



厦门大学
Xiamen University



大数据分析基础

数据库基础及应用

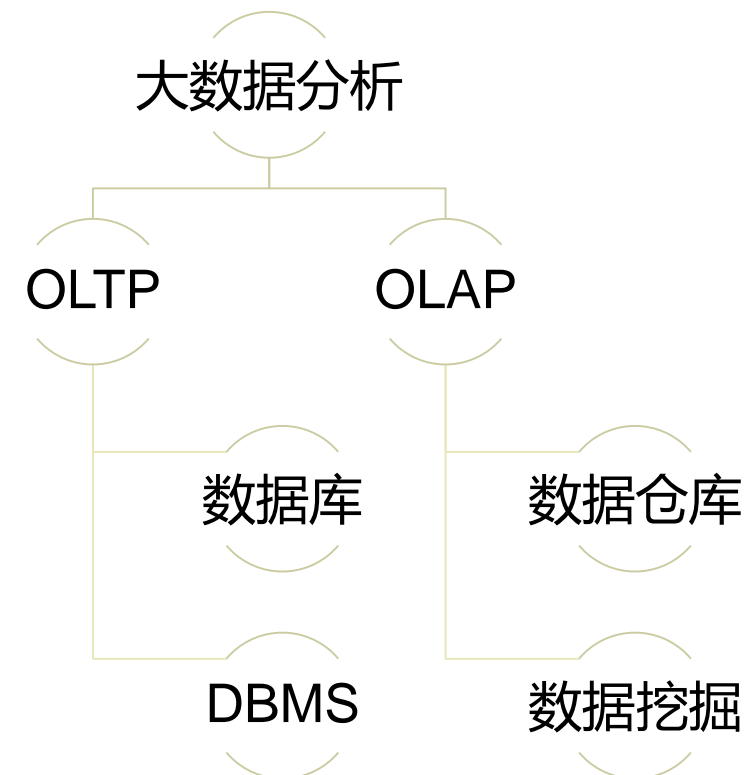
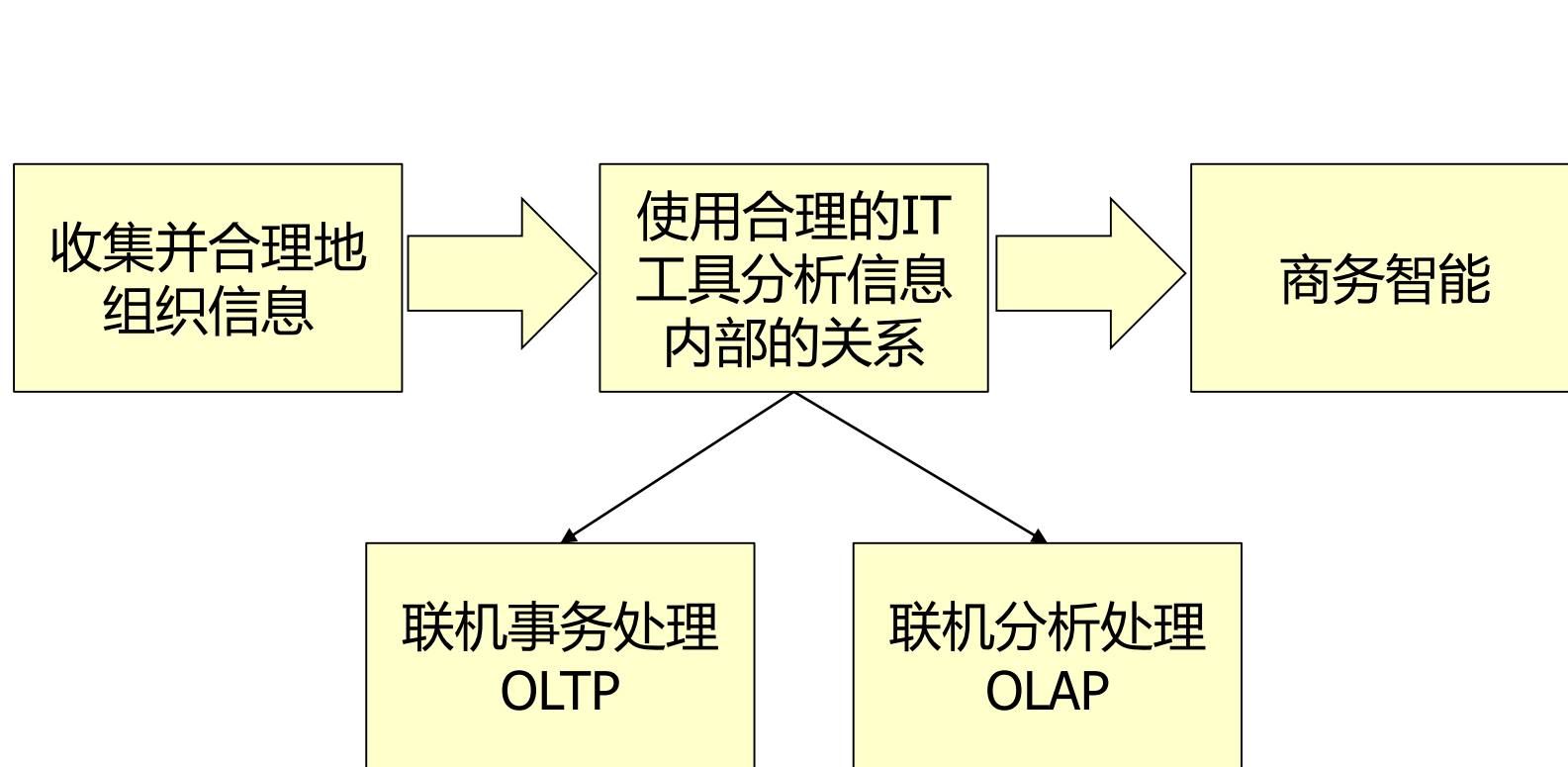




