

数据库操作

```
CREATE DATABASE My_Database | 创建数据库My_Database.  
DROP DATABASE My_Database | 删除数据库My_Database.
```

表操作

```
CREATE TABLE Person  
(Name varchar,Hobby varchar,  
Address varchar,Age int) | 创建表 Person(含有四列).  
DROP TABLE Person | 删除表Person.  
ALTER TABLE Person  
ADD Sex char(6) | 在表中再添加Sex列.  
ALTER TABLE Person  
DROP Sex char(6) | 删除表中Sex列.
```

索引操作

```
CREATE INDEX PersonIndex  
ON Person (Name, Hobby) | 创建索引.  
CREATE UNIQUE INDEX PersonIndex  
ON Person (Name) | 创建唯一索引.  
DROP INDEX Person.PersonIndex | 删除索引.
```

数据操作

```
INSERT INTO Person  
VALUES('XiaoXiao','football') | 在表中插入新的行.  
UPDATE Person SET Address='W house'  
WHERE Name='XiaoMing' | 更新行中的一个或多个列.  
DELETE FROM Person  
WHERE Name='XiaoMing' | 删除表中的行.  
TRUNCATE TABLE Person | 删除表中所有行.
```

```
INSERT INTO Person (Name,Age)  
VALUES ('Andy',30) | 插入Name为Andy, Age为30的数据.  
ALTER TABLE Person MODIFY Name VARCHAR(13) NULL | 修改Person表Name列.  
UPDATE Person SET Age= Age+1 WHERE  
Hobby='football' | 将Hobby为football的人Age加1.
```

SELECT操作

```
SELECT Name, Hobby FROM Person | 从表中选择Name, Hobby两列的数据.  
SELECT * FROM Person | 选择Person表中所有数据.  
SELECT * FROM Person  
WHERE Sex='female' | 从表中选择性别为female的数据.  
SELECT * FROM Person WHERE  
(Hobby='football' OR Hobby='movie')  
AND Age>18 | 从表中选择爱好是football或movie, 且年龄大于18的数据.  
SELECT * FROM Person  
WHERE Name LIKE '%m' | 从表中选择Name以m结尾的人.  
SELECT * FROM Person  
WHERE Name LIKE 'M%' | 从表中选择Name以M开头的人.  
SELECT * FROM Person  
WHERE Name LIKE '%js%' | 从表中选择Name含有js的人.  
SELECT Age FROM Person  
ORDER BY Name DESC | 从表中选择Age列, 且按Name降序排列.  
SELECT Hobby,AVG(Age) FROM Person  
GROUP BY Hobby | 从表中选择Hobby, Age列, 并按Hobby进行组合, 得到组均值.
```

别名操作

```
SELECT Name AS Xname, Hobby AS Xhobby  
FROM Person | 设置Name列别名为XName, Hobby列别名为Xhobby.  
SELECT Name, Hobby FROM Person AS POP | 设置Person表别名为POP.
```

JOIN操作



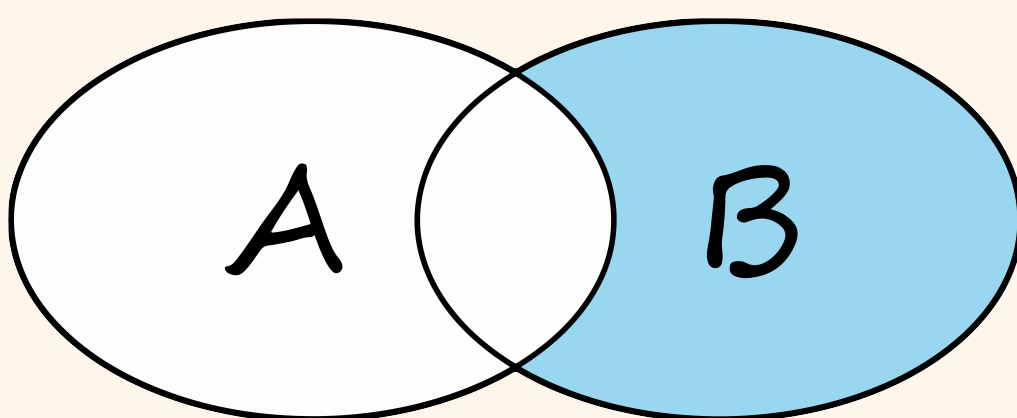
SELECT * FROM A INNER JOIN B ON A.KEY = B.KEY | 内连接.



