

# Improving Database Quality by Applying Consistency Aspects to Naming Fields and Tables

**Raissa Maringka**

Amikom University, Yogyakarta,  
55281, Indonesia  
raissa.1273@students.amikom.ac.id

**Aulia Khoirunnita\***

Amikom University, Yogyakarta,  
55281, Indonesia  
aulia.1270@students.amikom.ac.id  
\* Corresponding author

**Rodney Maringka**

Amikom University, Yogyakarta,  
55281, Indonesia  
rodney.1272@students.amikom.ac.id

**Ema Utami**

Amikom University, Yogyakarta,  
55281, Indonesia  
ema.u@amikom.ac.id

**Kusnawi**

Amikom University, Yogyakarta,  
55281, Indonesia  
khusnawi@amikom.ac.id

*Received: 2021-02-20; Revised: 2021-03-04; Accepted: 2021-03-05; Published: 2021-03-14*

**Abstract-** The database is one of the benchmarks that affect the quality of information systems. An effective information system certainly has a quality database. Aspects that can be measured to determine the quality of the database are aspects of truth, consistency, range, level of detail, completeness, minimalism, ability to integrate and readability. One of the mistakes that are often encountered in databases is related to the consistency aspect. Consistency aspects that are not paid much attention to its application can lead to data conflicts due to ambiguity and data duplication. This study aims to improve the quality of the database by applying consistency to the naming of fields and tables. A naming method to produce consistency in standardization was applied in this study.

**Keywords-** Database, Database Quality, SQL Server, Consistency Aspects

## I. INTRODUCTION

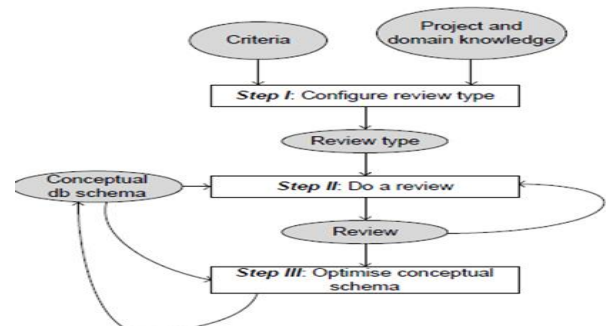
An information system can be considered as an effective system if it has a good database. Quality databases can produce quality information systems (Batini & Navathe 1986). A quality database is measured from several aspects, namely aspects of truth, aspects of consistency, aspects of reach, aspects of the level of detail, aspects of completeness, aspects of minimalism, aspects of ability to integrate and aspects of readability (Herden, 2001). One of the errors that is often encountered in databases is data conflicts due to ambiguity and data duplication or what is commonly called redundancy. Data conflicts arise due to the lack of attention to consistency aspects. Good database design is a design that can reduce data repetition (Marelli, 2018). The two main components in a database are tables and fields. The table contains several fields that hold related records. Field is a subject that has characteristics based on the table that contains it. Bad field names can show the characteristics of the table that contains them.

Therefore, good field naming also has an influence in designing a quality database (Pham, 2019). However, the database in an information system or application has become increasingly complex with the increasing number of tables, types of relations and data. In this case, it is necessary for information system or application developers who have a level of expertise in designing quality database schemes, because the risk of data redundancy, data ambiguity and errors in table relations can often occur. Inconsistent and long names of fields and tables can be one of the problems that can occur in complex databases (Taipalus, 2020). Too many tables and complicated relationships can lead to naming the same field in two different tables. To reduce the problems that can occur in naming fields and tables,

## II. RESEARCH METHODS

### A. Flow of Review

By conducting the Review stage, the database design concept can be tested for its accuracy (Herden, 2001). At this stage, the review flow in the research is carried out by determining the criteria to be tested through the results of data collection and literature review, making an implementation of the proposed method, optimizing the conceptual database scheme of the method that has been applied. The review flow can be seen in Picture 1.



Picture 1. Review flow

### B. Naming Method

In the implementation of this research, the authors designed a method that contains stages for naming tables and fields as shown in Picture 2. The first stage is identification. At this stage it is necessary to identify the number of databases, the function of each database, the tables to be named and the function of each table in the database schema.