- 数据库基础——关系型数据库
- MySQL
- 利用Python操作数据库
- 数据库在大数据时代的发展



大数据分析无法凭空出现





❖ 联机事务处理（Online transaction processing, OLTP）

- ❖ 输入信息收集、处理、更新
- ❖ 支持业务处理
- ❖ 销售订单、应收账款等
- ❖ 由业务数据库和数据库管理系统来支持

❖ 联机分析处理（Online analytical processing, OLAP）

- ❖ 一种提供决策支持的信息处理方式
- ❖ 帮助进行数据分析
- ❖ 由数据仓库和数据挖掘工具来支持



❖ 数据管理技术的三个阶段

- ❖ 人工管理阶段（20世纪50年代中期以前）
- ❖ 文件系统阶段（20世纪50年代后期—60年代中期）
- ❖ 数据库系统阶段（20世纪60年代后期—今）

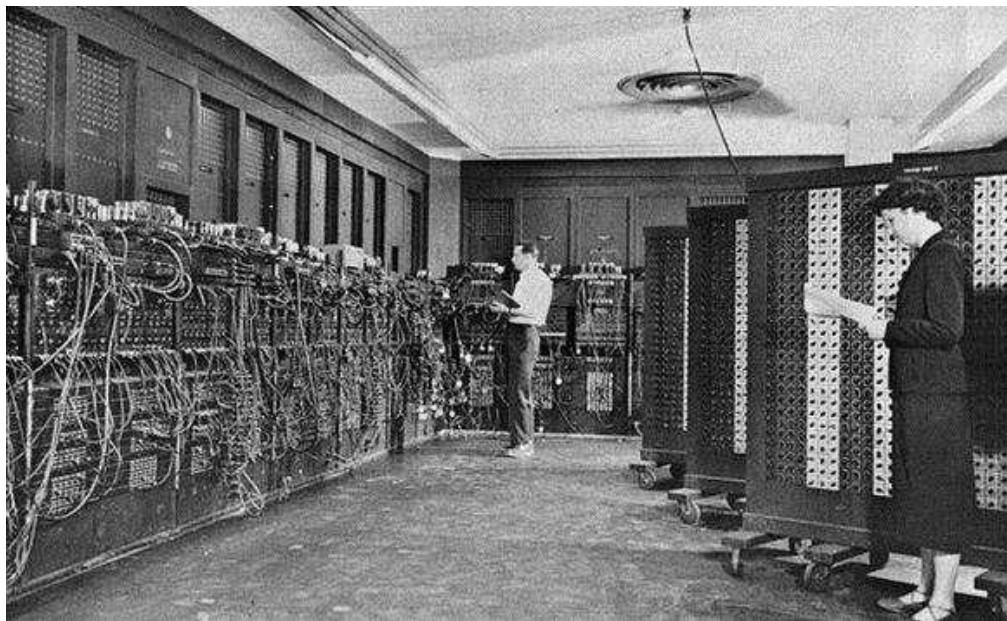


人工管理阶段 (~1950 中期)

背景

计算机主要用于科学计算

- 数据量小、结构简单，如高阶方程、曲线拟和等
 - 1946年ENIAC是美国奥伯丁武器试验场为了满足计算弹道需要而研制成的，这台计算器使用了17840支电子管，大小为80英尺×8英尺，重达28t（吨），功耗为170kW，运算速度为每秒5000次的加法运算





背景

❖ 外存为顺序存取设备

- 磁带、卡片、纸带，没有磁盘等直接存取设备。

❖ 没有操作系统及数据管理软件

- 用户用机器指令编码，通过纸带机输入程序和数据，程序运行完毕后，由用户取走纸带和运算结果，再让下一用户操作。

❖ 特点

- **数据不保存**：每个用户使用自己的数据，用完撤走不保存。
- **应用程序管理数据**：用户通过特定的应用程序完全负责数据管理工作，包括数据的组织、存储结构、存取方法、输入输出等。
- **数据不共享**：一组数据只能对应一组程序。
- **数据不具有独立性**：程序中存取数据的子程序随着存储结构的改变而改变。





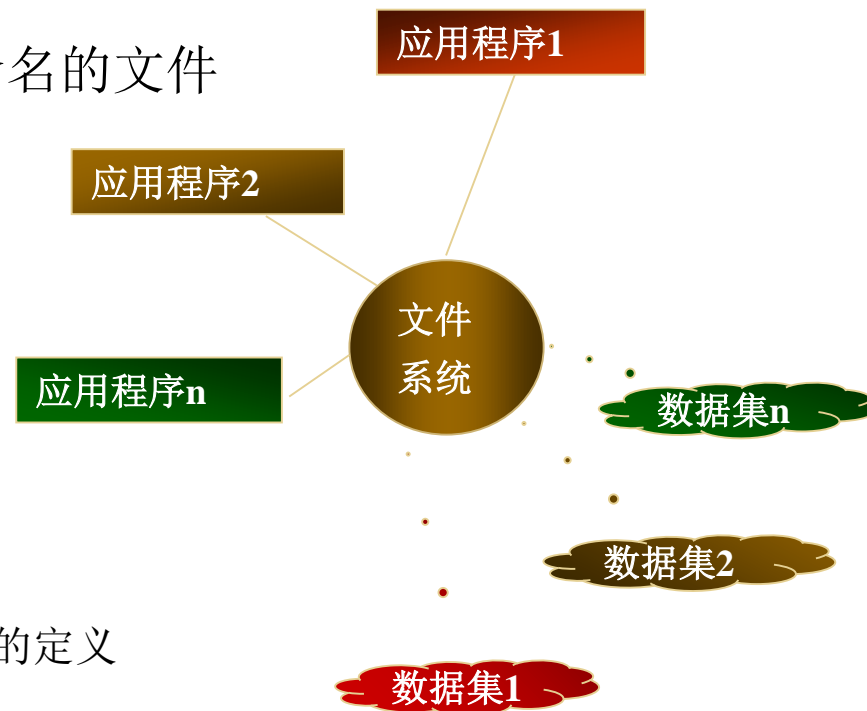
文件系统阶段 (1950 中期~1960 中期)

背景

- 计算机不但用于科学计算，还用于管理
- 外存有了磁盘、磁鼓等直接存取设备
- 有了专门管理数据的软件，一般称为文件系统
 - 在文件系统中，数据按其内容、结构和用途组成若干命名的文件

特点

- 数据可以长期保存
- 共享性差（一个文件对应一个程序）
 - 冗余、不一致
- 数据与程序有一定的独立性
 - 文件的逻辑结构与存储结构由系统进行转换
 - 数据在存储上的改变不一定反映在程序上
 - 文件的逻辑结构改变时，应用程序必须改变，同时修改文件结构的定义

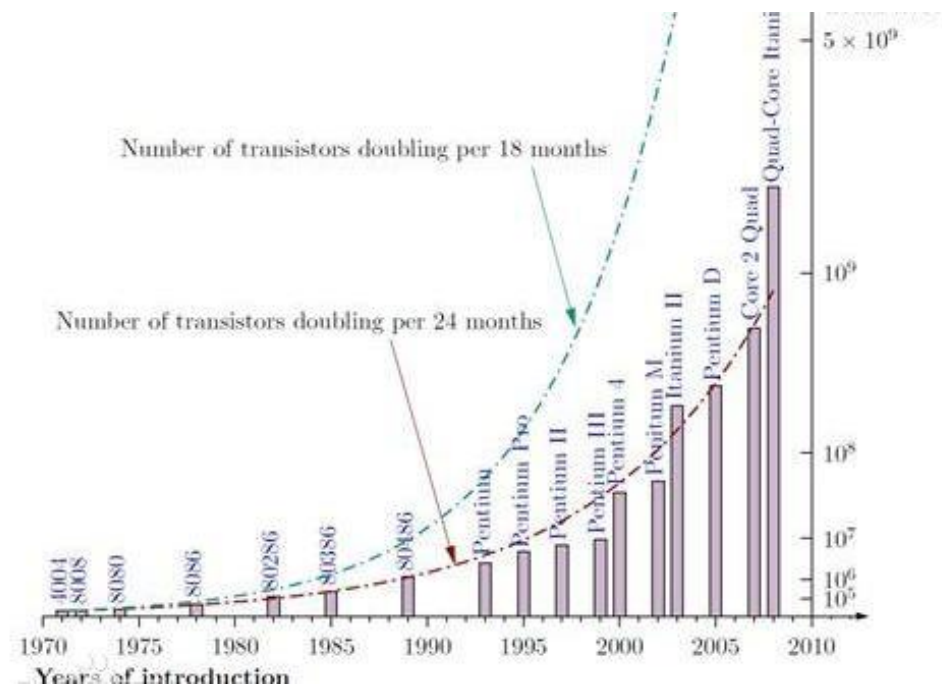




数据库系统阶段 (1960 中期~)

