AIRPORT PROJECT CONCEPTS

-Mayank.S.

1. After mapping ER to relational model

a. Mapping of Regular Entity types:

For each strong Entity type E, a relation R is created which includes all the attributes of the entity type E. One or more attributes are chosen as the primary key of the relation R. We have created a relation for each of the entity type.

b. Mapping of Weak Entity types:

For each weak Entity type E, a relation R is created which includes all the attributes of the entity type E. Now add a foreign key attribute in R which references to the primary key attribute of the owner entity type, which is described by the identifying relationship of E. We add Airport code for ATC and Employee number for DEPENDENTS.

c. Mapping of 1:1 binary relationship types:

Choose any relation, say S, and add a foreign key attribute in S which references to the primary key attribute of other participating entity type.

We have added a foreign key Ticket_number in the relation PASSENGER.

d. Mapping of 1:N binary relationship types:

Choose the relation, say S, which represents the participating entity type at the N-side of the relationship type. Add a foreign key attribute in S which references to the primary key attribute of the other participating entity type.

We have added Airport_code in EMPLOYEE, Head_PN (Passport_number of head of the family/friends) in PASSENGER.

e. Mapping of M:N binary relationship types:

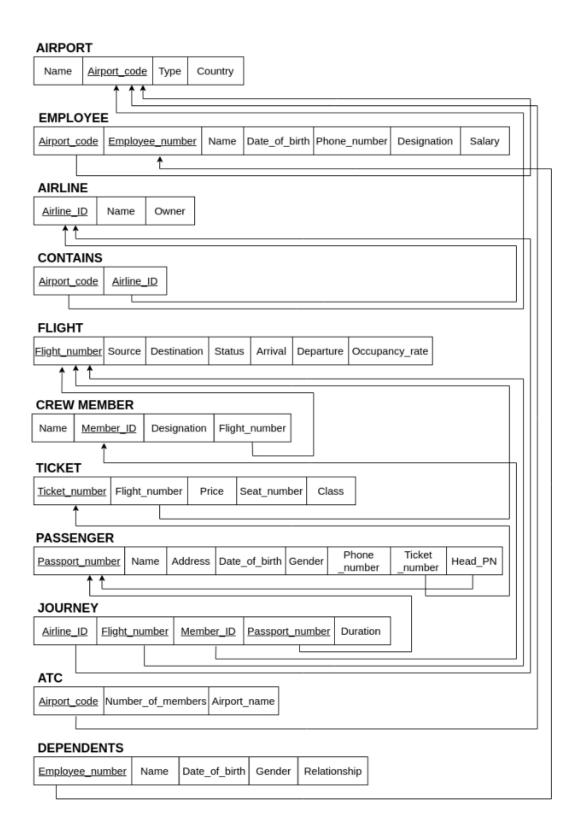
Create a new relation R for each of the M:N relationship types and include the primary keys of both the participating entity types as the foreign key attributes of R.

We have added a new relation CONTAINS which has Airport_code and Airline_ID as its attributes.

f. Mapping of N-ary relationship types:

For each n-ary relationship type R, create a new relationship relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

We have created a new relation JOURNEY which includes the primary key attributes of all the participating entity types.

