

CSCI235 – Database Systems

Database Design Quality

sjapit@uow.edu.au

28 March 2023



Acknowledgements

The following presentation were adapted from the lecture slides of:

CSCI235 – Database Systems, 01Database
Design Quality

By Dr Janusz R. Getta, University of Wollongong,
Australia

Spring, 2018

Outline

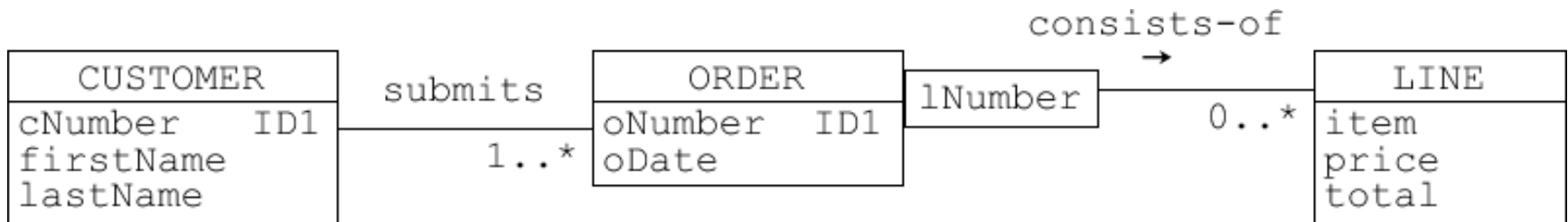
- Why not ONE BIG TABLE?
- Where is the problem?
- Functional dependency

Why not ONE BIG TABLE?

Let us consider the following database domain:

- A **customer** is described by a unique **customer number, first, and last name**
- **Customers** submit **orders**. An **order** is described by a **unique order number and order date**
- **Orders** consist of **lines**. A **line** contains information about a **name of ordered item, price per single item, and total number of ordered items**

Conceptual schema for the specified domain database:



Why not ONE BIG TABLE?

Logical design provides the following relational schemas:

CUSTOMER(cNumber, firstName, lastName)

Primary key: cNumber

ORDERS(oNumber, oDate, cNumber)

Primary key: oNumber

Foreign key: cNumber REFERENCES CUSTOMER(cNumber)

LINE(oNumber, lNumber, item, price, total)

Primary key: (oNumber, lNumber)

Foreign key: oNumber REFERENCES ORDERS(oNumber)

Why not ONE BIG TABLE?

Why not one relational schema?

CUSTOMER(cNumber, firstName, lastName, oNumber, oDate,
cNumber, oNumber, lNumber, item, price, totalQty)
Primary key: (cNumber, oNumber, lNumber)

<u>cNum</u>	fName	lName	<u>oNum</u>	oDate	<u>lNum</u>	Item	Price	total
7	James	Bond	7	2020-09-23	1	Bolt	23.04	5
7	James	Bond	7	2020-09-23	2	Screw	29.01	3
7	James	Bond	7	2020-09-23	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	1	Bolt	23.04	1
7	James	Bond	8	2020-09-28	2	Screw	29.01	1
7	James	Bond	8	2020-09-28	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	4	Lock	25.50	1

Why not ONE BIG TABLE?

Insertion of information about one customer who submitted 2 orders such that each order consists of several lines reveals a problem!

cNum	fName	lName	oNum	oDate	lNum	Item	Price	total
7	James	Bond	7	2020-0923	1	Bolt	23.04	5
7	James	Bond	7	2020-09-23	2	Screw	29.01	3
7	James	Bond	7	2020-09-23	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	1	Bolt	23.04	1
7	James	Bond	8	2020-09-28	2	Screw	29.01	1
7	James	Bond	8	2020-09-28	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	4	Lock	25.50	1

Why not ONE BIG TABLE?