背景

- 计算机管理的数据量大，关系复杂，共享性要求强（多种应用、不同语言共享数据）
- 外存有了大容量磁盘，光盘
- 软件价格上升，硬件价格下降
 - 摩尔定律





❖ 数据库（DataBase，DB）

- ❖ 长期储存在计算机内、有组织的、可共享的相关信息的集合。

❖ 数据库的特点

- ❖ 永久存储
- ❖ 有组织
- ❖ 可共享



❖ 数据库管理系统（DataBase Management System, DBMS）

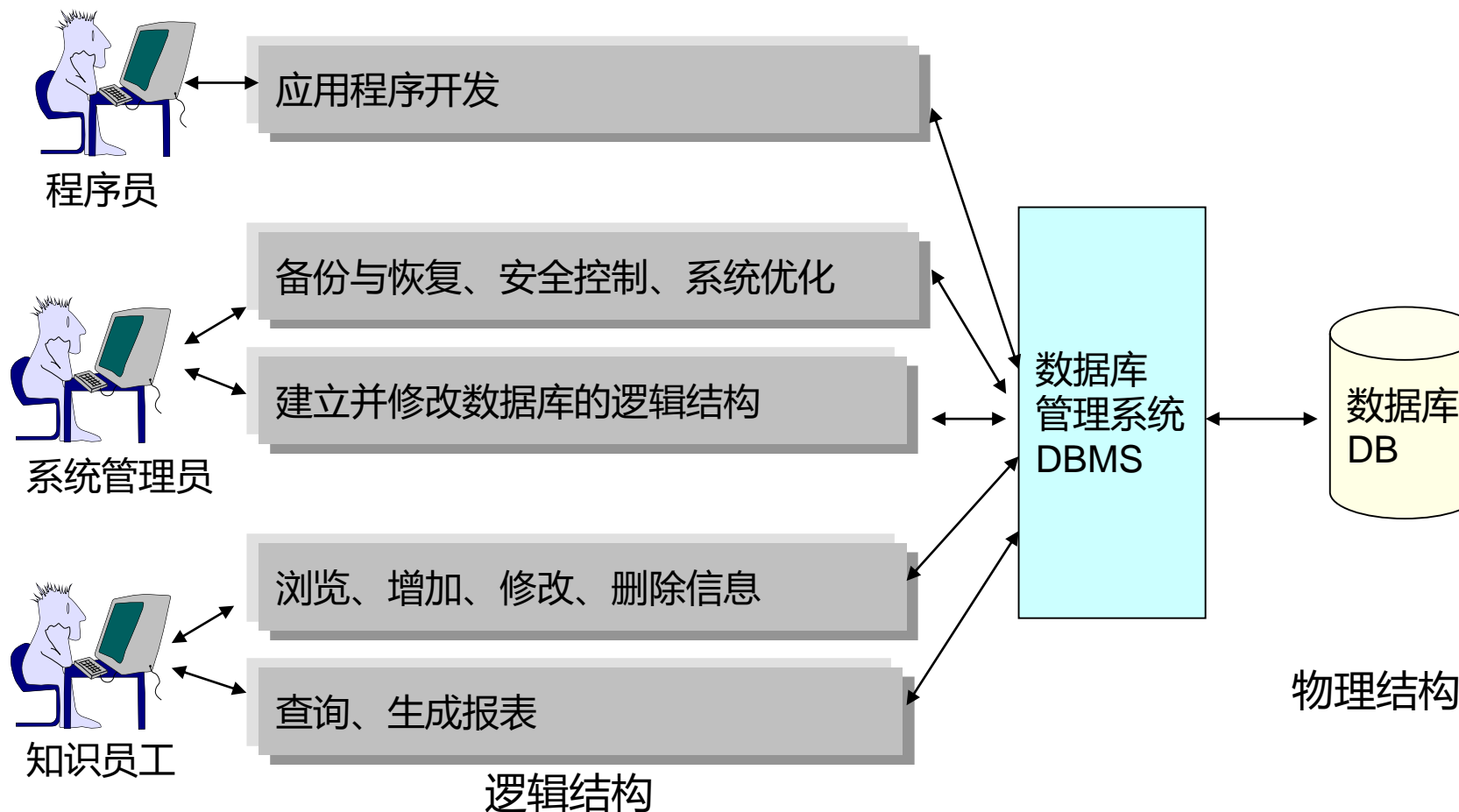
- ❑ 是介于用户与操作系统之间的一层数据管理软件。
- ❑ 是基于某种数据模型的。

❖ 数据库管理系统的功能

- ❑ 数据定义
- ❑ 数据组织、存储和管理
- ❑ 数据操纵
- ❑ 数据库的事务管理和运行管理
- ❑ 数据库的建立和维护
- ❑ 其他



数据库和DBMS的关系



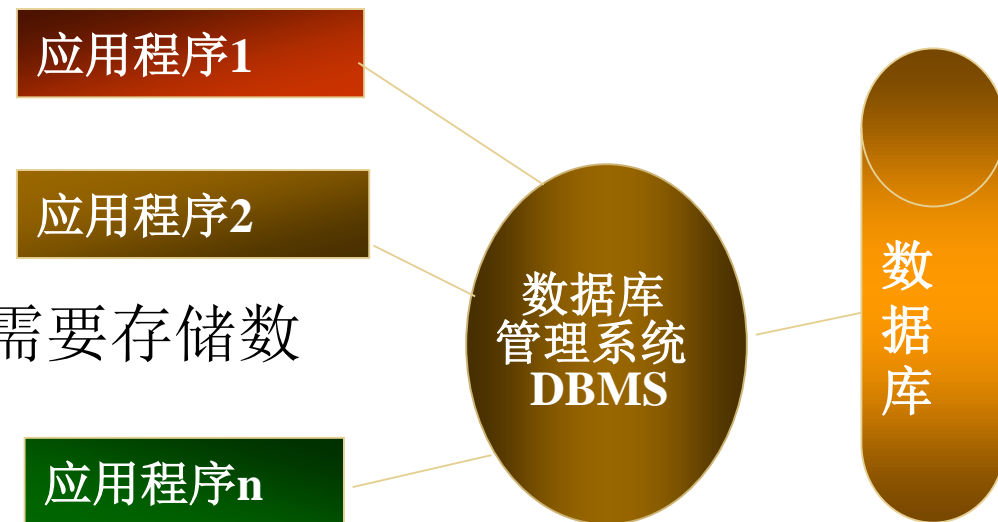


优点

- ❖ 数据有整体的结构性，面向全组织，面向现实世界
- ❖ 由数据库管理系统DBMS统一存取，维护数据语义及结构
- ❖ 数据共享性好
- ❖ 数据与程序完全相互独立

