CS2102 Database Systems

Project Report

(Part 2)

Project Team 61

Chen Anqi (A0188533W)

Chen Su (A0188119W)

Gu Yichen (A0204735J)

Shao Yufei (A0206427J)

NOTE:

1. FDs listed for a table are in the **minimal basis** form.
2. **Attributes in the primary key** of each table are highlighted in **yellow**.

Tables in 3NF:

Full_time_Emp (eid, monthly_salary) & Part_time_Emp (eid, hourly_rate) & Course_areas (name, eid) & Specializes (eid, area)

These tables have only 2 attributes so they must be in BCNF, and thus in 3NF.

Instructors (eid) & Full_time_Instructors (eid) & Part_time_Instructors (eid) & Administrators (eid) & Managers (eid)

These tables have only 1 attribute so they must be in BCNF, and thus in 3NF.

Rooms (rid, location, seating_capacity)

rid → location
rid → seating_capacity

// no two rooms should have the same location
location → rid

Since location is also a key of the table, all FDs have a key on the left hand side. Hence, the table is in 3NF.

Courses (course_id, title, description, course_area, duration)

course_id → title
course_id → description
course_id → course_area
course_id → duration

// Each course has a unique course title
title → course_id

Since title is also a key of the table, all FDs have a key on the left hand side. Hence, the table is in 3NF.

Offerings (course_id, launch_date, start_date, end_date, registration_deadline, target_number_registrations, seating_capacity, fees, eid)

(course_id, launch_date) → start_date
(course_id, launch_date) → end_date
(course_id, launch_date) → registration_deadline
(course_id, launch_date) → target_number_registrations
(course_id, launch_date) → seating_capacity
(course_id, launch_date) → fees
(course_id, launch_date) → eid

// By common sense, two offerings of the same course should not be offered over the same time period. Otherwise, their sessions can just be combined into one offering.
(course_id, start_date, end_date) → launch_date
(course_id, start_date, end_date) → registration_deadline
(course_id, start_date, end_date) → seating_capacity
(course_id, start_date, end_date) → fees
(course_id, start_date, end_date) → eid

All tuples on the left hand side of the FDs are keys of the table. Hence, the tabel is in 3NF.

Sessions (course_id, launch_date, sid, date, start_time, end_time, eid, rid)

(course_id, launch_date, sid) → date
(course_id, launch_date, sid) → start_time
(course_id, launch_date, sid) → end_time
(course_id, launch_date, sid) → eid
(course_id, launch_date, sid) → rid

// By requirement, no instructors can conduct two sessions at the same time.
(date, start_time, end_time, eid) → course_id
(date, start_time, end_time, eid) → launch_date
(date, start_time, end_time, eid) → sid
(date, start_time, end_time, eid) → rid

// By requirement, no rooms can hold two sessions at the same time.
(date, start_time, end_time, rid) → course_id
(date, start_time, end_time, rid) → launch_date
(date, start_time, end_time, rid) → sid
(date, start_time, end_time, rid) → eid

All tuples on the left hand side of the above FDs are keys of the table.

(course_id, start_time) → end_time
(course_id, end_time) → start_time
The right hand side of the two FDs are both prime attributes. Hence, the table is in 3NF.

Customers (cust_id, name, email, phone, address)

cust_id → name
cust_id → email
cust_id → phone
cust_id → address

// By common sense, no two persons should share the same phone number or email.
// We only allow update of personal information by UPDATE the existing tuple, but not INSERT a
// new tuple. This means a customer cannot insert a new entry with all information unchanged except a different email (or a different phone). So there cannot be two entries with the same email (or phone) but different cust_id.
email → cust_id
phone → cust_id

Since email and phone are also keys of the table, all FDs have a key on the left hand side. Hence, the table is in 3NF.

Credit_cards (number, expiry_date, CVV)

number → expiry date
number → cvv

All FDs have the primary key on the left hand side. Hence, the table is in 3NF.

Owns (cust_id, card_number, from_date)

card_number → cust_id
card_number → from_date

All FDs have the primary key on the left hand side. Hence, the table is in 3NF.

