

Project Phase - III

The mini-world is about the cab service in which customers book a ride and the driver nearby gets assigned to him, which is an innovative thought to simplify the transportation problems.

1. Converting EER model to Relational model.

→ Each regular (strong) entity type E in the ER schema, a relation R is created that includes all the simple attributes of E.

→ For each weak entity type W in the ER schema with owner entity type E, created a relation R and included all simple attributes of W as attributes of R. In addition, we included as foreign key attributes of R, the primary key attribute(s) of the relation that correspond to the owner entity type.

→ The composite attribute *Name* from the Entity_type Account is divided into two atomic attributes First name, Last name and are included in the relation Account.

→ The composite attribute *Description* from the Entity_type Vehicle_model, is divided into three atomic attributes 3-wheeler/4-wheeler, AC/Non-AC, numberOfSeats, and are included in the relation Vehicle_model.

→ The composite attribute *Driving License* from the Entity_type Driver, is divided into three atomic attributes LicenceNumber, DateOfIssue, ValidTill, and are included in the relation Driver.

→ The composite attribute *Valid_for* from the Entity_type Discount, is divided into two atomic attributes Valid_for_distance, Valid_for_amount, and are included in the relation Discount.

→ There is a multivalued attribute "Contact" in the entity-type "Account". For this a separate relation is created with contact number, Account Id (foreign key from other relation) as attributes.

→ There is a multivalued attribute "Contact" in the entity-type "Account". For this a separate relation is created with contact number, Account Id (foreign key from other relation) as attributes.

→ There is a multivalued attribute "Discount_code_applied" in the entity-type "Discount". For this a separate relation is created with Discount codes, discount Amount (foreign key from other relation) as attributes.

→ Binary relationships of 1:1 Type "FOR, OWNS" are mapped by including the primary key of one of the involved entities in the other as foreign key.

→ Binary relationship of n:1 Type "WITH" is mapped by including the primary key of Vehicle-model in the Vehicle.

→ N-ary relationships "GIVES, BOOKS" are mapped by creating a new-relationship R and including the primary keys of participating entity types as foreign keys in R.

Snapshot:

Link: <https://dbdiagram.io/d/6166e05e940c4c4eec937d33>

2. Conversion to first Normal form

First Normal Form states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.

→ All the composite attributes are decomposed into atomic attributes and multivalued attributes are mapped to the relational model as separate relations. So, the relational model is already in its first Normal form.

3. Converting 1NF to 2NF

The relational data is in 2NF if it is in 1NF and every non prime attribute is not partially dependent on any key.

The test for 2NF involves testing for functional dependencies whose LHS attributes are part of the key. If the key contains a single attribute, the test need not apply at all.

→ We have already converted it to 1NF, need to check only second condition

→ All the relations except "Discount" has only one attribute in all of their candidate keys. So it is required to check for Discount only. The candidate key of Discount is {Discount_code, Discount_Amount}, since we cannot uniquely identify Valid_for_distance or Valid_for_amount attributes with {Discount_code} or {Discount_Amount}, they are fully functionally dependent on {Discount_code, Discount_Amount}. Thus the condition satisfies

→ It is already in 2NF form.

4. Converting 2NF to 3NF

The relational data model is said to be in 3NF if it is in 2NF and there is no transitive dependency for non prime attributes in other terms if $x \rightarrow y$ then either x is super key or y is prime attribute.

→ In vehicle model relation we can see that the attributes {numberofseats, 3or4_wheeler and AC/_NON_AC} can determine the attribute cost_per_km but attributes {numberofseats, 3/4_wheeler and AC/_NON_AC} is not super key nor cost_per_km is prime attribute

→ In relation driver the only candidate keys are driver_id and licensenumber and minimal set of functional dependencies is

Driver_id \rightarrow {Dateofbirth, Licensenumber, Dateofjoining, DateofIssue, Valid_Till}

Licensenumber \rightarrow {Dateofbirth, Driver_id, Dateofjoining, DateofIssue, Valid_Till}

Since left side always has super key implies the relation is in 3NF

Snapshot:

Link: <https://dbdiagram.io/d/6166e05e940c4c4eec937d33>