缺点

- ❖ 数据库系统的设计非常复杂、困难并且耗时，需要存储数据间的关系
- ❖ 需要对所有的程序员和用户进行训练
- ❖ 需要为软硬件支付一定的成本
 - DB2、Oracle数十万美元起，MySQL数十万RMB起
- ❖ 单个程序所需运行时间较长





数据管理技术的比较

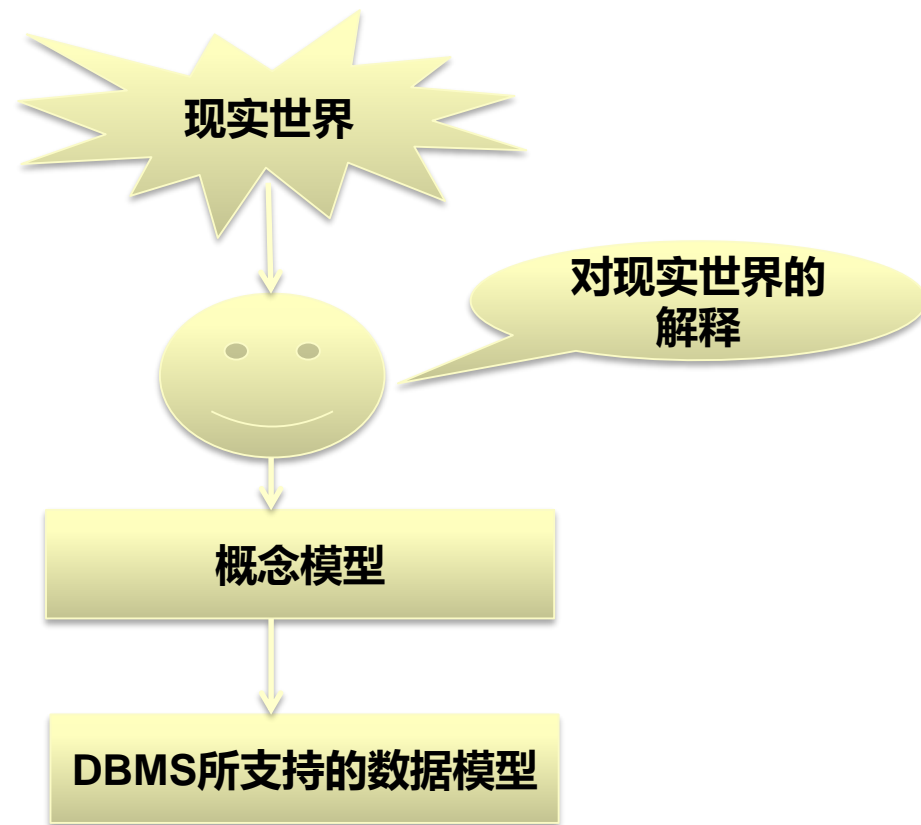
	人工	文件系统	数据库
硬件	无存储设备	硬盘	大容量硬盘
软件	无操作系统	文件系统	DBMS
数据管理者	用户	文件系统	DBMS
面向	应用程序	应用程序	组织
数据共享	No	低	高
数据独立性	No	低	高
结构化	No	半结构化	结构化
控制	应用程序	应用程序	DBMS



- ❖ 数据模型（Data Model）是对现实世界各种事物特征的数字化的模拟和抽象
- ❖ 数据模型应能满足三个方面的要求：
 - ❑ 能比较真实地模拟现实世界
 - ❑ 容易为人所理解
 - ❑ 便于在计算机上实现

信息世界

机器世界





✚ 现实世界

- ✚ 即客观世界，产生最原始的数据

✚ 信息世界

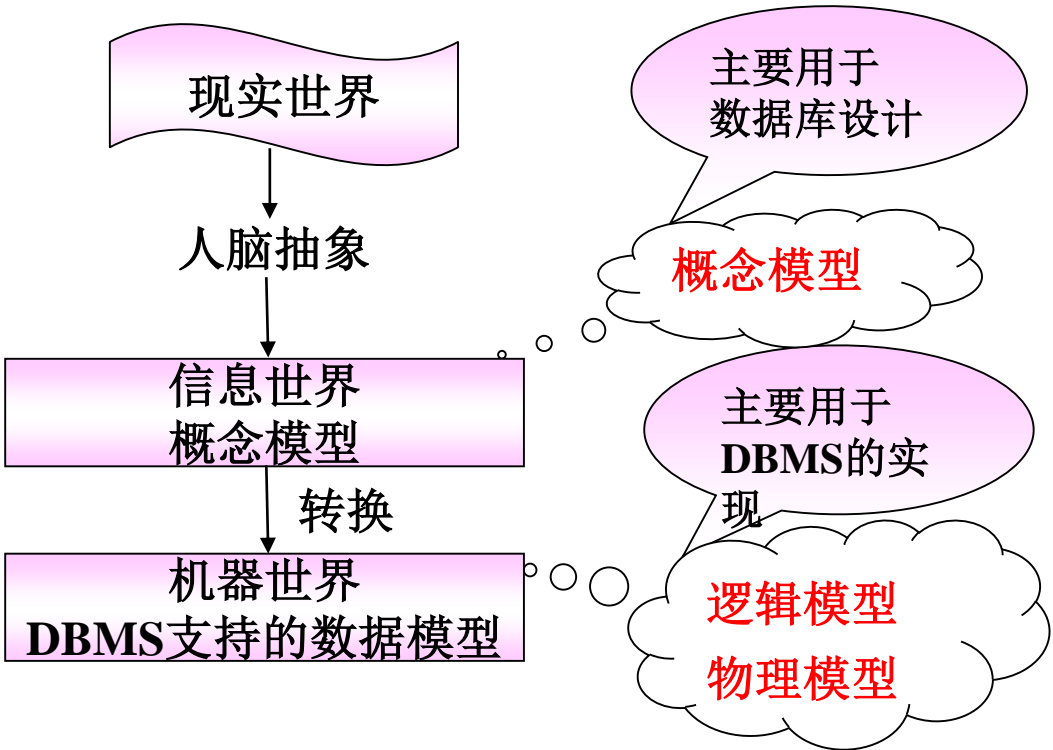
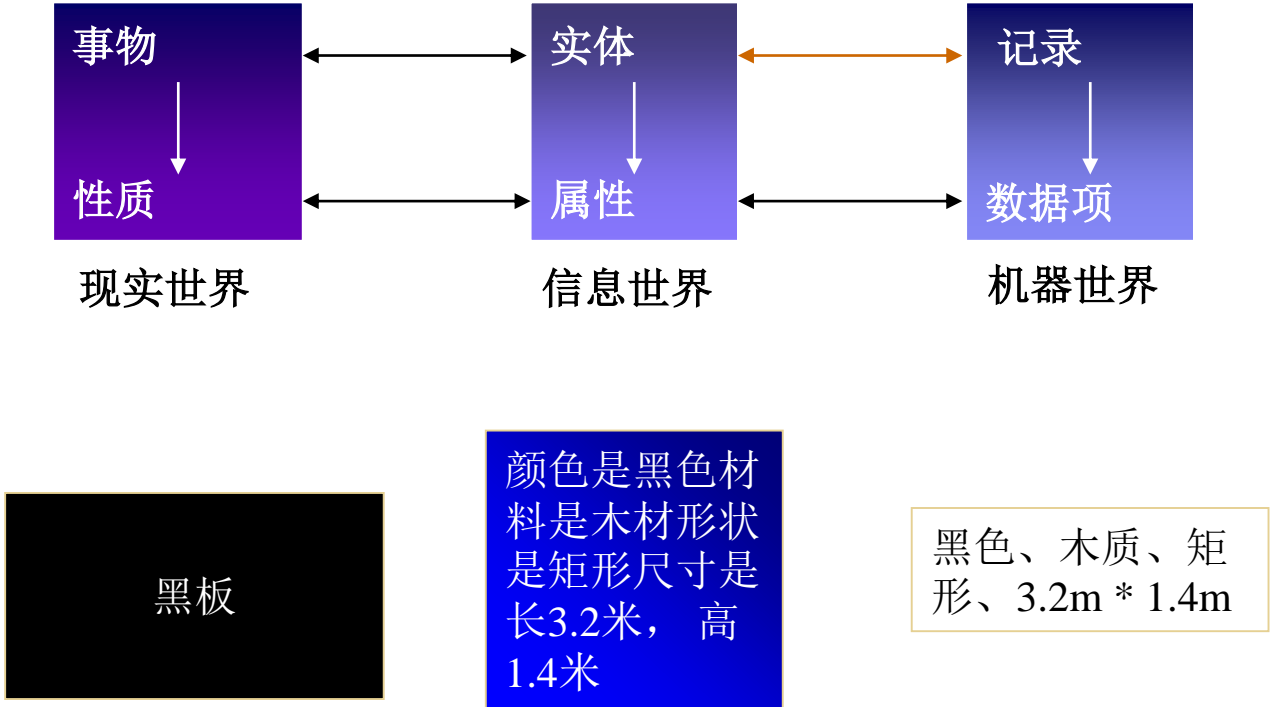
- ✚ 是现实世界在人们头脑中反映并用文字或符号记载下来，是人对现实世界的认识抽象过程，经选择、命名、分类等抽象工作后进入信息世界

