4a) COMMISSION (PromoterId, ProductId, TotalSale, CommissionPaid)

Find the functional dependencies and the minimal keys valid in the relational table and determine the highest normal form of the relational table.

TotalSale → CommissionPaid

CommissionPaid → TotalSale

(PromoterId, ProductId) → TotalSale, CommissionPaid

From the functional dependencies above, PromoterId and ProductId are the minimal super key of the relational table Commission which make them as composite key. TotalSale and CommissionPaid are non-primary keys which are functionally dependent on PromoterId and ProductId.

Attribute PromoterId in relation to TotalSale, CommissionPaid is fully functional dependant on attribute ProductId (composite key) of TotalSale, CommissionPaid where they are not functionally dependent upon the subset of ProductId.

Since the commission is computed based on a total sale for the product, for example, if the total sale for a product is below $1000, a promoter is paid 10% of the total sale for commission, this explains CommissionPaid→TotalSale.

Since the commission is computed based on a total sale for the product, for example, if the total sale for a product is between $1000 and $5000, a promoter is paid 20%, this explains TotalSale→CommissionPaid.

CommissionPaid→TotalSale and TotalSale→CommissionPaid show transitive dependency as they are attributes of a relation table Commission they are neither a candidate key nor a subset of any key of the relation table Commission.

The above functional dependencies are in 2NF as they are already in first normal form (1NF) and every non-primary-key attribute is fully functionally dependent on the primary key (PromoterId, ProductId).

4b) EmployeeProject (EmpeNum, ProjNum, HoursWork, DateStartWorkOnProj)

(EmpeNum, ProjNum) → HoursWork, DateStartWorkOnProj

Since the employees are involved in many projects and that each project may have many employees working on it, this makes EmployeeProject the association class making EmpeNum and ProjNum the minimal super key of the relation table EmployeeProject. With this, HoursWork and DateStartWorkOnProj attributes fully functionally dependent on the minimal super key. The relation table EmployeeProject is in Boyce-Codd Normal Form (BCNF) as every determinant is a candidate key.

4c) WallClimbDemo (LevelNum, WallNum, PlaceAvailable, DemoType)

LevelNum → WallNum

DemoType → LevelNum

(LevelNum, WallNum) → DemoType, PlaceAvailable

From the above functional dependencies, LevelNum, WallNum, DemoType attributes made up the minimal super key which are composite keys of the relational table WallClimbDemo that determines PlaceAvailable.

Level 1 consists of three different walls; each wall is identified as wall 1, wall 2, and wall 3. Level 2 consists of one wall, and it is identified as wall 1. This shows LevelNum is dependent on WallNum which explains LevelNum → WallNum.

Walls at level 1 are designed for easy and intermediate climbing, while wall at level 2 is designed for expert level of climbing. To promote this sport, the organizer organizes demonstration sessions inviting public for viewing. This shows DemoType → LevelNum, making it a non-trivial dependency that violates the Boyce-Codd Normal Form (BCNF) as DemoType is not a candidate key.

The above functional dependencies are in 3NF as there are no partial and transitive dependencies but not in BCNF as there exists a non-trivial dependency. Hence relational table WallClimbDemo is in 3NF as all non-primary key attributes are fully functional dependent on the minimal super key.

4d) CustomerCreditCard (custNum, custName, address, postalCode, cardType, cardNumber, cardExpDate)

custNum → custName

custNum → address

custNum → postalCode

address → postalCode

cardNumber → cardType

cardNumber → cardExpDate

(custNum, cardNumber) → custName, address, postalCode, cardType, cardExpDate

From the above functional dependencies, custNum and cardNumber made up the composite key of the relational table of CustomerCreditCard. The two attributes are the minimal super key. There are also partial dependencies which are custNum → custName, address, postalCode and cardNumber → cardType, cardExpDate. There is a transitive dependency where address → postalCode. Hence this shows that CustomerCreditCard is in 1NF as not every non-primary key attribute is fully functionally dependent on the primary key as there exists partial dependency.