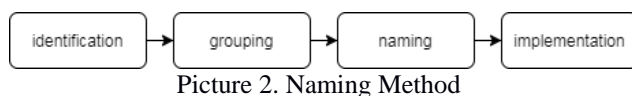
The second stage is grouping. Database schemes that have a number of databases more than one generally already have a good grouping based on the database. However, if the database is only one, it is necessary to group it based on table functions.

The third stage is naming or giving a name or label. The first naming is in the table, then the last naming is in the field. This naming process is based on table functions and groupings and still follows the applicable field naming standards. In this study, the naming method uses a combination of prefixes from each existing table or field function to form a specific code that will be easier to recognize and remember.

The next stage is documentation of the names of tables and fields that have been created to make it easier for application or system developers to search for field and function names if there are names that have not been remembered or for newcomers who will develop their applications.

The last stage is the implementation stage, namely by applying the names that have been made in tables and fields.

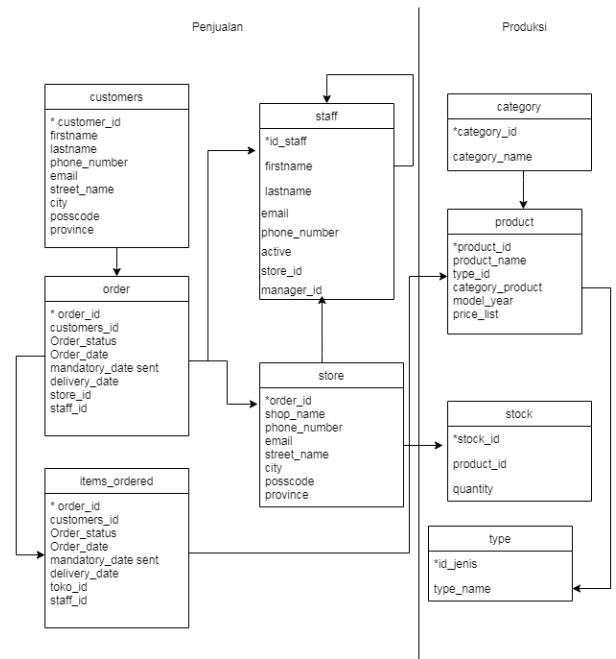
Naming fields and tables still follows the basic rules of database naming, namely not using spaces, if the name consists of two syllables, you can use the underscore to connect separate words or combine the two syllables, do not use words which are syntax of the database itself such as SELECT, CREATE, INSERT and so on, naming fields or tables must match the characteristics or functions of the fields and tables themselves and the latter can use prefixes and suffixes to simplify too long names (VM Le, 2019). The naming method can be seen in Picture 2.



Picture 2. Naming Method

### III. RESULT AND DISCUSSION

Picture 3 is a database diagram of the sales system. From the picture, it can be seen that the naming of fields is named according to the function and description of the field. Currently, the author will test the system to get a report on the remaining stock of goods by displaying the order date, order id, product name, type name and quantity. The sales database diagram can be seen in Picture 3.



Picture 3. Goods Sales Database Diagram

It can be seen that there is a repetition of the field names in which the goods\_porders and orders table have the same field name, namely dip Orders. Then in the goods\_produk table, products and stocks also have the same field name, namely id\_produk and finally the type table and product table have the same field, namely id\_jenis. Actually, these same fields are needed because they are what make the related tables. The same field is a link between tables and is distinguished by using an alias or AS. For the tables that are still not too many and the database is not too big, this kind of thing is not a problem. But for databases with high complexity and lots of tables, this will be a problem that adds to the burden on application or system developers.

One solution to this problem is to provide unique names that do not cause ambiguity to fields and tables. The stages of naming fields and tables designed by the author can be applied in this case. The test results before the application of the naming method can be seen in Picture 4.

```

SELECT tanggal_pesanan, A.id_pesanan, nama_jenis, nama_produk, D.kuantitas
FROM barang_pesanan AS A
JOIN pesanan AS B ON 1=1
AND A.id_pesanan = B.id_pesanan
JOIN produk AS C ON 1=1
AND A.id_produk = C.id_produk
JOIN stok AS D ON 1=1
AND C.id_produk = D.id_produk
JOIN jenis as E ON 1=1
AND C.id_jenis = E.id_jenis
    
```

tanggal_pesanan	id_pesanan	nama_jenis	nama_produk	kuantitas
2016-01-01	1	Electra	Electra Townie Original 7D EQ - Women's - 2016	26
2016-01-01	1	Electra	Electra Townie Original 7D EQ - Women's - 2016	20
2016-01-01	1	Electra	Electra Townie Original 7D EQ - Women's - 2016	19
2016-01-01	1	Trek	Trek Remedy 29 Carbon Frameset - 2016	0
2016-01-01	1	Trek	Trek Remedy 29 Carbon Frameset - 2016	1