✚ 机器世界

- ✚ 用数据模型描述现实世界中的事物及其联系



从现实世界到机器世界





❖ 概念模型（Conceptual Model）

- ❖ 又称信息模型、语义模型，是按用户的观点来对数据和信息建模。
- ❖ 利用实体、联系和约束描述现实世界的静态、动态和时态特征。
- ❖ 不注重数据的物理组织结构，着重表示数据模拟的语义，使数据模型更接近现实世界，便于用户理解。
- ❖ 常用的概念模型包括：
 - 实体-联系模型（E-R模型）
 - 面向对象数据模型
 - 函数数据模型
 - 扩展关系模型
 - 语义关联模型



逻辑模型

按计算机系统的观点对数据建模

主要包括：

- 层次模型（Hierarchical Model）
- 网状模型（Network Model）
- 关系模型（Relational Model）
- 面向对象模型（Object Oriented Model）
- 对象关系模型（Object Relational Model）

物理模型

对数据最低层的抽象

描述数据在系统内部的表示方式和存取方法，在磁盘上的存储方式和存取方法

面向计算机系统，由DBMS实现



数据模型的组成要素

❖ 数据结构

描述系统的静态特性

- ❖ 所研究的对象类型（Object type）的集合，包括：数据的类型、内容和性质的对象（事物）；数据之间联系的对象（联系）。

❖ 数据操作

描述系统的动态特性

- ❖ 对数据库中的各种对象的实例（值）允许执行的操作的集合。主要有检索和更新（插入、删除、修改）两类操作。

❖ 完整性约束

完整性规则（条件）的集合

- ❖ 给出数据及其联系所具有的制约、依赖和存储规则，用于限定数据库的状态和状态变化，保证数据库中的数据的正确、有效、完全和相容。



❖ 概念模型最常用的表示方法

- ❖ 实体-联系方法（Entity-Relationship Approach, ER模型）
- ❖ P.P.S. Chen（陈平山）于1976年提出
- ❖ 该模型直接从现实世界中抽象出实体类型及实体间联系，然后用E-R图表示现实世界的概念模型



❖ 基本概念

- ❖ 实体（entity）：客观存在，可以相互区别的东西称为实体。可以是具体的对象，也可以是抽象的事件。
- ❖ 实体集（entity set）：性质相同的同类实体的集合。
- ❖ 属性（attribute）：实体的某一方面的特征
- ❖ 属性域（domain）：属性的取值范围；含值的类型
- ❖ 码（key）：唯一标识每个实体的属性或属性集
- ❖ 实体型（entity type）：某一实体属性的集合
- ❖ 联系（relationship）：现实世界中事物内部以及事物之间的联系在信息世界中反映为实体内部的关系和实体之间的联系



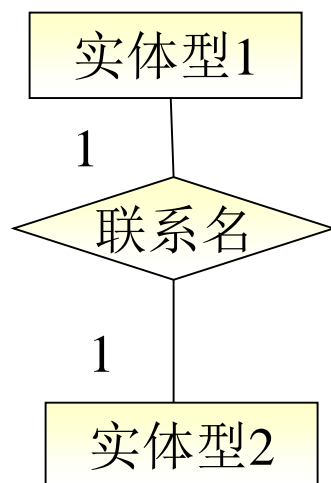
❖ 实体间的联系

两个实体型	}	[一对一联系 (1:1)
三个实体型			一对多联系 (1:n)
一个实体型			多对多联系 (m:n)

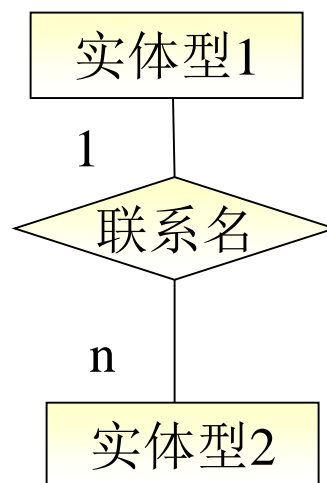


两个实体型之间的联系

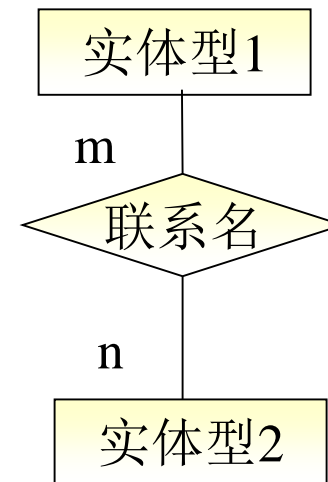
- 一对一的联系，记为 $1 : 1$
- 一对多的联系，记为 $1 : n$
- 多对多的联系，记为 $m : n$



