

西北工业大学

Northwestern Polytechnical University

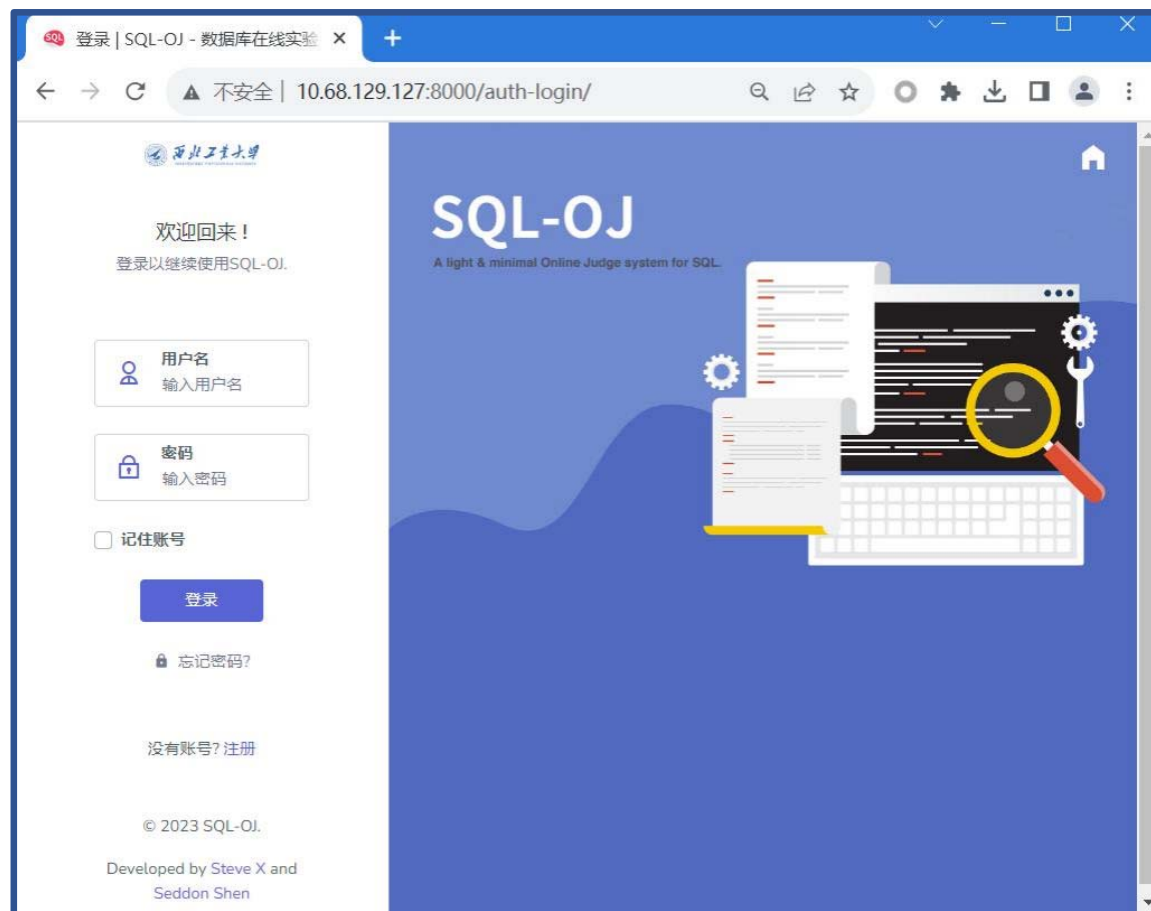
数据库系统原理

Database System

第三章 关系数据库标准语言SQL

李宁

2023.09



班级识别码: AB01

01

账号信息

02

设置密码

03

学校信息

学校全称

西北工业大学

班级识别码

联系老师获取

真实姓名

张三

学号

2021000000

完成注册

第1章 绪论

- 自学内容：DBS各类人员职责、层次模型、网状模型(共约20页PPT)
- 拓展阅读：
 1. 数据库图灵奖得主相关介绍、博客、漫画公众号文章等
 2. DB-GPT、SQL-GPT等相关内容
- 书面作业（100分）：
 1. 必做（100分）：第1章课后：第3, 14, 15题
- 实践作业（选做，平时成绩+2）
 1. 编程：复现教材P9扩展阅读中文件系统编程示例

第2章 关系代数

- 书面作业（100分）
 1. 必做（35分）：第2章课后：第6, 8题（仅用关系代数、ALPHA语言）
 2. 必做（65分）：补充作业：见PPT，两道关系代数大题（对应英文版教材6.14, 6.15）

新的国际经济、政治、科技形势

- 国际形势风云变幻
- 中美贸易摩擦，美国对中国制裁集中在高科技领域
- 科技战--高新技术“突围战”



引用：主题报告《国产数据库发展之我见》，李战怀教授，2021.6，SIGMOD2021，国产数据库发展高峰论坛

中国数据库发展历程

受益于市场需求和技术沉淀，进入百花齐放的快速发展期

从20世纪80年代起，我国数据库市场开始逐步发展起来。经历了初始的技术萌芽期和国外厂商垄断期，21世纪初，基于863计划、核高基计划等国家政策支持，一批拥有高校背景的国产厂商成立，打破了Oracle和IBM一统天下的格局。2010s，随着市场需求的增长、技术的沉淀，一批云厂商和新兴独立厂商开始提供数据库产品。近年来，借助国产化热潮，许多软件厂商、集成商、运营商等也开始入局，发展自己的数据库能力。

