Database Project Part-3
Conversion of ER Model into Relational Schema

Conversion of ER Model into Relational Schema:

Changes

- Entity “Tour_Guide” – Relation “Tour_Guide,” Relation “Languages_Spoken.”
- Entity “Tour_Packages” – Relation “Tour_Package,” Relation “Places.”
- Entity “Customized_Booking” – Relation “Customized_Booking,” Relation “Tourists_ID.”
- Entity “Bookings” – Relation “Bookings,” Relation “Booking_Tourists_ID.”

Reasoning: In the process of translating a conceptual data model into a relational database schema, strong entities are represented as relations. Single attributes are straightforwardly included, while the constituent parts of composite attributes are incorporated as separate attributes. For each multi-valued attribute, a new relation is established, consisting of two attributes: one referencing the strong entity and the other serving to store the attribute. Together, these attributes form the primary key of the newly created relation.

Changes

- Entity “Reviews_and_Feedbacks” – Relation “Reviews_and_Feedbacks.”
- Entity “Offers” – Relation “Offers.”
- Entity “Bookings” – Relation “Bookings.”
- Entity “Customizable_Transport_Booking” – Relation “Customizable_Transport_Booking.”
- Entity “Customizable_Hotel_Booking” – Relation “Customizable_Hotel_Booking.”
- Entity “Customizable_Bookings” – Relation “Customizable_Bookings.”

Reasoning: For every weak entity, a new relation is created with its primary key composed of the partial key of the weak entity combined with the primary key of the weak entity. Additionally, all other simple attributes of the weak entity are included as attributes of the relation. The primary key of the weak entity is also added to the new relation as a foreign key.

Changes

- “Reviews and Feedback – given by – Tourists” relationship – Foreign Key on Tourist relation
- “Tour_Packages – has offer – Offers” relationship – Foreign Key on Tour_Packages relation
- “Travel_Agents – assisting for booking – Tourists” relationship – Foreign Key on Travel_Agents relation
- “Tourists – has booked package for – Tourists” relationship – Foreign Key on Tourists relation
- “Tour_Guide – Assigned to – Tour_Packages” relationship – New Relation
- “Tourist – travels together with – Tourists” relationship – New Relation

Reasoning: For a 1:N or N:1 relationship type, include the primary key of the participating entity on the N-side as a foreign key in the participating entity on the 1-side. This is necessary because each entity instance on the N-side is related to at least one entity instance on the 1-side.

Changes

- "Tour_Guide – Assigned to – Tour_Packages" relationship – New Relation
- "Tourist – travels together with – Tourists" relationship – New Relation

Reasoning: For N:M relationship types R, a new relation S is created to represent R. The foreign keys in S are derived from the primary keys of the participating entities in R. The primary key of S is formed by combining these foreign keys. Since there are no other simple attributes for such relationship types, S does not include any additional attributes.

1. First Normal Form (1NF)

Goal: Ensure all columns contain atomic, indivisible values and each column contains values of a single type.

2. Second Normal Form (2NF)

Goal: Ensure the table is in 1NF and all non-key attributes are fully functionally dependent on the entire primary key, not just part of it.

3. Third Normal Form (3NF)

Goal: Ensure the table is in 2NF and all non-key attributes are only dependent on the primary key, removing any transitive dependencies.

4. Boyce-Codd Normal Form (BCNF)

Goal: A stronger version of 3NF where every determinant is a candidate key, addressing anomalies that 3NF might not cover.

5. Fourth Normal Form (4NF)

Goal: Ensure the table is in BCNF and has no multi-valued dependencies, where a single attribute is associated with multiple independent attributes.

6. Fifth Normal Form (5NF)

Goal: Ensure the table is in 4NF and that all join dependencies are implied by candidate keys, preventing redundancy from decomposition.

6. Sixth Normal Form (6NF)

Goal: Ensure the table is in 5NF and deals with temporal data, focusing on the decomposition of tables into smaller tables that capture changes over time.