1:1联系



1:n联系



m:n联系



✚ 一对一联系

- ✚ 实体集 E_1 中每个实体至多和实体集 E_2 中一个实体有联系，反之亦然，记为 $1 : 1$

✚ 一对多联系

- ✚ 实体集 E_1 中的每个实体与实体集 E_2 中任意个实体有联系，而 E_2 中每个实体至多和 E_1 中的一个实体有联系，记为 $1 : n$

✚ 多对多联系

- ✚ 实体集 E_1 中的每个实体与实体集 E_2 中任意个实体有联系，反之亦然，记为 $m : n$

ER图的四个基本成分

实体名

矩形框表示实体型

属性名

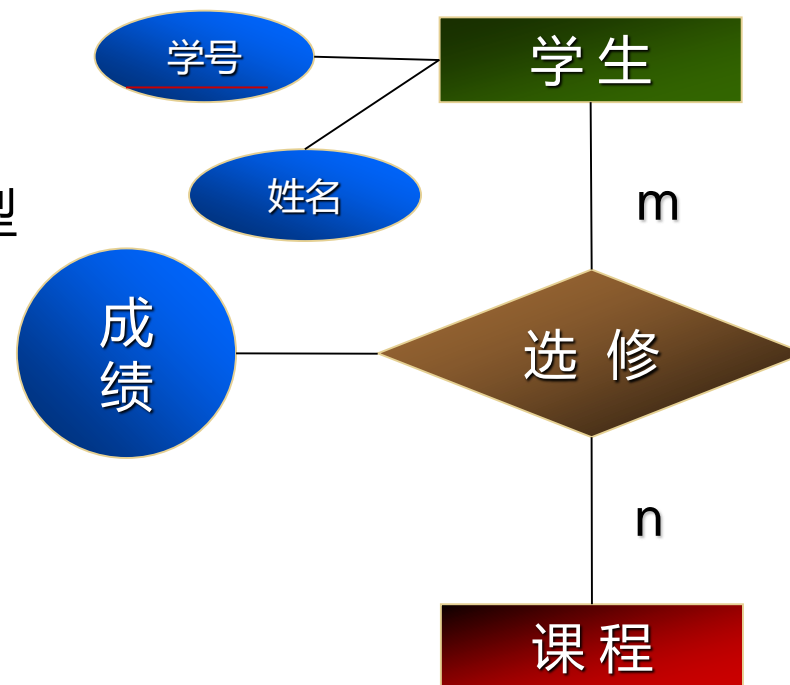
椭圆形表示属性

联系名

菱形表示联系

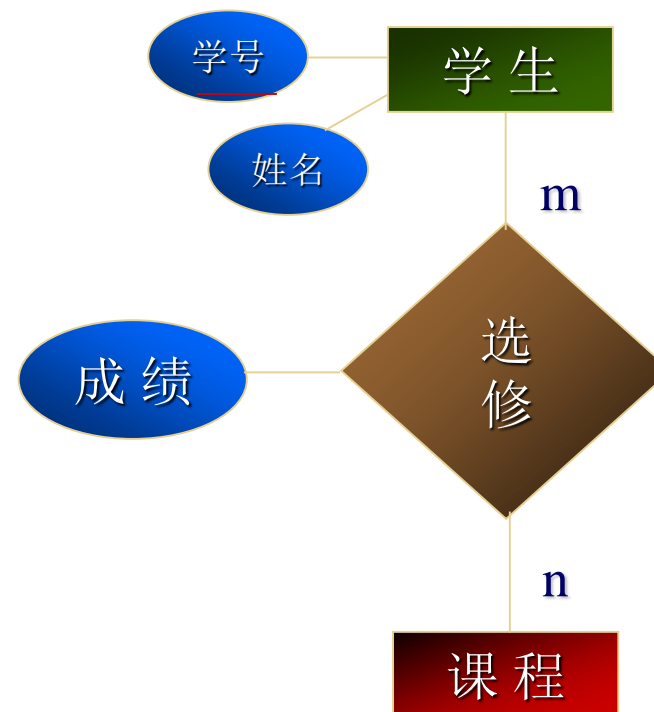
—

连接实体型与联系类型，也可用于表示实体与属性的联系并注明种类；对构成码的属性，在属性名下画一横线表示。

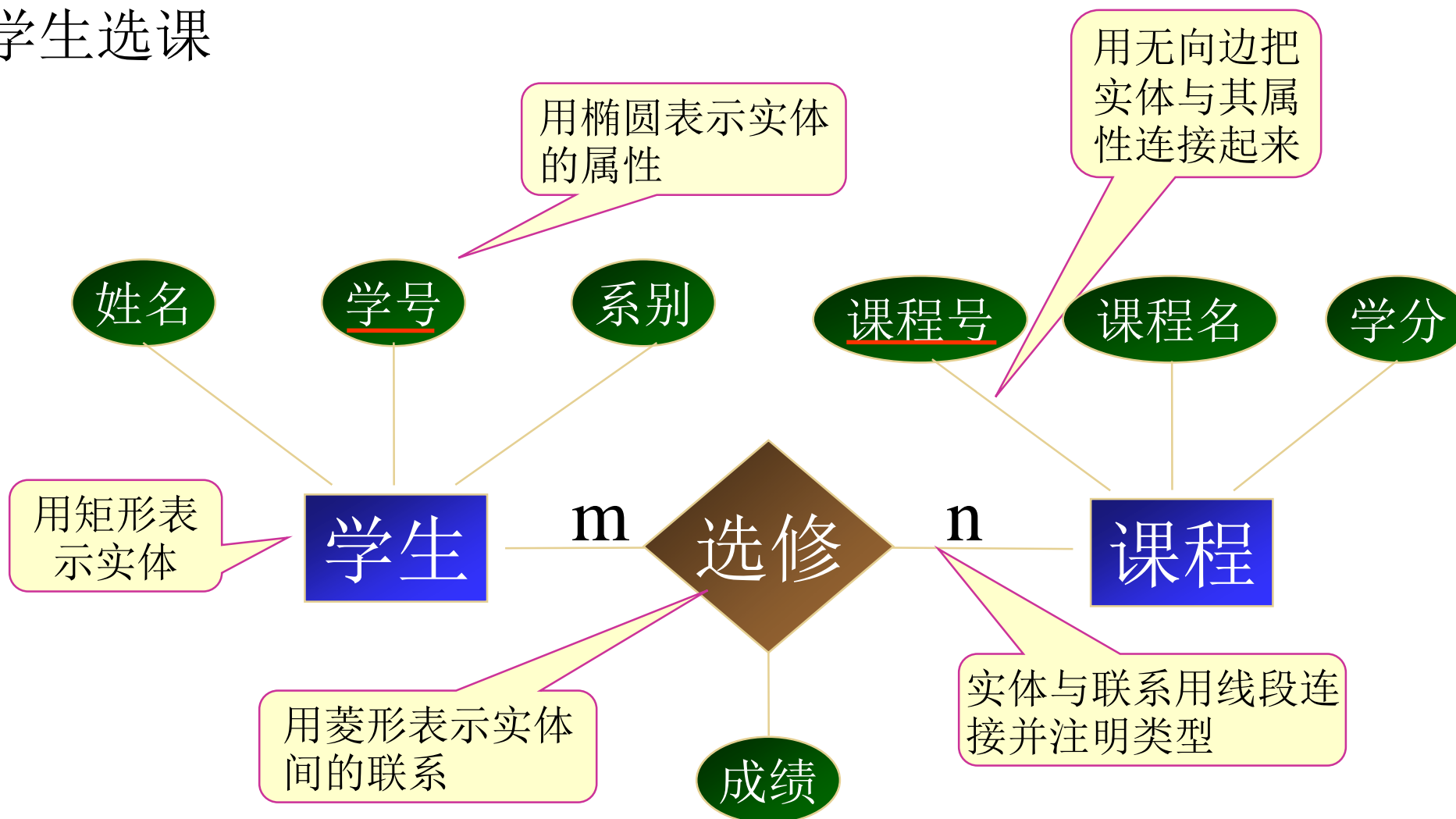


ER图设计过程

- ❖ 首先确定实体类型：几个实体类型及相应的实体名
- ❖ 确定联系类型：各实体类型之间是否有联系，是何种联系类型及相应的联系名
- ❖ 连接实体类型和联系类型，组合成E-R图
- ❖ 确定实体类型和联系类型的属性
- ❖ 确定实体类型的码
- ❖ 实体名和联系名均不能重复!



例：学生选课





- ❖ 实体、实体之间的联系均由二维表来表示
- ❖ 二维表称为关系

SC

学号 IDStu	姓名 NameStu	所在学院 Inst	院地址 Addr	课程号 IdCour	课程名 NameCour	学习成绩 Grade
00097001	张丽	管理	管201	C0001	高等数学	90
00097001	张丽	管理	管201	C0002	英语	87
00097002	王峰涛	电信	信103	C0001	高等数学	95
00097003	李一凡	能动	能301	C0002	英语	96



❖ 二维表，由行与列构成。

- ❖ 元组(Tuple)：在二维表中的一行，称为一个元组
- ❖ 属性(Attribute)：在二维表中的列，称为属性
- ❖ 码(Key)：如果在一个关系中存在唯一标识一个实体的一个属性或属性集称为实体的键，即使得在该关系的任何一个关系状态中的两个元组，在该属性上的值的组合都不同
- ❖ 域(Domain)：属性值的取值范围为值域
- ❖ 分量：每一行对应的列的属性值，即元组中的一个属性值

学号 IDStu	姓名 NameStu	所在学院 Inst	院地址 Addr	课程号 IdCour	课程名 NameCour	学习成绩 Grade
00097001	张丽	管理	管201	c0001	高等数学	90
00097001	张丽	管理	管201	c0002	英语	87
00097002	王峰涛	电信	信103	c0001	高等数学	95
