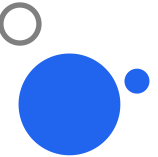


# 数据库基础

王立民



# 目录

**1.数据库的认识**

**2.主流数据库管理系统对比**

**3.数据库结构**

**4.SQL约束**

**5.SQL语句**

**6.SQL语句复杂查询**

**7.SQL语句多表查询**

内部资料

# 数据库的认识



# 数据的重要性

数据是最重要的原材料

数据是一切的基础

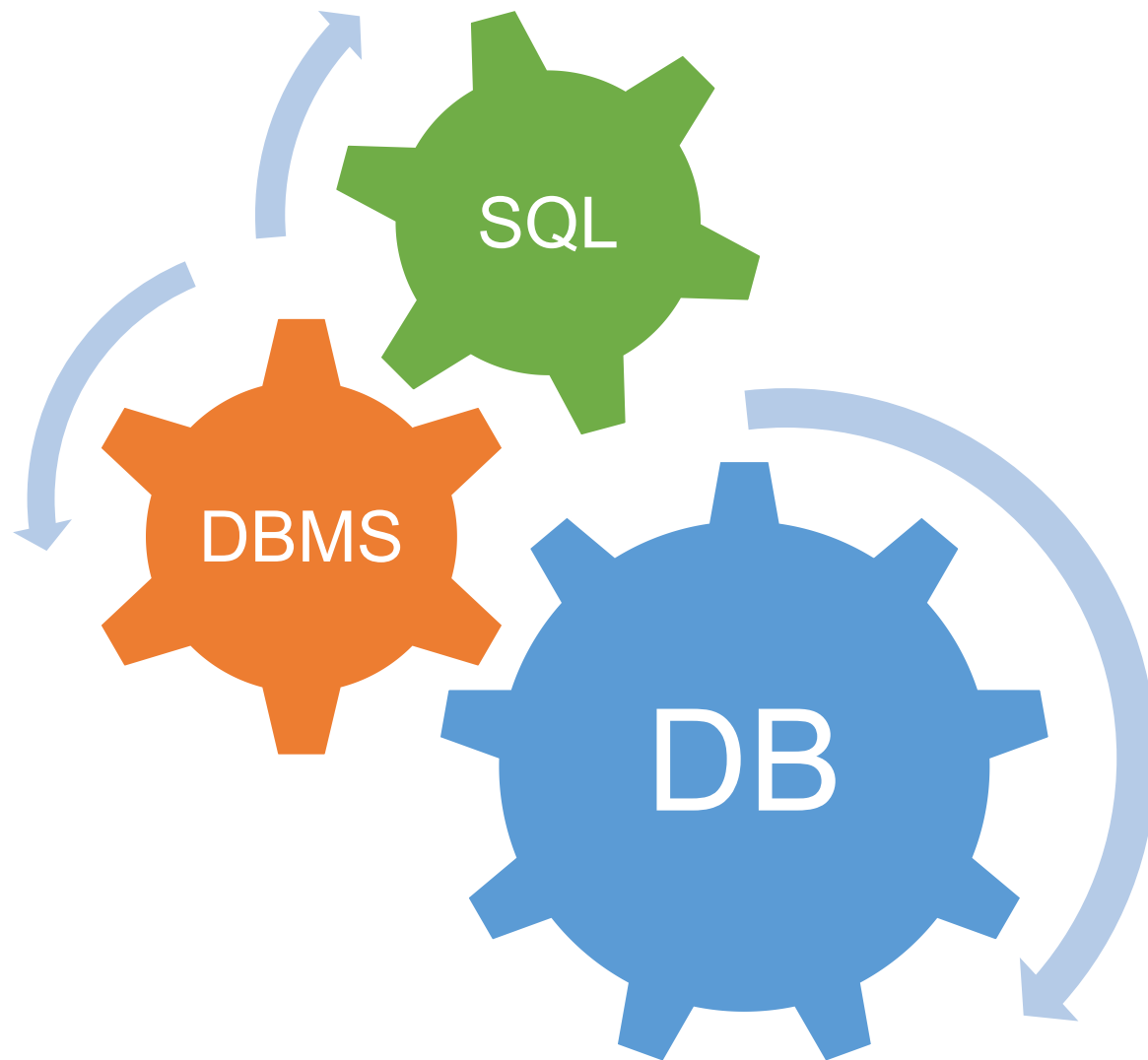




SQL 结构化查询语言

DBMS 数据库管理系统

DB 数据库





# 主流数据库管理系统对比

# 数据库管理系统对比



体积小、速度快、  
适配性好、灵活性  
好、支持多线程、  
支持大型数据库

不支持热备份  
没有存储过程

适合分布式组织的  
可伸缩性  
与同类软件的关联  
性和集成性较好

只能在windows  
上运行  
多用户数和数据卷  
伸缩性有限  
多用户时性能不佳

良好的兼容性、可  
移植性、开放性、  
安全性  
处理速度快，备份  
和负载处理优秀

对硬件的要求很高  
价格比较昂贵  
管理维护操作复杂  
日志和索引硬件开  
销大

优化了性能，增强  
了KILL命令，优化  
了磁盘访问，能够  
适配更多的引擎

版本控制较为混乱  
集群部署不够稳定

**OceanBase**  
A Powerful Database System

**TDSQL**



 **达梦数据库**



内部资料

# 数据库结构





think_classes	16.0 KiB
think_courses	160.0 KiB
think_groups	16.0 KiB
think_homework	80.0 KiB
think_key	32.0 KiB
think_labs	64.0 KiB
think_labserver	16.0 KiB
think_lessons	64.0 KiB
think_liveinfo	48.0 KiB
think_liverecord	32.0 KiB
think_question	16.0 KiB
think_register	48.0 KiB
think_request	16.0 KiB
think_reservation	192.0 KiB
think_resources	16.0 KiB
think_user	288.0 KiB
think_usercourseste...	16.0 KiB
think_usertime	16.0 KiB

	ID INT	NAME VARCHAR(50)	SEX VARCHAR(50)	AGE TINYINT	REMOVED TINYINT	CREATETIME DATETIME(6)
