基本数据类型

n：最多字符数量

p（有效位数）：字段的最大长度

s（标度）：小数点后面的位数

|  |  |  |  |
| --- | --- | --- | --- |
|  | CHARACTER(n) | | 通常用空格填充数量不足的字符。 |
|  | CHARACTER VARYING(n) | |  |
|  | BLOB | |  |
|  | TEXT | |  |
| 数值类型 | BIT(n) | 通用的数值类型NUMERIC(p) |  |
| BIT VARYING(n) |  |
| DECIMAL(p,s) |  |
| INTEGER |  |
| SMALLINT |  |
| BIGINT |  |
| FLOAT(p,s) |  |
| DOUBLE PRECISION(p,s) |  |
| REAL(s) |  |
| 日期和时间类型 | DATE | |  |
| TIME | |  |
| DATETIME | |  |
| TIMESTAMP | |  |
| 布尔值 |  | | 取值： TRUE、FALSE、NULL。 |
| NULL数据类型 |  | | NULL |

自定义类型

|  |  |
| --- | --- |
| CREATE TYPE **数据类型名** AS OBJECT  (**成员名** **数据类型**(,  **成员名** **数据类型**)\*); |  |

域：能够被使用的有效数据类型的集合。

|  |  |
| --- | --- |
| 创建域  CREATE DOMAIN **域名** AS **数据类型**; |  |
| 给域添加约束  ALTER DOMAIN **域名**  ADD CONSTRAINT **约束名**  CHECK(**关于VALUE的表达式**); | 1、**关于VALUE的表达式**的数据类型是布尔值。 |

|  |  |
| --- | --- |
| from 子句  FROM **表名**(,**表名**)\* |  |
| where子句  WHERE **关于列名**(,**列名**)\***的表达式** | 1、**关于列名**(,**列名**)\***的表达式**的数据类型是布尔值。 |

数据定义语言（DDL）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CREATE TABLE语句  CREATE TABLE **表名**  (**列名** **数据类型**( | NULL| NOT NULL)(,  **列名** **数据类型**( | NULL| NOT NULL))\*);  CREATE TABLE 表名;  AS **记录集合**; | 1、   |  |  | | --- | --- | | 无 | 字段可以为NULL。 | | NULL | | NOT NULL | 字段不可以为NULL。 |   2、例：   |  |  | | --- | --- | | MySQL、Oracle | CREATE TABLE **表名**  AS **SELECT语句去掉末尾的分号**; | | Microsoft SQL Server | SELECT (\*|**列名**(,**列名**)\*)  INTO **表名**  FROM **表名**(  **where子句**)?; | |
| ALTER TABLE命令  ALTER TABLE **表名**  ( modify)?( column **列名**)?( **数据类型**|NULL|NOT NULL)?  ( | restrict| cascade)  ( drop)?( constriant **约束名**)?  ( add)?( column)? **列名** **数据类型**( | NULL| NOT NULL); |  |
| DROP TABLE **表名**( | restrict| cascade) | 1、   |  |  | | --- | --- | | 无 |  | | restrict | 若表被视图或约束所引用，返回一个错误。 | | cascade | 若表被视图或约束所引用，删除所有引用视图和约束。 | |

完整性约束

|  |  |
| --- | --- |
| 主键约束  在CREATE TABLE语句中，在( | NULL| NOT NULL)后插入 PRIMARY KEY。  或  在CREATE TABLE语句中，在);前插入,PRIMARY KEY(**列名**(,**列名**)\*)。  或  ALTER TABLE **表名**  ADD CONSTRAINT **约束名** PRIMARY KEY(**列名**(,**列名**)\*); |  |
| 唯一性约束  在CREATE TABLE语句中，在( | NULL| NOT NULL)后插入 UNIQUE。 |  |
| 外键约束  在CREATE TABLE语句中，在);前插入,CONSTRAINT **约束名** FOREIGN KEY(**列名**) REFERENCES **表名**(**列名**)。  或  ALTER TABLE **表名**  ADD CONSTRAINT **约束名** FOREIGN KEY(**列名**) REFERENCES **表名**(**列名**); |  |
| NOT NULL约束 |  |
| 检查约束  在CREATE TABLE语句中，在);前插入,CONSTRAINT **约束名** CHECK(**关于列名的表达式**)。 | 1、**关于列名的表达式**的数据类型是布尔值。 |
| 去除约束  ALTER TABLE **表名**  DROP CONSTRAINT **约束名**; | 1、例：   |  |  | | --- | --- | | MySQL | ALTER TABLE **表名**  DROP PRIMAARY KEY; | |

数据操作语言（DML）

|  |  |
| --- | --- |
| INSERT INTO **表名**  VALUES(**值**(,**值**)\*);  INSERT INTO **表名**(**列名**(,**列名**)\*)  VALUES(**值**(,**值**)\*);  INSERT INTO **表名**  **SELECT语句去掉末尾的分号**; | 1、没有出现的列没有NOT NULL约束。  2、没有出现的列插入NULL。 |
| UPDATE **表名**  SET **列名**=**值**(,**列名**=**值**)\*(  **where子句**)?; |  |
| DELETE FROM **表名**(  **where子句**)?; |  |

数据查询语言（DQL）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT (\*|(|ALL |DISTINCT )**列名**(,**列名**)\*)  **from子句**(  **where子句**)?(  ORDER BY (**列名**|**列号**)(| ASC| DESC)(,(**列名**|**列号**)(| ASC| DESC))\*)?;   |  |  | | --- | --- | | 统计记录数量 | SELECT COUNT(\*)  FROM **表名**; | | 统计字段的值的数量 | SELECT COUNT(**列名**)  FROM **表名**; | | 统计列的值的集合的基数 | SELECT COUNT(DISTINCT **列名**)  FROM **表名**; | | 1、   |  |  | | --- | --- | | 无 | 不去重 | | ALL | | DISTINCT | 去重 |   2、   |  |  | | --- | --- | | 无 | 升序 | | ASC | | DESC | 降序 |   3、列号是列在SELECT后列列表里的位置，从1开始。 |
| 给字段创建临时别名  在SELECT语句中，在**列名**后插入 **别名**。 |  |

数据控制语言（DCL）

数据管理命令

事务控制命令

事务：若干个DML语句（INSERT、UPDATE、DELETE）。

|  |  |
| --- | --- |
| COMMIT( WORK)?; | 把事务所做的修改保存到数据库，清空临时回退区域。 |
| ROLLBACK( WORK)?; | 撤销上一个COMMIT或ROLLBACK命令之后的事务。 |
| ROLLBACK TO **保存点名**; | 回退到一个保存点。 |
| SAVEPOINT **保存点名**; | 1、保存点名互不相同。  2、例：   |  |  | | --- | --- | | Microsoft SQL Server | SAVE TRANSACTION**保存点名**; | |
| RELESE SAVEPOINT **保存点名**; | 删除保存点。 |
| SET TRANSACTION (READ WRITE|READ ONLY); | 指定事务的特性。READ ONLY避免对数据库对象进行加锁，可以提高事务的性能。 |