“team won” Phase 3 - Relational Model

Mapping of ER diagram to Relational Model

Step 1: Mapping of Strong Entity types:

- Employee(Employee_Id, Name, Salary, DOB)
- Building(Building_ID, Last_Maintenance)
- Ticket_Counter(Counter_ID, State)
- Platform(Platform_No, Length, Last_Cleaning)
- Passenger(Passenger_ID, Sex, DOB)
- Train(Train_Number, Train_Name, Train_Type, Num_Coach, Last_Maintenance)

Step 2: Mapping of Weak Entity Types:

- Ticket(Train_Number references Train(Train_Number) on delete restrict on update cascade, Ticket_Number, Ticket_Type, Berth, Coach, Date, Cost)
- TimeTable(Train_Number references Train(Train_Number) on delete cascade on update cascade, Platform_Number references Platform(Platform_No) on delete cascade on update cascade, Arrival_Time, Departure_Time)

Step 3: Mapping of 1:1 Binary Relationship

Mapping of “cleans” relationship is done by including “Cleaning_Staff_ID” as a foreign key in the Platform relation.

- Platform(Platform_No, Length, Last_Cleaning, Cleaning_Staff_ID references CleaningStaff(Employee_ID) on delete set null on update cascade)

Step 4: Mapping of 1:N Binary Relationships

- Mapping of “supervises” relationship is done by including Manager_ID as a foreign key in Employee.

- Employee(Employee_Id, Name, Salary, DOB, Manager_ID references Manager(Employee_ID) on delete set null on update cascade)

- Mapping of “works in” relationship is done by including Works_in_building as a foreign key in Employee.

- Employee(Employee_Id, Name, Salary, DOB, Manager_ID references Manager(Employee_ID) on delete set null on update cascade, Works_In_Building references Building(Building_ID) on delete set null on update cascade))

- Mapping of “sells” relationship is done by including “Sold_by” as a foreign key in Ticket entity.

- Ticket(Train_Number references Train(Train_Number) on delete restrict on update cascade, Ticket_Number, Berth, Coach, Date, Cost, Sold_by references Ticket_Counter(Counter_ID) on delete set nulll on update cascade)

Step 5: Mapping of M:N relationships

Mapping of “checks” relationship is done by cross-reference method by introducing a new relation “Checked_By”

- Checked_By(Employee_ID referencing Ticket_Checker(Employee_ID) on delete restrict on update cascade, Ticket_Number references Ticket(Ticket_Number) on delete restrict on update cascade, Time_Checked))

Step 6: Mapping of Multi-valued attributes

- Assigned_Buildings(Staff_ID references Cleaning_Staff(Employee_ID) on delete cascade on update cascade, Building_ID references Building(Building_ID) on delete cascade on update cascade)

- Usable_By(Platform_ID references Platform(Platform_No) on delete cascade on update cascade, Train_Type)

Step 7: Mapping of N-ary relationship

A new relationship “boards” is created.

- Boards(Train_Number references Train(Train_Number) on delete cascade on update cascade, Ticket_Number references Ticket(Ticket_Number) on delete cascade on update cascade, Platform_Number references Platform(Platform_No) on delete cascade on update cascade, Passenger_ID references Passenger(Passenger_ID) on delete cascade on update cascade)

Step 8: Mapping of Subclasses:

- Engineer(Employee_ID)
- Station_Master(Employee_ID)
- Cleaning_Staff(Employee_ID)
- Manager(Employee_ID)

- Ticket_Checker(Employee_ID)
- Food_Store(Building_ID)
- Enquiry(Building_ID)
- Toilet(Building_ID)
- Ticket_Counter(Building_ID, Counter_ID, state)

Final Relational Model:

1. Employee(Employee_Id, Name, Salary, DOB, Manager_ID references Manager(Employee_ID) on delete set null on update cascade, Works_In_Building references Building(Building_ID) on delete set null on update cascade)
2. Building(Building_ID, Last_Maintenance)
3. Passenger(Passenger_ID, Sex, DOB)
4. Train(Train_Number, Train_Name, Train_Type, Num_Coach, Last_Maintenance)
5. Ticket(Train_Number references Train(Train_Number) on delete restrict on update cascade, Ticket_Number, Berth, Coach, Date, Cost, Sold_by references Ticket_Counter(Counter_ID) on delete set null on update cascade)
6. Platform(Platform_No, Length, Last_Cleaning, Cleaning_Staff_ID references CleaningStaff(Employee_ID) on delete set null on update cascade)
7. TimeTable(Train_Number references Train(Train_Number) on delete cascade on update cascade, Platform_Number references Platform(Platform_No) on delete cascade on update cascade, Arrival_Time, Departure_Time)
8. Checked_By(Employee_ID referencing Ticket_Checker(Employee_ID) on delete restrict on update cascade, Ticket_Number references Ticket(Ticket_Number) on delete restrict on update cascade, Time_Checked))
9. Assigned_Buildings(Staff_ID references Cleaning_Staff(Employee_ID) on delete cascade on update cascade, Building_ID references Building(Building_ID) on delete cascade on update cascade)
10. Usable_By(Platform_ID references Platform(Platform_No) on delete cascade on update cascade, Train_Type)
11. Boards(Train_Number references Train(Train_Number) on delete cascade on update cascade, Ticket_Number references Ticket(Ticket_Number) on delete cascade on update cascade, Platform_Number references Platform(Platform_No) on delete cascade on update cascade, Passenger_ID references Passenger(Passenger_ID) on delete cascade on update cascade)
12. Engineer(Employee_ID)
13. Station_Master(Employee_ID)
14. Cleaning_Staff(Employee_ID)
15. Manager(Employee_ID)
16. Ticket_Checker(Employee_ID)
17. Food_Store(Building_ID)
18. Enquiry(Building_ID)
19. Toilet(Building_ID)
20. Ticket_Counter(Building_ID, Counter_ID, state)