中国数据库发展历程总览



来源：公开资料，专家访谈，艾瑞咨询研究院自主研究及绘制。

©2021.5 iResearch Inc.

www.iresearch.com.cn

42

引自：艾瑞咨询技术报告

中国数据库产业图谱

中国数据库产业图谱

国内厂商

传统厂商



初创厂商



云厂商



跨界厂商



国外厂商

商业数据库



开源数据库



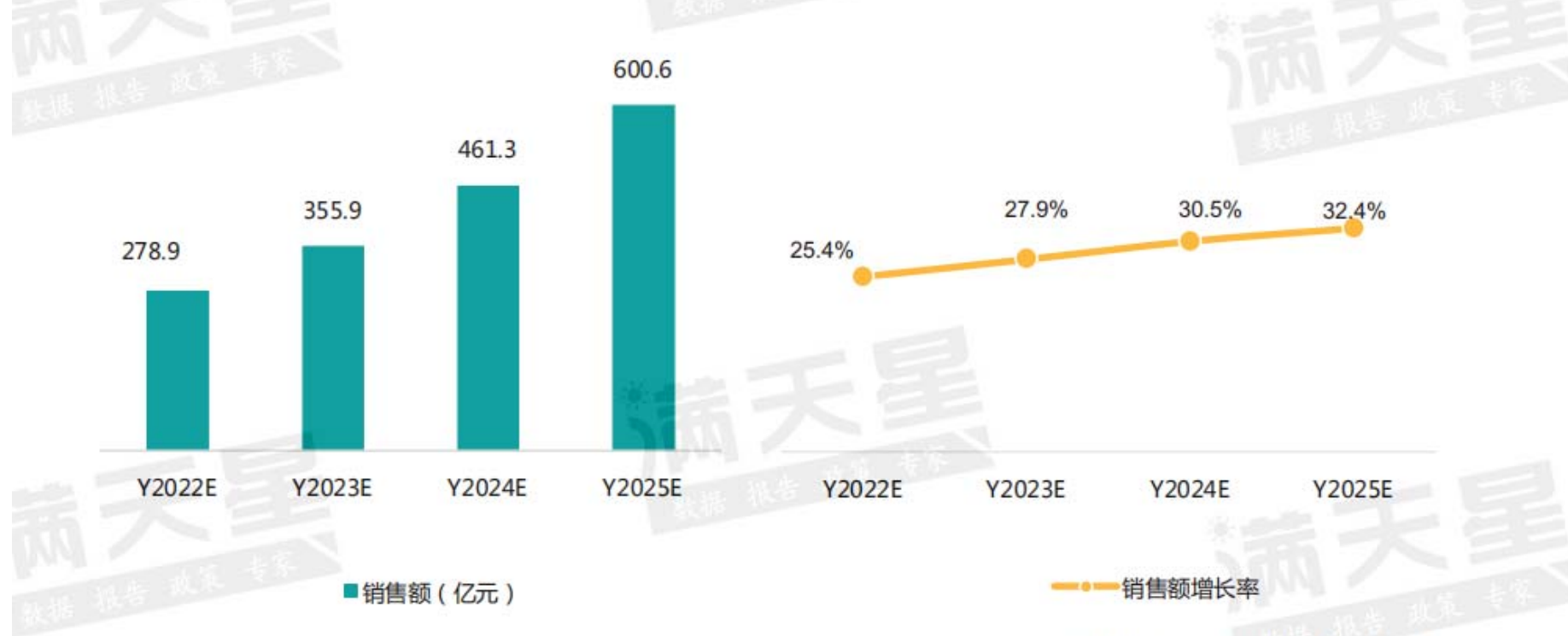
来源：公开资料，专家访谈，艾瑞咨询研究院自主研究及绘制。
注：图谱仅展示部分典型厂商，图谱展示顺序不代表市场份额排名。

©2021.5 iResearch Inc.