Picture 4. Test Results before Application of the Naming Method

The test results using Query on Sql Server are shown as shown in Picture 5. The queries used are as follows.

```
SELECT Order_Date, Order_id_id,
Type_Name, Product_Name, D.
Quantity
FROM items ordered US A
JOIN order US B ON 1 = 1
AND A.id_porders = B.id_porders
JOIN product US C ON 1 = 1
AND A.id_produk = C.id_produk
JOIN stock US D ON 1 = 1
AND C.id_produk = D.id_produk
JOIN type US E ON 1 = 1
AND C.id_jenis = E.id_jenis
```

Picture 5. Query Testing

It can be seen that there is a repetition of the field names in which the goods\_porders and orders table have the same field name, namely idap Orders. Then in the goods\_produk table, products and stocks also have the same field name, namely id\_produk and finally the type table and product table have the same field, namely id\_jenis. Actually, these same fields are needed because they are what make the related tables. The same field is a link between tables and is distinguished by using an alias or AS. For the tables that are still not too many and the database is not too big, this kind of thing is not a problem. But for databases with high complexity and lots of tables, this will be a problem that adds to the burden on application or system developers. One solution to this problem is to provide unique names that do not cause ambiguity to fields and tables. The stages of naming fields and tables designed by the author can be applied in this case.

**A. Identification Stage**

The stages of identification can be seen in Picture 3. The scheme of the sales data base has only one database with 9 tables, namely tables of customers, orders, goods\_porders, staff, stores, categories, products, stocks and types. The naming of table names in the first database schema is adjusted according to the function of each table. For example, the table containing the order data is named the order.

**B. Grouping Stage**

After passing the identification stage, the next step is to identify the table based on its function. The grouping of tables based on their functions can be seen in table 1. The explanation of the table is that there are two main functions in the database schema, namely the sales function and the production function, then the tables that have the same characteristics as the sales or production function can be made into one group same. The orders, goods Orders, customers, staff and shop tables are in the same group of table functions, namely the sales function. Tables of categories, products, stocks and types are in the same table function group, namely the Production table.

Table 1. Identification of Tables by Function.

Sales	Production
order	category
items ordered	product
customer	stock
staff	type
store	

**C. The Naming Stage**

The next stage is to give unique names to fields and tables. The unique names of fields and tables are derived from the universal language of a term. In this study, the Universal language used is English which can be seen in Table 2 in the Name column.

Table 2. Naming Tables by Function

Function	Prefix	Name	Data Dictionary	Table	Prefix Table
Sales	S	Customer	CUS	SCUS	SC
Sales	S	Staffs	STF	SSTF	SF
Sales	S	Store	STR	SSTR	SS
Sales	S	Order	ORD	SORD	SO
Sales	S	Order Item	OIT	SOIT	SI
Production	P	Category	CAT	PCAT	PC
Production	P	Product	PRO	PPRO	PP

. From combining the table function prefix and the Data Dictionary, a new unique table name is created with characteristics according to its function.

**D. Field Naming**

The process of naming fields is almost the same as naming a table. Each Table Name is made a Data Dictionary which is used as a standardization of naming systems. For example, for every name that ends with the word ID, the standard data dictionary is NO. So if the name is Customer ID, then the Data Dictionary is CUNO,

if the name is Category ID, then the Data Dictionary is CANO. After the Data Dictionary in each field has been obtained, the next step is to combine the Data Dictionary with the table prefix that has been made in table 2. Start with the table prefix, then continue with the data dictionary. The naming process is accompanied by recording or bookkeeping on the data dictionary and prefixes of newly created fields and tables as a guide to standardizing the naming of fields and tables in the system. The existence of complete bookkeeping or data collection also supports aspects of database quality,

namely aspects of the ability to integrate and readability. The field table can be seen in Table 3.

