Name: Gia Dao

Student ID: 1001747062

Honors Contract

Introduction:

**The normalization process** takes a relation schema through a series of tests to certify whether it satisfies a certain normal form. The process, which proceeds in a top-down fashion by evaluating each relation against the criteria for normal forms and decomposing relations as necessary, can thus be considered as relational design by analysis.

**Normalization of data** can be considered a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of (1) minimizing redundancy and (2) minimizing the insertion, deletion, and (3) update anomalies. It can be considered as a “filtering” or “purification” process to make the design have successively better quality. An unsatisfactory relation schema that does not meet the condition for a normal form—the normal form test—is decomposed into smaller relation schemas that contain a subset of the attributes and meet the test that was otherwise not met by the original relation. Thus, the normalization procedure provides database designers with the following:

* A formal framework for analyzing relation schemas based on their keys and on the functional dependencies among their attributes
* A series of normal form tests that can be carried out on individual relation schemas so that the relational database can be normalized to any desired degree

**Definition**: **The normal form** of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized.

**First normal form**: First normal form (1NF) is now considered to be part of the formal definition of a relation in the basic (flat) relational model; historically, it was defined to disallow multivalued attributes, composite attributes, and their combinations. In other words, 1NF disallows relations within relations or relations as attribute values within tuples. The only attribute values permitted by 1NF are single atomic (or indivisible) values.

**Second normal form**: Second normal form (2NF) is based on the concept of full functional dependency. A functional dependency X → Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold anymore; that is, for any attribute A ε X, (X − {A}) does not functionally determine Y. A functional dependency X → Y is a partial dependency if some attribute A ε X can be removed from X and the dependency still holds; that is, for some A ε X, (X − {A}) → Y.

Definition: A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

**Third normal form**: Third normal form (3NF) is based on the concept of transitive dependency. A functional dependency X → Y in a relation schema R is a transitive dependency if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R, and both X → Z and Z → Y hold.

Definition: According to Codd’s original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.

CUSTOMER TABLE:

|  |  |  |
| --- | --- | --- |
| CustID | Name | Phone |
| 201 | A. Parks | (214) 555-0127 |
| 202 | S. Patel | (849) 811-6298 |
| 203 | A. Hernandez | (355) 572-5385 |
| 204 | G. Carver | (753) 763-8656 |
| 205 | Sh. Byers | (912) 925-5332 |
| 206 | L. Lutz | (931) 966-1775 |
| 207 | L. Bernal | (884) 727-0591 |
| 208 | I. Whyte | (811) 979-7345 |
| 209 | L. Lott | (954) 706-2219 |
| 210 | G. Clarkson | (309) 625-1838 |
| 211 | Sh. Dunlap | (604) 581-6642 |
| 212 | H. Gallegos | (961) 265-8638 |
| 213 | L. Perkins | (317) 996-3104 |
| 214 | M. Beach | (481) 422-0282 |
| 215 | C. Pearce | (599) 881-5189 |
| 216 | A. Hess | (516) 570-6411 |
| 217 | M. Lee | (369) 898-6162 |
| 218 | R. Booker | (730) 784-6303 |
| 219 | A. Crowther | (325) 783-4081 |
| 220 | H. Mahoney | (212) 262-8829 |
| 221 | J. Brown | (644) 756-0110 |
| 222 | H. Stokes | (931) 969-7317 |
| 223 | J. Reeves | (940) 981-5113 |
| 224 | A. Mcghee | (838) 610-5802 |
| 225 | L. Mullen | (798) 331-7777 |
| 226 | R. Armstrong | (325) 783-4081 |
| 227 | J. Greenaway | (212) 262-8829 |
| 228 | K. Kaiser Acosta | (228) 576-1557 |
| 229 | D. Kirkpatrick | (773) 696-8009 |
| 230 | A. Odonnell | (439) 536-8929 |
| 231 | K. Kay | (368) 336-5403 |

From the CUSTOMER table above, there’s no column that consist of composite or multi-valued data and each line is unique; therefore, the CUSTOMER table is already in 1st NF.