www.iresearch.com.cn

43

图 20 2022—2025 年中国数据库市场规模预测

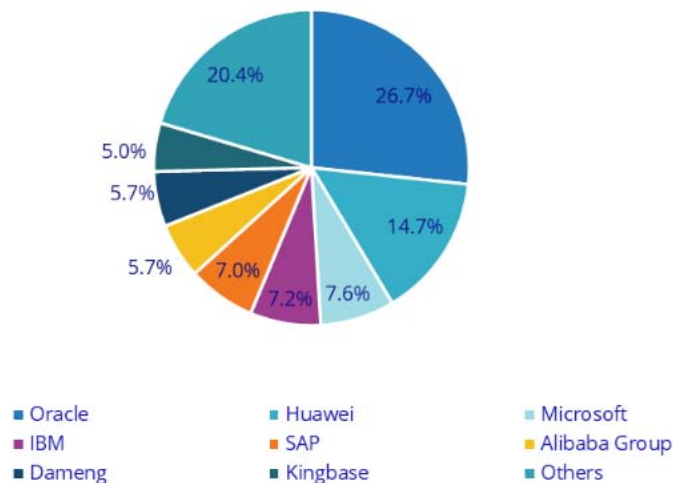


数据来源：赛迪顾问，2022.5



中国关系型数据库软件市场厂商份额，2021H1

—— 本地部署模式

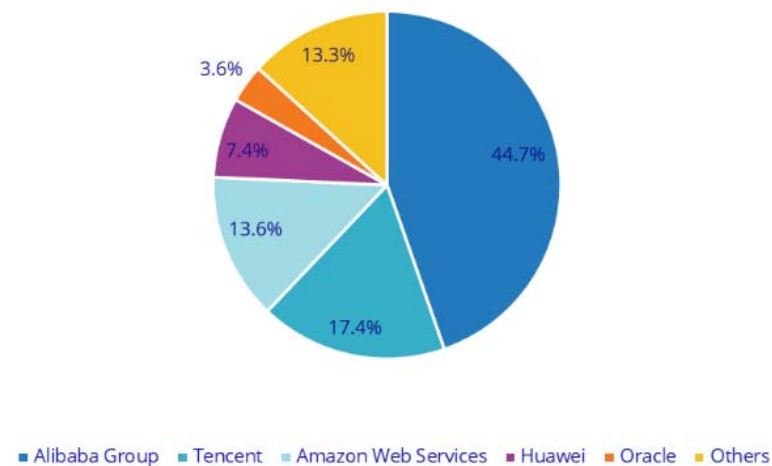


来源：IDC中国，2022



中国关系型数据库软件市场厂商份额，2021H1

—— 公有云模式



来源：IDC中国，2022

➤ 关系代数

- 投影, 选择, 连接, 除、重命名、聚集操作
- 关系代数的综合应用
- 关系代数的用途 (查询实现与查询优化)

➤ 关系演算: 元组关系演算语言 ALPHA

- 检索语句: GET
- 更新语句: PUT, HOLD, UPDATE, DELETE, DROP

要求：求至少用了供应商S1所供应的全部零件的工程号。

关系代数思路：

1. 查询供应商S1所供应的全部零件号。
2. 查询某工程中使用的零件号包含S1供应商供应的所有零件号。

可能两种含义（任选一种解答即可，第二种相对复杂）

含义1：使用了S1供应的零件P1和P2（满足该类型）即可；

含义2：使用的P1和P2必须是S1供应的（其他供应商不可以）

关系演算思路：

1. 对于任意的供应商S1所供应的零件号x，某项目都使用了（存在该项目使用该零件的记录）零件x，则这样的项目是我们期望得到的结果。

关系 R 包含属性{A1,A2,A3,A4,A5}，其中(A1,A2)组合为 R 的候选码，则下面的说法**正确**的是：

- ☐ A A1 或者有可能单独成为 R 的候选码。
- ☐ B (A1,A2,A3)必然也是 R 的候选码
- ☒ C R 中绝不可能出现两个在 A1 和 A2 上取值完全相同的元组
- ☐ D R 的所有元组中，A1 或者 A2 的值都是不能重复的

提交

3.1.SQL概述

3.2.学生-课程数据库

3.3.数据定义

3.4.数据查询

3.5.数据更新

3.6 空值的处理

3.7.视图



➤重点

- ✓ 数据定义DDL
- ✓ 数据查询DQL
- ✓ 数据更新DML

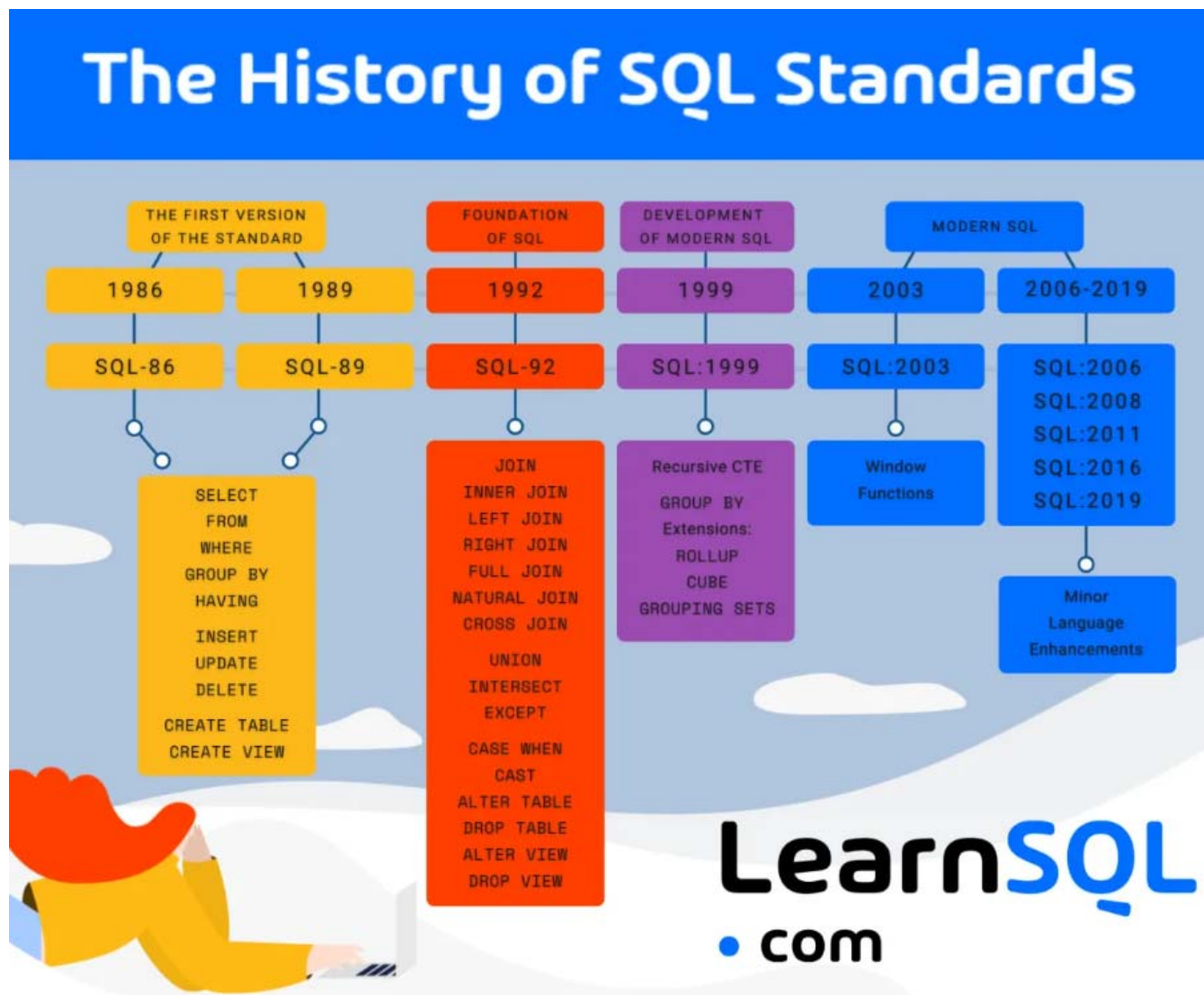
数据控制语言DCL
(第四章 安全性)

