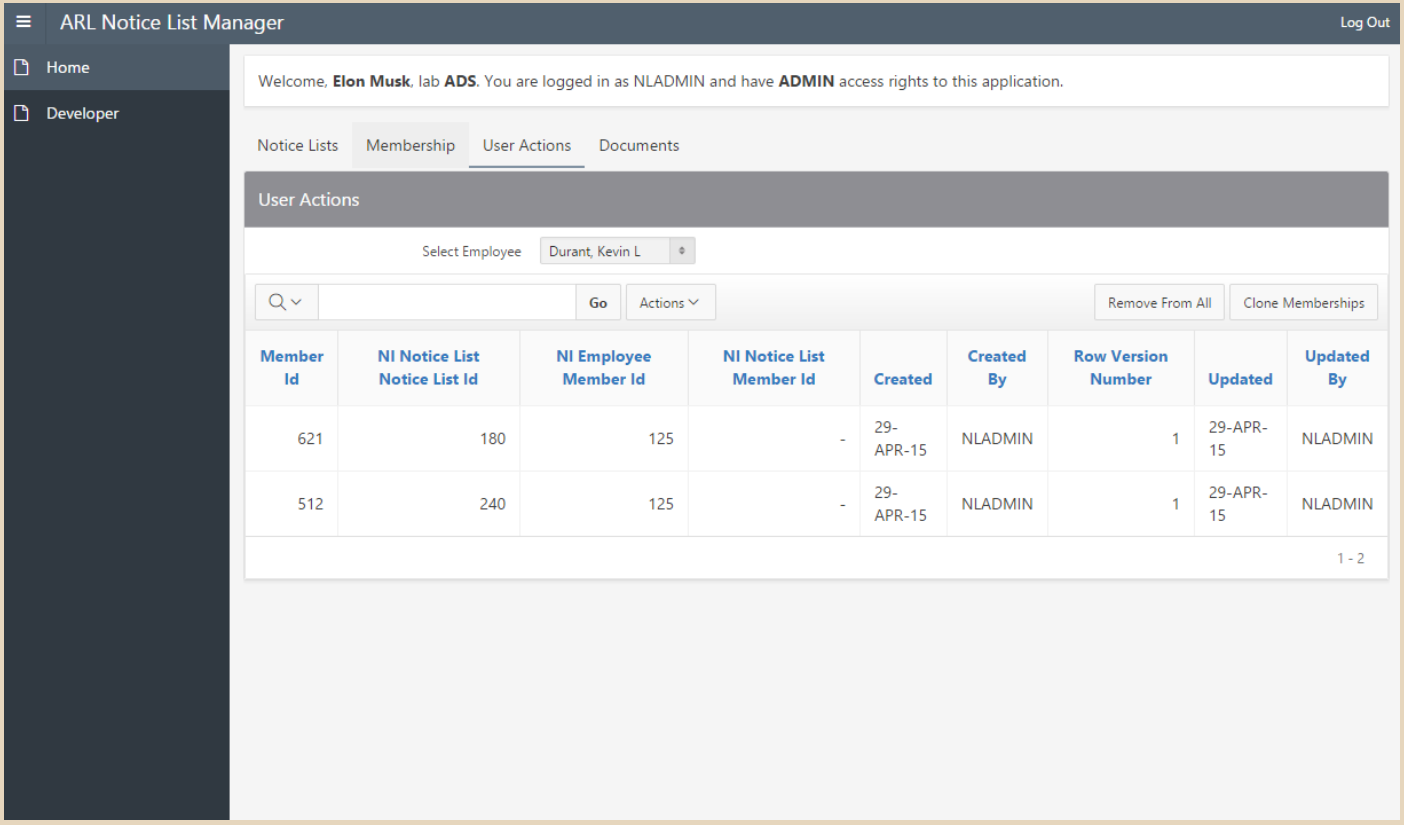
Members Tab

- Manage memberships
- Filter by last name
- Filter by list name
- ADMINS can control any list
- STANDARD users can only control lists in their lab
- Tree view of list membership
- Multiselect vis shuttle controls