1	1	张三	男	12	0	2021-05-22 22:04:43.383000
2	2	李四	男	14	0	2021-05-22 22:04:43.383000
3	3	李玲	女	13	0	2021-05-22 22:04:43.383000
4	4	王五	男	14	0	2021-05-22 22:04:43.383000
5	5	赵华	女	15	0	2021-05-22 22:04:43.383000

# 数据库结构

字段构成记录    记录构成表    表构成数据库

记录是操作的单位，也是展示的单位

	ID INT	STUDENTID INT	COURSENAME VARCHAR(50)	SCORE INT	REMOVED TINYINT	CREATETIME DATETIME(6)
1	1	1	语文	85	0	2021-05-22 22:04:43.383000
2	2	2	数学	60	0	2021-05-22 22:04:43.383000
3	3	4	政治	90	0	2021-05-22 22:04:43.383000
4	4	1	英语	70	0	2021-05-22 22:04:43.383000
5	5	3	数学	90	0	2021-05-22 22:04:43.383000
6	6	5	语文	80	0	2021-05-22 22:04:43.383000
7	7	5	政治	60	0	2021-05-22 22:04:43.383000
8	8	3	历史	75	0	2021-05-22 22:04:43.383000

内部资料

# SQL约束





# 简单的约束

NOT NULL – 指示某列不能存储 NULL 值。

UNIQUE – 保证某列的每行必须有唯一的值。

PRIMARY KEY – NOT NULL 和 UNIQUE 的结合。

确保某列有唯一标识，有助于更容易更快速地找到表中的一个特定的记录。

FOREIGN KEY – 保证一个表中的数据匹配另一个表中的值的参照完整性。

DEFAULT – 规定没有给列赋值时的默认值。



	ID INT	NAME VARCHAR(50)	SEX VARCHAR(50)	AGE TINYINT	REMOVED TINYINT	CREATETIME DATETIME(6)
1	1	张三	男	12	0	2021-05-22 22:04:43.383000
2	2	李四	男	14	0	2021-05-22 22:04:43.383000
3	3	李玲	女	13	0	2021-05-22 22:04:43.383000
4	4	王五	男	14	0	2021-05-22 22:04:43.383000
5	5	赵华	女	15	0	2021-05-22 22:04:43.383000