➤难点

- ✓ 数据查询



- SQL (Structured Query Language) 发展过程





SQL2023含
11个部分

ISO/IEC 9075-1 信息技术 – 数据库语言 – SQL – 第 1 部分：框架 (SQL/框架)

ISO/IEC 9075-2 信息技术 – 数据库语言 – SQL – 第 2 部分：基本原则 (SQL/基本原则)

ISO/IEC 9075-3 信息技术 – 数据库语言 – SQL – 第 3 部分：调用级接口 (SQL/CLI)

ISO/IEC 9075-4 信息技术 – 数据库语言 – SQL – 第 4 部分：持久存储模块 (SQL/PSM)

ISO/IEC 9075-9 信息技术 – 数据库语言 – SQL – 第 9 部分：外部数据管理 (SQL/MED)

ISO/IEC 9075-10 信息技术 – 数据库语言 – SQL – 第10 部分：对象语言绑定 (SQL/OLB)

ISO/IEC 9075-11 信息技术 – 数据库语言 – SQL – 第 11 部分：信息与定义概要 (SQL/Schemata)

ISO/IEC 9075-13 信息技术 – 数据库语言 – SQL – 第 13 部分：使用 Java 编程语言的 SQL 程序与类型 (SQL/JRT)

ISO/IEC 9075-14 信息技术 – 数据库语言 – SQL – 第 14 部分：XML 相关规范 (SQL/XML)

ISO/IEC 9075-15 信息技术 – 数据库语言 – SQL – 第 15 部分：多维数组 (SQL/MDA)

ISO/IEC 9075-16 信息技术 – 数据库语言 – SQL – 第 16 部分：属性图查询 (SQL/PGQ)

标准SQL2023 (2023年6月1日发布)与前一版相比，最大的新增是哪部分？

A 16

B 15

C 13

提交

SQL语言的特点

- (1) 综合统一 (DDL, DML, DCL)
- (2) 高度非过程化
- (3) 面向集合的操作方式
- (4) 以同一种语法结构提供两种使用方式

SQL语言既是自含式语言，又是嵌入式语言

- (5) 语言简捷，易学易用。

SQL语言所使用的动词

SQL功能	动词
数据定义DDL	CREATE, DROP, ALTER
数据查询DQL	SELECT
数据更新DML	INSERT, UPDATE, DELETE
数据控制DCL	GRANT, REVOKE

注意通常SQL语言中不区分大小写！
部分数据库提供参数可配置

3.1.SQL概述

3.2.学生-课程数据库

3.3.数据定义

3.4.数据查询

3.5.数据更新

3.6 空值的处理

3.7.视图



3.2 学生-课程示例数据库



综合运算例题（本小节的所有查询均用到以下关系实例）

Student

sno	sname	Sgender	smajor	sbirthdate
801	张三	女	01	19
802	李四	男	01	20
803	王五	男	01	20
804	赵六	女	02	20
805	钱七	男	02	19

SC

sno	cno	grade
801	C4	92
801	C3	78
801	C2	85
802	C3	82
802	C4	90
803	C4	88

Course

cno	cname	credit	cpno
C1	数据库	3.5	C2
C2	数据结构	4	C4
C3	编译原理	4	C6
C4	C语言	3	

关系模式：

S(sno, sname, sgender, smajor, sbirthdate)

C(cno, cname, credit, cpno)

SC(sno, cno, grade)

3.1.SQL概述

3.2.学生-课程数据库

3.3.数据定义

3.4.数据查询

3.5.数据更新

3.6 空值的处理

3.7.视图



□ MySQL 的数据库结构

名称	解释
数据库集群cluster	可包含1个或多个数据库
数据库database	可包含1个或多个基本表
模式schema	= database (mysql中等价, 但其他DBMS可能不相同)
表table	存放数据的基本表
表空间tablespace	<p>MySQL的Innodb:</p> <ul style="list-style-type: none">- 共享表空间: 所有数据保存在一个单独的表空间里面, 而这个表空间可以由很多个文件组成, 一个表可以跨多个文件存在。- 独立表空间 (默认) (每个表都有独立的多个数据文件, 而且做到了索引和数据的分离)

```
mysql> show variables like '%innodb_file_per_table%';
```

