

sql基础

select top

规定要返回的记录数目

注：并非所有数据库都支持select top子句。

syntax：

```
select top number| percent column_name(s) from table_name;
```

MySQL syntax：

```
select column_name(s) from table_name limit number;
```

eg：

```
select * from Persons limit 5;
```

Oracle syntax：

```
select column_name(s) from table_name where rownum <= number;
```

eg：

```
select * from Persons where rownum <= 5;
```

再Microsoft SQL Server中还可以用百分比作为参数。

eg：

```
select top 50 percent from Websites;
```

SQL 通配符

%：替代0个或多个字符

-：替代一个字符

[charlist]：字符列中的任何单一字符

[^charlist] 或 [!charlist]：不在字符列的任何单一字符

MySQL 中使用 REGEXP 或 NOT REGEXP 运算符来操作正则表达式。

eg：

选取name以G, F, 或s开始的所有网站

```
select * from Websites where name REGEXP '^[GFs]';
```

eg：

选取name以A到H字母开头的网站

```
select * from Websites where name REGEXP '^[A-H]';
```

eg：

选取name不以A到H字母开头的网站

```
select * from Websites where name REGEXP '^[^A-H]';
```

SQL别名，列别名，表别名

eg：

吧三个列组合在一起，并创建一个名为 site_info的别名

```
select name, CONCAT(url, ', ', alexa, ', ', country) as site_info from Websites;
```

eg：

```
select w.name, w.url, a.count, a.date from Websites as w, access_log as a where a.site_id = w.id and w.name="菜鸟教程";
```

SQL JOIN

INNER JOIN（同JOIN）：如果表中有至少一个匹配，则返回行

eg：

```
select Websites.id, Websites.name, access_log.count, access_log.date from Websites inner join access_log on  
Websites.id=access_log.site_id;
```

eg：

```
select Websites.name, access_log.count, access_log.date from Websites inner join access_log on Website.id=access_log.site_id  
order by access_log.count;
```

LEFT JOIN：即使右表中没有陪陪，也从左表返回所有的行

eg：

```
select Websites.name, access_log.count, access_log.date from Websites left join access_log on Websites.id=access_log.site_id
order by access_log.count desc;
```

RIGHT JOIN : 及时左表中没有匹配, 也从右表返回所有的行

eg :

```
select Websites.name, access_log.count, access_log.date from access_log right join Websites on access_log.site_id=Websites.id
order by access_log.count desc;
```

FULL JOIN (FULL OUTER JOIN): 只要其中一个表中存在匹配, 则返回行

MySQL 不支持FULL OUTER JOIN.

可以再SQL Server中测试

eg :

```
select Websites.name, access_log.count, access_log.date from Websites full outer join access_log on
Websites.id=access_log.site_id order by access_log.count desc;
```

SQL UNION 操作符

用于合并两个或多个SELECT语句的结果集(去重)

注 : UNION内部的每个select 语句必须拥有相同数量的列, 列也必须拥有想死的数据类型。同时, 每个SELECT语句的列的顺序必须相同。

默认地, UNION操作符选取不同的值, 若要允许重复值, 使用UNION ALL (全部加起来, 不去重)

eg :

```
select * from Websites;
select * from apps;
select country from Websites union select country from apps order by country;
```

eg :

```
select country from Websites union all select country from apps order by country;
```

eg :

```
select country, name from Websites where country='CN' union all select country, app_name from apps where country='CN' order by
country;
```

SQL SELECT INTO 语句

从一个表复制数据, 并把数据插入到另一个新表中

syntax :

```
select * into newtable [in externaldb] from table1;
select column_name(s) into newtable [in externaldb] from table1;
```

eg :

创建Websites的备份复件

```
select * into WebsitesBackup2016 from Websites;
```

eg:

只复制一些列插入到新表中

```
select name, url into WebsitesBackup2016 from Websites;
```

eg :

只复制中国的网站插入到其中

```
select * into WebsitesBackup2016 from Websites where country='CN' ;
```

eg :

复制多个表中的数据插入到新表中

```
select Websites.name, access_log.count, access_log.date into WebsitesBackup2016 from Websites left join access_log on
Websites.id=access_log.site_id;
```

eg :

创建一个新的空表, 只需要添加促使查询没有数据返回的WHERE子句即可

```
select * into newtable from table1 where 1=0;
```

SQL INSERT INTO SELECT语句

从一个表复制数据, 并把数据插入到一个已存在的表中

syntax :

```
insert into table2 select * from table1;
insert into table2 (column_name(s)) select column_name(s) from table1;
```

eg :

复制apps中的数据插入到Websites中

```
insert into Websites (name, country) select app_name, country from apps;
insert into Websites (name, country) select app_name, country from apps where id=1;
```

SQL CREATE DATABASE语句

syntax :

create database dbname;

eg :

创建一个名为 my_db 的数据库

create database my_db;

MySQL 创建UNIQUE 约束

eg :

create table Persons

(

P_Id int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Address varchar(255),

City varchar(255),

UNIQUE (P_Id)

);

eg :

SQL Server / Oracle / MS Access 创建UNIQUE约束

create table Persons

(

P_Id int NOT NULL UNIQUE,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

eg :

MySQL / SQL Server / Oracle / MS Access 定义多个列的UNIQUE约束

create table Persons

(

P_Id int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Address varchar(255),

City varchar(255),

CONSTRAINT uc_PersonID UNIQUE (P_Id, LastName)

);

eg :

MySQL / SQL Server / Oracle / MS Access

alter table Persons add unique (P_Id);

alter table Persons add constraint uc_PersonID UNIQUE (P_Id, LastName);

eg :

MySQL 撤销UNIQUE约束

alter table Persons drop index uc_PersonID;

eg :

SQL Server / Oracle / MS Access 撤销UNIQUE约束

alter table Persons drop constraint uc_PersonID;

SQL DEFAULT约束

eg :

create table Persons

(

P_Id int NOT NULL,

```
LastName varchar(255) NOT NULL,  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255) default 'Sandnes'  
);
```

eg :

```
create table Orders  
(  
O_Id int NOT NULL,  
OrderNo int NOT NULL,  
P_Id int,  
OrderDate date DEFAULT GETDATE()  
);
```

SQL CREATE UNIQUE INDEX

eg :

```
create unique index index_name on table_name (column_name);
```

SQL DROP INDEX, DROP TABLE, DROP DATABASE

eg :

用于MS Access

```
drop index index_name on table_name;
```

eg :

用于MS SQL Server

```
drop index table_name.index_name;
```

用于DB2/Oracle

```
drop index index_name;
```

用于MySQL

```
alter table table_name drop index index_name;
```

eg :

```
drop table table_name;
```

```
drop database database_name;
```

```
truncate table table_name;
```

SQL ALTER TABLE

```
alter table table_name add column_name datatype;
```

```
alter table table_name drop column column_name;
```

eg :

SQL Server / MS Access

```
alter table table_name alter column column_name datatype;
```

eg :

My SQL / Oracle

```
alter table table_name modify column column_name datatype;
```

Oracle 10G 之后的版本

```
alter table table_name modify column_name datatype;
```

SQL AUTO INCREMENT字段

SQL Date 函数

SQL NULL 值

SQL NULL 函数 isnull(), nvl(), ifnull(), coalesce()

SQL 通用数据类型

SQL 用于各种数据库的数据类型

参考：<http://www.runoob.com/sql/sql-tutorial.html>