

Основы проектирования баз данных

Нормализация данных

Проектирование таблиц

Ловим рыбу

fish_info

common	species	location	weight
большеротый окунь	M. salmoides	Монтгомери Лейк, GA	22 фт 4 унц
судак	S. vitreus	Олд Хикори Лейк, TN	25 фт 0 унц
лосось Кларка	O. Clarki	Пирамид Лейк, NV	41 фт 0 унц
желтый окунь	P. Flavescens	Бордентаун, NJ	4 фт 3 унц
синежаберник	L. Macrochirus	Кетона Лейк, AL	4 фт 12 унц
панцирник	L. Osseus	Тринити Ривер, TX	50 фт 5 унц
белый краппи	P. annularis	Дамба Энид, MS	5 фт 3 унц
красноперая щука	E. americanus	Дьюарт Лейк, IN	1 фт 0 унц
серебристый карась	C. auratus	Лейк Ходжес, CA	6 фт 10 унц
чавыча	O. Tshawytscha	Кенай Ривер, AK	97 фт 4 унц

fish_records

first_name	last_name	common	location	state	weight	date
Джордж	Перри	большеротый окунь	Монтгомери Лейк	GA	22 фт 4 унц	2/6/1932
Мабри	Харпер	судак	Олд Хикори Лейк	TN	25 фт 0 унц	2/8/1960
Джон	Скиммерхорн	лосось Кларка	Пирамид Лейк	NV	41 фт 0 унц	1/12/1925
С.С.	Эббот	желтый окунь	Бордентаун	NJ	4 фт 3 унц	1/5/1865
Т.С.	Хадсон	синежаберник	Кетона Лейк	AL	4 фт 12 унц	9/4/1950
Таунсенд	Миллер	панцирник	Тринити Ривер	TX	50 фт 5 унц	30/7/1954
Фред	Брайт	белый краппи	Дамба Энид	MS	5 фт 3 унц	31/7/1957
Майк	Берг	красноперая щука	Дьюарт Лейк	IN	1 фт 0 унц	9/6/1990
Флорентино	Абена	серебристый карась	Лейк Ходжес	CA	6 фт 10 унц	17/4/1996
Лес	Андерсон	чавыча	Кенай Ривер	AK	97 фт 4 унц	17/5/1985

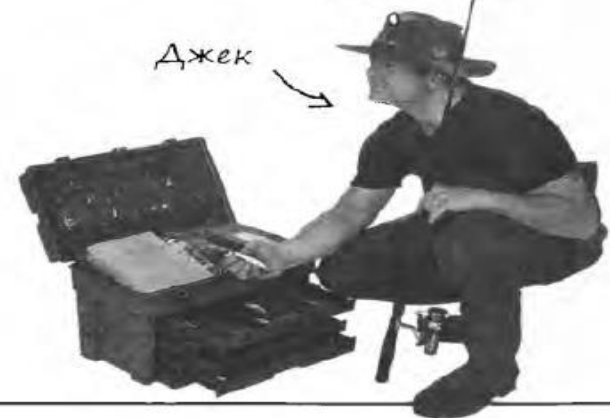
Я ихтиолог. В своей таблице я буду проводить поиск только по бытовому или научному названию, чтобы узнать вес и место вылова рыбы.

А я пишу статьи для рыболовного журнала. И мне нужно знать имена рыбаков, даты и места рекордного вылова.

Напишите запрос для каждой таблицы, возвращающий все записи для штата Нью-Джерси.



Марк



Джек

Результат

Запрос напрямую
обращается
к столбцу state.

А мне часто приходится искать по штату, поэтому я выделил название штата в отдельный столбец при создании таблицы.

```
SELECT * FROM fish_records
```

```
WHERE state = 'NJ';
```

first_name	last_name	common	location	state	weight	date
C.C.	Эббот	желтый окунь	Бордентаун	NJ	4 фт 3 унц	1/5/1865

Основные принципы

1. Какой основной объект описывает Ваша таблица?
2. Как будет использоваться Ваша таблица?
3. Как проще всего запросить данные из таблицы

Атомарные данные

1. Какой один объект описывает ваша таблица?
2. Как вы предполагаете ИСПОЛЬЗОВАТЬ таблицу для получения информации о ее объекте?
3. Содержат ли столбцы таблицы атомарные данные, чтобы запросы были короткими и конкретными?



Анализируем прошлые ошибки

ПРАВИЛО 1. Столбец, содержащий атомарные данные, не может состоять из нескольких однотипных элементов.

ПРАВИЛО 2. Таблица с атомарными данными не может содержать несколько однотипных столбцов

fish_records



first_name	last_name	common	location	state	weight	date
Джордж	Перри	большеротый окунь	Монтгомери Лейк	GA	22 фт 4 унц	2/6/1932
Мабри	Харпер	судак	Олд Хикори Лейк	TN	25 фт 0 унц	2/8/1960
Джон	Скиммерхорн	лосось Кларка	Пирамид Лейк	NV	41 фт 0 унц	1/12/1925
С.С.	Эббот	желтый окунь	Бордентаун	NJ	4 фт 3 унц	1/5/1865
Т.С.	Хадсон	синежаберник	Кетона Лейк	AL	4 фт 12 унц	9/4/1950
Таунсенд	Миллер	панцирник	Тринити Ривер	TX	50 фт 5 унц	30/7/1954
Фред	Брайт	белый краппи	Дамба Энид	MS	5 фт 3 унц	31/7/1957
Майк	Берг	красноперая щука	Дьюарт Лейк	IN	1 фт 0 унц	9/6/1990
Флорентино	Абена	серебристый карась	Лейк Ходжес	CA	6 фт 10 унц	17/4/1996
Лес	Андерсон	чавыча	Кенай Ривер	AK	97 фт 4 унц	17/5/1985

Преимущества нормализованных таблиц

1. Нормализованные таблицы не содержат дубликатов данных, а это сокращает размер базы данных.