innodb_file_per_table	ON
-----------------------	----

表示当前是独立表空间。

若为OFF, 则表示为共享表空间。

同样: 1表示ON、0表示OFF

3.3. 数据定义 - MySQL存储引擎



MySQL的存储引擎

存储引擎：如何存储数据、如何为存储的数据建立索引和如何更新、查询数据等技术的**实现方法**。

```
1 • show engines;
```

Engine	Support	Comment	Transactions	XA	Savepoints
MEMORY	YES	Hash based, stored in memory, useful for temp...	NO	NO	NO
MRG_MYISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
CSV	YES	CSV storage engine	NO	NO	NO
FEDERATED	NO	Federated MySQL storage engine	NULL	NULL	NULL
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO
MyISAM	YES	MyISAM storage engine	NO	NO	NO
InnoDB	DEFAULT	Supports transactions, row-level locking, and fo...	YES	YES	YES
BLACKHOLE	YES	/dev/null storage engine (anything you write to ...	NO	NO	NO
ARCHIVE	YES	Archive storage engine	NO	NO	NO

3.3. 数据定义 - MySQL存储引擎



MySQL的存储引擎

Table 16.1 Storage Engines Feature Summary

Feature	MyISAM	Memory	InnoDB	Archive	NDB
B-tree indexes	Yes	Yes	Yes	No	No
Backup/point-in-time recovery (note 1)	Yes	Yes	Yes	Yes	Yes
Cluster database support	No	No	No	No	Yes
Clustered indexes	No	No	Yes	No	No
Compressed data	Yes (note 2)	No	Yes	Yes	No
Data caches	No	N/A	Yes	No	Yes
Encrypted data (note 3)	Yes	Yes	Yes	Yes	Yes
Foreign key support	No	No	Yes	No	Yes (note 4)
Full-text search indexes	Yes	No	Yes (note 5)	No	No
Geospatial data type support	Yes	No	Yes	Yes	Yes
Geospatial indexing support	Yes	No	Yes (note 6)	No	No
Hash indexes	No	Yes	No (note 7)	No	Yes
Index caches	Yes	N/A	Yes	No	Yes
Locking granularity	Table	Table	Row	Row	Row
MVCC	No	No	Yes	No	No
Replication support (note 1)	Yes	Limited (note 8)	Yes	Yes	Yes
Storage limits	256TB	RAM	64TB	None	384EB
T-tree indexes	No	No	No	No	Yes
Transactions	No	No	Yes	No	Yes
Update statistics for data dictionary	Yes	Yes	Yes	Yes	Yes

<https://blog.csdn.net/seagal890>

<https://blog.csdn.net/seagal890/article/details/82055219>

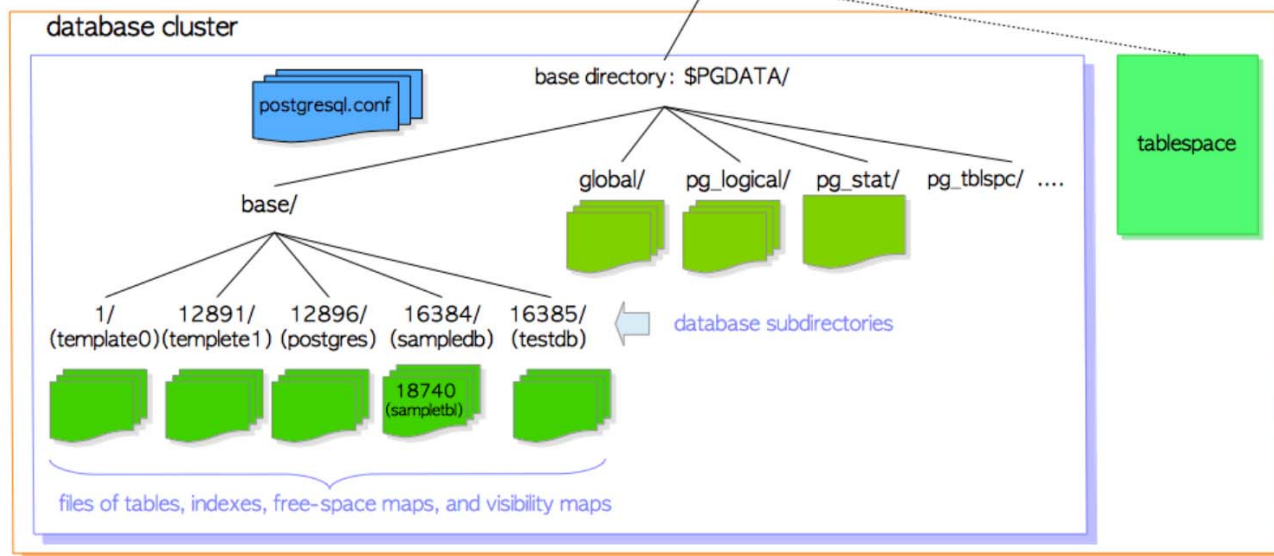
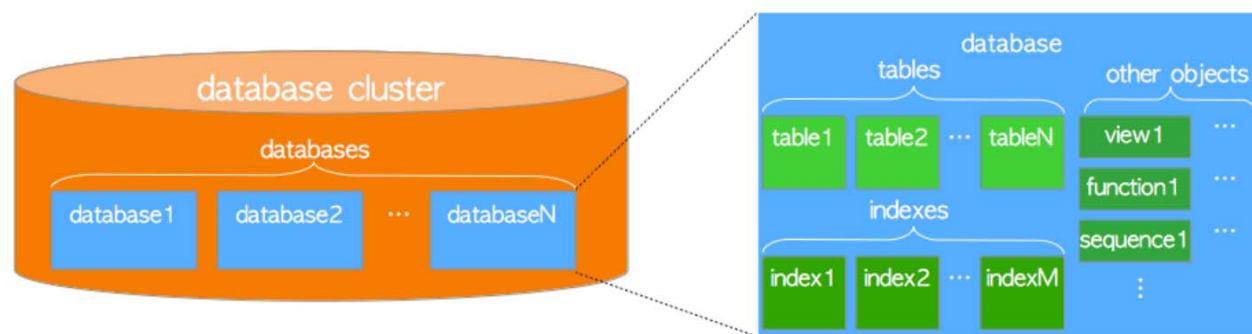
PostgreSQL 的数据库结构