Insertion of information about one customer who submitted 2 orders such that each order consists of several lines reveals a problem!

cNum	fName	lName	oNum	oDate	oQty	iName	iPrice	total
7	James	Bond	1	2020-09-28	2	Screw	25.01	5
7	James	Bond	1	2020-09-28	3	Nut	4.55	3
7	James	Bond	1	2020-09-28	4	Lock	25.50	2
7	James	Bond	1	2020-09-28	1			1
7	James	Bond	1	2020-09-28	1			1
7	James	Bond	8	2020-09-28	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	4	Lock	25.50	1

The number, first name, and last name of a customer is repeated as many times as the total number of different items purchased in all orders and ...

Why not ONE BIG TABLE?

Insertion of information about one customer who

...and order number is repeated together with order date as many times as the total number of different items purchased in an order.

cNum	fName	lName	oNum	oDate	lNum	Item	Price	total
7	James	Bond	7	2020-0923	1	Bolt	23.04	5
7	James	Bond	7	2020-09-23	2	Screw	29.01	3
7	James	Bond	7	2020-09-23	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	1	Bolt	23.04	1
7	James	Bond	8	2020-09-28	2	Screw	29.01	1
7	James	Bond	8	2020-09-28	3	Nut	4.55	2
7	James	Bond	8	2020-09-28	4	Lock	25.50	1

Why not ONE BIG TABLE?

A multi-table design does not have such a problem:

CUSTOMER(cNumber, firstName, lastName)

Primary key: cNumber

<u>cNumber</u>	firstName	lastName
7	James	Bond

Why not ONE BIG TABLE?

ORDERS(oNumber, oDate, cNumber)

Primary key: oNumber

Foreign key: cNumber REFERENCES CUSTOMER(cNumber)

<u>oNumber</u>	oDate	cNumber
7	2019-01-03	7
8	2019-04-05	7

Why not ONE BIG TABLE?

LINE(oNumber, INumber, item, price, total)

Primary key: (oNumber, INumber)

Foreign key: oNumber REFERENCES ORDERS(oNumber)

<u>oNumber</u>	<u>INumber</u>	item	price	total
7	1	bolt	23.04	5
7	2	Screw	29.01	3
7	3	Nut	4.55	2
8	1	Bolt	23.04	1
8	2	Screw	28.00	1
8	3	Nut	5.20	2
8	4	Lock	30.10	1

Where is the problem?

Why do we get redundancies in an incorrectly designed relational table?

TABLE_NAME

COLUMN_1	COLUMN_2	...	COLUMN_N
Green	Red	...	Blue
Green	Red	...	Orange
Green	Red	...	Red
Blue	Yellow	...	Yellow
Blue	Yellow	...	Purple
Orange	Red	...	Yellow
Orange	Red	...	Green

Data dependencies:

- If **COLUMN_1** is green then **COLUMN_2** is red
- If **COLUMN_1** is blue then **COLUMN_2** is yellow
- If **COLUMN_1** is orange then **COLUMN_2** is red

For any colour x , if **COLUMN_1** is x then **COLUMN_2** is y .

Why not ONE BIG TABLE?

Data dependencies can be represented as a separate relational table...

TABLE_1

COLUMN_1	COLUMN_2
Green	Red
Blue	Yellow
Orange	Red

... and COLUMN_2 can be removed from the original table

TABLE_2

COLUMN_1	...	COLUMN_N
Green	...	Blue
Green	...	Orange
Green	...	Red
Blue	...	Yellow
Blue	...	Magenta
Orange	...	Yellow
Orange	...	Green

Why not ONE BIG TABLE?

Do data dependencies exist in BIG TABLE?

cNumber	fName	IName	oNumber	oDate	INumber	Item	Price	Total
7	James	Bond	7	2019-01-03	1	Bolt	23.04	5
7	James	Bond	7	2019-01-03	2	Screw	29.01	3
7	James	Bond	7	2019-01-03	3	Nut	4.55	2
7	James	Bond	8	2019-04-05	1	Bolt	23.04	1
7	James	Bond	8	2019-04-05	2	Screw	28.00	1
7	James	Bond	8	2019-04-05	3	Nut	5.20	2
7	James	Bond	8	2019-04-05	4	Lock	30.10	1

Data dependencies:

- If **cNumber** = 7 then **fName** = James
- If **cNumber** = 7 then **IName** = Bond

For any customer number x if
cNumber = x then **fName** = y
and **IName** = z

Why not ONE BIG TABLE?