Table 3. Field Naming

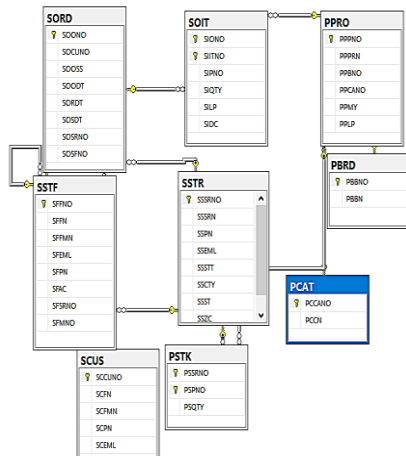
Table	Description	Name	Dictionary Data	Prefix Table	Field Name
SCUS	id pelanggan	customer ID	CUNO	SC	SCCUNO
SCUS	nama depan	first name	FN	SC	SCFN
SCUS	nama belakang	family name	FMN	SC	SCFMN
SCUS	nomor telfon	phone number	PN	SC	SCPN
SCUS	email	email	EML	SC	SC EML
SCUS	nama jalan	street	STT	SC	SCSTT
SCUS	kota	city	CTY	SC	SCCTY
SCUS	kode pos	ZIP code	ZC	SC	SCZC
SCUS	provinsi	state	ST	SC	SCST
SORD	id pesanan	Order ID	ONO	SO	SOONO
SORD	id pelanggan	Customer ID	CUNO	SO	SOCUNO
SORD	status pesanan	Order Status	OSS	SO	SOOSS
SORD	tanggal pesanan	Order Date	ODT	SO	SOODT
SORD	tgl wajib kirim	Required Date	RDT	SO	SORDT
SORD	tgl pengiriman	Shipped Date	SDT	SO	SOSDT
SORD	id toko	Store ID	SRNO	SO	SOSRNO
SORD	id staff	staff ID	SFNO	SO	SOSFNO
SOIT	id pesanana	order id	ONO	SI	SIONO
SOIT	id barang	item id	ITNO	SI	SIITNO
SOIT	id produk	product id	PNO	SI	SIPNO
SOIT	kuantitas	quantity	QTY	SI	SIQTY
SOIT	daftar harga	list price	LP	SI	SILP
SOIT	diskon	discount	DC	SI	SIDC
SSTF	id staff	staff ID	SFNO	SF	SEFNO
SSTF	nama depan	first name	FN	SF	SFFN
SSTF	nama belakang	family name	FNM	SF	SFFNM
SSTF	email		EML	SF	SFEML
SSTF	nomor telfon	phone number	PN	SF	SFPN
SSTF	aktif	active	AC	SF	SFAC
SSTF	id toko	store id	SRNO	SF	SFSRNO
SSTF	id manager	manager id	MNO	SF	SFMNO
SSTR	id toko	Store ID	SRNO	SS	SSSRNO
SSTR	nama toko	store name	SRN	SS	SSSRN
SSTR	nomor telfon	phone number	PN	SS	SSPN
SSTR	email		EML	SS	SSEML
SSTR	nama jalan	street	STT	SS	SSSTT
SSTR	kota	city	CTY	SS	SSCTY
SSTR	kode pos	ZIP code	ZC	SS	SSZC
SSTR	provinsi	state	ST	SS	SSST
PCAT	id kategori	category id	CANO	PC	PCCANO
PCAT	nama kategori	category name	CN	PC	PCCN
PPRO	id product	product id	PNO	PP	PPPNO
PPRO	nma produk	product name	PRN	PP	PPPRN
PPRO	id jenis	brand id	BNO	PP	PPBNO
PPRO	id kategori	category id	CANO	PP	PPCANO
PPRO	tahun model	model year	MY	PP	PPMY
PPRO	daftar harga	list price	LP	PP	PPLP
PSTK	id stok	stock id	STNO	PS	PSSTNO
PSTK	id produk	product id	PNO	PS	PSPNO
PSTK	kuantitas	quantity	QTY	PS	PSQTY
PBRD	id jenis	brand id	BNO	PB	PBBNO
PBRD	nama jenis	brand name	BN	PB	PBBN

### E. Implementation Stage

The last stage is implementation by creating tables and fields according to the names that have been designed in the documentation.

