Henrique Lima

Fall 2019

**Normalized Logical Schema Version.1 Based on: Logical Schema Version.3**

**Assistant**

|  |  |  |
| --- | --- | --- |
| AssistID | Name | HoursWorked |

**Candidate Keys:** AssistID is the only CK since any combination of Name and HoursWorked will not guarantee any uniqueness.

**Functional Dependencies:**

AssistID - > Name

AssistID - > HoursWorked

**1NF:** Assistant is already in first normal form because there are no composites and all non-key attributes need a key attribute.

**2NF:** Assistant is already in second normal form because there are no partial candidate keys.

**3NF:** Assistant is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Assistant is already in BCNF because there is only one key attribute.

**Groomer**

|  |  |  |
| --- | --- | --- |
| GroomID | Name | HoursWorked |

**Candidate Keys:** GroomID is the only CK since any combination of Name and HoursWorked will not guarantee any uniqueness.

**Functional Dependencies:**

GroomID - > Name

GroomID - > HoursWorked

**1NF:** Groomer is already in first normal form because there are no composites and all non-key attributes need a key attribute.

**2NF:** Groomer is already in second normal form because there are no partial candidate keys.

**3NF:** Groomer is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Groomer is already in BCNF because there is only one key attribute.

**Customer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Street | City | Zip | State | DateOfService |

**Candidate Keys:** ID is the only CK since any combination of Name, Street, City, Zip, State, and DateOfService will not guarantee any uniqueness. Also, DateOfService is optional.

**Functional Dependencies:**

ID - > Name

ID - > Street

ID - > City

ID - > Zip

ID - > State

ID - > DateOfService

**1NF:** Customer is not in first normal form because there are composites in the table.

**2NF:** Customer is already in second normal form because there are no partial candidate keys.

**3NF:** Customer is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Customer is already in BCNF because there is only one key attribute

**Pet**

|  |  |  |
| --- | --- | --- |
| AniID | AnimalName | RabiesDone |

**Candidate Keys:** AniID is the only CK since any combination of AnimalName and RabiesDone will not guarantee any uniqueness.

**Functional Dependencies:**

AniID - > AnimalName

AniID - > RabiesDone

**1NF:** Pet is already in first normal form because there are no composites and all non-key attributes need a key attribute.

**2NF:** Pet is already in second normal form because there are no partial candidate keys.

**3NF:** Pet is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Pet is already in BCNF because there is only one key attribute.

**Visit**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GroomID | AssistID | StartOfAppointment | EndOfAppointment | Total |

**Candidate Keys:** GroomID is the only CK since any combination of AssistID and StartOfAppointment, EndOfAppointment, and Total will not guarantee any uniqueness. Also, AssistID is optional.

**Functional Dependencies:**

GroomID - > AssistID

GroomID - > StartOfAppointment

GroomID - > EndOfAppointment

GroomID - > Total

**1NF:** Visit is already in first normal form because there are no composites and all non-key attributes need a key attribute.

**2NF:** Visit is already in second normal form because there are no partial candidate keys.

**3NF:** Visit is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Visit is already in BCNF because there is only one key attribute.

**Visit\_NameOfProcedures**

|  |  |
| --- | --- |
| GroomID | NameOfProcedure |

**Candidate Keys:** GroomID is the only CK since any combination of NameOfProcedure will not guarantee any uniqueness.

**Functional Dependencies:**

GroomID - > NameOfProcedure

**1NF:** Visit\_NameOfProcedures is already in first normal form because there are no composites and all non-key attributes need a key attribute.

**2NF:** Visit\_NameOfProcedures is already in second normal form because there are no partial candidate keys.

**3NF:** Visit\_NameOfProcedures is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Visit\_NameOfProcedures is already in BCNF because there is only one key attribute.

**Visit\_CostOfProcedures**

|  |  |
| --- | --- |
| GroomID | CostOfProcedure |

**Candidate Keys:** GroomID is the only CK since any combination of CostOfProcedure will not guarantee any uniqueness.

**Functional Dependencies:**

GroomID - > CostOfProcedure

**1NF:** Visit\_CostOfProcedure is already in first normal form because there are no composites and all non-key attributes need a key attribute.

**2NF:** Visit\_CostOfProcedure is already in second normal form because there are no partial candidate keys.

**3NF:** Visit\_CostOfProcedure is already in third normal form because there are no non-key attributes that need anything other than a key attribute.

**BCNF:** Visit\_CostOfProcedure is already in BCNF because there is only one key attribute.

**Normalized Logical Schema**

**Groomer**(GroomID, Name\*, HoursWorked\*)

**Assistant**(AssistID, Name\*, HoursWorked\*)

**Customer**(ID, Name\*, Street\*, City\*, State\*, Zip\*, DateOfService)

**Pet**(AniID, AnimalName\*, RabiesDone\*)

**Visit**(GroomID, AssistID, StartOfAppointment\*, EndOfAppointment\*, Total\*)

**GroomID references Groomer.GroomID; AssistID references Assistant.AssistID;**

**Visit\_NameOfProcedures**(GroomID, NameOfProcedure)

**GroomID references Groomer.GroomID;**

**Visit\_CostOfProcedures**(GroomID, CostOfProcedure)

**GroomID references Groomer.GroomID;**