

# Основы проектирования баз данных

**Изменения данных**

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

FIRST

;

С ключевым словом FIRST столбец phone размещается перед всеми остальными столбцами.

phone	contact_id	last_name	first_name	email
-------	------------	-----------	------------	-------

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

LAST

;

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

FIFTH

;

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

;

С ключевым словом LAST столбец phone размещается после всех остальных столбцов. То же происходит при использовании ключевого слова FIFTH и без указания позиции.

contact_id	last_name	first_name	email	phone
------------	-----------	------------	-------	-------

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

SECOND

;

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

BEFORE

last\_name

;

С ключевым словом SECOND столбец phone размещается во второй позиции; то же происходит при использовании ключевого слова BEFORE (с именем столбца last\_name).

contact_id	phone	last_name	first_name	email
------------	-------	-----------	------------	-------

```
ALTER TABLE my_contacts
```

```
ADD COLUMN phone VARCHAR(10)
```

AFTER

last\_name

;

Конструкция AFTER last\_name размещает столбец phone в третьей позиции. Если бы в наборе был магнит THIRD, то он сделал бы то же самое.

contact_id	last_name	phone	first_name	email
------------	-----------	-------	------------	-------

Название столбца ничего не говорит о его содержимом.

Не содержит информации о том, какие данные должны храниться в этой таблице.

С парой символов подчеркивания имя столбца станет более понятным.

**projekts**

<b>number</b>	<b>descriptionofproj</b>	<b>contractoronjob</b>
1	покраска дома	Мэрфи
2	перестройка кухни	Вальдес
3	укладка паркета	Келлер
4	кровельные работы	Джексон



зайте воспользуемся командой DESCRIBE и посмотрим, какую структуру имеет таблица. Из ее результатов мы узнаем, есть ли у таблицы первичный ключ

Имена таблицы и столбцов оставляют желать лучшего, но с данными все в порядке. Нам хотелось бы их сохранить.

---

```
ALTER TABLE project_list
```

```
MODIFY COLUMN proj_desc VARCHAR(120);
```

Имя изменяемого  
столбца.

Новый тип данных.

И не забудьте проследить за тем,  
чтобы новый тип не привел  
к усечению существующих данных!

---

```
ALTER TABLE название_таблицы [WITH CHECK | WITH NOCHECK]
{ ADD название_столбца тип_данных_столбца [атрибуты_столбца] |
  DROP COLUMN название_столбца |
  ALTER COLUMN название_столбца тип_данных_столбца [NULL|NOT NULL] |
  ADD [CONSTRAINT] определение_ограничения |
  DROP [CONSTRAINT] имя_ограничения}
```

# Заполнение нового столбца существующими данными

```
SELECT RIGHT(location, 2) FROM my_contacts;
```

Выделение символов от ПРАВОГО края значения (функция LEFT выделяет символы от левого края).

Используемый столбец.

Количество символов, выделяемых от правого края значения.

## Выборка всех символов до запятой

Функция `SUBSTRING_INDEX()` находит все символы текстового значения, **предшествующие заданному символу или подстроке**. Запятая заключается в апострофы, а функция `SUBSTRING_INDEX()` возвращает все символы, стоящие перед запятой.

```
SELECT SUBSTRING_INDEX(location, ',', 1) FROM my_contacts;
```

Функция выделяет часть содержимого столбца (подстроку). Она ищет текст, заключенный в апострофы (запятая в данном случае), и возвращает все символы, предшествующие ему.

И снова имя столбца.

Запятая, которая ищется в тексте.

«1», потому что ищется первая запятая. С параметром «2» функция найдет вторую запятую в строке и вернет все предшествующие ей символы.

**Строковые функции выделяют часть значения текстового столбца.**

```
UPDATE my_contacts
```

```
SET state = RIGHT(location, 2);
```

Новый столбец для сокращенного обозначения штата.

Этот вызов функции выделяет два последних символа в столбце location.

# Фильмотека

«Д» и «Н» — сокращения  
для «Да» и «Нет».

Дата приобретения  
фильма.

movie\_table

movie_id	title	rating	drama	comedy	action	gore	scifi	for_kids	cartoon	purchased
1	Корпорация монстров	G	Н	Д	Н	Н	Н	Д	Д	6-3-2002
2	Крестный отец	R	Н	Н	Д	Д	Н	Н	Н	5-2-2001
3	Унесенные ветром	G	Д	Н	Н	Н	Н	Н	Н	20-11-1999
4	Американский пирог	R	Н	Д	Н	Н	Н	Н	Н	19-4-2003
5	Кошмар на улице Вязов	R	Н	Н	Д	Д	Н	Н	Н	19-4-2003
6	Касабланка	PG	Д	Н	Н	Н	Н	Н	Н	5-2-2001

Все эти столбцы существуют для того, чтобы  
работники могли отвечать на вопросы о содер-  
жании конкретных фильмов.

To: Персоналу видеотеки  
From: Директор  
Subject: Новым полкам — новые категории!

Всем привет,

Новые полки уже привезли, и я хочу привести  
в порядок наши фильмы. Мы будем использо-  
вать следующие категории:

- Боевики и приключения
- Драма
- Комедия
- Семейное кино
- Ужасы
- Фантастика и Фэнтези
- Разное

Разберитесь сами, как заставить нашу текущую  
таблицу работать с новыми категориями.

А мне пора на обед.

