DATABASE TABLES SCHEMA NOTES: . The naming convention is camelCase. • All purple variables are countable (substitutes for arrays); front-end should expect functions to retrieve count and actual list. • Blue boxes are notes for the front-end team; red for the back-end team. · Yellow boxes are character limits / hard-coded formats / other generated by front-end restrictions that both teams should be aware of. operations before . The first field is the unique identifier for that item. sending to database • Dotted arrows are the same variable for one instance of the item. **APPLICATIONS TABLE** varchar(50) varchar(50) varchar(50) varchar(10) varchar(100) varchar(100) varchar(100) username year attachment firstName lastName grade320 referenceName referenceEmail status gpa (String) (String) (String) (String) (String) (String) (String) (float) (char) (int) (S3 link) (1 of 4 options) (unique) 0: accepted, 1: rejected 'Freshman' etc. 2: pending Name will be asked for during registration. This STUDENTS TABLE name can be retrieved So, if a student prefers section 7 from here to greet user. varchar(50) and then section 8, the pairings username would be (1, 7), (2, 8). conflict ranking sectionRanked matched (int = section ID) (int = 1, 2, ...)(int = section ID) (int = section ID) **USERS TABLE** (unique) varchar(50) varchar(255) varchar(50) varchar(50) varchar(50) Send out notification when this type username value changes from -1 to password firstName lastName (Boolean; 0 = email (String) section ID, since the student **PROFESSORS TABLE** (String) student, 1 = (String) (String) (String) (unique) is now matched. professor) varchar(50) varchar(50) appRanked username section ranking (String = student's (String) (int = section ID) (int = 1, 2, ...)(unique) username) **MATCHINGS TABLE SECTIONS TABLE** varchar(50) varchar(50) varchar(50) varchar(10) varchar(30) username id username applicant Retrieve capacity and section (String = capacity numEnrolled day time (integer) (String = Professor's (String = student's numEnrolled from here. (int = section ID) student's (unique, taken (String) (String) (int) username) username) username) from schedule) 'MW', 'MWF', .g. '2:30 PM Update numEnrolled 'TuTh', etc. 3:45 PM' as the matching algorithm progresses. Used to be String = Professor's Every student who has name. Simplifies back-end submitted application operations. Front-end will have choosing this section function to retrieve instructor's Update with every name when displaying submission. section info.