SQLite 23 - ER diagram

Database and programming

LILLEBAELT ACADEMY OF
PROFESSIONAL HIGHER EDUCATION

Author
Martin Grønholdt
mart80c7@edu.eal.dk

Tuesday 13 June 2017

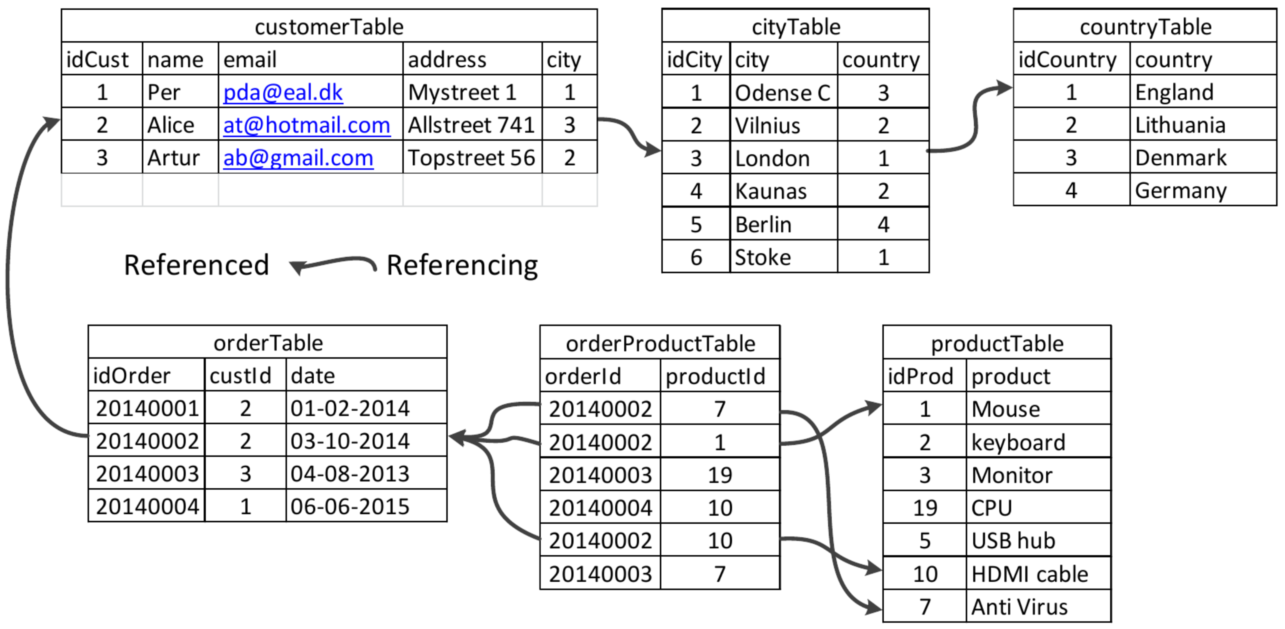
Table of Contents

1. Introduction

This hand in is about using ER (Entity Releation) diagrams for visualising database structures.

All hand ins for this course are available on GitHub at:
https://github.com/deadbok/eal_programming

2. Database structure and corresponding ER diagram.



customerTable				
idCust	name	email	address	city
1	Per	pda@eal.dk	Mystreet 1	1
2	Alice	at@hotmail.com	Allstreet 741	3
3	Artur	ab@gmail.com	Topstreet 56	2

cityTable		
idCity	city	country
1	Odense C	3
2	Vilnius	2
3	London	1
4	Kaunas	2
5	Berlin	4
6	Stoke	1