学号 ID  
名字 NAME  
性别 SEX  
年龄 AGE

内部资料

# SQL语句



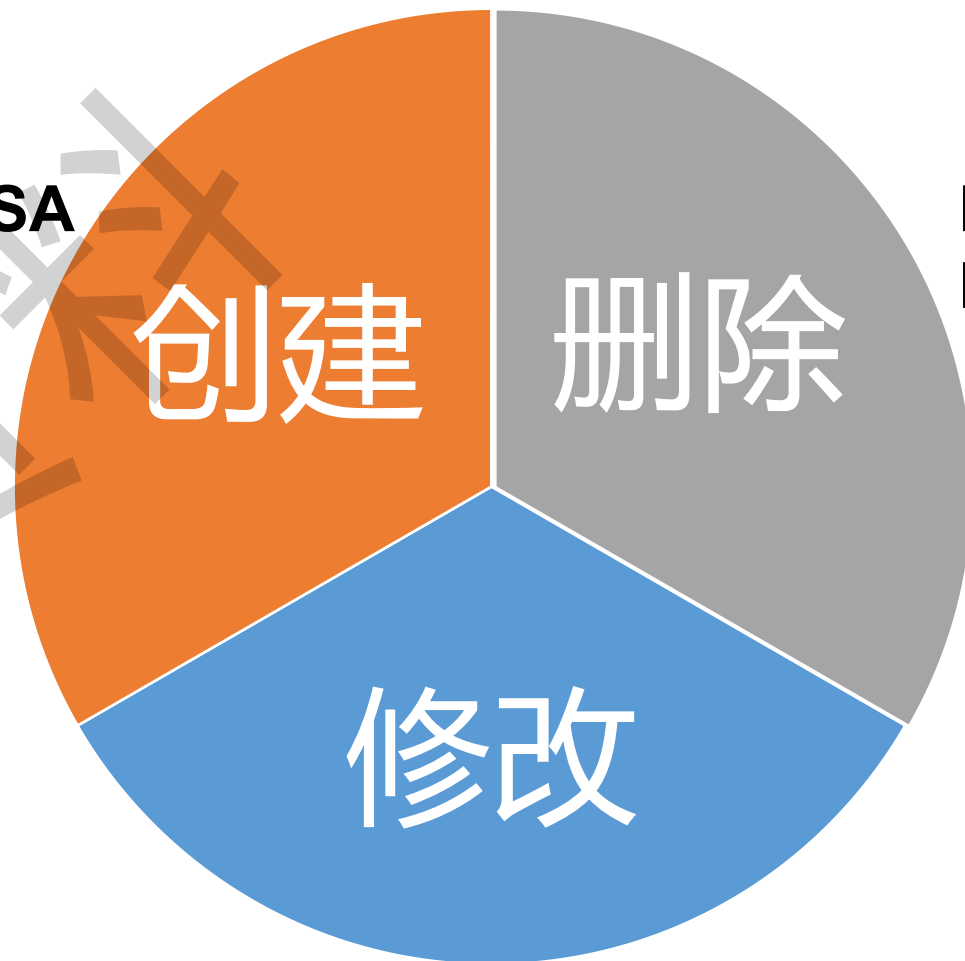
SELECT \* FROM "school"."student" ;

	ID INT	NAME VARCHAR(50)	SEX VARCHAR(50)	AGE TINYINT	REMOVED TINYINT	CREATETIME DATETIME(6)
1	1	张三	男	12	0	2021-05-22 22:04:43.383000
2	2	李四	男	14	0	2021-05-22 22:04:43.383000
3	3	李玲	女	13	0	2021-05-22 22:04:43.383000
4	4	王五	男	14	0	2021-05-22 22:04:43.383000
5	5	赵华	女	15	0	2021-05-22 22:04:43.383000



