

CS2102 Project Report

Team 4

Names and matric numbers:

- Lim Junxue A0204757B
- Chang Rui Feng A0201575L
- Bikramjit Dasgupta A0200600J
- Soh Xin Wei A0199847B

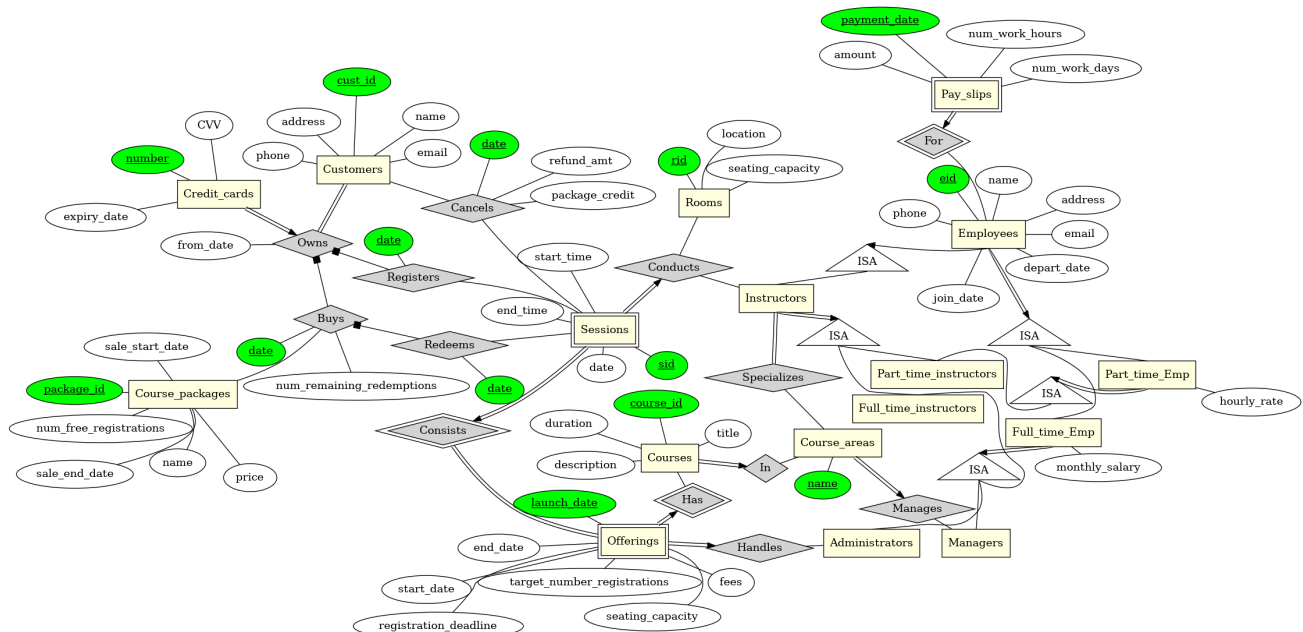
Listing of project responsibilities:

	Functions	Schema	Data	Triggers	Report
Lim Junxue	1, 5, 9, 13, 17, 21, 25, 29	Entities: part time instr, full time instr, admin, managers Relationships: manages, handles	Entities: part time instr, full time instr, admin, managers, Course_packages Relationships: manages, handles, Consists	Triggers on insert/update/delete on all employee related tables, register_before_deadline_trigger, redeem_before_deadline_trigger	Equal
Bikramjit	4, 8, 10, 12, 16, 20, 24, 28	Entities: Courses, Offerings, Course_areas, Customers, Rooms Relationships: Conducts, Consists, Has, In	Entities: Courses, Offerings, Course_areas, Customers, Rooms Relationships: Conducts, Consists, Has, In	session_delete_trigger, session_insert_update_trigger	
Rui Feng	2, 6, 14, 18, 22, 26, 30	Entities: Customers, Credit_cards, Course_packages, Relationships: Owns, Buys,	Entities: Customers, Credit_cards, Instructors	remove_employee_trigger, check_customer_registration_trigger, check_customer_redemption_trigger	

		Registers, Redeems, Cancels, Consists, Conducts	Relationships: Owns, Buys, Specializes		
Xin Wei	3, 7, 11, 15, 19, 23, 27	Entities: Pay_slips, Employees, Part_time_Emp, Full_time_Emp, Sessions Relationships: For, Registers, Redeems, Cancels	Entities: Pay_slips, Employees, Part_time_Emp, Full_time_Emp, Sessions Relationships: For, Registers, Redeems, Cancels	assign_instructor_to_s ession	

ER model:

We used the sample ER diagram given to us by the teaching staff. Here is the diagram for easier reference:



Non-trivial design decisions:

1. This design models Credit Cards as an entity (rather than as an attribute of Customers) using the Owns relationship to relate to Customers.
 - The reason is that the customer's credit card information may change over time (by the routine update_credit_card). The current active credit card is determined by the from_date attribute of Owns.
2. This design models the Registers, Redeems, and Cancels relationships with a date attribute each to record the date of the transaction.
 - By having this attribute as part of the key of the relationship, it supports multiple instances of the relationship to exist for the same set of relationship participants.

