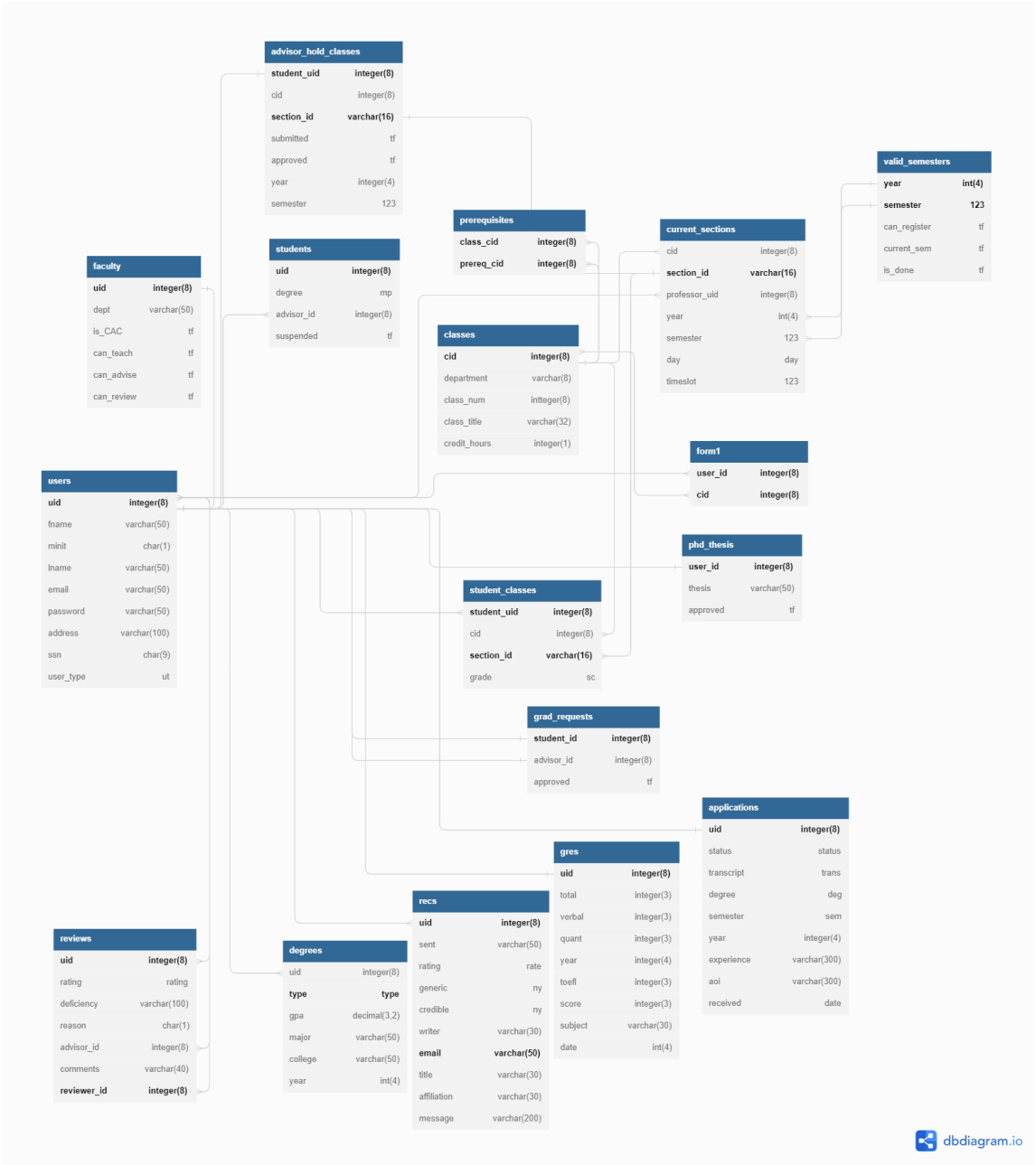
DB DESIGN:

- **ER DIAGRAM**

The majority of our tables are in BCNF, but we have two tables which are in 3NF which means overall our database is in 3NF. We believe that is this true as (except for the two tables which will be mentioned later) as for any X → A in our tables, X is the superkey for that table. This is helped because the majority of our tables only use one column as the primary/super key, and any dependencies have that column as the X. The two tables for which BCNF is not met are in the student_classes and advisor_hold_classes tables. For these tables section_id, cid, as well as student_id are the superkey, and there exists the dependency section_id → cid. This is still in 3NF as cid is a member of the superkey, but is not in BCNF as section_id is not the whole superkey for the table.