Отсутствие дубликатов экономит дисковое пространство.

2. Уменьшение объема данных, по которым ведется поиск, ускоряет выполнение запросов.



1 нормальная форма

Мы уже знаем,
как это дела-
ется.



Каждая запись должна содержать атомарные значения.



Чтобы таблица
была полностью
нормализована,
каждой записи
необходимо при-
своить первич-
ный ключ.

Каждая запись должна обладать уникальным идентификатором, который называется первичным ключом.

Ключ

Ключ — это поле или комбинация полей, однозначно определяющие запись.

Могут ли эти данные быть ключом?

- ~~фамилия~~
- ~~имя~~
- номер паспорта
- ~~номер дома~~
- регистрационный номер автомобиля
- ~~город проживания~~
- адрес электронной почты
- ~~дата выполнения работы~~
- марка стиральной машины ?



Может ли быть несколько ключей в таблице?

Ключ

Первичный ключ — это ключ, выбранный в качестве основного.

Простой ключ — состоит из одного поля.

Составной ключ — состоит из нескольких полей.

Дата	Время	Температура	Влажность	Скорость ветра
21.07.2012	12:00	25	75	4
21.07.2012	15:00	23	70	3
...



Какой ключ?

Составной ключ ← Дата + Время

Свойства ключа:

- уникальность
- несократимость (Дата + Время + Влажность)

Требования к первичному ключу

1. Первичный ключ не может содержать NULL
2. Значение первичного ключа должно задаваться при вставке записи
3. Первичный ключ должен быть компактным
4. Значения первичного ключа должны оставаться неизменными
5. Помните, что первичный ключ должен быть уникальным.

Суррогатный ключ

Фамилия	Имя	Адрес	Телефон
Иванов	Петр	Суворовский пр., д. 32, кв. 11	275-75-75
Петров	Василий	Кутузовский пр., д. 12, кв. 20	276-76-76
Васильев	Иван	Нахимовский пр., д. 23, кв. 33	277-77-77



Какой ключ?

Суррогатный ключ – это дополнительное поле, которое служит первичным ключом.

- операции с числами выполняются быстрее
- уникальность поддерживается СУБД (поле-счётчик)

Добавляем первичный ключ

```
CREATE TABLE [dbo].[my_contacts](  
    contact_id int not null,  
    [last_name] [varchar](30) not NULL,  
    [first_name] [varchar](20) not NULL,  
    [email] [varchar](50) not NULL,  
    [gender] [char](1) not NULL,  
    [birthday] [date] not NULL,  
    [profession] [varchar](50) not NULL,  
    [locations] [varchar](50) not NULL,  
    [statuss] [varchar](20) not NULL,  
    [interests] [varchar](100) not NULL,  
    [seeking] [varchar](100) not NULL,  
    primary key (contact_id)  
)
```

Автоматическая нумерация

```
CREATE TABLE [dbo].[my_contacts](  
    [contact_id] [int] IDENTITY(1,1) NOT NULL,  
    [last_name] [varchar](30) NOT NULL,  
    [first_name] [varchar](20) NOT NULL,  
    [email] [varchar](50) NOT NULL,  
    [gender] [char](1) NOT NULL,  
    [birthday] [date] NOT NULL,  
    [profession] [varchar](50) NOT NULL,  
    [locations] [varchar](50) NOT NULL,  
    [statuss] [varchar](20) NOT NULL,  
    [interests] [varchar](100) NOT NULL,  
    [seeking] [varchar](100) NOT NULL,  
    primary key (contact_id)
```

```
)
```

Упражнение



Упражнение

1. Напишите команду `CREATE TABLE` для создания приведенной ниже таблицы, в которой хранятся имена и фамилии. Таблица должна содержать столбец первичного ключа с ключевым `AUTO_INCREMENT` и два атомарных столбца.

2. Откройте терминал SQL или графический интерфейс, выполните команду `CREATE TABLE`.

3. Попробуйте выполнить каждую из приведенных ниже команд `INSERT`. Обведите кружком команды, которые были успешно выполнены.

```
INSERT INTO your_table (id, first_name, last_name)
VALUES (NULL, 'Марсия', 'Брэди');
```

```
INSERT INTO your_table (id, first_name, last_name)
VALUES (1, 'Джен', 'Брэди');
```

```
INSERT INTO your_table
VALUES ('', 'Бобби', 'Брэди');
```

```
INSERT INTO your_table (first_name, last_name)
VALUES ('Синди', 'Брэди');
```

```
INSERT INTO your_table (id, first_name, last_name)
VALUES (99, 'Питер', 'Брэди');
```

4. Все ли команды были выполнены успешно? Напишите, как будет выглядеть содержимое таблицы после выполнения команд `INSERT`.

your_table

<i>id</i>	<i>first_name</i>	<i>last_name</i>

Конец фильма