countryTable	
idCountry	country
1	England
2	Lithuania
3	Denmark
4	Germany

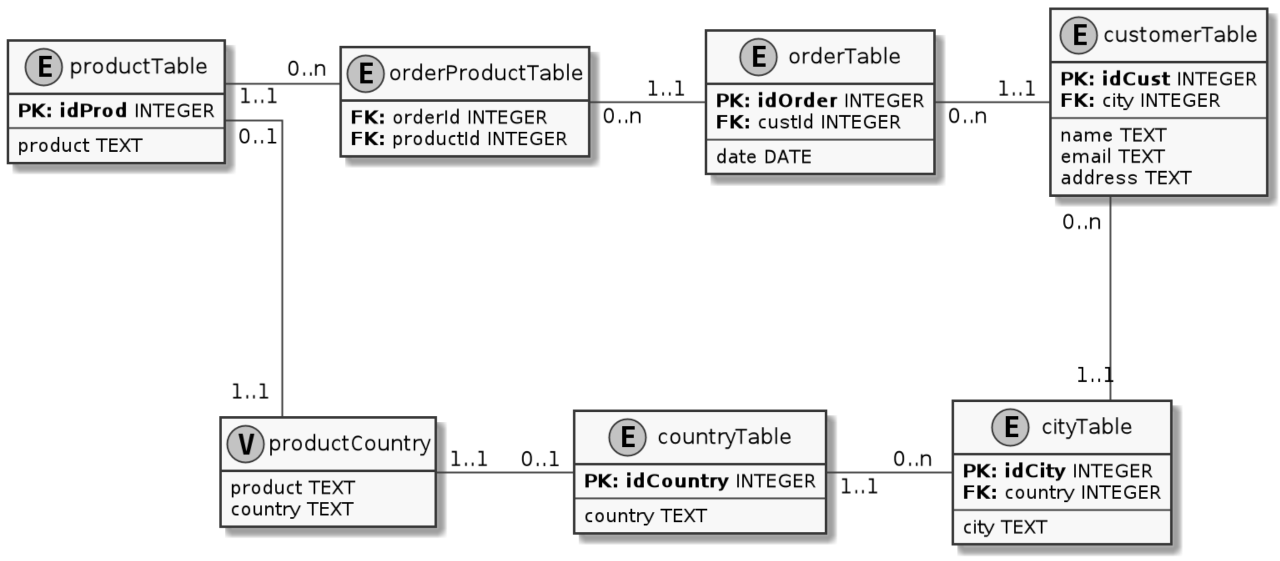
orderTable		
idOrder	custId	date
20140001	2	01-02-2014
20140002	2	03-10-2014
20140003	3	04-08-2013
20140004	1	06-06-2015

orderProductTable	
orderId	productId
20140002	7
20140002	1
20140003	19
20140004	10
20140002	10
20140003	7

productTable	
idProd	product
1	Mouse
2	keyboard
3	Monitor
19	CPU
5	USB hub
10	HDMI cable
7	Anti Virus

The tables and relations given to create the ER diagram

Above is the information used as a base for the diagram, relations are designated by the arrows being foreign keys creating a relation to a primary key at the arrow end. All fields beginning width the designation "id" is used as a primary key.



ER diagram (including the productCountry view).

The above ER diagram primary keys are designated "**PK:**" and foreign keys "**FK:**". This diagram uses 0..1, and 0..n number notation, instead of crow foot notation. Referring to the "productTable" in the illustration above, the relation going from "productTable" to "orderedProductTable" is marked "1..1" and "0..n". This tells that the primary key "idProd" in "productTable" relates to the foreign key "productId" in "orderProductTable", in a 1 to many relations ship. Meaning the “idProd” in “productTable” instance on one side of the relationship can have many related “productId” in “orderProductTable” instances on the other side.

3. View in ER diagrams.

I have found no conclusive documentation about illustrating view in an ER diagram, but since the view relates to entities in tables, I see no reason it should not be done. In the ER diagram above the “productCountry” is a view that has a relation to

4. PlantUML for creating ER diagrams.

I am continuing to investigate the PlantUML in relation to database diagrams. I have added and changed the macros for creating the key entries, to look more like the style of VISIO (there is no way to use the crow foot relation type).

Below is the PlantUML source file for the above ER diagram.

```
@startuml
skinparam monochrome true
skinparam linetype ortho
scale 2

!define table(x) class x << (T,#FFAAAA) >>
!define view(x) class x << (V,#FFAAAA) >>
!define ent(x) class x << (E,#FFAAAA) >>
!define primary_key(x) <b>PK: x</b>
!define foreign_key(x) <b>FK: </b>x

hide methods
hide stereotypes

ent(customerTable) {
    primary_key(idCust) INTEGER
    foreign_key(city) INTEGER
    ---
    name TEXT
    email TEXT
    address TEXT
}

ent(cityTable) {
    primary_key(idCity) INTEGER
    foreign_key(country) INTEGER
    ---
    city TEXT
}

ent(countryTable) {
    primary_key(idCountry) INTEGER
    ---
    country TEXT
}

ent(orderTable) {
    primary_key(idOrder) INTEGER
    foreign_key(custId) INTEGER
    ---
    date DATE
}

ent(orderProductTable) {
    foreign_key(orderId) INTEGER
    foreign_key(productId) INTEGER
}

ent(productTable) {
    primary_key(idProd) INTEGER
    ---
    product TEXT
}

productTable "1..1" - "0..n" orderProductTable : "        "
orderProductTable "0..n" - "1..1 " orderTable : "        "
orderTable "0..n  " - "1..1" customerTable : "        "
customerTable "0..n" -- "1..1" cityTable
countryTable "1..1" - "0..n " cityTable : "        "

view(productCountry) {
    product TEXT
    country TEXT
}

productCountry "1..1" --- "0..1" productTable
productCountry "1..1" - "0..1" countryTable
@enduml
```

5. Conclusion

ER diagrams are really good for their task of seeing and planning relations in databases. Since I am still using PlantUML to create the database diagram, and since PlantUML has no specific support for ER diagrams, I hope to get some comments related to the correctness and if this type of diagrams are usable and looks correct.