Definition of Relation Schema

- OrderDetails (orderDetailID, quantity, price)
- IsAssociatedWith(*orderDetailID*, *orderID*)
- Order(orderID, orderDate, totalAmount, status)
- IsMadeBy(*orderID*, *customerID*)
- Customer(customerID, firstName, lastName, email, phone, address)
- IsOwnedBy(*customerID*, *petID*)
- Pet(petID, petName, age, species, breed)

BCNF Proof of Relation Schema:

- OrderDetails (orderDetailID, quantity, price)

In this scenario, orderDetailID is the candidate key, and it determines all the other attributes. There are no partial dependencies because the candidate key uniquely determines all the non-prime attributes.
Since all non-prime attributes (quantity, price) are fully functionally dependent on the candidate key (orderDetailID), the OrderDetails relation schema is already in BCNF. There are no violations of BCNF in this schema.
In summary, the OrderDetails relation schema is in BCNF because it meets the conditions of BCNF: no partial dependencies and all non-prime attributes are dependent on the candidate key.

- IsAssociatedWith(*orderDetailID*, *orderID*)

Functional Dependencies:
{*orderDetailID*, *orderID*} -> {other non-prime attributes, if any}
The composite key {*orderDetailID*, *orderID*} is indeed the candidate key, and there won't be any partial dependencies, and it satisfies BCNF, as long as the functional dependencies are satisfied. If there are other non-prime attributes dependent on {*orderDetailID*, *orderID*}, then it's in BCNF.

- Order(orderID, orderDate, totalAmount, status)

In this scenario, orderID is the candidate key, and it determines all the other attributes. There are no partial dependencies because the candidate key uniquely determines all the non-prime attributes.
Since all non-prime attributes (orderDate, totalAmount, status) are fully functionally dependent on the candidate key (orderID), the Order relation schema is already in BCNF. There are no violations of BCNF in this schema.

In summary, the Order relation schema is in BCNF because it meets the conditions of BCNF: no partial dependencies and all non-prime attributes are dependent on the candidate key.

- IsMadeBy(*orderID*, *customerID*)

Functional Dependencies:
{*orderID*, *customerID*} -> {other non-prime attributes, if any}
The composite key {*orderID*, *customerID*} is indeed the candidate key, and there won't be any partial dependencies, and it satisfies BCNF, as long as the functional dependencies are satisfied. If there are other non-prime attributes dependent on {*orderID*, *customerID*}, then it's in BCNF.

- Customer(customerID, firstName, lastName, email, phone, address)

In this scenario, customerID is the candidate key, and it determines all the other attributes. There are no partial dependencies because the candidate key uniquely determines all the non-prime attributes.
Since all non-prime attributes (firstName, lastName, email, phone, address) are fully functionally dependent on the candidate key (customerID), the Customer relation schema is already in BCNF. There are no violations of BCNF in this schema.
In summary, the Customer relation schema is in BCNF because it meets the conditions of BCNF: no partial dependencies and all non-prime attributes are dependent on the candidate key.

- IsOwnedBy(*customerID*, *petID*)

Functional Dependencies:
{*customerID*, *petID*} -> {other non-prime attributes, if any}
The composite key {*customerID*, *petID*} is indeed the candidate key, and there won't be any partial dependencies, and it satisfies BCNF, as long as the functional dependencies are satisfied. If there are other non-prime attributes dependent on {*customerID*, *petID*}, then it's in BCNF.

- Pet(petID, petName, age, species, breed)

In this scenario, petID is the candidate key, and it determines all the other attributes. There are no partial dependencies because the candidate key uniquely determines all the non-prime attributes.
Since all non-prime attributes (petName, age, species, breed) are fully functionally dependent on the candidate key (petID), the Pet relation schema is already in BCNF. There are no violations of BCNF in this schema.
In summary, the Pet relation schema is in BCNF because it meets the conditions of BCNF: no partial dependencies and all non-prime attributes are dependent on the candidate key.