# 数据定义语言 DDL

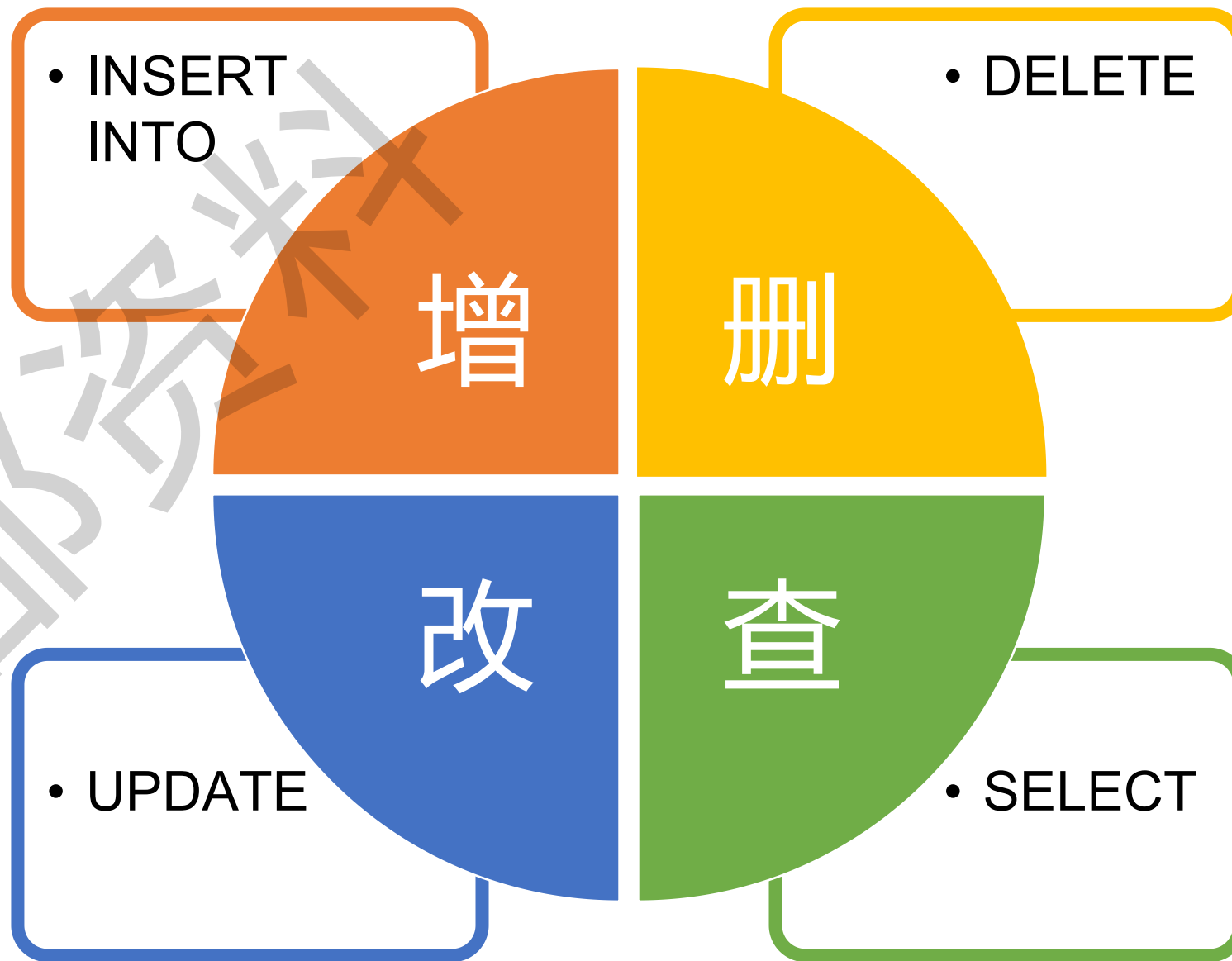
**CREATE DATABASESA  
CREATE TABLE**



**DROP DATABASESA  
DROP TABLE**

**ALTER DATABASESA  
ALTER TABLE**

内部资料







**INSERT INTO** "school"."student" (NAME,SEX,AGE) VALUES ('赵六','男',12);

操作关键词

库名

表名

字段名

组成词

值

**DELETE** FROM "school"."student" WHERE NAME='赵六';

操作关键词

组成词

库名

表名

组成词

字段名

值

**UPDATE** "school"."student" SET NAME='赵六' WHERE NAME='张三';

操作关键词

库名

表名

组成词

字段名

值

组成词

字段名

值

**SELECT** \* FROM "school"."student" ;

操作关键词

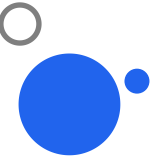
字段名

组成词

库名

表名

# SQL语句复杂查询



# 模糊查询和过滤查询

```
SELECT * FROM "school"."student" WHERE NAME like '%王%';
```

```
SELECT ID,NAME FROM "school"."student" LIMIT 3;
```

```
SELECT * FROM "school"."student" WHERE AGE>10 and AGE < 12;
```

```
SELECT NAME FROM "school"."student" WHERE AGE in (12,14);
```

查询带有“理”的课程名称

显示分数在80以上的所有信息

查询男生或者女生的年龄和名字

查询分数为90和70的所有信息



# 排序查询

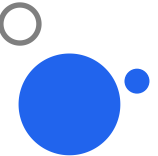
学生年龄排序

```
select * from "school"."student" order by AGE ASC|DESC
```

成绩由高到低排序

```
select * from "school"."mark" order by SCORE DESC
```

内部资料



## 分组查询

查询每个学生总分数

```
SELECT STUDENTID, SUM(SCORE) AS scores FROM "school"."mark" GROUP BY STUDENTID;
```

查询学生选课的数量

```
SELECT STUDENTID, COUNT(COURSENAME) AS nums FROM "school"."mark" GROUP BY STUDENTID;
```

查询每个课程的总分数

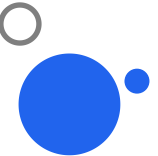
```
SELECT COURSENAME, SUM(SCORE) AS scores FROM "school"."mark" GROUP BY COURSENAME;
```