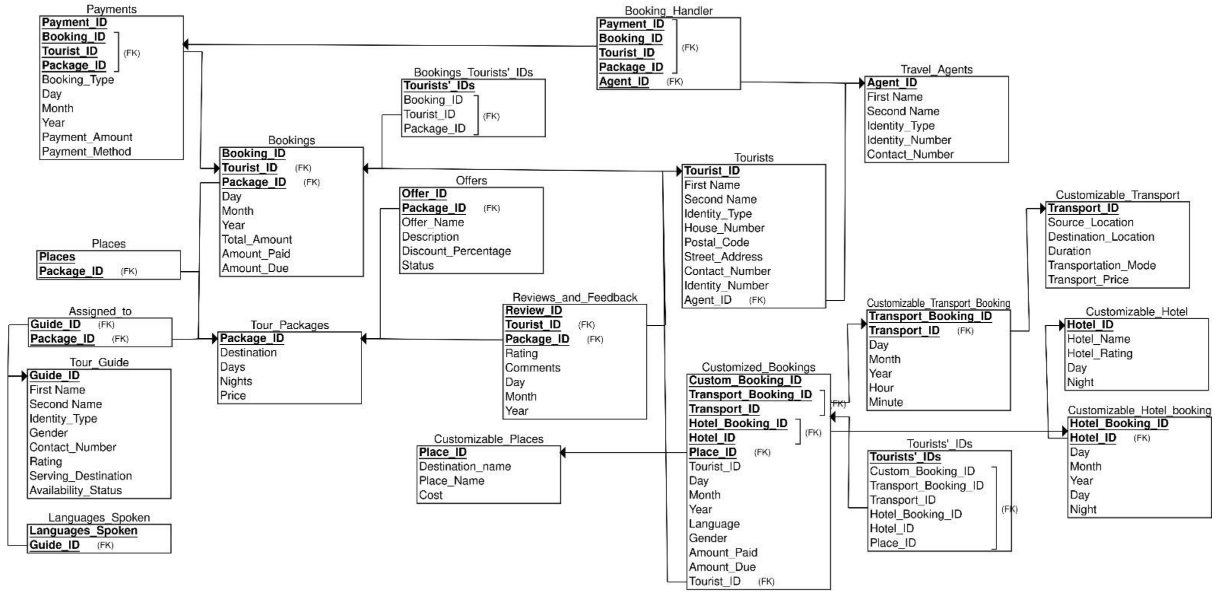

Figure 1: Initial Schema

Normalization

Changes made to schema while converting to 1NF:

The database tables adhere to the principles of First Normal Form, where each attribute is assigned a well-defined data type, ensuring that all values are atomic and singular within their respective domains. Every relation possesses a primary key, and there are no instances of duplicate or repeating attributes. Therefore, there are no changes required while converting to 1NF form.

Changes made to schema while converting to 2NF:

The relations Payments, Bookings, Offers, Reviews_and_Feedback, Customized_Bookings, Customized_Transport Booking, and Customized Hotel Booking have been decomposed into Payments1 and Payments2, Bookings1 and Bookings2, and so on, respectively. This was because the attributes in the second relation of each were functionally dependent on only a part of the primary key of the original relation, thus a separate relation has been created for them.

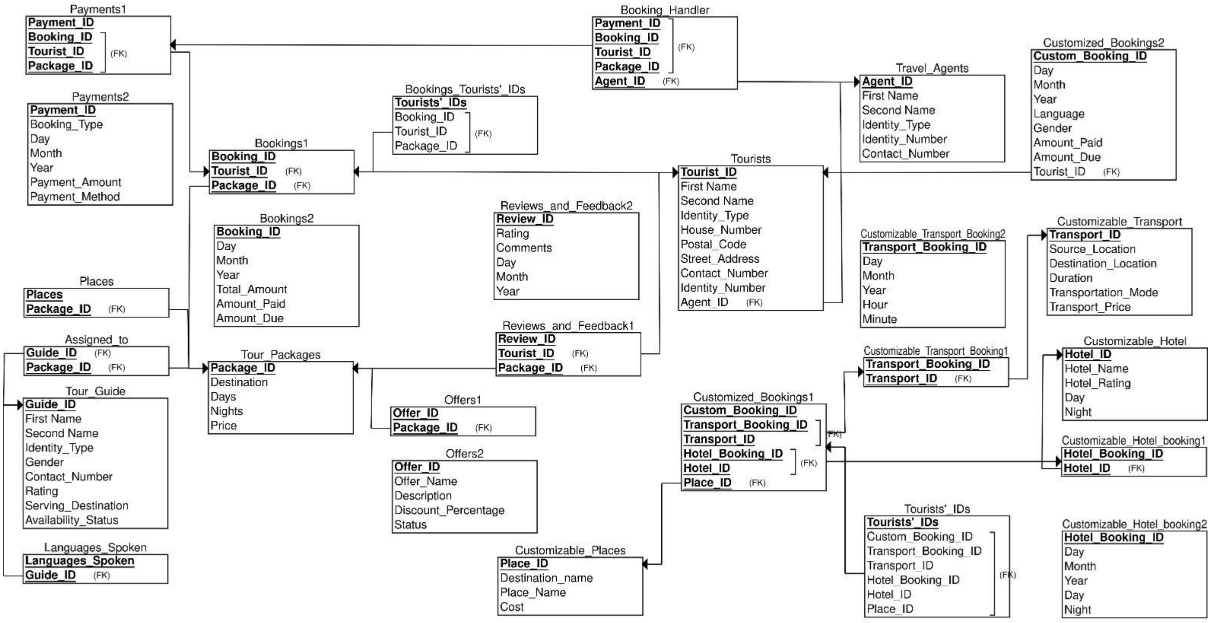

Figure 2: Schema After 2NF

Changes made to schema while converting to 3NF:

We don't need to go to Third Normal Form (3NF) because the tables are already cool without any transitive dependencies. We've already made sure that non-prime attributes totally depend on the primary key, and any potential extra dependencies have been sorted out. Therefore, we're sticking to a design that avoids redundancy and makes our database easy to handle.