名称	解释
数据库集群cluster	可包含1个或多个数据库
数据库database	可包含1个或多个模式schema
模式schema	默认为public, 包含表/视图/索引等。 每个数据库除了public和用户自定义的模式外, 都包含一个pg_catalog模式, 包含系统表和所有内置数据类型、函数等。
表 table	存放数据的基本表
表空间tablespace	数据库在逻辑上分成的多个存储单元 例: 一个数据库可以使用一个表空间或多个表空间, 如一个很频繁使用的索引可以被放在非常快并且非常可靠的磁盘上, 一个很少使用的或者对性能要求不高的存储归档数据的表可以存储在一个便宜但比较慢的磁盘系统上。

3.3. 数据定义 - PG存储引擎



PostgreSQL 的数据库结构



- ✓ Servers (1)
 - ✓ PostgreSQL 10
 - ✓ Databases (3)
 - > postgres
 - > spj
 - > student
 - > Login/Group Roles
 - > Tablespaces

PostgreSQL > 10 > data

名称

- pg_commit_ts
- pg_dynshmem
- pg_multixact
- pg_replslot
- pg_serial
- pg_snapshots
- pg_stat
- pg_tblspc
- pg_twophase
- pg_subtrans
- pg_wal
- pg_xact
- log
- pg_notify
- base
- global
- pg_logical
- pg_stat_tmp
- PG_VERSION
- pg_hba.conf
- pg_ident.conf
- postgresql.auto.conf
- postgresql.conf
- current_logfiles
- postmaster.opts
- postmaster.pid

本页部分图来源: <http://www.interdb.jp/pg/pgsql01.html>

3.3 数据定义语言DDL

- 1、数据库的定义，修改和删除（模式）
- 2、基本表的定义、修改和删除（模式）
- 3、索引的定义、修改和删除（内模式）
- 4、视图的定义、修改和删除（外模式）

SQL注释符号：

单行注释：--（双连字符）

多行注释：/* ... */

操作对象	操作方式(以MySQL8.0 为例)		
	创建	删除	修改
库	CREATE DATABASE	DROP DATABASE	ALTER DATABASE
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	ALTER VIEW
索引	CREATE INDEX	DROP INDEX	—

1. 数据库的定义与查看

命令： 创建数据库

CREATE DATABASE database_name

```
1  CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
2      [create_option] ...
3
4  create_option: [DEFAULT] {
5      CHARACTER SET [=] charset_name
6      | COLLATE [=] collation_name
7      | ENCRYPTION [=] {'Y' | 'N'}
8  }
```

例子：

create database student;

<https://dev.mysql.com/doc/refman/8.0/en/create-table.html>

character set: 字符集 utf8mb4

collate: 排序规则 utf8mb4_0900_ai_ci

1. 数据库的定义与查看

□ 查看数据库的信息

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| spj |
| spj_test |
| student |
| sys |
| world |
+-----+
9 rows in set (0.00 sec)
```

<https://www.cnblogs.com/kerrycode/p/11170266.html>

□ 数据库的删除

DROP DATABASE [IF EXISTS] name

□ 数据库的修改

```
1 ALTER {DATABASE | SCHEMA} [db_name]
2     alter_option ...
3
4 alter_option: {
5     [DEFAULT] CHARACTER SET [=] charset_name
6     | [DEFAULT] COLLATE [=] collation_name
7     | [DEFAULT] ENCRYPTION [=] {'Y' | 'N'}
8     | READ ONLY [=] {DEFAULT | 0 | 1}
9 }
```

例子:

DROP DATABASE if exists student

ALTER DATABASE mydb

READ ONLY = 0

DEFAULT COLLATE utf8mb4_bin;

2. 基本表的定义 CREATE TABLE

- 语法格式:

```
CREATE TABLE <表名> (  
    <列名>      <数据类型>      [完整性约束]  
    [, <列名>    <数据类型>      [完整性约束]] ...  
    [, <表级完整性约束>] ) ;
```

- <表名>: 所要定义的基本表的名字
- <列名>: 组成该表的各个属性 (列)
- <列级完整性约束条件>: 涉及相应属性列的完整性约束条件
- <表级完整性约束条件>: 涉及一个或多个属性列的完整性约束条件

□ MySQL中的常用数据类型

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

整数类型	字节数	说明	范围
smallint	2BYTE	小范围整数	-32768 ~ 32767
int	4BYTE	常用的整数	-2^{31} ~ $2^{31}-1$
bigint	8BYTE	大范围整数	-2^{63} ~ $2^{63}-1$

浮点类型	字节数	说明	范围
decimal	可变长	用户指定的精度, 精确	小数点前131072位; 小数点后16383 位 salary DECIMAL(5, 2) -999.99 to 999.99
numeric	可变长	用户指定的精度, 精确 mysql中: 等价于 decimal	
float	4BYTE	可变精度, 不精确	6 位十进制数字精度

□ MySQL中的常用数据类型

字符类型	作用	说明
char(n)	定长字符数据	不足补空白
varchar(n)	变长字符数据	有长度限制
text	变长字符数据，无长度限制	

日期类型	描述
DATE	日期: 2021-01-01 范围: '1000-01-01' to '9999-12-31'.
DATETIME	日期时间: 2021-01-01 10:10:29.000000 or 2021-01-01 10:10:29 范围: '1000-01-01 00:00:00.000000' to '9999-12-31 23:59:59.999999',

完整性约束条件

Primary key, Foreign key, Unique, Null, default

CHECK约束, auto_increment

- 这些完整性约束条件被存入系统的**数据字典**中