The functional dependencies from the table above: CustID -> Name, CustID -> Phone

From the table, the ‘phone’ column does not necessarily depend on the name, as we can have two different persons but shared the same name; therefore, we split the table into 2 sub-tables, which satisfy the condition of 2nd NF that the non-prime attributes have to be fully functional dependent on the primary key.

|  |  |
| --- | --- |
| CustID | Name |
| 201 | A. Parks |
| 202 | S. Patel |
| 203 | A. Hernandez |
| 204 | G. Carver |
| 205 | Sh. Byers |
| 206 | L. Lutz |
| 207 | L. Bernal |
| 208 | I. Whyte |
| 209 | L. Lott |
| 210 | G. Clarkson |
| 211 | Sh. Dunlap |
| 212 | H. Gallegos |
| 213 | L. Perkins |
| 214 | M. Beach |
| 215 | C. Pearce |
| 216 | A. Hess |
| 217 | M. Lee |
| 218 | R. Booker |
| 219 | A. Crowther |
| 220 | H. Mahoney |
| 221 | J. Brown |
| 222 | H. Stokes |
| 223 | J. Reeves |
| 224 | A. Mcghee |
| 225 | L. Mullen |
| 226 | R. Armstrong |
| 227 | J. Greenaway |
| 228 | K. Kaiser Acosta |
| 229 | D. Kirkpatrick |
| 230 | A. Odonnell |
| 231 | K. Kay |

|  |  |
| --- | --- |
| CustID | Phone |
| 201 | (214) 555-0127 |
| 202 | (849) 811-6298 |
| 203 | (355) 572-5385 |
| 204 | (753) 763-8656 |
| 205 | (912) 925-5332 |
| 206 | (931) 966-1775 |
| 207 | (884) 727-0591 |
| 208 | (811) 979-7345 |
| 209 | (954) 706-2219 |
| 210 | (309) 625-1838 |
| 211 | (604) 581-6642 |
| 212 | (961) 265-8638 |
| 213 | (317) 996-3104 |
| 214 | (481) 422-0282 |
| 215 | (599) 881-5189 |
| 216 | (516) 570-6411 |
| 217 | (369) 898-6162 |
| 218 | (730) 784-6303 |
| 219 | (325) 783-4081 |
| 220 | (212) 262-8829 |
| 221 | (644) 756-0110 |
| 222 | (931) 969-7317 |
| 223 | (940) 981-5113 |
| 224 | (838) 610-5802 |
| 225 | (798) 331-7777 |
| 226 | (325) 783-4081 |
| 227 | (212) 262-8829 |
| 228 | (228) 576-1557 |
| 229 | (773) 696-8009 |
| 230 | (439) 536-8929 |
| 231 | (368) 336-5403 |

There are no transitive relationships from the 2nd NF tables above, therefore, they cannot be normalized to 3rd NF.

RATING table:

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Category | Weekly | Daily |
| 1 | 0 | 480 | 80 |
| 1 | 1 | 600 | 100 |
| 2 | 0 | 530 | 90 |
| 2 | 1 | 660 | 110 |
| 3 | 0 | 600 | 100 |
| 3 | 1 | 710 | 120 |
| 4 | 0 | 685 | 115 |
| 4 | 1 | 800 | 135 |
| 5 | 0 | 780 | 130 |
| 6 | 0 | 685 | 115 |

From the RATING table, no column is composited or multi-valued, and each row is unique; therefore, RATING TABLE is in 1st NF.

The Weekly and Daily columns, which represent the cost of the car rental based on their types, and they are both dependent on Type and Category, therefore, they are fully functional dependencies. The RATING table is already in 2nd NF. In addition, there is are no transitive dependents in RATING table, so it’s already in 3rd NF.

