

Assignment 7

According to Bernstein's Algorithm, there are 4 steps to derive 3NF (which is lossless and dependency preserving)

Step 1 : Find out facts about the real world

The domains for the ticket database system are customer, event manager, artists and venues.

- customer(customer_id, first_name, last_name, email, phone_no, DOB, gender)
- ticket_type(ticket_id, ticket_number, ticket_class, section_number, seat_number, show_name, show_time_date)
- showinfo(show_id, show_title, artist_performers, location, all_show_times_dates, show_description, available_ticket_classes)
- ticket_distribution(distribution_receipt, ticket_id, customer_fname, customer_lname, customer_email, customer_phone_no, show_time_date, location)
- ticket_restriction(restriction_id, ticket_id, show_id, reason_for_restriction)

Step 2 : Reduce the list of functional dependencies (takes out redundancies)

1. customer

a) Rewrite FDs so RH is 1 attribute only:

customer_id → first_name
customer_id → last_name
customer_id → email
customer_id → phone_no
customer_id → DOB
customer_id → gender

b) Remove redundant FDs:

No redundant FDs

c) Minimize LH side:

customer_id \rightarrow first_name
customer_id \rightarrow last_name

A = customer_id
Q = customer_id

Customer_id+ = {first_name, last_name}

Already minimal dependency, repeat for all attributes.

customer_id \rightarrow first_name, last_name, email, phone_no, DOB, gender

2. ticket_type

ticket_type(ticket_id, ticket_number, ticket_class, section_number, seat_number, show_name)

- {ticket_id} \rightarrow ticket_number, ticket_class, section_number, seat_number, show_name

ticket_showinfo(show_name, show_time_date)

- {show_name} \rightarrow show_time_date

3. showinfo

{show_id} \rightarrow show_title, artist_performers, location, all_show_times_dates, show_description, available_ticket_classes

4. ticket_distribution

ticket_distribution(distribution_reciept, ticket_id, customer_email)

- {distribution_reciept} \rightarrow ticket_id
- {customer_email} \rightarrow distribution_reciept

ticket_distribution_customer(customer_fname, customer_lname, customer_email, customer_phone_no)

- {customer_email} \rightarrow customer_fname, customer_lname, customer_phone_no

ticket_distribution_ticket(ticket_id, show_time_date, location)

- {ticket_id, show_time_date, location} → distribution_receipt

5. ticket_restriction

- {restriction_id} → ticket_id, show_id, reason_for_restriction

Step 3 : Finding keys

The candidate keys for the tables are as follows:

customer : {customer_id}

customer_id → first_name

customer_id → last_name

customer_id → email

customer_id → phone_no

customer_id → DOB

customer_id → gender

The only attribute never on the RHS of any FD is customer_id.

ticket_type : {ticket_id}

Showinfo : {show_id}

ticket_distribution : {distribution_receipt,
 {customer_email},
 {ticket_id, show_time_date, location}}

ticket_restriction : {restriction_id}

Step 4 : Deriving the final schema

Since most of the tables were anyway in 3NF and there's very few redundancies, the final schema is almost the same. After combining FDs through union and eliminating smaller attribute relations, the final schemas are as follows:

customer

R1(customer_id, first_name, last_name, email, phone_no, DOB, gender)

ticket_type

R2(ticket_id, ticket_number, ticket_class, section_number, seat_number, show_name)

showinfo

R3(show_id, show_title, artist_performers, location, all_show_times_dates, show_description, available_ticket_classes)

ticket_distribution

R4(distribution_receipt, ticket_id)

R5(customer_email, customer_fname, customer_lname, customer_phone_no)

R6(ticket_id, show_time_date, location)

ticket_restriction

R7(restriction_id, ticket_id, show_id, reason_for_restriction)