

CS 4750 – Database Systems

DB Project Normal Forms

Last updated: 11/21/2013 by Hong Moon

First draft completed for all tables

Functional Dependencies

Customer:

$\text{customer_id} \rightarrow \text{customer_id}, \text{password}, \text{First name}, \text{Last name}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}, \text{phone_number}$

$\text{phone_number} \rightarrow \text{street_addr}, \text{city}, \text{state}, \text{zip_code}$

$\text{zip_code} \rightarrow \text{state}$

Product:

$\text{product_id} \rightarrow \text{product_id}, \text{product_name}, \text{amount_left}, \text{price}, \text{vendor_name}, \text{supplied_date}, \text{product_description}, \text{URL}$

$\text{product_name} \rightarrow \text{price}$

Order:

$\text{order_id} \rightarrow \text{order_id}, \text{customer_id}, \text{list_id}, \text{order_date}, \text{total_order_price}, \text{order_status}, \text{payment_method_id}$

Product_list:

$\text{list_id}, \text{product_id} \rightarrow \text{list_id}, \text{product_id}, \text{amount}$

Order_history:

$\text{customer_id}, \text{order_id} \rightarrow \text{customer_id}, \text{order_id}$

Vendor:

$\text{vendor_name} \rightarrow \text{vendor_name}, \text{phone_number}, \text{address}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{address}$

(Note: As explained in the Requirements document, we treat the vendor address as atomic so we don't break it apart as in Customer table)

Supply:

product_id, vendor_name → product_id, vendor_name, supplied_date, supplied_amount

Message:

message_id → message_id, customer_id, message_title, message_content, date

Write_Msg:

customer_id, message_id → customer_id, message_id

Write_Prod:

customer_id, product_id → customer_id, product_id

Product_review:

product_review_id → product_review_id, product_id, customer_id, date, star_rating

Payment_method:

payment_method_id → payment_method_id, customer_id, credit_card_num, credit_card_type, csv

credit_card_num, credit_card_type → csv

First Normal Form

All the attributes in the tables are atomic, so they are in first normal forms.

Third Normal Forms

(1) Customer

Customer:

$\text{customer_id} \rightarrow \text{customer_id}, \text{password}, \text{First name}, \text{Last name}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}, \text{phone_number}$

$\text{phone_number} \rightarrow \text{street_addr}, \text{city}, \text{state}, \text{zip_code}$

$\text{zip_code} \rightarrow \text{state}$

Calculate F+:

① Copy FDs

$\text{customer_id} \rightarrow \text{customer_id}, \text{password}, \text{First name}, \text{Last name}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}, \text{phone_number}$

$\text{phone_number} \rightarrow \text{street_addr}, \text{city}, \text{state}, \text{zip_code}$

$\text{zip_code} \rightarrow \text{state}$

② Add Reflexives

$\text{customer_id} \rightarrow \text{customer_id}, \text{password}, \text{First name}, \text{Last name}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}, \text{phone_number}$

$\text{phone_number} \rightarrow \text{phone_number}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}$

$\text{zip_code} \rightarrow \text{zip_code}, \text{state}$

③ Add Rules

(No rules to apply)

Therefore, F+:

$\text{customer_id} \rightarrow \text{customer_id}, \text{password}, \text{First name}, \text{Last name}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}, \text{phone_number}$

$\text{phone_number} \rightarrow \text{phone_number}, \text{street_addr}, \text{city}, \text{state}, \text{zip_code}$

$\text{zip_code} \rightarrow \text{zip_code}, \text{state}$

Calculate Fc:

① Eliminate Reflexives from F+

$\text{customer_id} \rightarrow \text{password, First name, Last name, street_addr, city, state, zip_code, phone_number}$

$\text{phone_number} \rightarrow \text{street_addr, city, state, zip_code}$

$\text{zip_code} \rightarrow \text{state}$

② Eliminate Transitive Dependencies

Since we have

$\text{customer_id} \rightarrow \text{phone_number}$

$\text{phone_number} \rightarrow \text{street_addr, city, state, zip_code}$

$\text{customer_id} \rightarrow \text{street_addr, city, state, zip_code}$

Eliminate

$\text{customer_id} \rightarrow \text{street_addr, city, state, zip_code}$

Intermediate result:

$\text{customer_id} \rightarrow \text{password, First name, Last name, phone_number}$

$\text{phone_number} \rightarrow \text{street_addr, city, state, zip_code}$

$\text{zip_code} \rightarrow \text{state}$

③ Eliminate Transitive Dependencies

Since we have

$\text{phone_number} \rightarrow \text{zip_code}$

$\text{zip_code} \rightarrow \text{state}$

$\text{phone_number} \rightarrow \text{state}$

Eliminate

$\text{phone_number} \rightarrow \text{state}$

Therefore, Fc:

$\text{customer_id} \rightarrow \text{password, First name, Last name, phone_number}$

$\text{phone_number} \rightarrow \text{street_addr, city, zip_code}$

$\text{zip_code} \rightarrow \text{state}$

From Fc, the 3NF tables for Customer table are:

customer_id, password, First name, Last name, phone_number //

phone_number, street_addr, city, zip_code //

zip_code, state

(Answer)

customer_id, password, First name, Last name, phone_number //

phone_number, street_addr, city, zip_code //

zip_code, state

(2) Product

Product:

$\text{product_id} \rightarrow \text{product_id}, \text{product_name}, \text{amount_left}, \text{price}, \text{vendor_name}, \text{supplied_date}, \text{product_description}, \text{URL}$

$\text{product_name} \rightarrow \text{price}$

Calculate F+:

① Copy FDs

$\text{product_id} \rightarrow \text{product_id}, \text{product_name}, \text{amount_left}, \text{price}, \text{vendor_name}, \text{supplied_date}, \text{product_description}, \text{URL}$

$\text{product_name} \rightarrow \text{price}$

② Add Reflexives

$\text{product_id} \rightarrow \text{product_id}, \text{product_name}, \text{amount_left}, \text{price}, \text{vendor_name}, \text{supplied_date}, \text{product_description}, \text{URL}$

$\text{product_name} \rightarrow \text{product_name}, \text{price}$

③ Add Rules

(No rules to apply)

Therefore, F+:

$\text{product_id} \rightarrow \text{product_id}, \text{product_name}, \text{amount_left}, \text{price}, \text{vendor_name}, \text{supplied_date}, \text{product_description}, \text{URL}$

$\text{product_name} \rightarrow \text{product_name}, \text{price}$

Calculate Fc:

- ① Eliminate Reflexives from F+

$\text{product_id} \rightarrow \text{product_name, amount_left, price, vendor_name, supplied_date, product_description, URL}$

$\text{product_name} \rightarrow \text{price}$

- ② Eliminate Transitive Dependencies

Since we have

$\text{product_id} \rightarrow \text{product_name}$

$\text{product_name} \rightarrow \text{price}$

$\text{product_id} \rightarrow \text{price}$

Eliminate

$\text{product_id} \rightarrow \text{price}$

Therefore, Fc:

$\text{product_id} \rightarrow \text{product_name, amount_left, vendor_name, supplied_date, product_description, URL}$

$\text{product_name} \rightarrow \text{price}$

From Fc, the 3NF tables are:

$\text{product_id, product_name, amount_left, vendor_name, supplied_date, product_description, URL //$

$\text{product_name, price}$

(Answer)

$\text{product_id, product_name, amount_left, vendor_name, supplied_date, product_description, URL //$

$\text{product_name, price}$

(3) Order

Order:

$\text{order_id} \rightarrow \text{order_id}, \text{customer_id}, \text{list_id}, \text{order_date}, \text{total_order_price}, \text{order_status}, \text{payment_method_id}$

Explanation:

order_id is a super key of the Order table. So the table is already in 3NF.

(4) Product_list

Product_list:

$\text{list_id, product_id} \rightarrow \text{list_id, product_id, amount}$

Explanation:

The combination of attributes list_id and product_id is a super key of the Product_list table. So the table is already in 3NF.

(5) **Order History**

Order_history:

$\text{customer_id, order_id} \rightarrow \text{customer_id, order_id}$

Explanation:

The combination of attributes `customer_id` and `order_id` is a super key of the `Order_history` table. So the table is already in 3NF.

