

# SQL Problems and solutions

*<https://sql-ex.ru/>*

*S. I. Moiseenko*





# SQL Problems and solutions

S. I. Moiseenko



 back go forward 






## Appendixes

Here are given accounts of all the databases which are used for solving problems at [SQL-EX.RU](http://SQL-EX.RU) website as well as the list of exercises intended for the learning stage. Exercises considered in the book are marked by sign "+".





### Content:



-  Appendix 1. Databases description
-  Appendix 2. Exercises list
-  Appendix 3. Chronicles of Torus

 sql-ex.ru learning stage

 Previous [Querying graph database data]

[Appendix 1. Databases description] Next 



# SQL Problems and solutions

S. I. Moiseenko



◀ back go forward ▶



## Appendixes






### Appendix 1. Databases description

---

SHARE    ...

#### Content:



-  (1.1.) Computer firm
-  (1.2.) Recycled firm
-  (1.3.) Ships
-  (1.4.) Airport
-  (1.5.) Painting



Computer firm ships Airport Painting recycled materials



Previous [Appendixes]

[Computer firm] Next ▶

ENHANCED BY Google





# SQL Problems and solutions

S. I. Moiseenko



◀ back go forward ▶

Common mistakes in solving SELECT statement exercises

## Computer Firm Database

The database schema consists of four tables (see Figure 1.1):

- Product(maker, model, type)
- PC(code, model, speed, ram, hd, cd, price)
- Laptop(code, model, speed, ram, hd, screen, price)
- Printer(code, model, color, type, price)

The Product table contains data on the maker, model number, and type of product ('PC', 'Laptop', or 'Printer'). It is assumed that model numbers in the Product table are unique for all makers and product types. Each personal computer in the PC table is unambiguously identified by a unique code, and is additionally characterized by its model (foreign key referring to the Product table), processor speed (in MHz) – speed field, RAM capacity (in Mb) – ram, hard disk drive capacity (in Gb) – hd, CD-ROM speed (e.g., '4x') – cd, and its price. The Laptop table is similar to the PC table, except that instead of the CD-ROM speed, it contains the screen size (in inches) – screen. For each printer model in the Printer table, its output type ('y' for color and 'n' for monochrome) – color field, printing technology ('Laser', 'Jet', or 'Matrix') – type, and price are specified.

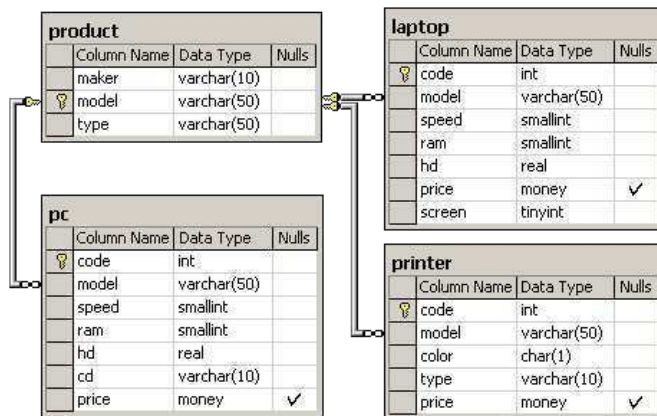


Fig. 1.1. Computer firm database schema


















Additional information can be obtained from the logical database schema shown in Fig. 1.1. Each table containing data on products of a specific type (PCs, laptops, or printers) has a foreign key (model) referencing the Product table. The one-to-many relationship means that a model present in the Product table may be missing in the other tables. On the other hand, a model with the same number, even with identical technical characteristics may occur in these tables several times, since the code column represents the primary key here. The latter fact calls for additional explanation, since different people interpret the term model in different ways. Within the framework of this schema, a model is considered to comprise products with a uniform manufacturer and production technology. E.g., models having the same number may be equipped with storage devices that are technically identical but have different capacities, i.e., 60 and 80 GB. Among other things, this means that, say, two personal computers having identical model numbers yet different prices may be present in the PC table.

In terms of the subject area this schema may imply that the Product table holds information about all known suppliers of good types under consideration and models supplied by them, while the other tables contain models currently available (offered for sale). Therefore, it is quite possible there is a maker producing models none of which is in stock at the moment.