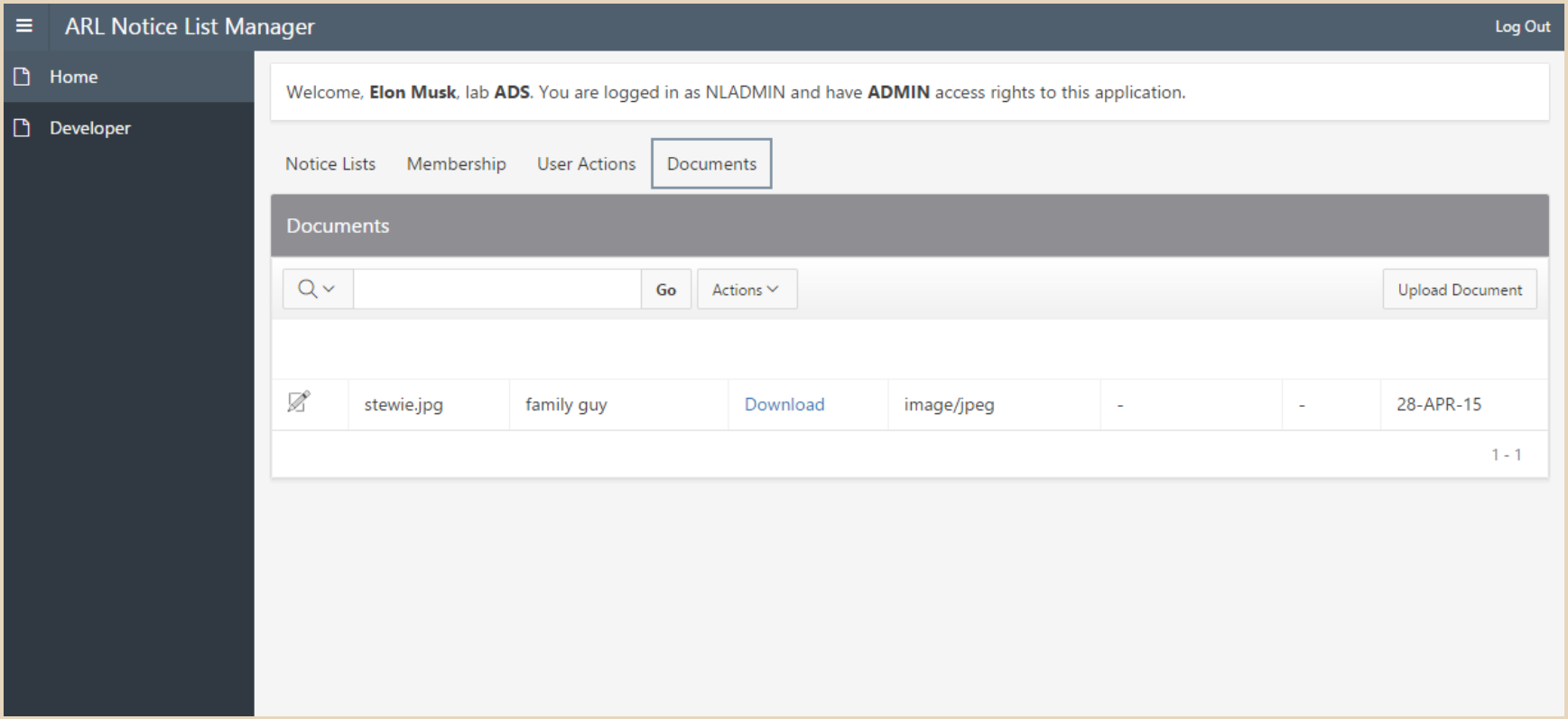
User Action Tab

- Remove user from all lists
- Clone user membership to another user
- ADMINs can manage any user, STANDARD can only manage users in their lab



