**MUSICAL INSTRUMENTS SHOP**

**Ece Yücesoy**

Functional Dependencies & Normal Forms

Functional dependencies are examined table by table, provided with an explanation of why/why not they satisfy 3NF. All of the following tables and their keys have single values which makes them suitable under 1NF, which fulfills a part of 2NF. Keeping this in mind, each table is analyzed for 2NF and 3NF. Some questionable attributes are also explained further on why there are considered as they have been.

Customer:

FD: {cust\_id} → {cust\_name, cust\_surname, cust\_phone, cust\_address, cust\_city, cust\_mail}

The candidate key of the “Customer” table is {cust\_id} and the rest of the attributes are non-prime attributes. In this case, each non-prime attribute is functionally dependent on {cust\_id} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on another but only dependent on the {cust\_id}, which also satisfies 3NF.

*i.e. phone attributes are not considered as candidate keys because when our daily life shopping habits are discussed, I have decided that some family members choose to give the same phone number or even though not very common phone numbers can be changed and may not define different people always. The following tables that contain phone attributes are examined in consideration of this explanation.*

Suppliers:

FD: {sup\_id} → {sup\_name, sup\_phone, sup\_mail, sup\_address}

The candidate key of the “Suppliers” table is {sup\_id} and the rest of the attributes are non-prime attributes. In this case, each non-prime attribute is functionally dependent on {sup\_id} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on another but only dependent on the {sup\_id}, which also satisfies 3NF.

Supplies:

FD:{inst\_id, sup\_id} → {supply\_date}

The candidate key of the “Supplies” table is {inst\_id, sup\_id} and {supply\_date] attribute is a non-prime attribute. In this case, the non-prime attribute is only functionally dependent on {inst\_id, sup\_id} which supports 2NF, there are no partial dependencies. There are no transitive functional dependencies either, {supply\_date} attribute is not transitively dependent on another only dependent on the {inst\_id, sup\_id}, which also satisfies 3NF.

Category:

FD:{category\_id} → {category\_name}

The candidate key of the “Category” table is {category\_id} and the rest of the attributes are non-prime attributes. In this case, each non-prime attribute is functionally dependent on {category\_id} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on another, only dependent on the {category\_id}, which also satisfies 3NF.

Instruments:

FD: {inst\_id} → {category\_id, inst\_name, inst\_price, inst\_stock, inst\_sold}

The candidate key of the “Instruments” table is {inst\_id} and the rest of the attributes are non-prime attributes. In this case, each non-prime attribute is functionally dependent on {inst\_id} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on another only dependent on the category\_id, which also satisfies 3NF.

*i.e. inst\_name attribute is not considered as a primary and candidate key, because I believe in practice an instrument can have different stocks and prices depending on other variables. For instance, the musical instrument shop might decide to define a different price for a batch of instruments for a time or different colors of an instrument can be recorded under the same inst\_name in which case can be addressed by a different inst\_id. So, in long term, there might be tuples with different inst\_id and the same inst\_name, which is why the presence of inst\_id in the Instrument table does not violate 3NF.*

Sales:

FD: {sales\_id} →{inst\_id, emp\_id, cust\_id, sales\_date, sales\_quantity}

The candidate key of the “Sales” table is {sales\_id} and the rest of the attributes are non-prime attributes. In this case, each non-prime attribute is functionally dependent on {sales\_id} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on another only dependent on the category\_id, which also satisfies 3NF.

Employees:

FD: {emp\_id} → {emp\_name, emp\_surname, emp\_marriage, emp\_dob, emp\_gender, emp\_education, emp\_position, emp\_phone, emp\_address, emp\_mail}

The candidate key of the “Employees” table is {emp\_id} and the rest of the attributes are non-prime attributes. In this case, each non-prime attribute is functionally dependent on {emp\_id} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on another but only dependent on the {emp\_id}, which also satisfies 3NF.

*i.e. emp\_mail attribute is not considered as the primary key, because in long term an employee may leave and a person with similar initials or surnames can come. If a musical instrument shop decides to give out emails, they can choose to give the email address of a former employee to a new one.*

Salaries:

FD: {emp\_position} → {salary}

The candidate key of the Salaries table is {emp\_position} and the attribute is a non-prime attribute. In this case, the non-prime attribute is functionally dependent on {emp\_position} which supports 2NF. There are no transitive functional dependencies either, no attributes are transitively dependent on one another only dependent on the emp\_position, which also satisfies 3NF.