## Content:




-  (1.2) Exercise #2
-  (1.3) Exercise #3
-  (1.4) Exercise #5
-  (1.5) Exercise #6
-  (1.6) Exercise #7
-  (1.7) Exercise #8
-  (1.8) Exercise #10
-  (1.8) Exercise #11
-  (1.8) Exercise 71
-  (1.10) Exercise #13
-  (1.21) Exercise 3 (rating)
-  (1.11) Exercise #15
-  (1.12) Exercise #16
-  (1.13) Exercise #17
-  (1.14) Exercise #18
-  (1.15) Exercise #20
-  (1.16) Exercise #23
-  (1.17) Exercise #24
-  (1.18) Exercise #25
-  (1.19) Exercise #26
-  (1.20) Exercise #27
-  (1.21) Exercise #28

 Product printer laptop PC Computer firm

 [Previous](#) [Common mistakes in solving SELECT statement exercises]

[Exercise #1] [Next](#) 

ENHANCED BY Google



Provided by RibSage



# SQL Problems and solutions

S. I. Moiseenko



◀ back go forward ▶

## Common mistakes in solving SELECT statement exercises

### Database «Recycled materials company»

The company receiving recycled and has several points of reception. Each point receives money to pay to suppliers for raw materials. In fact, the diagram shows two databases. In each task under this scheme uses only one pair of tables (with the suffix «\_o», or without it).

In tables Income\_o and Outcome\_o the primary key is a pair of attributes (point, date) - point number and date of reception. This key must simulate a situation when information about receiving money at the receiving point, and their extradition to suppliers recorded in the database no more than once a day.

Income			
	Column Name	Data Type	Nulls
	code	int	
	point	tinyint	
	[date]	datetime	
	inc	smallmoney	

Outcome			
	Column Name	Data Type	Nulls
	code	int	
	point	tinyint	
	[date]	datetime	
	out	smallmoney	

Income_o			
	Column Name	Data Type	Nulls
	point	tinyint	
	[date]	datetime	
	inc	smallmoney	

Outcome_o			
	Column Name	Data Type	Nulls
	point	tinyint	
	[date]	datetime	
	out	smallmoney	

Fig. 2.1.The scheme of «Recycled materials company» database.

#### Note

The data values in the column date does not contain the time, for example, 2001-03-22 00:00:00.000. Unfortunately, the use for the datetime data type of column may cause confusion, since it is obvious that taking into account the time does not allow to limit multiple input values with the same date (and the item number) and different time of day. This deficiency associated with the lack of specific data types for date and time have been overcome in SQL Server 2008. To ensure the correct input in SQL Server 2000 the CK\_Income\_o restriction can be used:

```

1. ALTER TABLE Income_o ADD
2. CONSTRAINT PK_Income_o PRIMARY KEY
3. (
4. [point],
5. [date]
6. ),
7. CONSTRAINT CK_Income_o CHECK
8. (
9. DATEPART(hour,[date]) + DATEPART(minute,[date]) +
10. DATEPART(second,[date]) + DATEPART(millisecond,[date]) = 0
11. )

```

This restriction (the amount of hours, minutes, seconds and milliseconds is equal to zero) would not set any time other than 00:00:00.000. With such restriction the primary key will actually ensure that there is only one row per day for each point of reception.

The Income\_o (point, date, inc) table contains information on the income (inc) per point of reception (point). Similar table — Outcome\_o (point, date, out) — is used to control the expenses (out).





The second pair of tables — Income (code, point, date, inc) and Outcome (code, point, date, out) — simulates the situation where the income and expenses of money can be recorded several times a day. It should be noted that if the record in the last

table only date without time (that occurs), then no natural combination of attributes can be used as the primary key, since the amount of money can be the same. Therefore, we must take into account the time, or add an artificial key. We used the second option, adding an integer column code only in order to ensure uniqueness of records in the table.



Content:



-  (2.1) Exercise #30
-  (2.2) Exercise #59
-  (2.3) Exercise #60
-  (2.4) Exercise 128

 Recycled materials company income outcome

 Previous [Exercise #28]

[Exercise #30] Next 

ENHANCED BY Google 



**Last added:**  
Explicit join operations  
Exercise #28  
Exercise #8 (tips and solutions) page 2



# SQL Problems and solutions

S. I. Moiseenko



◀ back go forward ▶

Common mistakes in solving SELECT statement exercises

## Database «Ships» page 1

The database of naval ships that took part in World War II is under consideration. The database has the following relations:

1. Classes (class, type, country, numGuns, bore, displacement)
2. Ships (name, class, launched)
3. Battles (name, date)
4. Outcomes (ship, battle, result)

Ships in classes are arranged to a single project. A class is normally assigned the name of the first ship in the class under consideration (head ship); otherwise, the class name does not coincide with any ship name in the database.

The Classes relation includes the class name, type (bb for a battle ship, or bc for a battle cruiser), country where the ship was built, number of main guns, gun caliber (diameter of the gun barrel, in inches), and displacement (weight in tons).

The Ships relation includes the ship name, its class name, and launch year.

The Battles relation covers the name and date of a battle the ships participated; while the result of their participation in the battle (sunk, damaged, or unharmed - OK) is in the Outcomes relation.

