数据库：Oracle，Sybase，SQL Server，DB2，Access

SQL对大小写不敏感

数据操作语言（DML）和数据定义语言（DDL）

* **SELECT** - 从数据库表中获取数据
* **UPDATE** - 更新数据库表中的数据
* **DELETE** - 从数据库表中删除数据
* **INSERT INTO** - 向数据库表中插入数据
* **CREATE DATABASE** - 创建新数据库
* **ALTER DATABASE** - 修改数据库
* **CREATE TABLE** - 创建新表
* **ALTER TABLE** - 变更（改变）数据库表
* **DROP TABLE** - 删除表
* **CREATE INDEX** - 创建索引（搜索键）
* **DROP INDEX** - 删除索引

SELECT DISTINCT 列名称 FROM 表名称 //选择唯一的不重复

SELECT 列名称 FROM 表名称 WHERE 列 运算符 值

|  |  |
| --- | --- |
| **操作符** | **描述** |
| = | 等于 |
| <> | 不等于 |
| > | 大于 |
| < | 小于 |
| >= | 大于等于 |
| <= | 小于等于 |
| BETWEEN | 在某个范围内 |
| LIKE | 搜索某种模式 |

在某些版本的 SQL 中，操作符 <> 可以写为 !=

常见的错误发生，以下为正确形式

SELECT \* FROM Persons WHERE Year>1965

SELECT \* FROM Persons WHERE FirstName='Bush'

**AND 和 OR 运算符**

## ORDER BY 语句(可以使用 DESC 关键字,ASC)

ORDER BY Company, OrderNumber

ORDER BY Company DESC

## INSERT INTO 语句

INSERT INTO table\_name (列1, 列2,...) VALUES (值1, 值2,....)

## Update 语句

UPDATE 表名称 SET 列名称 = 新值 WHERE 列名称 = 某值

## DELETE 语句

DELETE FROM 表名称 WHERE 列名称 = 值

## 删除所有行

DELETE FROM table\_name

DELETE \* FROM table\_name

## TOP 子句 //TOP 子句用于规定要返回的记录的数目

## SQL Server 的语法：

SELECT TOP number|percent column\_name(s) //

FROM table\_name

SELECT TOP 50 PERCENT \* FROM Persons //50%的记录

**Mysql语法**

SELECT column\_name(s)

FROM table\_name

LIMIT number

### Oracle 语法

SELECT column\_name(s)

FROM table\_name

WHERE ROWNUM <= number

### SQL LIKE 操作符语法

SELECT column\_name(s)

FROM table\_name

WHERE column\_name LIKE pattern //pattern需要'%~~%' 也可以加 not like

## SQL 通配符

|  |  |
| --- | --- |
| **通配符** | **描述** |
| % | 替代一个或多个字符 |
| \_ | 仅替代一个字符 |
| [charlist] | 字符列中的任何单一字符// '[ALN]%' |
| [^charlist]  或者  [!charlist] | 不在字符列中的任何单一字符 |

SELECT \* FROM Persons

WHERE City LIKE '[!ALN]%' //不以ALN其中任意一个开头

### SQL IN 语法

SELECT \* FROM Persons

WHERE LastName IN ('Adams','Carter') //判断是否在其中

### SQL BETWEEN 语法(可以加not关键字)

SELECT column\_name(s)

FROM table\_name

WHERE column\_name

BETWEEN value1 AND value2

## SQL Alias(重命名 alias\_name为新的名字)//结果都是以新名字为准的

SELECT column\_name AS alias\_name

FROM table\_name

**SQL join 用于根据两个或多个表中的列之间的关系，从这些表中查询数据**

* JOIN: 如果表中有至少一个匹配，则返回行
* LEFT JOIN: 即使右表中没有匹配，也从左表返回所有的行
* RIGHT JOIN: 即使左表中没有匹配，也从右表返回所有的行
* FULL JOIN: 只要其中一个表中存在匹配，就返回行

### INNER JOIN 关键字语法(INNER JOIN 与 JOIN 是相同的)

SELECT column\_name(s)

FROM table\_name1

INNER JOIN table\_name2

ON table\_name1.column\_name=table\_name2.column\_name

### LEFT JOIN 关键字语法

SELECT column\_name(s)

FROM table\_name1

LEFT JOIN table\_name2

ON table\_name1.column\_name=table\_name2.column\_name

### RIGHT JOIN 关键字语法

SELECT column\_name(s)

FROM table\_name1

RIGHT JOIN table\_name2

ON table\_name1.column\_name=table\_name2.column\_name

### FULL JOIN 关键字语法

SELECT column\_name(s)

FROM table\_name1

FULL JOIN table\_name2

ON table\_name1.column\_name=table\_name2.column\_name

### SQL UNION 语法（union all 即使重复也列出）

SELECT column\_name(s) FROM table\_name1

UNION

SELECT column\_name(s) FROM table\_name2

SELECT column\_name(s) FROM table\_name1

UNION ALL

SELECT column\_name(s) FROM table\_name2

## SELECT INTO 语句

SELECT \*

INTO new\_table\_name [IN externaldatabase]

FROM old\_tablename

### SQL CREATE DATABASE 语法

CREATE DATABASE database\_name

### SQL CREATE TABLE 语法

CREATE TABLE 表名称

(

列名称1 数据类型,

列名称2 数据类型,

列名称3 数据类型,

....

