

Project 2 Design Document

proj2_package.sql

- function show_students return ref_cursor; - Returns the full contents of table students to a ref cursor to be printed by proj2.java.
- function show_courses return ref_cursor; - Returns the full contents of table courses to a ref cursor to be printed by proj2.java.
- function show_prerequisites return ref_cursor; - Returns the full contents of table prerequisites to a ref cursor to be printed by proj2.java.
- function show_classes return ref_cursor; - Returns the full contents of table classes to a ref cursor to be printed by proj2.java.
- function show_enrollments return ref_cursor; - Returns the full contents of table enrollments to a ref cursor to be printed by proj2.java.
- function show_logs return ref_cursor; - Returns the full contents of table logs to a ref cursor to be printed by proj2.java.
- procedure add_student(sid_in in students.sid%type, fn_in in students.firstname%type, ln_in in students.lastname%type, status_in in students.status%type, gpa_in in students.gpa%type, email_in in students.email%type); - takes all 6 columns of a student tuple as input to add to students table.
- function find_student(sid_in in students.sid%type, ln_out out students.lastname%type,

- `status_out out students.lastname%type) return ref_cursor;` - takes SID as input to find a student. Binds lastname and status as out variables and returns all classes taken by the student to a `ref_cursor`.
- `function find_prereqs(dc_in in prerequisites.dept_code%type, cn_in in prerequisites.course_no%type) return ref_cursor;` - takes course number and department code to find all prerequisites required by class and returns them to a `ref_cursor`.
- `function class_roster(cid_in in classes.classid%type, title_out out courses.title%type) return ref_cursor;` - takes classid as input to find a class. Binds course title to out variable and returns all students enrolled in the class as a `ref_cursor`.
- `function enroll_student(sid_in in students.sid%type, cid_in in classes.classid%type) return number;` - Takes a SID and a CID as input to enroll student in the class. If successful returns 0. If not, the student is not enrolled in the class and function returns error code so `proj2.java` can figure out what went wrong and display output.
 - If SID is invalid returns 1.
 - If CID is invalid returns 2.
 - If the class is currently full returns 3.
 - If student is already enrolled in provided class returns 4.
 - If student is already enrolled in 5 classes in the current semester and year returns 5.
 - If student has not gotten at least a C+ in all required prerequisite courses returns 6.
- `function drop_class(sid_in in students.sid%type, cid_in in classes.classid%type)`

return number; - Takes a SID and a CID as input to drop student from the class. If successful returns 0. If not, the student is not dropped from the class and function returns error code so proj2.java can figure out what went wrong and display output.

- If SID is invalid returns 1.
- If CID is invalid returns 2.
- If the student is not enrolled in the class returns 3.
- If this is the only class the student is enrolled in returns 4.
- If student dropping the course would remove required prerequisites from enrollment returns 5.
- If the class is now empty returns 6 but still drops student from class.
- function delete_student(sid_in in students.sid%type) return number; - Deletes student with input SID from table. If successful returns 0, if SID is invalid return 1.

proj2_sequences.sql

- create sequence logid_sequence - starts at 100001 and increments by 1 every time sequence is called. Used to generate logid on insertion to table logs.

proj2_triggers.sql

- create or replace trigger generate_logid - calls logid_sequence to generate next logid on insertion to table logs.
- create or replace trigger add_log_insert - adds a log to table logs every time a student is added to table students. Tracks timestamp, table added to, operation performed, and SID of student being added.

- create or replace trigger add_log_delete - adds a log to table logs every time a student is deleted from table students. Tracks timestamp, table added to, operation performed, and SID of student being deleted.
- create or replace trigger add_log_update - adds a log to table logs every time a student is updated in table students. Tracks timestamp, table added to, operation performed, and SID of student being updated.
- create or replace trigger update_class_size - increments class size by 1 every time a student is added to a class.
- create or replace trigger update_class_size - decreases class size by 1 every time a student is deleted from a class.

proj2.java

Connects to database. Lists every function/procedure from proj2_package.sql and assigns a number to each. Waits for user to input a number. 1-8 selects corresponding function/procedure, 0 exits program, anything else throws error or requests proper input and redisplay menu.

- Picking 1 displays sub menu of every possible table to display. 1-6 displays corresponding table and returns to main menu, 0 returns to main menu, improper input either returns to main menu or throws error.
- Picking 2 asks user to input values for student tuple and passes input to proj2.add_students(). Prints “Added student [sid] to table.” on success and returns to main menu.
- Picking 3 asks user to input a SID. Passes input sid to proj2.find_student() and prints SID, lastname, status, and classid, department code, and course number of all classes student is enrolled in. If SID is invalid, program prints message and returns to main

menu. If student is not enrolled in any classes print “[sid] hasn’t taken any courses.”

Returns to main menu on succes.

- Picking 4 asks user to input a Department Code and Course Number. Passes input to proj2.find_prereqs() and prints all prerequisites required by the course, if any. Returns to main menu when done.
- Picking 5 asks user to input Class ID. Passes input to proj2.class_roster() and print Class ID, Course Title, and the SID, lastname, and email for every student enrolled in the class. If CID is invalid, prints message and returns to main menu. If class is empty, prints message instead of students. Returns to main menu when done.
- Picking 6 asks user to input SID and Class ID and attempts to roll the student in the provided class by passing input to proj2.enroll_student().

Prints error messages if

- SID or Class ID are invalid
- class is full
- the student is already enrolled in 5 classes
- student is already enrolled in the provided class
- the student is missing prerequisites.

On success prints “[sid] enrolled in [class id] successfully” and returns to main menu when done.

- Picking 7 asks user to input SID and Class ID and attempts to drop the student from the provided class by passing input to proj2.drop_class().

Prints error messages if

- SID or Class ID are invalid
- student is not enrolled in the provided class

- provided class is the only class student is enrolled in
- dropping class would remove required prerequisites

On success print “[sid] enrolled in [class id] successfully”. Prints if the student was the last student enrolled in the class. Returns to main menu when done.

- Picking 8 asks user to input SID and passes input to proj2.delete_student() to delete student from table. If the SID is invalid program prints message and returns to main menu. If student is found program prints “Student [sid] deleted successfully” and returns to main menu.

Program continues to loop and take input until user enters 0 or program encounters error. The while loop runs while int flag (set to 0 by default) is less than 1. When the user inputs 0, flag is set to 1 to break while loop.