Директор

# Преобразование таблицы

movie\_table

title	rating	drama	comedy	action	gore	scifi	for_kids	cartoon	category
Большое приключение	G	Н	Н	Н	Н	Н	Д	Н	семейное
Грег: Неизвестные истории	PG	Н	Н	Д	Н	Н	Н	Н	боевик
Безумные клоуны	R	Н	Н	Н	Д	Н	Н	Н	ужасы
Параскеведекатриа-фобия	R	Д	Д	Д	Н	Д	Н	Н	?
Крыса по имени Дарси	G	Н	Н	Н	Н	Н	Д	Н	семейное
Конец очереди	R	Д	Н	Н	Д	Д	Н	Д	разное
Блестящие вещи	PG	Д	Н	Н	Н	Н	Н	Н	драма
Заберите обратно	R	Н	Д	Н	Н	Н	Н	Н	комедия
Наживка для акул	G	Н	Н	Н	Н	Н	Д	Н	?
Разгневанный пират	PG	Н	Д	Н	Н	Н	Н	Д	разное
Планета пригодна для жизни	PG	Н	Д	Н	Н	Д	Н	Н	?

Вопросительным знаком помечены столбцы, измененные более чем одной командой UPDATE. Значение столбца зависит от порядка выполнения UPDATE.



```
UPDATE movie_table
SET category =
CASE
```

```
  WHEN drama = 'Д' THEN 'драма'
  WHEN comedy = 'Д' THEN 'комедия'
  WHEN action = 'Д' THEN 'боевик'
  WHEN gore = 'Д' THEN 'ужасы'
  WHEN scifi = 'Д' THEN 'фантастика'
  WHEN for_kids = 'Д' THEN 'семейное'
  WHEN cartoon = 'Д' THEN 'семейное'
  ELSE 'разное'
```

```
END;
```

Всем записям, не подходящим ни под одно из перечисленных условий, назначается категория «разное».

То же самое, что UPDATE movie\_table SET category = 'драма' WHERE drama = 'Д' — но намного компактнее!

Значения, которые оставались неопределенными при заполнении нового столбца отдельными командами UPDATE, теперь определены.

Но также обратите внимание на новые значения для фильмов «Разгневанный пират» и «Конец очереди».

movie\_table

title	rating	drama	comedy	action	gore	scifi	for_kids	cartoon	category
Большое приключение	PG	Н	Н	Н	Н	Н	Н	Д	семейное
Грег: Неизвестные истории	PG	Н	Н	Д	Н	Н	Н	Н	боевик
Безумные клоуны	R	Н	Н	Н	Д	Н	Н	Н	ужасы
Параскеведекатриафобия	R	Д	Д	Д	Н	Д	Н	Н	драма
Крыса по имени Дарси	G	Н	Н	Н	Н	Н	Д	Н	семейное
Конец очереди	R	Д	Н	Н	Д	Д	Н	Д	драма
Блестящие вещи	PG	Д	Н	Н	Н	Н	Н	Н	драма
Заберите обратно	R	Н	Д	Н	Н	Н	Н	Н	комедия
Наживка для акул	G	Н	Н	Н	Н	Н	Д	Н	семейное
Разгневанный пират	PG	Н	Д	Н	Н	Н	Н	Д	комедия
Планета пригодна для жизни	PG	Н	Т	Н	Н	Т	Н	Н	комедия

# Задание

movie\_table

movie_id	title	rating	category	purchased
83	Большое приключение	G	семейное	6-3-2002
84	Грег: Неизвестные истории	PG	боевик	5-2-2001
85	Безумные клоуны	R	ужасы	20-11-1999
86	Параскеведекатриафобия	R	боевик	19-4-2003
87	Крыса по имени Дарси	G	семейное	19-4-2003
88	Конец очереди	R	разное	5-2-2001
89	Блестящие вещи	PG	драма	6-3-2002
90	Заберите обратно	R	комедия	5-2-2001
91	Наживка для акул	G	разное	20-11-1999
92	Разгневанный пират	PG	разное	19-4-2003
93	Планета пригодна для жизни	PG	фантастика	5-2-2001

## Order By

---

```
SELECT title, category  
FROM movie_table  
WHERE  
title LIKE 'A%'  
AND  
category = 'семейное'  
ORDER BY title;
```

Что следует убрать из этого запроса, чтобы расширить его возможности?

```
SELECT title, category, purchased
FROM movie_table
ORDER BY category, purchased;
```

movie_id	title	rating	category	purchased
84	Грег: Неизвестные истории	PG	боевик	5-2-2001
86	Параскеведекатриафобия	R	боевик	19-4-2003
89	Блестящие вещи	PG	драма	6-3-2002
90	Заберите обратно	R	комедия	5-2-2001
91	Наживка для акул	G	разное	20-11-1999
88	Конец очереди	R	разное	5-2-2001
92	Разгневанный пират	PG	разное	19-4-2003
83	Большое приключение	G	семейное	6-3-2002
87	Крыса по имени Дарси	G	семейное	19-4-2003
85	Безумные клоуны	R	ужасы	20-11-1999

```
SELECT * FROM movie_table
ORDER BY category, purchased, title;
```

# Начинаем с НОВЫХ

---

```
SELECT title, purchased  
FROM movie_table
```

```
ORDER BY title ASC, purchased DESC;
```

Здесь можно указать ключевое слово ASC, но это не обязательно. Достаточно помнить, что по умолчанию данные сортируются по возрастанию.

Чтобы данные были отсортированы от Я до А (или от 9 до 1), используйте ключевое слово DESC.

# Конец фильма

---