)

|  |  |
| --- | --- |
| **数据类型** | **描述** |
| * integer(size) * int(size) * smallint(size) * tinyint(size) | 仅容纳整数。在括号内规定数字的最大位数。 |
| * decimal(size,d) * numeric(size,d) | 容纳带有小数的数字。  "size" 规定数字的最大位数。"d" 规定小数点右侧的最大位数。 |
| char(size) | 容纳固定长度的字符串（可容纳字母、数字以及特殊字符）。  在括号中规定字符串的长度。 |
| varchar(size) | 容纳可变长度的字符串（可容纳字母、数字以及特殊的字符）。  在括号中规定字符串的最大长度。 |
| date(yyyymmdd) | 容纳日期。 |

# SQL 约束(Constraints)

可以在创建表时规定约束（通过 CREATE TABLE 语句），或者在表创建之后也可以（通过 ALTER TABLE 语句）。

* NOT NULL
* UNIQUE
* PRIMARY KEY
* FOREIGN KEY
* CHECK
* DEFAULT

## SQL NOT NULL 约束

Id\_P int NOT NULL,

## SQL UNIQUE 约束

创建表时

UNIQUE (Id\_P) //mysql

Id\_P int NOT NULL UNIQUE //sql server/oracle

CONSTRAINT uc\_PersonID UNIQUE (Id\_P,LastName) //通用

创建表后

ALTER TABLE Persons

ADD UNIQUE (Id\_P)

## 撤销 UNIQUE 约束

ALTER TABLE Persons

DROP INDEX uc\_PersonID // mysql

ALTER TABLE Persons

DROP CONSTRAINT uc\_PersonID //sql server/oracle/MS Access

## SQL PRIMARY KEY 约束

创建表时

PRIMARY KEY (Id\_P) //mysql

Id\_P int NOT NULL PRIMARY KEY //sql server/oracle/MS Access

CONSTRAINT pk\_PersonID PRIMARY KEY (Id\_P,LastName)//通用

创建表后

ALTER TABLE Persons

ADD PRIMARY KEY (Id\_P)

ALTER TABLE Persons

ADD CONSTRAINT pk\_PersonID PRIMARY KEY (Id\_P,LastName)//命名方法

## 撤销 PRIMARY KEY 约束

ALTER TABLE Persons

DROP PRIMARY KEY //mysql

ALTER TABLE Persons

DROP CONSTRAINT pk\_PersonID //sql server/oracle/MA Access

## SQL FOREIGN KEY 约束

创建表时

FOREIGN KEY (Id\_P) REFERENCES Persons(Id\_P) //mysql

Id\_P int FOREIGN KEY REFERENCES Persons(Id\_P) //sql server/oracle/MS Access

CONSTRAINT fk\_PerOrders FOREIGN KEY (Id\_P) REFERENCES Persons(Id\_P) //需要命名时

创建表后

ALTER TABLE Orders

ADD FOREIGN KEY (Id\_P)

REFERENCES Persons(Id\_P)//通用

ALTER TABLE Orders

ADD CONSTRAINT fk\_PerOrders

FOREIGN KEY (Id\_P)

REFERENCES Persons(Id\_P)

## 撤销 FOREIGN KEY 约束

ALTER TABLE Orders

DROP FOREIGN KEY fk\_PerOrders //mysql

ALTER TABLE Orders

DROP CONSTRAINT fk\_PerOrders //sql server/oracle/MS Access

## SQL CHECK 约束

CHECK 约束用于限制列中的值的范围

如果对单个列定义 CHECK 约束，那么该列只允许特定的值。

如果对一个表定义 CHECK 约束，那么此约束会在特定的列中对值进行限制。

创建表时：

CHECK (Id\_P>0) //mysql

Id\_P int NOT NULL CHECK (Id\_P>0)//sql server/oracle/MS Access

CONSTRAINT chk\_Person CHECK (Id\_P>0 AND City='Sandnes')//通用，可以同时定多个check

创建表后：

ALTER TABLE Persons

ADD CHECK (Id\_P>0)//通用

ALTER TABLE Persons

ADD CONSTRAINT chk\_Person CHECK (Id\_P>0 AND City='Sandnes')//通用，加名字

## 撤销 CHECK 约束

ALTER TABLE Persons

DROP CHECK chk\_Person //mysql

ALTER TABLE Persons

DROP CONSTRAINT chk\_Person //sql server/oracle/MS Access

## SQL DEFAULT 约束

City varchar(255) DEFAULT 'Sandnes'