Documents Tab

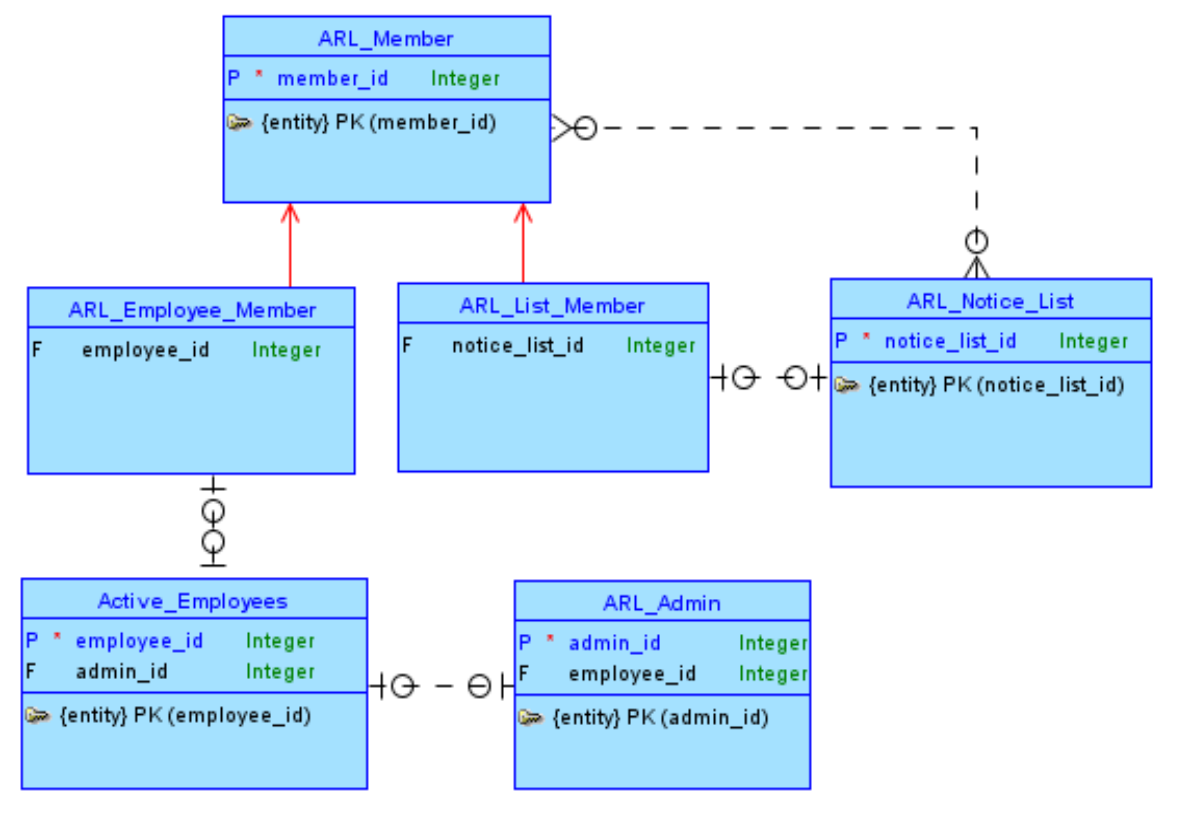
- ADMIN/STANDARD/BASIC users can upload documents associated to their NL_EMPLOYEE row



BACKUP SLIDES

Conceptual Model B

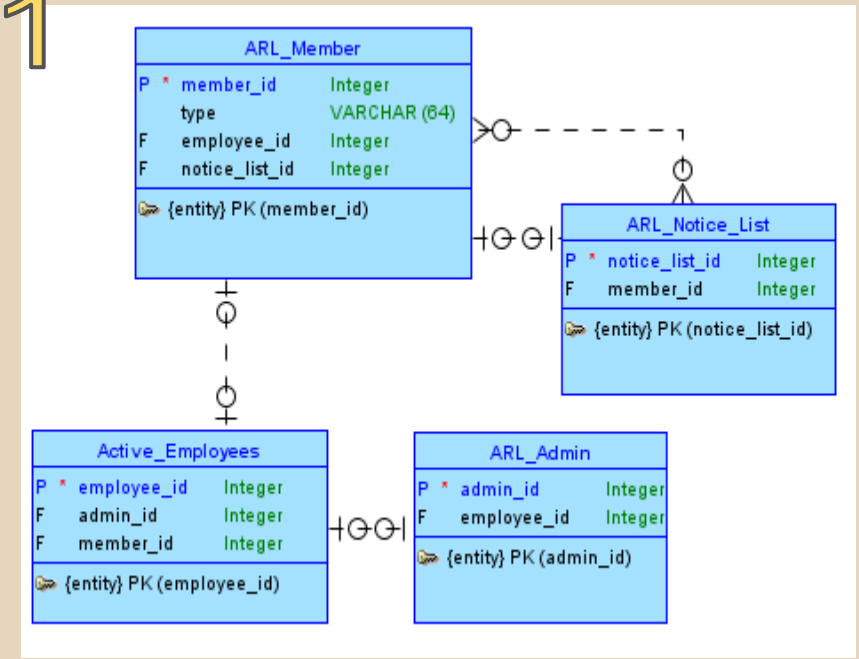
- Use inheritance to model “member” hierarchy
- Downside is it creates a new member table that is unnecessary and can be simplified down to the same thing as logical model A (explained in next slide)



Logical Model B

1. Collapse the member hierarchy
2. Add junction table for M:N relationship
3. Notice that ARL_Member can be coalesced into ARL_NL2M and **type** can be made implicit by using employee_id OR notice_list_id. This change brings us back to logical model A. (It doesn't seem like there are any downsides to doing this UNLESS we need to have views into the junction table to guarantee that the right id attribute is filled in.)

1



2 3