customer(<u>customer_id</u>, first_name, last_name, email, phone_no, DOB, gender)

{customer_id} → first_name, last_name, email, phone_no, DOB, gender

This table is in 3NF because all non primary keys depend on the candidate key customer_id.

ticket_type(<u>ticket_id</u>, ticket_number, ticket_class, section_number, seat_number, show_name, show_time_date)

This table is not in 3NF since {show_name} → show_time_date. To make it 2NF, all non primary keys must functionally depend on the primary key. To make it 3NF, all non primary keys must be directly dependant on candidate keys. The relation must be split:

ticket_type(<u>ticket_id</u>, ticket_number, ticket_class, section_number, seat_number, show_name)

ticket_showinfo(<u>show_name</u>, show_time_date)

In this way, {show_name} → show_time_date and {ticket_id} → ticket_number, ticket_class, section_number, seat_number, show_name. Therefore show_name is dependant on ticket_id through show_name.

showinfo(<u>show_id</u>, show_title, artist_performers, location, all_show_times_dates, show_description, available_ticket_classes)

{show_id } → show_title, artist_performers, location, all_show_times_dates, show_description, available_ticket_classes

This table is in 3NF because all non primary keys are functionaly and directly dependant on the primary key.

ticket_distribution(<u>distribution_reciept</u>, <u>ticket_id</u>, customer_fname, customer_lname, customer_email, customer_phone_no, show_time_date, location)

This table is not in 3NF because {distribution_reciept} → ticket_id. Thus, it's not 2NF and can't be 3NF too. The relation should be split into:

ticket_distribution(<u>distribution_reciept</u>, <u>ticket_id</u>, customer_email)

ticket_distribution_customer(customer_fname, customer_lname, <u>customer_email</u>, customer_phone_no)

ticket_distribution_ticket(<u>ticket_id</u>, show_time_date, location)

ticket_restriction(<u>restriction_id</u>, <u>ticket_id</u>, <u>show_id</u>, reason_for_restriction)

This table is in 3NF because the non primary key reason_for_restriction requires the ID of the ticket and show, and its unique ID number to identify the restriction. Therefore, it is functionally and directly dependant on the candidate keys.