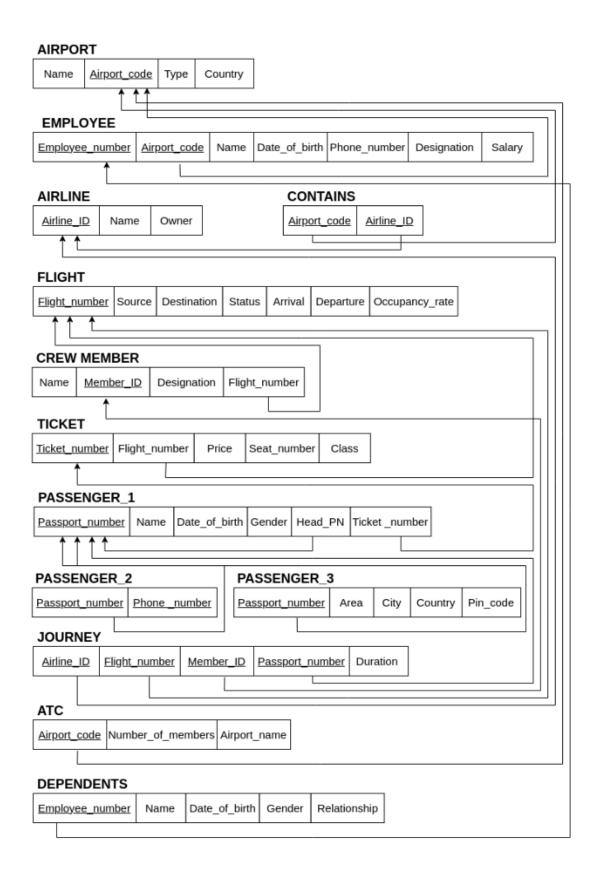


Relational model after conversion to 1NF

In order to remove multivalued attributes of the relation R, we create a new relation S and include primary key attribute of R and multivalued attribute of R as the primary key attributes of S. Also, remove the multivalued attribute from the relation R.

In order to remove composite attributes of the relation R, we create a new relation S and include primary key attribute of R and add some attributes (which describe the composite attribute of R) to S. Also, remove the composite attribute from the relation R.

We have split PASSENGER into PASSENGER_1, PASSENGER_2 and PASSENGER_3 Where PASSENGER_2 handles multivalued attribute Phone_number and PASSENGER_3 handles composite attribute Address. We have added attributes Area, City, Country and Pin_code in relation PASSENGER_3 to describe the address of a passenger.

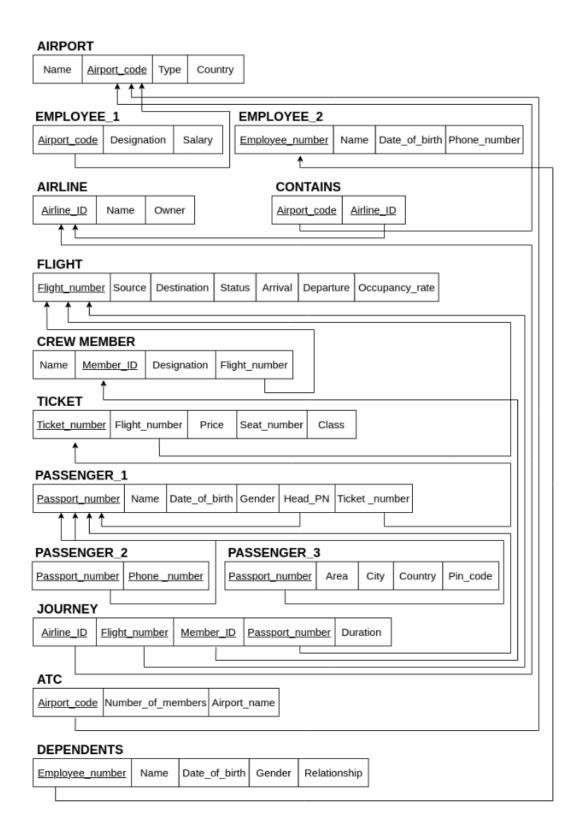


3. Relational model after conversion to 2NF

Every non-prime attribute should be fully functionally dependent on a primary key or a set of primary key attributes.

In order to follow 2NF, we have split EMPLOYEE into EMPLOYEE_1 and EMPLOYEE_2. EMPLOYEE_1 has Airport_code as its primary key and other attributes Designation and Salary are fully functionally dependent on it.

EMPLOYEE_2 has Employee_Number as its primary key and attributes Name, Date_of_birth and Phone_number are fully functionally dependent on it.



4. Relational model after conversion to 3NF

There should not be any transitive dependency in the relations.

In the relation TICKET, we can see that Ticket_number, Class are fully functionally dependent on Ticket_number & Price is fully functionally dependent on Ticket_number, Class.

So, Price is transitively dependent on the primary key Ticket_number.

In order to convert to 3NF, we split the relation TICKET into Ticket_1 and TICKET_2.

TICKET_1 contains Flight_number, Class as primary keys and Price as the attribute. TICKET_2 contains Ticket_number as the primary key and Flight_number, Seat_number and Class as the other attributes.

