CSCI235 Database Systems

Database Design Quality

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

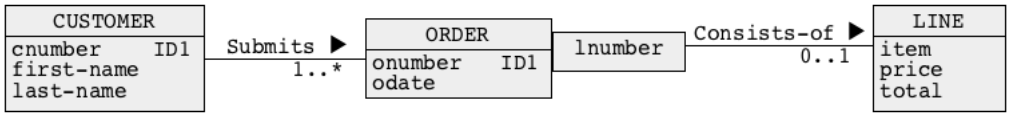
Database Design Quality

Outline

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



Logical design provides the following relational schemas:

Relational schemas

```
CUSTOMER(cnumber, first-name, last-name)
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 ?

Big relational schema

```
CUSTOMER(cnumber, first-name, last-name, onumber, odate, cnumber, onumber, lnumber, item, price total)
PRIMARY KEY = (cnumber, onumber, lnumber)
```

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

Big relational table

cnumber	fname	lname	onumber	odate	lnumber	item	price	total
7	James	Bond	7	2017-01-01	1	bolt	23.04	5
7	James	Bond	7	2017-01-01	2	screw	29.01	3
7	James	Bond	7	2017-01-01	3	nut	4.55	2
7	James	Bond	8	2018-01-01	1	bolt	23.04	1
7	James	Bond	8	2018-01-01	2	screw	23.04	1
7	James	Bond	8	2018-01-01	3	nut	23.04	2
7	James	Bond	8	2018-01-01	4	lock	23.04	1

Why not ONE BIG TABLE !?

Big relational table

cnumber	fname	lname	onumber	odate	lnumber	item	price	total
7	James	Bond	7	2017-01-01	1	bolt	23.04	5
7	James	Bond	7	2017-01-01	2	screw	29.01	3
7	James	Bond	7	2017-01-01	3	nut	4.55	2
7	James	Bond	8	2018-01-01	1	bolt	23.04	1
7	James	Bond	8	2018-01-01	2	screw	23.04	1
7	James	Bond	8	2018-01-01	3	nut	23.04	2
7	James	Bond	8	2018-01-01	4	lock	23.04	1

A 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 ...

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

Why not ONE BIG TABLE !?

A multitable design does not have such a problem:

CUSTOMER schema

```
CUSTOMER(cnumber, first-name, last-name)
PRIMARY KEY = (cnumber)
```

CUSTOMER table

```
+---------+-------+-------+
| cnumber | fname | lname |
+---------+-------+-------+
|       7 | James | Bond  |
+---------+-------+-------+
```

ORDERS schema

```
ORDERS(onumber, odate, cnumber) PRIMARY KEY = (onumber)
FOREIGN KEY = (cnaumber) REFERENCES CUSTOMER(cnumber)
```

ORDERS table

```
+---------+------------+---------+
| onumber | odate      | cnumber |
+---------+------------+---------+
|       7 | 2017-01-01 |       7 |
|       8 | 2018-01-01 |       7 |
+---------+------------+---------+
```

Why not ONE BIG TABLE !?

A multitable design does not have such a problem:

LINE schema