VEHICLE table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VehicleID | Description | Year | Type | Category |
| 19VDE1F3XEE414842 | Acura ILX | 2014 | 1 | 1 |
| 1FDEE3FL6EDA29122 | Ford E 350 | 2014 | 6 | 0 |
| 1FDRF3B61FEA87469 | Ford Super Duty Pickup | 2015 | 5 | 0 |
| 1FTNF1CF2EKE54305 | Ford F Series Pickup | 2014 | 5 | 0 |
| 1G1JD5SB3E4240835 | Chevrolet Optra | 2014 | 1 | 0 |
| 1GB3KZCG1EF117132 | Chevrolet Silverado | 2014 | 5 | 0 |
| 1HGCR2E3XEA305302 | Honda Accord | 2014 | 2 | 0 |
| 1N4AB7AP2EN855026 | Nissan Sentra | 2014 | 1 | 0 |
| 1N6BA0EJ9EN516565 | Nissan Titan | 2014 | 5 | 0 |
| 1N6BF0KM0EN101134 | Nissan NV | 2014 | 6 | 0 |
| 1VWCH7A3XEC037969 | Volkswagen Passat | 2014 | 2 | 1 |
| 2HGFB2F94FH501940 | Honda Civic | 2015 | 1 | 0 |
| 2T3DFREV0FW317743 | Toyota RAV4 | 2015 | 4 | 0 |
| 3MZBM1L74EM109736 | Mazda 3 | 2014 | 1 | 0 |
| 3N1CE2CP0FL409472 | Nissan Versa Note | 2015 | 1 | 0 |
| 3N1CN7APXEK444458 | Nissan Versa | 2014 | 1 | 0 |
| 3VW2A7AU1FM012211 | Volkswagen Golf | 2015 | 1 | 0 |
| 4S4BRCFC1E3203823 | Subaru Outback | 2014 | 4 | 0 |
| 4S4BSBF39F3261064 | Subaru Outback | 2015 | 4 | 0 |
| 4S4BSELC0F3325370 | Subaru Outback | 2015 | 4 | 0 |
| 5J6RM4H90FL028629 | Honda CR-V | 2015 | 4 | 0 |
| 5N1AL0MM8EL549388 | Infiniti JX35 | 2014 | 4 | 1 |
| 5NPDH4AE2FH565275 | Hyundai Elantra | 2015 | 1 | 0 |
| 5TDBKRFH4ES26D590 | Toyota Highlander | 2014 | 4 | 0 |
| 5XYKT4A75FG610224 | Kia Sorento | 2015 | 4 | 0 |
| 5XYKU4A7XFG622415 | Kia Sorento | 2015 | 4 | 0 |
| 5XYKUDA77EG449709 | Kia Sorento | 2014 | 4 | 0 |
| JF1GPAA61F8314971 | Subaru Impreza | 2015 | 1 | 0 |