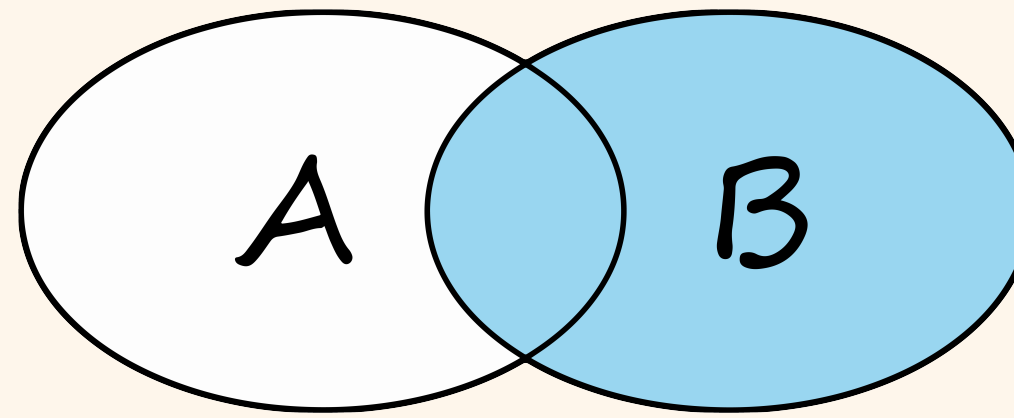
SELECT * FROM A LEFT JOIN B ON A.KEY = B.KEY
WHERE B.KEY IS NULL | 左连接（不含重叠部分）.



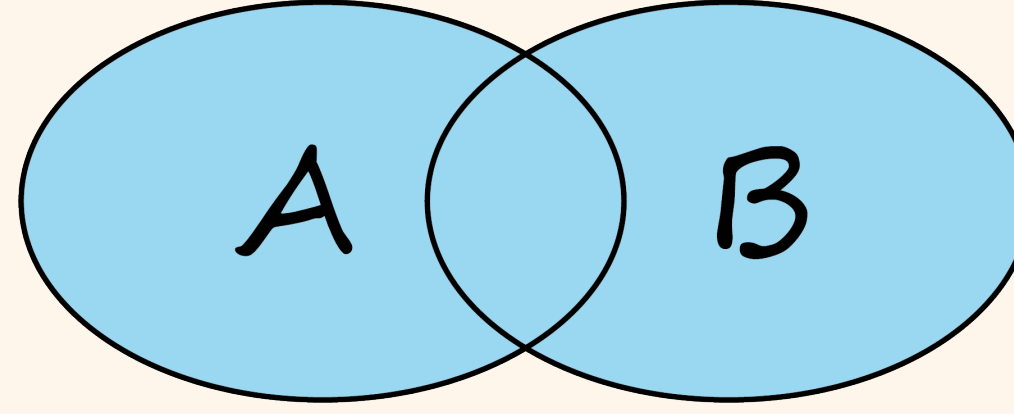
SELECT * FROM A LEFT JOIN B ON A.KEY = B.KEY | 左连接.



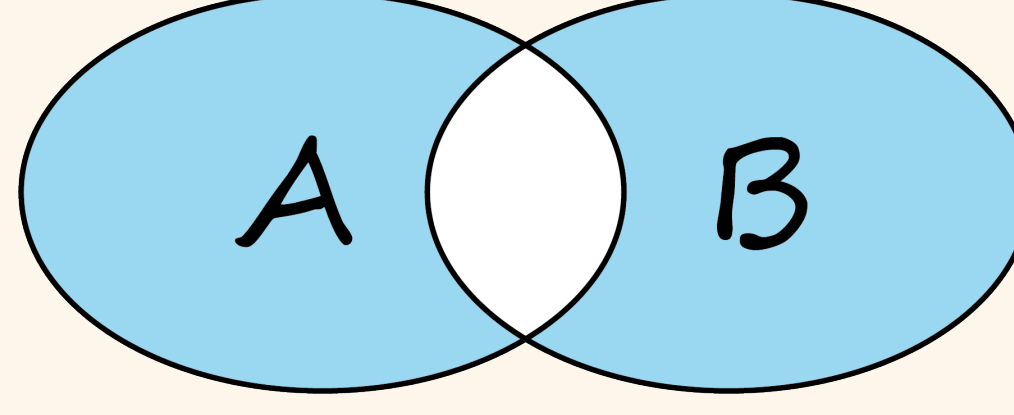
SELECT * FROM A RIGHT JOIN B ON A.KEY = B.KEY
WHERE B.KEY IS NULL | 右连接（不含重叠部分）.



SELECT * FROM A RIGHT JOIN B ON A.KEY = B.KEY | 右连接.



SELECT * FROM A FULL JOIN B ON A.KEY = B.KEY | 全连接.



SELECT * FROM A FULL JOIN B ON A.KEY = B.KEY
WHERE A.KEY IS NULL OR B.KEY IS NULL
全连接（不含重叠部分）.

UNION操作

```
SELECT Name FROM Person1 UNION  
SELECT Name FROM Person2 | 从表Person1和表Person2选取所有不同的Name值.  
SELECT Name FROM Person1 UNION  
ALL SELECT Name FROM Person2 | 从表Person1和表Person2选取所有的Name值.
```

常用统计函数

功能	描述
AVG(列)	返回列的平均值
COUNT(列)	返回列的行数(不含NULL值)
MAX(列)	返回列的最大值
MIN(列)	返回列的最小值
SUM(列)	返回列的总和