00097003	李一凡	能动	能301	c0002	英语	96

码

分量

属性

元组



码

主键 (Primary Key, PK)

- 用来区别一张关系表内的不同元组

外键 (Foreign Key, FK)

- 用于与另一张表的关联, 是能确定另一张表记录的字段, 用于保持数据的一致性

	主键	外键
定义:	唯一标识一条记录, 不能有重复的, 不允许为空	表的外键是另一表的主键, 外键可以有重复的, 可以是空值
作用:	用来保证数据完整性	用来和其他表建立联系用的
个数:	主键只能有一个	一个表可以有多个外键



对关系的描述称为关系模式，一般表示为：

关系名（属性1，属性2，...，属性n）

学号 IDStu	姓名 NameStu	所在学院 Inst	院地址 Addr	课程号 IdCour	课程名 NameCour	学习成绩 Grade
00097001	张丽	管理	管201	C0001	高等数学	90
00097001	张丽	管理	管201	C0002	英语	87
00097002	王峰涛	电信	信103	C0001	高等数学	95
00097003	李一凡	能动	能301	C0002	英语	96

学生课程（学号，姓名，所在学院，院地址，课程号，课程名，学习成绩）



❖ 为了建立冗余较小、结构合理的数据库，设计数据库时必须遵循一定的规则。在关系型数据库中这种规则就称为范式。

❖ 第一范式(确保每列保持原子性)

- 数据库表中的所有字段值都是不可分解的原子值

❖ 第二范式(确保表中的每列都和主键相关)

- 第二范式需要确保数据库表中的每一列都和主键相关，而不能只与主键的某一部分相关（主要针对联合主键而言）
- 在一个数据库表中，一个表中只能保存一种数据，不可以把多种数据保存在同一张数据库表中
 - 比如要设计一个订单信息表，因为订单中可能会有多种商品，所以要将订单编号和商品编号作为数据库表的联合主键。这样在该表中商品名称、单位、商品价格等信息不与该表的主键相关，而仅仅是与商品编号相关。所以在这里违反了第二范式的设计原则。

❖ 第三范式(确保每列都和主键列直接相关,而不是间接相关)

- 每一列数据都和主键直接相关，而不能间接相关
 - 比如在设计一个订单数据表的时候，可以将客户编号作为一个外键和订单表建立相应的关系。而不可以再在订单表中添加关于客户其它信息（比如姓名、所属公司等）的字段



❖ 数据操作

- ❖ 关系模型的数据操作主要包括：查询、插入、删除和修改数据
- ❖ 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
- ❖ 存取路径对用户隐蔽，用户只要指出“干什么”或“找什么”，不必详细说明“怎么干”或“怎么找”

❖ 完整性约束

❖ 实体完整性

- 实体完整性(Entity integrity)是指关系的主关键字不能重复也不能取“空值”