| JH4KC1F50EC800004 | Acura RLX | 2014 | 3 | 1 |
| JH4KC1F56EC000095 | Acura RLX | 2014 | 3 | 1 |
| JM1BM1V35E1210570 | Mazda 3 | 2014 | 1 | 0 |
| JM3KE4DY4F0441471 | Mazda CX5 | 2015 | 4 | 0 |
| JM3TB3DV0E0015742 | Mazda CX9 | 2014 | 4 | 0 |
| JN8AS5MV0FW760408 | Nissan Rogue Select | 2015 | 4 | 0 |
| JTEZUEJR7E5081641 | Toyota 4Runner | 2014 | 4 | 0 |
| JTHBW1GG1F120DU53 | Lexus ES 300h | 2015 | 2 | 1 |
| JTHCE1BL3F151DE04 | Lexus GS 350 | 2015 | 2 | 1 |
| JTHDL5EF9F5007221 | Lexus LS 460 | 2015 | 3 | 1 |
| JTHFF2C26F135BX45 | Lexus IS 250C | 2015 | 1 | 1 |
| JTJHY7AX2F120EA11 | Lexus LX 570 | 2015 | 4 | 1 |
| JTJJM7FX2E152CD75 | Lexus GX460 | 2014 | 4 | 1 |
| JTMBFREV1FJ019885 | Toyota RAV4 | 2015 | 4 | 0 |
| KM8SN4HF0FU107203 | Hyundai Santa Fe | 2015 | 4 | 0 |
| KMHJT3AF1FU028211 | Hyundai Tucson | 2015 | 4 | 0 |
| KMHTC6AD8EU998631 | Hyundai Veloster | 2014 | 1 | 0 |
| KNAFZ4A86E5195865 | KIA Sportage | 2014 | 4 | 0 |
| KNAFZ4A86E5195895 | KIA Forte | 2014 | 1 | 0 |
| KNAGN4AD2F5084324 | Kia Optima Hybrid | 2015 | 2 | 0 |
| KNALN4D75E5A57351 | Kia Cadenza | 2014 | 3 | 0 |
| KNALU4D42F6025717 | Kia K900 | 2015 | 3 | 0 |
| KNDPCCA65F7791085 | KIA Sportage | 2015 | 4 | 0 |
| WA1LGAFE8ED001506 | Audi Q7 | 2014 | 4 | 1 |
| WAU32AFD8FN005740 | Audi A8 | 2015 | 3 | 1 |
| WAUTFAFH0E0010613 | Audi A5 | 2014 | 1 | 1 |
| WBA3A9G51ENN73366 | BMW 3 Series | 2014 | 1 | 1 |
| WBA3B9C59EP458859 | BMW 3 Series | 2014 | 1 | 1 |
| WBAVL1C57EVR93286 | BMW X1 | 2014 | 4 | 1 |
| WDCGG0EB0EG188709 | Mercedes Benz GLK | 2014 | 1 | 1 |
| YV440MDD6F2617077 | Volvo XC60 | 2015 | 4 | 1 |
| YV4940NB5F1191453 | Volvo XC70 | 2015 | 4 | 1 |