5 constraints not enforced by this ER model:

1. Each session is conducted by an instructor on a specific weekday (Monday to Friday).
2. Each session can start earliest at 9am and end latest by 6pm, no session can be conducted between 12pm to 2pm.
3. A customer can register for at most one session per course offering, and can only either register or redeem a session, but not both.
4. Registration deadline for a course offering must be at least 10 days before its start date.
5. Seating capacity of a course offering is equal to the sum of the seating capacities of its sessions.

Schema:

Non-trivial design decisions:

1. Instead of creating a schema for the Conducts relationship, we merged the relationship into our Sessions schema.
 - To enforce the key and total participation relationship - implying that there won't be any null values relating to the data from the Conducts relationship.
2. Instead of creating a schema for the Handles relationship, we merged the relationship into our Offerings schema.
 - To enforce the key and total participation relationship - implying that there won't be any null values relating to the data from the Handles relationship.
3. Instead of creating a schema for the Manages relationship, we merged the relationship into our Course_area schema.
 - To enforce the key and total participation relationship - implying that there won't be any null values relating to the data from the Manages relationship.
4. Instead of creating a schema for the In relationship, we merged the relationship into our Courses schema.
 - To enforce the key and total participation relationship - implying that there won't be any null values relating to the data from the In relationship.

5. The type serial was used for ids that have to be decided by the system, because it has to be unique and so that users will not need to check when inserting entries.

5 constraints not enforced by our schema:

1. Part time instructors must not teach more than 30 hours each month.
2. A customer can register for at most one session per course.
3. An instructor must specialize in the course area of a session that he teaches.
4. A course offering's start and end date is determined by the dates of its earliest and latest sessions.
5. Instructors need a one hour break in between 2 consecutive sessions.

3 most interesting triggers:

A lot of our constraints not enforced by the schema are enforced within our functions and procedures to manipulate the data in our database. Triggers are used for constraints that are not present within our functions or are used in multiple functions (to consolidate).

1. Assigning an instructor to a session

Name: assign_instructor_to_session

Usage: To enforce

- a. A part time instructor must not teach more than 30 hours each month.
- b. An instructor must specialize in the course area of a session that he teaches.
- c. Instructors need a one hour break in between 2 consecutive sessions.
- d. Instructor can only teach at most one session at the same hour/time

Justifications: It was difficult to enforce these constraints in the schema created, and whether an instructor is assigned to a session affects a few routines we had to implement, thus making it into a trigger was the most effective way to enforce this constraint

2. **Name:** session_insert_update_trigger

Usage: To enforce the below conditions when adding, updating or deleting a session

- a. A course offering's start and end date is determined by the dates of its earliest and latest sessions.
- b. Seating capacity of a course offering is equal to the sum of the seating capacities of its sessions.
- c. No two sessions for the same course offering can be conducted on the same day and at the same time.
- d. Rooms cannot be used by more than 1 session at the same time

Justifications: Since each session is linked to a course offering, and a course offering can have multiple sessions, setting the course offering's start and end date based on the earliest and latest session linked to that offering would be best implemented with a trigger when a session that belongs to that offering is modified.

3. **Name:** check_customer_registration_trigger

Usage: To enforce

- a. A customer can register for at most one session per course offering.
- b. A customer can only either register or redeem a session, but not both.

Justifications: The brief states that customers cannot join more than a session per course offering, and can only register or redeem for a session, but not both. However, this is difficult to implement with the schema alone, due to the fact that both (a) and (b) need to check multiple records from multiple tables (Registers and Redeems).

Difficulties

- It was difficult to come up with mock data as there were many foreign key constraints that were all across the schemas.
- Certain conditions were tough to implement and test since the test data due to the large scale of the ER model.
- Certain routines had a lot of information and requirements to meet making it difficult to come up with everything.
- The dependency of data within parent tables made functions which insert into child tables very tricky.
- We worked on our mock data before putting our check constraints and triggers, which resulted in several errors after the data was validated. It was very difficult to change the data or add new data based and validated on our check constraints and triggers after we made them.

Lessons

- Communicating with one another made implementing routines easier as certain members worked on specific areas of the schemas, so whenever there are routines that overlap with other member's portions of the schema they were able to help ensure the constraints were still held together.
- Coming up with a comprehensive and feasible schema takes a lot of time and effort, and making sure certain relationships go well with the entities are important to make creation of the schema easier.
- Not all constraints and requirements for routines have to be implemented in the function itself; instead they can be translated into triggers and schema constraints more easily for better scalability.
- Understanding the several alternatives to enforce constraints that are not enforced by an ER model, and finding their strengths and weaknesses.