Course_packages (package_id, num_free_registrations, sale_start_date, sale_end_date, name, price)

package_id → num_free_registrations
package_id → sale_start_date
package_id → sale_end_date
package_id → name
package_id → price

// UNIQUE constraint on schema: there should not be two entries that only differ by package_id
(num_free_registrations, sale_start_date, sale_end_date, name, price) → package_id

Since (num_free_registrations, sale_start_date, sale_end_date, name, price) is also a key of the table, all FDs have a key on the left hand side. Hence, the table is in 3NF.

Buys (package_id, card_number, date, num_remaining_redemptions)

(package_id, card_number, date) → num_remaining_redemptions

All FDs have the primary key on the left hand side. Hence, the table is in 3NF.

Redeems (package_id, card_number, buy_date, course_id, launch_date, sid, date)

All attributes in primary key. Regardless of the FDs this table has, any attribute on the right hand side is a prime attribute. Therefore, the table must be in 3NF.

Registers (card_number, course_id, launch_date, sid, date)

All attributes in primary key. Regardless of the FDs this table has, any attribute on the right hand side is a prime attribute. Therefore, the table must be in 3NF.

Cancels (cust_id, course_id, launch_date, sid, date, refund_amt, package_credit)

(cust_id, course_id, launch_date, sid, date) → refund_amt
(cust_id, course_id, launch_date, sid, date) → package_credit

All FDs have the primary key on the left hand side. Hence, the table is in 3NF.

Tables not in 3NF:

Employees (eid, name, email, phone, address, join_date, depart_date)

eid → email
eid → phone
eid → address
eid → join_date
eid → depart_date

// UNIQUE constraint on schema: there should not be two entries that only differ by eid
(name, email, phone, address, join_date, depart_date) → eid

FDs that violate the 3NF constraint:
email → name
phone → name

Explanation: There might be cases when the same employee departed from the company earlier and re-join on a later date. Then, there might be two tuples with the same name, email, and phone. Here, we assume that whenever an employee wants to update their personal information, we do it by UPDATE existing tuple instead of INSERT a new tuple tuple. We do not update the join_date directly when the employee re-joins the company since companies usually would want to keep a record on the ex-employees. Since **no two persons should share the same email address or phone number, we have the above two FDs**. The LHS of these FDs are not superkeys of the table (due to the multiple records of the same employee who re-joins), and the RHS of the FDS do not contain any prime attribute. Hence, it violates 3NF.

BCNF Decomposition:
Closure causing the split: $\{email\}^+ = \{email, name\}$
R1(email, name): only 2 attributes so must be in BCNF.
R2 (eid, email, phone, address, join_date, depart_date): No violation of BCNF.
Not dependency preserving since **phone → name** is broken into two tables.

3NF Decomposition:
eid → email, phone, address, join_date, depart_date
email → name
phone → name

R1(eid, email, phone, address, join_date, depart_date)
R2(email, name)
R3(phone, name)
Since R1 contains eid which is a key of the original Employees table, no new table needed.

Pay_slips (eid, payment_date, amount, num_work_hours, num_work_days)

(eid, payment_date) → num_work_hours
(eid, payment_date) → num_work_days

FDs that violate the 3NF constraint:
(payment_date, num_work_hours, num_work_days) → amount

Explanation:
We can tell whether an employee is part-time or full-time from the NULL-ness of num_work_hours and num_work_days based on our schema implementation:

- full-time: num_work_hours is NULL, num_work_days is NOT NULL
- part-time: num_work_hours is NOT NULL, num_work_days is NULL

We can further get the total number of days in that month from payment_date. Then we can calculate the amount that should be paid based on the information above. Since the respective algorithms for calculating salary are fixed for every part-time / full-time employee, it is regardless of their eid. So eid need not appear on the left hand side. The LHS of this FD is not a superkey, and the RHS of this FD is not a prime attribute. Hence it violates 3NF.

BCNF Decomposition:
Closure causing the split: {payment_date, num_work_hours, num_work_days}$^+$ = {payment_date, num_work_hours, num_work_days, amount}
R1 (payment_date, num_work_hours, num_work_days, amount): No violation of BCNF
R2 (eid, payment_date, num_work_hours, num_work_days): No violation of BCNF
This decomposition is dependency-preserving since all FDs in minimal basis can be derived from these two tables.