Normalization:

1NF:

The schema is already in 1NF as we have removed multi-valued attributes during the creation of the relational model.

2NF:

The schema is already in 2NF. No relations have a functional dependency from a partial key to some non-prime attribute.

3NF:

The schema is already in 3NF. No relations have a functional dependency from a set of non-prime attributes to some other non-prime attribute.

Changes made:

1. Removed train_name as a primary key.
2. Fixed cardinality of “checks” relationship in ER diagram to M:N
3. Removed “counter ID” attribute
4. Renamed Customer ID attribute to Passenger ID in Passenger relation.

Example Database state:

Employee(Employee_Id, Name, Salary, DOB, Manager_ID references Manager(Employee_ID) on delete set null on update cascade, Works_In_Building references Building(Building_ID) on delete set null on update cascade)

a. Employee(1, ”A”, 100000, 10-11-1994, 1, 1)
b. Employee(2, ”B”, 200000, 25-04-1995, 2, 2)
c. Employee(3, ”C”, 100000, 26-05-1996, 3, 3)
d. Employee(4, ”D”, 100000, 15-07-1990, 2, 3)

Building(Building_ID, Last_Maintenance)

a. Building(1, 30-10-2019)

b. Building(2, 31-10-2019)
c. Building(3, 01-10-2019)
d. Building(4, 01-10-2019)

Platform(Platform_No, Length, Last_Cleaning, Cleaning_Staff_ID references CleaningStaff(Employee_ID) on delete set null on update cascade)

a. Platform(1, 100,30-10-2019, 3)
b. Platform(2, 200,31-10-2019, 3)
c. Platform(3, 300,29-10-2019, 3)

Passenger(Passenger_ID, Sex, DOB)

a. Passenger(1, Male,10-11-1999)
b. Passenger(2, Female,11-08-1999)
c. Passenger(3, Male,02-09-1997)

Train(Train_Number, Train_Name, Train_Type, Num_Coach, Last_Maintenance)

a. Train(12345, Budha_Express, Express, 11, 01-11-2019)
b. Train(16543, Amaravati_Superfast_Express, Superfast, 12, 31-10-2019)
c. Train(13425, Hyderabad_Local, Local, 10, 31-10-2019)

Ticket(Train_Number references Train(Train_Number) on delete restrict on update cascade, Ticket_Number, Berth, Coach, Date, Cost, Sold_by references Ticket_Counter(Counter_ID) on delete set null on update cascade)

a. Ticket(16543, 0011, Sleeper, 35, 5, 02-11-2019, 350, 1)

TimeTable(Train_Number references Train(Train_Number) on delete cascade on update cascade, Platform_Number references Platform(Platform_No) on delete cascade on update cascade, Arrival_Time, Departure_Time)

a. Timetable(12345, 1, 17:20, 17:30)

Checked_By(Employee_ID referencing Ticket_Checker(Employee_ID) on delete restrict on update cascade, Ticket_Number references Ticket(Ticket_Number) on delete restrict on update cascade, Time_Checked))

a. Checked_By(2, 0011, 11:50)

Assigned_Buildings(Staff_ID references Cleaning_Staff(Employee_ID) on delete cascade on update cascade, Building_ID references Building(Building_ID) on delete cascade on update cascade)

a. Assigned_Buildings(3, 1)

Usable_By(Platform_ID references Platform(Platform_No) on delete cascade on update cascade, Train_Type)

a. Usable_By(1, Express)
b. Usable_By(1, Local)
c. Usable_By(1, Superfast)

Boards(Train_Number references Train(Train_Number) on delete cascade on update cascade, Ticket_Number references Ticket(Ticket_Number) on delete cascade on update cascade, Platform_Number references Platform(Platform_No) on delete cascade on update cascade, Passenger_ID references Passenger(Passenger_ID) on delete cascade on update cascade)

a. Boards(16543, 0011, 3, 1)

Engineer(Employee_ID)

a. Engineer(4)

Station_Master(Employee_ID)

a. Station_Master(1)

Cleaning_Staff(Employee_ID)

a. Cleaning_Staff(3)

Manager(Employee_ID)

a. Manager(1)

Ticket_Checker(Employee_ID)

a. Ticket_Checker(2)

Food_Store(Building_ID)

a. Food_Store(2)

Enquiry(Building_ID)

a. Enquiry(3)

Toilet(Building_ID)

a. Toilet(4)

Ticket_Counter(Building_ID, state)

a. Ticket_Counter(1, OPEN)