(6) Vendor

Vendor:

$\text{vendor_name} \rightarrow \text{vendor_name}, \text{phone_number}, \text{address}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{address}$

Calculate F+:

① Copy FDs

$\text{vendor_name} \rightarrow \text{vendor_name}, \text{phone_number}, \text{address}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{address}$

② Add Reflexives

$\text{vendor_name} \rightarrow \text{vendor_name}, \text{phone_number}, \text{address}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{phone_number}, \text{address}$

③ Add rules

(No rules to apply)

Therefore, F+:

$\text{vendor_name} \rightarrow \text{vendor_name}, \text{phone_number}, \text{address}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{phone_number}, \text{address}$

Calculate Fc:

① Eliminate Reflexives from F+

$\text{vendor_name} \rightarrow \text{phone_number}, \text{address}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{address}$

② Eliminate Transitive Dependencies

Since we have

$\text{vendor_name} \rightarrow \text{phone_number}$

$\text{phone_number} \rightarrow \text{address}$

$\text{vendor_name} \rightarrow \text{address}$

Eliminate

$\text{vendor_name} \rightarrow \text{address}$

Therefore, Fc:

$\text{vendor_name} \rightarrow \text{phone_number}, \text{contact_personnel_email}$

$\text{phone_number} \rightarrow \text{address}$

From Fc, the 3NF tables are:

vendor_name, phone_number, contact_personnel_email //

phone_number, address

(Answer)

vendor_name, phone_number, contact_personnel_email //

phone_number, address

(7) Supply

Supply:

$\text{product_id, vendor_name} \rightarrow \text{product_id, vendor_name, supplied_date, supplied_amount}$

Explanation:

The combination of attributes `product_id` and `vendor_name` is a super key of the Supply table. So the table is already in 3NF.

(8) Message

Message:

message_id → message_id, customer_id, message_title, message_content, date

Explanation:

message_id is a super key of the Message table. So the table is already in 3NF.

(9) Write_Msg

Write_Msg:

customer_id, message_id \rightarrow customer_id, message_id

Explanation:

The combination of attributes customer_id and message_id is a super key of the Write_Msg table. So the table is already in 3NF.

(10) Write_Prod

Write_Prod:

$\text{customer_id, product_id} \rightarrow \text{customer_id, product_id}$

Calculate F+:

The combination of attributes `customer_id` and `product_id` is a super key of the `Write_Prod` table. So the table is already in 3NF.

(11) Product_review

Product_review:

product_review_id → product_review_id, product_id, customer_id, date, star_rating

Explanation:

product_review_id is a super key of the Product_review table. So the table is already in 3NF.

(12) Payment_method

Payment_method:

$\text{payment_method_id} \rightarrow \text{payment_method_id}, \text{customer_id}, \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

$\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{csv}$

Calculate F+:

① Copy FDs

$\text{payment_method_id} \rightarrow \text{payment_method_id}, \text{customer_id}, \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

$\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{csv}$

② Add Reflexives

$\text{payment_method_id} \rightarrow \text{payment_method_id}, \text{customer_id}, \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

$\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

③ Add Rules

(No rules to apply)

Therefore, F+:

$\text{payment_method_id} \rightarrow \text{payment_method_id}, \text{customer_id}, \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

$\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

Calculate Fc:

① Eliminate Reflexives from F+:

$\text{payment_method_id} \rightarrow \text{customer_id}, \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

$\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{csv}$

② Eliminate Transitive Dependencies

Since we have

$\text{payment_method_id} \rightarrow \text{credit_card_num}, \text{credit_card_type}, \text{csv}$

$\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{csv}$

$\text{payment_method_id} \rightarrow \text{csv}$

Eliminate

$\text{payment_method_id} \rightarrow \text{csv}$

Therefore, Fc:

$\text{payment_method_id} \rightarrow \text{credit_card_num}, \text{credit_card_type}$
 $\text{credit_card_num}, \text{credit_card_type} \rightarrow \text{csv}$

From Fc, the 3NF tables are:

$\text{payment_method_id}, \text{credit_card_num}, \text{credit_card_type} //$
 $\text{credit_card_num}, \text{credit_card_type}, \text{csv}$

(Answer)

$\text{payment_method_id}, \text{credit_card_num}, \text{credit_card_type} //$
 $\text{credit_card_num}, \text{credit_card_type}, \text{csv}$