Do data dependencies exist in BIG TABLE?

cNumber	fName	lName	oNumber	oDate	lNumber	Item	Price	Total
7	James	Bond	7	2019-01-03	1	Bolt	23.04	5
7	James	Bond	7	2019-01-03	2	Screw	29.01	3
7	James	Bond	7	2019-01-03	3	Nut	4.55	2
7	James	Bond	8	2019-04-05	1	Bolt	23.04	1
7	James	Bond	8	2019-04-05	2	Screw	28.00	1
7	James	Bond	8	2019-04-05	3	Nut	5.20	2
7	James	Bond	8	2019-04-05	4	Lock	30.10	1

Data dependencies:

- If **oNumber** = 7 then **oDate** = 2019-01-03
- If **oNumber** = 8 then **oDate** = 2019-04-09

For any order number x if
oNumber = x then **oDate** = y

Functional dependency

What does it mean: **if a value in column A is x then a value in column B is always y** ?

It means that **every value x in a column A is associated with only one value y in a column B.**

For example, every customer number in a column **cNumber** is associated with only one first name in a column **fName**, i.e. a customer has only one first name

For example, every customer number in a column **cNumber** is associated with only one last name in a column **lName** i.e. a customer has only one last name

Functional dependency

- For example, every **order number** in a column **oNumber** is associated with only one **order date** in a column **oDate** i.e. an order has only one date
- Such data dependency does not hold for **item name** and **order number** because an **item name** in a column **item** can be associated with many **order numbers** in a column **oNumber** and the opposite ...
- ... an **order number** in a column **oNumber** can be associated with only one **item name**.

Functional dependency

- If every value in a column **A** is associated with only one value in a column **B** then it means that the columns **A** and **B** represent a function *f* that maps the values in a column **A** into the values in a column **B**

$$f: \text{domain}(A) \rightarrow \text{domain}(B)$$

- If every value in a column **cNumber** is associated with only one value in a column **fName** then the columns **cNumber** and **fName** represent a function

$$f: \text{domain}(cNumber) \rightarrow \text{domain}(fname)$$

Functional dependency

If every value in a column **cNumber** is associated with only one value in a column **lName** then the column **cNumber** and **lName** represent a function

$$f: \text{domain}(cNumber) \rightarrow \text{domain}(lName)$$

If every value in a column **oNumber** is associated with only one value in a column **oDate** then the columns **oNumber** and **oDate** represent a function

$$f: \text{domain}(oNumber) \rightarrow \text{domain}(oDate)$$

Functional dependency

If the columns **A** and **B** in a relational table **R** represent a function

$$f: \text{domain}(A) \rightarrow \text{domain}(B)$$

then in the future it will be denoted by $A \rightarrow B$, and we shall say that a **functional dependency** $A \rightarrow B$ is valid in a **relational table** **R** or that **A functionally determines B**

Functional dependency

Therefore, the following functional dependencies are valid in a big table **CUSTOMER**:

cNumber \rightarrow *fName*

cNumber \rightarrow *lName*

oNumber \rightarrow *oDate*

... and the others

Functional dependency

- **Functional dependency** is a special kind of so called **data dependency** which is a reflection of the real world **consistency constraint**.
- **Functional dependencies** can be used to describe the **semantics** (meaning) of data.
- **Functional dependencies** can be used to determine whether a relational schema (header of relational table) is constructed in a correct way.
- **Functional dependencies** can be used to design a database.

References

- T. Connolly, C. Begg, Database Systems, A Practical Approach to Design, Implementation, and Management, Chapter 14.1 The Purpose of Normalization, Chapter 14.2 How Normalization Supports Database Design, 14.3 Data Redundancies and Update Anomalies, Pearson Education Ltd, 2015