### Notes:

*The Outcomes relation may include the ships not included in the Ships relation.*

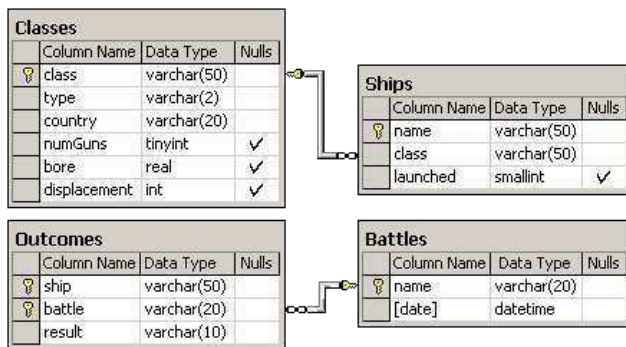


Fig. 3.1. Schema of database «Ships»

[Next page](#)



### Content:

- (3.7.) Exercise #70
- (3.2.) Exercise #37
- (3.3.) Exercise #38
- (3.4.) Exercise #39







# SQL Problems and solutions

S. I. Moiseenko



◀ back go forward ▶

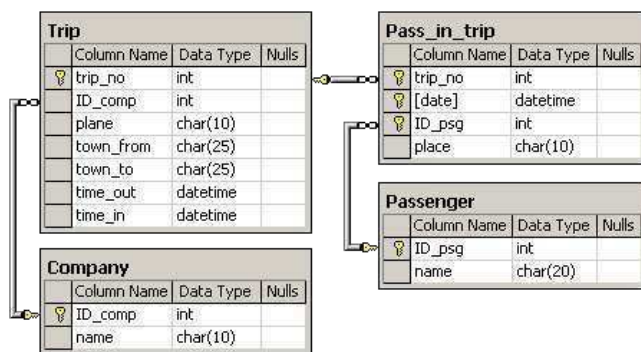
## Appendixes

### Appendix 1. Databases description

## Airport

Database schema consists of 4 tables:

1. Company (ID\_comp, name)
2. Trip (trip\_no, id\_comp, plane, town\_from, town\_to, time\_out, time\_in)
3. Passenger (ID\_psg, name)
4. Pass\_in\_trip (trip\_no, date, ID\_psg, place)



Company table has ID and name of the company, which transports passengers.

Trip table has information about trips: trip number, company ID, plane type, departure city, arrival city, departure time, and arrival time.

The Passenger table has passenger's ID and passenger's name.

Pass\_in\_trip table has information about the flights: trip number, departure date (day), passenger's ID and his place during the flight. We should note that,

- Any trip is being accomplished every day; duration of a flight is less than a calendar-day (24 hours);
- Time and date are considered comparatively one time zone;
- The departure time and the arrival time are given to within a minute;
- There can be the passengers bearing the same names (for example, Bruce Willis);
- The place during the flight is a number followed by a letter; the number defines the row number, the letter (a - d) - the place in the row (from the left to the right) in the alphabetical order;
- Relationships and restrictions are shown in the data schema.



database schema Airport Trip Passenger Pass\_in\_trip Company

◀ Previous [Ships]

[Painting] Next ▶



# SQL Problems and solutions

S. I. Moiseenko



◀ back go forward ▶

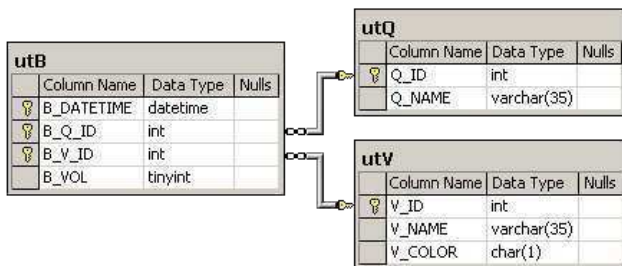
## Appendixes

### Appendix 1. Databases description

## Painting

Database schema consists of 3 tables:

1. utQ (Q\_ID, Q\_NAME)
2. utV (V\_ID, V\_NAME, V\_COLOR)
3. utB (B\_Q\_ID, B\_V\_ID, B\_VOL, B\_DATETIME)



The utQ table includes square identifier, square name. Note that non-painted square is black.

The table utV includes balloon (the can) identifier, balloon name, and paint color.

The table utB shows information on painting square with paint balloon and includes the square identifier, the balloon identifier, the paint quantity, and time of painting.

It should be noted that

- balloon may be of one from three colors: red (V\_COLOR='R'), green (V\_COLOR='G'), or blue (V\_COLOR='B');
- any balloon is originally full with volume of 255;
- square color is defined in accordance with RGB rule, i.e. R=0, G=0, B=0 is black, whereas R=255, G=255, B=255 is white;
- any record in the table utB decreases paint quantity in the balloon by B\_VOL and increase paint quantity in the square by the same value;
- B\_VOL must be more than 0 and less or equal 255;
- paint quantity of the same color in one square may not be over 255;
- paint quantity in a balloon may not be less than 0.



☁ database schema Painting utB utQ utV balloon

◀ Previous [Airport]

[Appendix 2. Exercises list] Next ▶