查询每个年龄段的人数

```
SELECT AGE, COUNT(ID) AS nums FROM "school"."student" GROUP BY AGE;
```



# SQL语句多表查询



查询选了语文课程的学生信息

```
select * from "school"."student"  
where ID in  
(select STUDENTID from "school"."mark" where COURSENAME='语文')
```

查询没有选数学课的学生信息

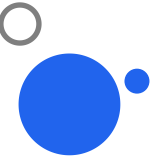
```
select * from "school"."student"  
where ID not in  
(select STUDENTID from "school"."mark" where COURSENAME='数学')
```

查询没有人选的课程

①查出所有被选的课程名

②将①得到的结果作为筛选条件

```
select NAME from "school"."course" where NAME not in (select  
COURSENAME from "school"."mark")
```



展示学生ID、学生姓名、选择的课程以及成绩 并排序

```
SELECT "school"."student".ID,"school"."student".NAME,  
"school"."mark".COURSENAME,"school"."mark".SCORE  
FROM "school"."student"  
LEFT JOIN "school"."mark"  
ON "school"."student".ID="school"."mark".STUDENTID
```

查询学生信息及选择的课程数量，显示学生姓名、性别、年龄和课程数量

```
select "school"."student".NAME,"school"."student".SEX,  
"school"."student".AGE,a.COURSENUMS  
from "school"."student"  
LEFT JOIN  
(  
SELECT STUDENTID, COUNT(COURSENAME) AS coursenums  
FROM "school"."mark"  
GROUP BY STUDENTID  
) AS a  
ON "school"."student".ID=a.STUDENTID;
```

查询学生的总分数，显示学生ID、姓名、年龄、总分

```
select "school"."student".ID,"school"."student".NAME,"school"."student".AGE,a.Scores
from "school"."student"
LEFT JOIN
(
SELECT STUDENTID,SUM(SCORE) AS scores
FROM "school"."mark"
GROUP BY STUDENTID
) AS a
ON "school"."student".ID=a.STUDENTID
```

内部资料

百度智能云 百度智能云

# THANK YOU

CLOUD.BAIDU.COM

ABCXUEYUAN.BAIDU.COM