The results of the implementation of the new naming process can be seen in Picture 5. The longest name is only limited to six characters and the shortest name is only four characters. The names of the new tables with sales or sales functions are SORD (Order), SOIT (Order Item), SSTF (Staff), SSTR (Store), SCUS (Customer). The names of the new tables with production or production functions are PPRO (Product), PBRD (Brand), PCAT (Category)

After the implementation, the test is again carried out to get the same results as the initial test, but with a different syntax or query due to the new table and field names. The Database Diagram after the Naming Process can be seen in Picture 6 and the Test Results after the Naming Process can be seen in Picture 7.



Picture 6. Database Diagram after the Naming Process

```

SELECT SOODT AS "ORDER DATE", SIONO AS "ORDER ID",
PBBN AS "BRAND NAME", PPPRN AS "PRODUCT NAME",
PSQTY AS "QUANTITY"
FROM SOIT
JOIN SORD ON 1=1
AND SOONO = SIONO
JOIN PPRO ON 1=1
AND PPPNO = SIPNO
JOIN PSTK ON 1=1
AND PSPNO = SIPNO
JOIN PBRD ON 1=1
AND PBBNO = PPBNO
    
```

ORDER DATE	ORDER ID	BRAND NAME	PRODUCT NAME	QUANTITY
2016-01-01	1	Electra	Electra Townie Original 7D EQ - Women's - 2016	26
2016-01-01	1	Electra	Electra Townie Original 7D EQ - Women's - 2016	20
2016-01-01	1	Electra	Electra Townie Original 7D EQ - Women's - 2016	19
2016-01-01	1	Trek	Trek Remedy 29 Carbon Frameset - 2016	0

Picture 7. Testing Results after the Naming Process

You can see the difference before and after the naming process is applied where there is no ambiguity between the field names and the simpler syntax. Connecting fields between tables are still easily identified by looking at the similarities of the data dictionary. For example, the SOONO field and the SIONO field. Both of these fields have the same data dictionary, namely 'ONO' so it can be concluded that these fields. Image Testing Query 2 can be seen in Picture 8.

```

SELECT SOODT AS "Order Date",
SIONO AS "ORDER ID", PBBN US
"BRAND NAME", PPPRN AS "PRODUCT
NAME", PSQTY AS "QUANTITY"
FROM SOIT
JOIN SORD ON 1 = 1
AND SOONO = SIONO
JOIN PPRO ON 1 = 1
AND PPPNO = SIPNO
JOIN PSTK ON 1 = 1
AND PSPNO = SIPNO
JOIN PBRD ON 1 = 1
AND PBBNO = PPBNO
    
```

Picture 8. Testing Query 2

Based on the research conducted, it can be concluded that the method of naming fields and tables that have

been designed is very suitable for databases that are still in the design stage and have not been implemented. If it has already been implemented, the process of changing table and field names will change all database schemas from the start, except for simple database schema types that don't have too many tables and fields, this is still possible. This method also makes it easier for developers to develop databases with a high level of complexity where the database has many tables, databases and many relationships.

Through this research, it is hoped that it can be used as a reference in database design. It is also hoped that further research will pay special attention to the effect of naming fields and tables in the database using other methods and approaches. It is recommended to use algorithms and tools that can generate field and table names automatically so that it makes it easier for developers to design databases that have a high level of complexity.

REFERENCES

Batini, C., Lenzerini, M. & Navathe, SB, 1986. A comparative analysis of methodologies for database schema integration. ACM computing surveys (CSUR), 18 (4), pp.323-364.

Herden, O., 2001. Measuring quality of database schemas by reviewing-concept, criteria and tool. Oldenburg Research and Development Institute for Computer Science Tools and Systems, Escherweg, 2, p. 26121.

Le, VM, Soares, GA and Gulwani, S., Microsoft Technology Licensing LLC, 2019. SQL Query Formatting By Examples. US Patent Application 16 / 007,396.

Marelli, M. & Amenta, S., 2018. A database of orthography-semantics consistency (OSC) estimates for 15,017 English words. Behavior research methods, 50 (4), pp. 1482-1495.

Pham, N., van Heck, RG, van Dam, JC, Schaap, PJ, Saccenti, E. and Suarez-Diez, M., 2019. Consistency, inconsistency, and ambiguity of metabolite names in biochemical databases used for genome-scale metabolic modeling. Metabolites, 9 (2), p. 28.

Taipalus, T., 2020, August. The Effects of Database Complexity on SQL Query Formulation (journal-first). In 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 185-185). IEEE.