

Таблицы

Типы таблиц:

- Постоянные таблицы (т.е. созданные с помощью инструкции create table)
- Производные таблицы (т.е. строки, возвращаемые подзапросом и хранящиеся в памяти)
- Временные таблицы (т.е. изменяемые данные, хранящиеся в памяти)
- Виртуальные таблицы (т.е. созданные с помощью инструкции create view).

Создание таблицы

```
create table person (  
    person_id smallint unsigned,  
    fname varchar(20),  
    lname varchar(20),  
    eye_color char(2),  
    constraint pk_person primary key (person_id)  
);
```

изменение структуры таблицы

```
ALTER TABLE person MODIFY person_id SMALLINT UNSIGNED AUTO_INCREMENT;
```

какие таблицы ссылаются по внешним ключам на таблицу X

```
SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE REFERENCED_TABLE_NAME = 'X';
```

```
DROP TABLE favorite_food;
```

Ограничения (constraint)

Первичный ключ, может быть любое кол-во столбцов

```
constraint pk_person primary key (person_id)
```

Создание связи

```
CONSTRAINT fk_fav_food_person_id FOREIGN KEY (person_id) REFERENCES person (person_id)
```

Ограничение набора в столбце

```
enum('ht', 'hy')  
create table person (... eye_color ENUM('BR','BL','GR', ....);
```

AUTO_INCREMENT в описании ключа автоматическое увеличение номера

desc person - описание таблицы

not null при создании столбца запрещает быть пустым

проверка ограничений

```
SELECT * FROM information_schema.check_constraints WHERE table_schema = 'db name' and table_name =  
'table name';
```

просмотр скрипта создания таблицы со всеми внесенными изменениями

```
SHOW CREATE TABLE mytable;
```

Получение/изменение данных

```
INSERT INTO person (person_id, fname, lname, eye_color, birth_date)
```

```
VALUES (null, 'William','Turner', 'BR', '1972-05-27');
```

ORDER BY - группировка в селекте

```
UPDATE person
```

```
SET street = '1225 Tremont St.', city = Boston', state = 'MA', country = 'USA', postal_code = '02138' WHERE  
person_id = 1;
```

```
DELETE FROM person WHERE person_id = 2;
```

Структура select

select	Определяет,какие столбцы следует включить в результирующий набор запроса
from	Определяет таблицы, из которых следует выбирать данные, а также таблицы, которые должны быть соединены
where	Отсеивает ненужные данные
group by	Используется для группировки строк по общим значениям столбцов
having	Отсеивает ненужные данные
order by	Сортирует строки окончательного результирующего набора по одному или нескольким столбцам

Фильтрация

Логические операции в where: and, or, not, для группировки круглые скобки, операторы сравнения, такие как =, !=, <, >, like, in и between; арифметические операторы, такие как +, -, * и /.

Подзапросы в фильтрах:

```
... where film_id = (SELECT film_id FROM film WHERE title = 'RIVER OUTLAW') ...
```

Двухсторонний диапазон. Сначала нижняя граница включительно.

```
WHERE rental date BETWEEN '2005-06-14 AND '2005-06-16';
```

```
WHERE amount BETWEEN 10.0 AND 11.99;
```

```
WHERE last name BETWEEN 'FA' AND 'FR' - FRA не войдет. Можно 'FRB'
```

множество IN, NOT IN - WHERE rating IN ('G', 'PG');

первая буква Q - WHERE left(last name, 1) = 'Q';

регулярки: WHERE last_name REGEXP '^[QY]'

WHERE return_date IS NULL; (is not null) Если WHERE return_date = NULL; - ошибки не будет, но пустая выборка

он результат BETWEEN 180 AND 240;

exist - наличие хотя-бы одного значения в подмножестве

```
SELECT c.first_name, c.last_name  
FROM customer c WHERE EXISTS  
  (SELECT 1 FROM rental r  
   WHERE r.customer_id = c.customer id  
   AND date(r.rental date) < '2005-05-25');
```

Revision #2

Created 7 April 2025 03:04:33 by Admin

Updated 7 April 2025 05:10:20 by Admin