```
CREATE TABLE person
( id int not null auto_increment
  primary key,
  name varchar(8),
  index ix_person_name(name));
```

```
mysql> desc person;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| name  | varchar(8)    | YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
```

[例1] 建立一个“学生”表Student，它由学号Sno、姓名Sname、性别Sgender、出生日期Sbirthdate、所在系Smajor五个属性构成。其中学号不能为空，并且姓名取值也唯一。

```
CREATE TABLE Student
( Sno      CHAR(8) NOT NULL,
  Sname    CHAR(20) unique,
  Sgender  CHAR(6),
  Sbirthdate Date,
  Smajor   CHAR(40),
  PRIMARY KEY(Sno) )
```

```
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Sno   | char(8) | NO | PRI | NULL |  |
| Sname | char(20) | YES | UNI | NULL |  |
| Sgender | char(6) | YES |  | NULL |  |
| Sbirthdate | date | YES |  | NULL |  |
| Smajor | char(40) | YES |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

[例2] 建立一个“学生选课”表SC，它由学号Sno、课程号Cno，修课成绩Grade组成，其中(Sno, Cno)为主码。

```
CREATE TABLE SC (  
    Sno CHAR(8) ,  
    Cno CHAR(3) ,  
    Grade int ,  
    PRIMARY KEY (Sno, Cno) ,  
    FOREIGN KEY (Sno) REFERENCES S(Sno) ,  
    FOREIGN KEY (Cno) REFERENCES C(Cno))
```

```
mysql> desc scl;
```

Field	Type	Null	Key	Default	Extra
Sno	char(5)	NO	PRI	NULL	
Cno	char(3)	NO	PRI	NULL	
Grade	int	YES		NULL	

3 rows in set (0.00 sec)

□基本表的修改 ALTER TABLE

- 语法格式:

ALTER TABLE <表名>

[ADD <新列名> <数据类型> [完整性约束]]

[ADD <完整性约束名> <列名>]

[DROP <完整性约束名> <列名>]

[ALTER COLUMN <列名> <数据类型>];

- <表名>: 要修改的基本表
- ADD子句: 增加新列和新的完整性约束条件
- DROP子句: 删除指定列或者列的完整性约束条件
- ALTER COLUMN子句: 用于修改列名和数据类型

[例3] 向学生表增加“邮箱地址”列，数据类型为字符型。

```
ALTER TABLE Student ADD Semail VARCHAR(30) ;
```

注：不论基本表中原来是否已有数据，新增加的列一律为空值。

[例4] 将student表中出生日期的数据类型改为字符型

```
ALTER TABLE Student ALTER column Sbirthdate VARCHAR(20) ;
```

注：修改原有的列定义有可能会破坏已有数据

[例5] 增加课程名称必须取唯一值的约束条件。

```
ALTER TABLE Student ADD UNIQUE (Cname) ;
```

[例6] 删除学生姓名必须取唯一值的约束。

```
ALTER TABLE Student DROP constraint IX_sname ;
```


□基本表的删除 DROP TABLE

- 语法格式:

DROP TABLE <表名>;

[例7] 删除Student表

DROP TABLE Student ;

删除一定要慎重！



3.3 数据定义 - 索引是什么



- 无索引的查询：顺序依次遍历表中的每行(全表扫描)
- 索引：加速数据查询的辅助数据结构

select * from s where 姓='李'				
(以页为单位进行全表遍历)				
学 号	姓	名	性别	成绩
2010302562	贺	媛媛	女	
2010302563	计	羽	女	
2010302564	陆	丹	女	
2010302565	魏	苗凤	女	
2010302566	李	航	男	
2010302567	陈	文嘉	男	
2010302568	陈	昕	男	
2010302569	高	高翔	男	
2010302570	李	佩旭	男	

□ 有索引的检索

```
select * from user where id = 103
```



索引表

数据表

音序查字法，部首查字法

索引的定义 CREATE INDEX

- 语法格式:

```
CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX index_name  
    [index_type]  
    ON tbl_name (key_part,...)  
    [index_option]  
    [algorithm_option | lock_option] ...
```

```
key_part: {col_name [(length)] | (expr)} [ASC | DESC]
```

例子:

CREATE INDEX idx_name ON s (sname);

mysql> show index from s;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
s	0	PRIMARY	1	sno	A	4	NULL	NULL		BTREE			YES	NULL
s	1	IX_ngd	1	sname	A	4	NULL	NULL	YES	BTREE			YES	NULL
s	1	IX_ngd	2	sgender	A	4	NULL	NULL	YES	BTREE			YES	NULL
s	1	IX_ngd	3	sdept	A	4	NULL	NULL	YES	BTREE			YES	NULL
s	1	idx_name	1	sname	A	4	NULL	NULL	YES	BTREE			YES	NULL