OrderDate date DEFAULT GETDATE() //通用

ALTER TABLE Persons

ALTER City SET DEFAULT 'SANDNES' //mysql

ALTER TABLE Persons

ALTER COLUMN City SET DEFAULT 'SANDNES' //sql server/oracle/MS Access

## 撤销 DEFAULT 约束

ALTER TABLE Persons

ALTER City DROP DEFAULT //mysql

ALTER TABLE Persons

ALTER COLUMN City DROP DEFAULT //sql server/oracle/MS Access

### SQL CREATE INDEX 语法//创建索引

CREATE INDEX index\_name

ON table\_name (column\_name) //创建索引

CREATE UNIQUE INDEX index\_name

ON table\_name (column\_name)//创建唯一的索引

CREATE INDEX PersonIndex

ON Person (LastName DESC)//降序来索引

CREATE INDEX PersonIndex

ON Person (LastName, FirstName)//创建多个索引

## SQL DROP INDEX 语句 //删除索引

ALTER TABLE table\_name DROP INDEX index\_name //mysql

DROP INDEX table\_name.index\_name//sql server

DROP INDEX index\_name//oracle

DROP INDEX index\_name ON table\_name //Access

## SQL DROP语句

DROP TABLE 表名称

DROP DATABASE 数据库名称

TRUNCATE TABLE 表名称 //仅仅删除表内的数据，而不删除表

## ALTER TABLE 语句

ALTER TABLE table\_name

ADD column\_name datatype //增加列

ALTER TABLE table\_name

DROP COLUMN column\_name//删除列

## AUTO INCREMENT 字段

P\_Id int NOT NULL AUTO\_INCREMENT //mysql,从1开始

ALTER TABLE Persons AUTO\_INCREMENT=100//设定从100开始

P\_Id int PRIMARY KEY IDENTITY//sql server

IDENTITY(20,10)

P\_Id int PRIMARY KEY AUTOINCREMENT//Access

AUTOINCREMENT(20,10)

**CREATE SEQUENCE seq\_person**

**MINVALUE 1**

**START WITH 1**

**INCREMENT BY 1**

**CACHE 10 //oracle 从1开始，每次递增1，创建10个缓存**

### SQL CREATE VIEW 语法

CREATE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition

## SQL 日期

## MySQL Date 函数

|  |  |
| --- | --- |
| **函数** | **描述** |
| [NOW()](http://www.w3school.com.cn/sql/func_now.asp) | 返回当前的日期和时间 |
| [CURDATE()](http://www.w3school.com.cn/sql/func_curdate.asp) | 返回当前的日期 |
| [CURTIME()](http://www.w3school.com.cn/sql/func_curtime.asp) | 返回当前的时间 |
| [DATE()](http://www.w3school.com.cn/sql/func_date.asp) | 提取日期或日期/时间表达式的日期部分 |
| [EXTRACT()](http://www.w3school.com.cn/sql/func_extract.asp) | 返回日期/时间按的单独部分 |
| [DATE\_ADD()](http://www.w3school.com.cn/sql/func_date_add.asp) | 给日期添加指定的时间间隔 |
| [DATE\_SUB()](http://www.w3school.com.cn/sql/func_date_sub.asp) | 从日期减去指定的时间间隔 |
| [DATEDIFF()](http://www.w3school.com.cn/sql/func_datediff_mysql.asp) | 返回两个日期之间的天数 |
| [DATE\_FORMAT()](http://www.w3school.com.cn/sql/func_date_format.asp) | 用不同的格式显示日期/时间 |

## SQL Server Date 函数

|  |  |
| --- | --- |
| **函数** | **描述** |
| [GETDATE()](http://www.w3school.com.cn/sql/func_getdate.asp) | 返回当前日期和时间 |
| [DATEPART()](http://www.w3school.com.cn/sql/func_datepart.asp) | 返回日期/时间的单独部分 |
| [DATEADD()](http://www.w3school.com.cn/sql/func_dateadd.asp) | 在日期中添加或减去指定的时间间隔 |
| [DATEDIFF()](http://www.w3school.com.cn/sql/func_datediff.asp) | 返回两个日期之间的时间 |
| [CONVERT()](http://www.w3school.com.cn/sql/func_convert.asp) | 用不同的格式显示日期/时间 |

**SQL Date 数据类型**

MySQL 使用下列数据类型在数据库中存储日期或日期/时间值：

* DATE - 格式 YYYY-MM-DD
* DATETIME - 格式: YYYY-MM-DD HH:MM:SS
* TIMESTAMP - 格式: YYYY-MM-DD HH:MM:SS
* YEAR - 格式 YYYY 或 YY

SQL Server 使用下列数据类型在数据库中存储日期或日期/时间值：

* DATE - 格式 YYYY-MM-DD
* DATETIME - 格式: YYYY-MM-DD HH:MM:SS
* SMALLDATETIME - 格式: YYYY-MM-DD HH:MM:SS
* TIMESTAMP - 格式: 唯一的数字

**http://www.w3school.com.cn/sql/sql\_dates.asp**

## SQL NULL 值

无法比较 NULL 和 0；它们是不等价的

## SQL ISNULL()、NVL()、IFNULL() 和 COALESCE() 函数