The VEHICLE table is already in 1st NF since all the rows are unique.

Identify functional dependencies:

VehicleID -> Description

VehicleID -> Year

VehicleID, Description -> Type, Category

All the dependencies above are all fully functional dependencies, therefore, the VEHICLE table is also already in 2nd NF. The VEHICLE table is also in 3rd NF since there are no transitive functional dependencies.

RENTAL table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CustID | VehicleID | StartDate | OrderDate | RentalType | Qty | ReturnDate | TotalAmount | PaymentDate |
| 203 | JM3KE4DY4F0441471 | 9/9/2019 | 5/22/2019 | 1 | 4 | 9/13/2019 | 460 | 9/9/2019 |
| 210 | 19VDE1F3XEE414842 | 11/1/2019 | 10/28/2019 | 7 | 2 | 11/15/2019 | 1200 | NULL |
| 210 | JTHFF2C26F135BX45 | 5/1/2019 | 4/15/2019 | 7 | 1 | 5/8/2019 | 600 | 5/8/2019 |
| 210 | JTHFF2C26F135BX45 | 11/1/2019 | 10/28/2019 | 7 | 2 | 11/15/2019 | 1200 | NULL |
| 210 | WAUTFAFH0E0010613 | 11/1/2019 | 10/28/2019 | 7 | 2 | 11/15/2019 | 1200 | NULL |
| 210 | WBA3A9G51ENN73366 | 11/1/2019 | 10/28/2019 | 7 | 2 | 11/15/2019 | 1200 | NULL |
| 210 | WBA3B9C59EP458859 | 11/1/2019 | 10/28/2019 | 7 | 2 | 11/15/2019 | 1200 | NULL |
| 210 | WDCGG0EB0EG188709 | 11/1/2019 | 10/28/2019 | 7 | 2 | 11/15/2019 | 1200 | NULL |
| 212 | 19VDE1F3XEE414842 | 6/10/2019 | 4/15/2019 | 7 | 3 | 7/1/2019 | 1800 | 6/10/2019 |
| 216 | 1N6BF0KM0EN101134 | 8/2/2019 | 3/15/2019 | 7 | 4 | 8/30/2019 | 2740 | 8/2/2019 |
| 216 | 1N6BF0KM0EN101134 | 8/30/2019 | 3/15/2019 | 1 | 2 | 9/1/2019 | 230 | 8/2/2019 |
| 221 | 19VDE1F3XEE414842 | 7/1/2019 | 6/12/2019 | 7 | 1 | 7/8/2019 | 600 | 7/1/2019 |
| 221 | 19VDE1F3XEE414842 | 7/9/2019 | 6/12/2019 | 1 | 2 | 7/11/2019 | 200 | 7/1/2019 |
| 221 | 19VDE1F3XEE414842 | 1/1/2020 | 12/15/2019 | 7 | 4 | 1/29/2020 | 2400 | NULL |
| 221 | JTHFF2C26F135BX45 | 1/1/2020 | 12/15/2019 | 7 | 4 | 1/29/2020 | 2400 | NULL |
| 221 | WAUTFAFH0E0010613 | 7/1/2019 | 6/12/2019 | 7 | 1 | 7/8/2019 | 600 | 7/1/2019 |
| 221 | WAUTFAFH0E0010613 | 7/9/2019 | 6/12/2019 | 1 | 2 | 7/11/2019 | 200 | 7/1/2019 |
| 221 | WAUTFAFH0E0010613 | 1/1/2020 | 12/15/2019 | 7 | 4 | 1/29/2020 | 2400 | NULL |
| 221 | WBA3A9G51ENN73366 | 1/1/2020 | 12/15/2019 | 7 | 4 | 1/29/2020 | 2400 | NULL |
| 221 | WBA3B9C59EP458859 | 1/1/2020 | 12/15/2019 | 7 | 4 | 1/29/2020 | 2400 | NULL |
| 221 | WDCGG0EB0EG188709 | 1/1/2020 | 12/15/2019 | 7 | 4 | 1/29/2020 | 2400 | NULL |
| 229 | 19VDE1F3XEE414842 | 5/6/2019 | 4/12/2019 | 1 | 4 | 5/10/2019 | 400 | 5/6/2019 |
| 229 | WAUTFAFH0E0010613 | 5/6/2019 | 4/12/2019 | 1 | 4 | 5/10/2019 | 400 | 5/6/2019 |

RENTAL table is already in 1st NF because all the rows are unique and there are no multi-valued or composited data in the table

(Primary key: CustID, VehicleID)

Identify the dependencies:

CustID -> OrderDate

VehicleID -> RentalType

CustID -> Qty

StartDate, RentalType, Qty -> ReturnDate

VehicleID, Qty, RentalType -> TotalAmount

CustID -> PaymentDate

The RENTAL table is not in 2nd NF, since the PaymentDate and OrderDate only depend on CustID, therefore, we split the table and create new ones.

|  |  |  |
| --- | --- | --- |
| CustID | PaymentDate | OrderDate |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CustID | VehicleID | StartDate | RentalType | Qty | ReturnDate | TotalAmount |

There exists a transitive dependency between the VehicleID, RentalType, and ReturnDate. Also, the CustID and Qty and ReturnDate is transitive, therefore, we can normalize it to 3rd NF

|  |  |  |  |
| --- | --- | --- | --- |
| CustID | PaymentDate | OrderDate | Qty |

|  |  |  |  |
| --- | --- | --- | --- |
| VehicleID | RentalType | Qty | TotalAmount |

|  |  |  |  |
| --- | --- | --- | --- |
| StartDate | RentalType | Qty | ReturnDate |

# Works Cited

Navathe, E. (2016). *Fundamentals of Database Systems.* Pearson.