VISUAL COMPONENT:

- screenshots of our project are included in six zip files called:
    - screenshots_g2_p2_1
    - screenshots_g2_p2_2
    - screenshots_g2_p2_3
    - screenshots_g2_p2_4
    - screenshots_g2_p2_5
    - screenshots_g2_p2_6

  which are in a folder called screenshots

DESIGN JUSTIFICATION:

There were three main things we had to do when integrating our components. The first was changing the database, the second was changing the back-end to allow users to pass through the system from applicant to alumni (and adapt the code to the new database), and the third was changing the front-end to incorporate all possible actions/privileges from our different sections into one cohesive portal for the different user types.

The main change to our database was adapting how we included faculty, as we had to support faculty being a CAC, a reviewer, an advisor, a teacher, or some combination of the four. To do this we added a new table (faculty) which uses the user_id as the primary key and includes their department, and has T or F values for is_cac, can_teach, can_review, and can_advise. These values are used to determine the privileges the faculty has.

Integration in the backend was fairly straightforward although it took a lot of time. Other than changing the code to adapt to the new database and adding new functionality and features, we also ensured that once an application is accepted, the user_type of an applicant in the users table is changed to a Student, and once a student graduated the user_type is updated to alumni. REGS and ADS also had to work closely together to ensure that the class registration information from REGS replaced the hard coded information used in ADS.

For the front end, when dealing with users that had combined functionality between the different parts, we created a universal portal which had links to all the different functionalities. This allowed us to not have to rewrite all our code, as we could reuse HTML pages from phase 1 instead of having to rewrite all our pages. Another way we integrated our projects was by ensuring that we had consistent CSS across all our pages, which gives our project a unified look and feel even when navigating through pages originally belonging to different phase 1 groups.

Lastly, regarding the queries required by the project specs, we satisfied these in two ways. The first way was adding search bars (by things such as UID, last name, degree, year, etc.) to any tables we had which display applicants, students etc. This allows the GS, Sysadmin, and other users who have access to those tables to perform queries on the users they can see. For queries which involved information we did not already have tables for, we created new pages and a new section in the user portals called something like “manage queries”.

SPECIAL FEATURES:

We implemented a valid_semesters table, which allows the registrar and sysadmin to control what semester is the current semester, as well as decide whether or not registration is open for specific semesters. Students can only register for classes for which registration is open, but can view their schedule for all classes.

We implemented the Registrar, who (along with sysadmin) can add sections to classes from the course catalog, as well as edit sections for future semesters where registration is closed, and add sections to any semester.

We implemented the advisor hold, which doesn't allow a user to register for classes until they have submitted a form to their advisor. This acts similarly to a UAF form but only for the first semester.

Users are able to download transcripts to which they have access.

We added javascript form validation to all relevant fields.

WORK BREAKDOWN:

In phase 1, Matias worked on ADS, Freya worked on REGS, and Benedek worked on APPS. Aside from changing our code from phase 1 to work with the new database (which was mainly just fixing sql queries), we did most of our work together in person, which meant that there was no clear separation of work between who did what as we helped each other and sometimes even worked on solving issues on just one computer. Where we did have slightly different responsibilities was near the end, where Matias focused on CSS, Benedek did some CSS as well and did the javascript form validation, and Freya implemented the queries and advisor hold forms.

BIG ASSUMPTIONS:

- The project specs include some repetition issues (two Ringo Starrs and two matching uids), we changed this and described these changes in the git issue. Another issue was with diana krall uid which we were supposed to set as 999999999. Our current system auto increments uid when we add a new user, which would break the system.We changed her uid to 999999990 to allow new users to be added. This would not be a problem in a real system where we don't manually input users, as we would likely not reach 100 million users.
- Form 1 and graduating a student would involve manual oversight by the GS, such as making sure that form1 and transcript make sense.
- Advisor hold form works like the UAF form, where the student does not need to register for the exact classes they submitted on the UAF form. They do automatically register for these classes, but it is possible for the classes to be dropped and for different classes to be added.