□索引的类型

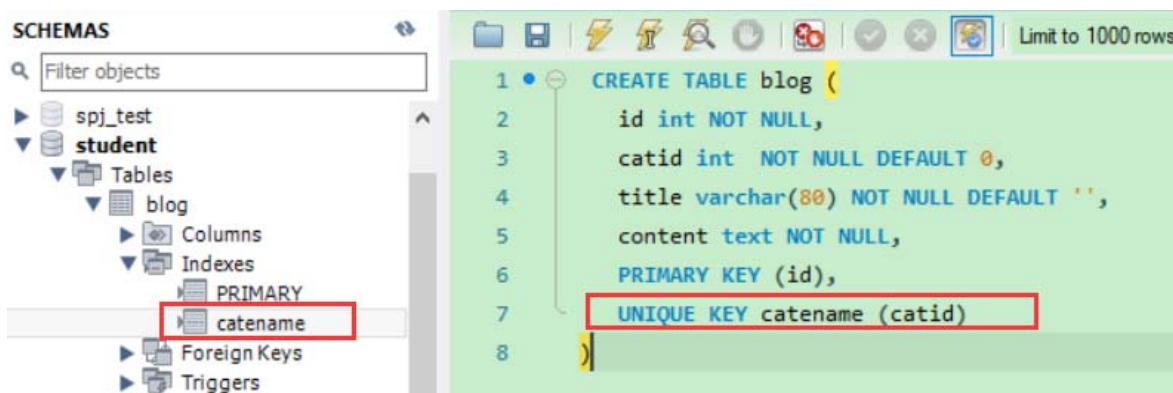
- 功能逻辑：
 - 主键索引 (唯一且非空, 每张表1个, 聚簇索引)
 - 普通索引
 - 唯一索引
 - 全文索引
- 物理实现方式：
 - 聚簇索引
 - 非聚簇索引 (二级索引或者辅助索引, 需要回表)
- 字段个数：
 - 单一索引
 - 联合索引: 最左原则, index: (a, b, c), 则支持 a; a, b; a, b, c查询

3.3 数据定义 - 索引定义-例

[例7] 为学生-课程数据库中的Student, Course, SC三个表建立索引。其中Student表按学号升序建唯一索引, Course表按课程号升序建唯一索引, SC表按学号升序和课程号降序建唯一索引。

```
CREATE UNIQUE INDEX Idx_Stusno ON Student(Sno);  
CREATE UNIQUE INDEX Idx_Coucno ON Course(Cno);  
CREATE UNIQUE INDEX Idx_SCno ON SC(Sno ASC, Cno DESC);
```

注意：也可以在创建表的时候直接指定：



```
1 mysql> show index from tblname;  
2 mysql> show keys from tblname;
```

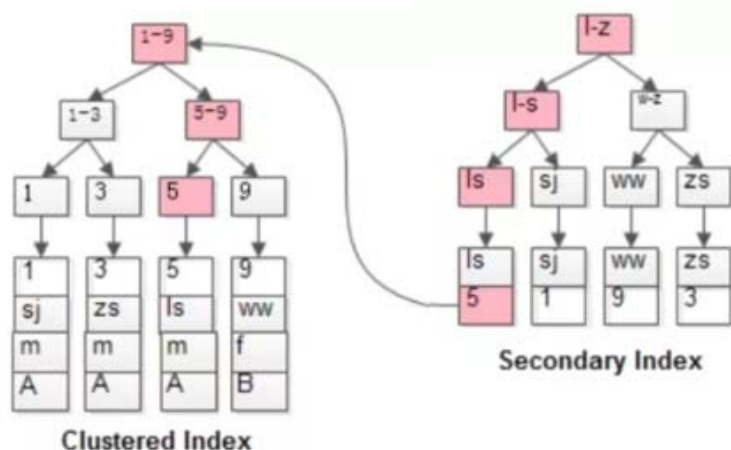
问题：索引一定可以提高查询效率吗？如何使用索引才能提高效率？

作答

[例8] 在Student表的Sname（姓名）列上建立一个聚簇索引，
而且Student表中的记录将按照Sname值的升序存放。

CREATE **CLUSTER** INDEX Stusname ON Student (Sname);
(MySQL不支持单独的聚簇索引，主键索引即聚簇索引)

- t(id **PK**, name **KEY**, gender, flag);
- 四条数据（name上创建辅助索引）



select * from t where name='lisi';

(1) 先通过普通索引(辅助索引)定位到主键值id=5;

(2) 在通过聚集索引定位到行记录; (**回表操作**)

如何减少回表?
如果查询结果是多行记录,
是每条记录都回表还是若干
次后回表1次?

<https://zhuanlan.zhihu.com/p/401198674>

□索引的删除 DROP INDEX

- 语法格式:

DROP INDEX <索引名> ON <表名>

[例9] 删除Student表的Stusname索引。

DROP INDEX Stusname ON Student

或者

ALTER TABLE student drop index Stusname;

SQL标准中没有定义对索引的修改功能，而采用删除后重新定义索引的方式实现。

判断正误：

聚集索引决定了索引所在表的物理存储结构，一张表中只能有一个聚集索引，但是一个聚集索引可以包含很多列。

☒ A 对

☐ B 错

提交

唯一键 和 主键 的区别是什么？

作答

正常使用主观题需2.0以上版本雨课堂