CS2102 Database Systems

ER Data Model

Project Team 61

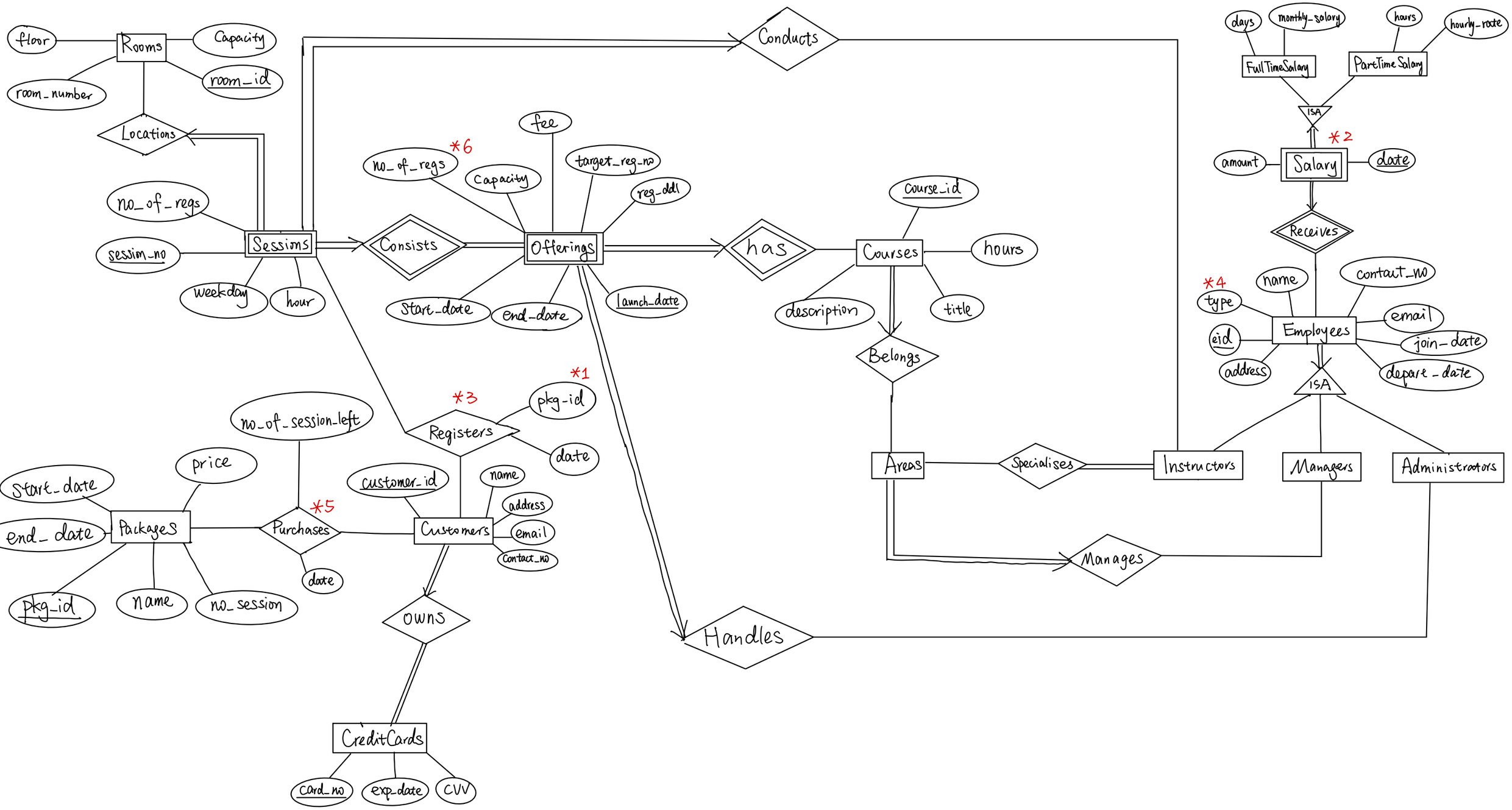
Chen Anqi (A0188533W)

Chen Su (A0188119W)

Gu Yichen (A0204735J)

Shao Yufei (A0206427J)

ER DATA MODEL



Justification for Non-Trivial Design Decisions Made

1. “For each registered course, a customer pays for the course fees by either making a credit card payment or redeeming a session from his/her active course package.” To capture this constraint, we used an attribute **package_id** in Registers relationship (between Customers and Sessions) to determine the method of payment. The package_id refers to the course package from which one session was redeemed. If the package_id is NULL, it means the customer paid for the registration using a credit card. This design is simple and is able to distinguish between the two payment methods (credit-card payment and package redeem) for each registration. Furthermore, as each Customer has exactly one credit card, we could trace the customer to find the credit card used for payment.
   An alternative design would be to add another two relationships Redeem (between Sessions and Packages) and Pay (Between Sessions and CreditCards) to show there were two different ways of payment, but we felt this was unnecessary as it would complicate the model, and it would be difficult to link the two relationships with the Registers relationship together.

2. Salary is made to be a weak entity of employees. Because it is difficult to uniquely identify a salary instance with its own attributes like **working_days, monthly_salary, hours, hourly_rate, amount, date**. On the same day, there might be multiple salary payments with the same amount. In our design, with the primary key **eid** from the Employees table, the salary of an employee then can be uniquely identified together with **date** in Salary, since for a particular date, there can only be one salary received by an employee.

3. We do not store the refundability of a registered session as an attribute in the ER model. Instead, we check whether a session could be refunded by dynamically computing the difference in date between the date of cancellation and the attribute **date** of the session(in the Sessions Entity). The rationale is that if we add a “status” attribute to the Sessions entity, then we’ll need to continuously check against the current date and the start date of the session and update its potential refundability every day. If the Sessions table contains a huge amount of entries, this would incur a large time complexity. If such queries for refundability status are rare, computing the status at runtime would be more efficient.

4. Since salary payments for part-time employees are calculated differently from those for full-time employees, we add the attribute **type** into Employees Entity to indicate whether an employee is full-time or part-time. As such, we can easily refer to the two types of Salaries when needed by any of the three employee types (Instructor, Manager, Administrator).

5. “A customer’s course package is classified as either active if there is at least one unused session in the package, partially active if all the sessions in the package have been redeemed but there is at least one redeemed session that could be refunded if it is cancelled, or inactive otherwise.” For higher efficiency, this will also be dynamically computed when there is an incoming query (similar to point 3). We can enumerate sessions redeemed in this package and see if the date of redeem is more than 7 days before session time. If it is, then the package is partially active.

6. We do not store the availability of a Session/Offering as an attribute because it can be computed dynamically easily. An Offering is said to be available if the current **num_of_reg** (number of people registered) is no more than its seating capacity; otherwise, we say that an Offering is fully booked. The information of availability only matters when a Customer wants to register for a new Session, and therefore we can just perform the simple computation on the go.

5 Constraints That Are Not Captured By The Proposed ER Data Model

1. An instructor who is assigned to teach a course session must be specialized in that course area.
2. Each instructor can teach at most one course session at any hour.
3. Each instructor must not be assigned to teach two consecutive course sessions; i.e., there must be at least one hour of break between any two course sessions that the instructor is teaching.
4. No two sessions for the same course offering can be conducted on the same day and at the same time.
5. Each room can be used to conduct at most one course session at any time.