```
LINE(onumber, lnumber, item, price total)
PRIMARY KEY = (onumber, lnumber)
FOREIGN KEY = (onumber) REFERENCES ORDERS(onumber)
```

LINE table

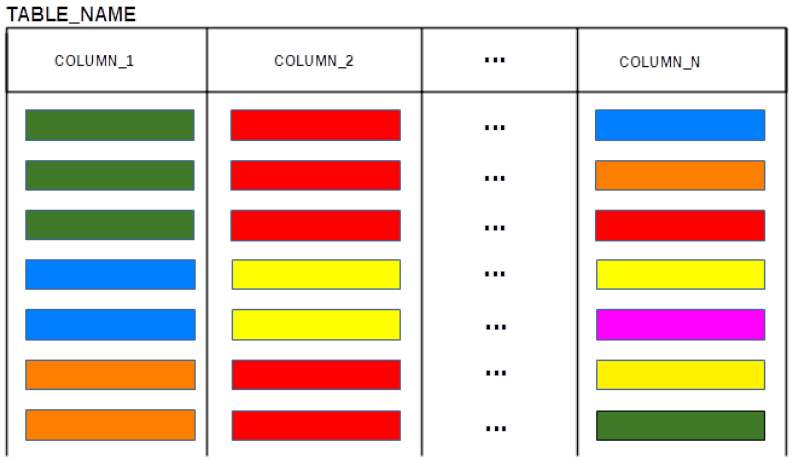
onumber	lnumber	item	price	total
7	1	bolt	23.04	5
7	2	screw	29.01	3
7	3	nut	4.55	2
8	1	bolt	23.04	1
8	2	screw	23.04	1
8	3	nut	23.04	2
8	4	lock	23.04	1

Database Design Quality

Outline

Where is a problem ?

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



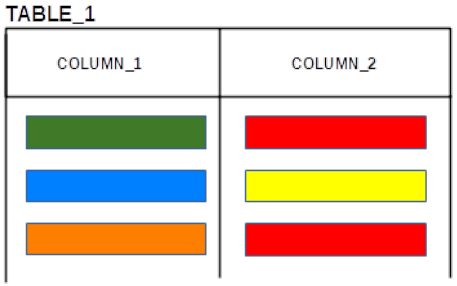
Data dependencies:

- 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

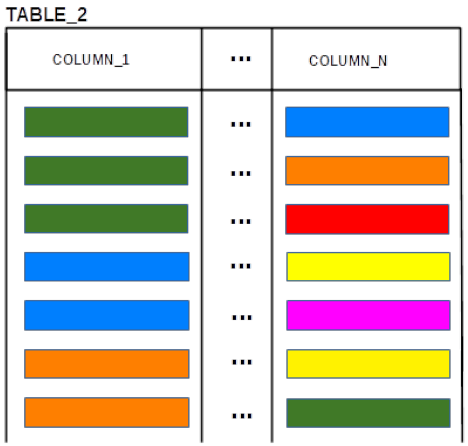
For any colour x if `COLUMN_1` is x then `COLUMN_2 is` y

Where is a problem ?

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



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



Where is a problem ?

Do data dependencies exist in BIG TABLE ?

Big relational table

cnumber	fname	lname	onumber	odate	lnumber	item	price	total
7	James	Bond	7	2017-01-01	1	bolt	23.04	5
7	James	Bond	7	2017-01-01	2	screw	29.01	3
7	James	Bond	7	2017-01-01	3	nut	4.55	2
7	James	Bond	8	2018-01-01	1	bolt	23.04	1
7	James	Bond	8	2018-01-01	2	screw	23.04	1
7	James	Bond	8	2018-01-01	3	nut	23.04	2
7	James	Bond	8	2018-01-01	4	lock	23.04	1

Data dependencies:

- If `cnumber` = 7 then `fname` = James
- If `cnumber` = 7 then `lname` = Bond

For any customer number x if `cnumber` = x then `fname` = y and `lname` = z

Where is a problem ?

Do data dependencies exist in BIG TABLE ?

Big relational table

cnumber	fname	lname	onumber	odate	lnumber	item	price	total
7	James	Bond	7	2017-01-01	1	bolt	23.04	5
7	James	Bond	7	2017-01-01	2	screw	29.01	3
7	James	Bond	7	2017-01-01	3	nut	4.55	2
7	James	Bond	8	2018-01-01	1	bolt	23.04	1
7	James	Bond	8	2018-01-01	2	screw	23.04	1
7	James	Bond	8	2018-01-01	3	nut	23.04	2
7	James	Bond	8	2018-01-01	4	lock	23.04	1

Data dependencies:

- If `onumber` = 7 then `odate` = 2017-01-01
- If `onumber` = 8 then `odate` = 2018-01-01

For any order number x if `onumber` = x then `odate` = y

Database Design Quality

Outline

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

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

Functional dependency

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 many item names in a column `item`

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

$$f : domain(A) \rightarrow domain(B)$$

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

$$f : domain(cnumber) \rightarrow domain(fname)$$

If every value in a column `cnumber` is associated with only one value in a column `lname` then the columns `cnumber` and `lname` represent a function

$$f : domain(cnumber) \rightarrow domain(lname)$$

Functional dependency

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

$$f : domain(onumber) \rightarrow domain(odate)$$

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

$$f : domain(A) \rightarrow domain(B)$$

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`:

$$\text{cnumber} \rightarrow \text{fname}$$

$$\text{cnumber} \rightarrow \text{lname}$$

$$\text{onumber} \rightarrow \text{odate}$$

$$\text{onumber} \rightarrow \text{cnumber}$$

$$\text{onumber} \rightarrow \text{fname, lname}$$

$$\text{onumber, lnumber} \rightarrow \text{item}$$

$$\text{onumber, lnumber} \rightarrow \text{price, total}$$

... 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, however such approach is used very rarely

References

T. Connoly, 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