❖ 参照完整性

- 参照完整性(Referential Integrity)是定义建立关系之间联系的主关键字与外部关键字引用的约束条件。

❖ 用户定义完整性

- 用户定义完整性(user defined integrity)则是根据应用环境的要求和实际的需要，对某一具体应用所涉及的数据提出约束性条件

❖ 存储结构

- ❖ 在DB的物理组织中，表以文件形式存储



✚ 优点

- ✚ 严格的数学概念
- ✚ 结构单一、简单
- ✚ 用户易懂易用存取路径透明
- ✚ 数据独立性强
- ✚ 安全性高
- ✚ 易于开发

✚ 缺点

- ✚ 查询效率低



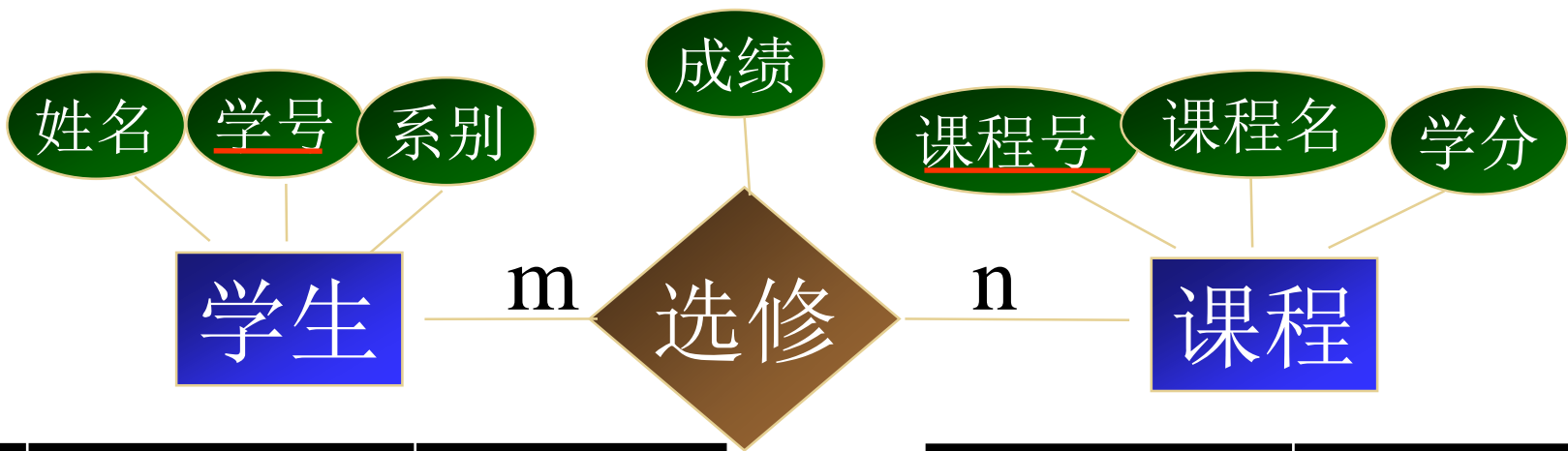
- 数据字典是指对数据的数据项、数据结构、数据流、数据存储、处理逻辑、外部实体等进行定义和描述，其目的是对数据流程图中的各个元素做出详细的说明，使用数据字典为简单的建模项目。

表名	描述
@100	100
@AAFT1	AAFT1
@AAFT2	AAFT2
@ACGS1	ACGS1
@ACGSZ1	ACGSZ1
@ACGX1	ACGX1
@ACGZJ1	ACGZJ1
@ACPG1	ACPG1
@ACPJ1	ACPJ1
@ACPJ21	ACPJ21
@ACPJ31	ACPJ31
@AFT1	二次分摊统计
@AFT2	二次分摊统计
@AFYA1	AFYA1

字段ID	字段描述	字段类型	字段长度	是否可以为空	连接向
ItemCode	物料编号	0	20	0	
ItemName	物料描述	0	100	0	
FrgrName	外文描述	0	100	0	
ItmsGrpCod	物料组	2	6	0	OITB
CstGrpCode	关税组	2	6	0	OARG
VatGourpSa	销项税定义	0	8	0	OVTG
CodeBars	条形码	0	16	0	
VATLiable	税定义	0	1	0	
PrchseItem	采购物料 [是...	0	1	0	
SellItem	销售物料 [是...	0	1	0	
InvntItem	库存物料 [是...	0	1	0	
OnHand	存货量	4	40	0	
IsCommitted	客户订购的数量	4	40	0	
OnOrder	供应商订购数量	4	40	0	



- 对每个实体（Entity）均建立一张关系表
- 对每个关系（Relationship）也同样建立一张关系表
 - 但是，有时候为了在性能和规范中取得平衡，并不一定要这么做



姓名	学号 (PK)	系别
张三	35620191150001	物流工程

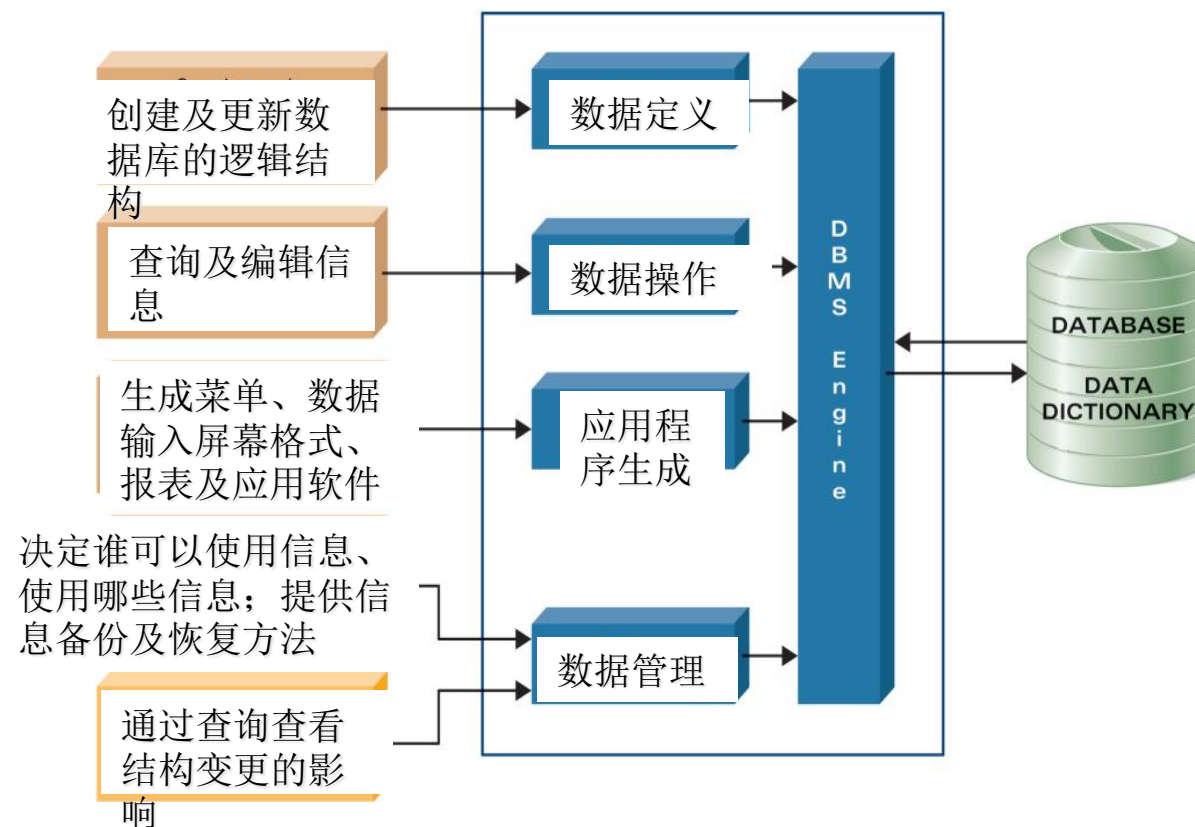
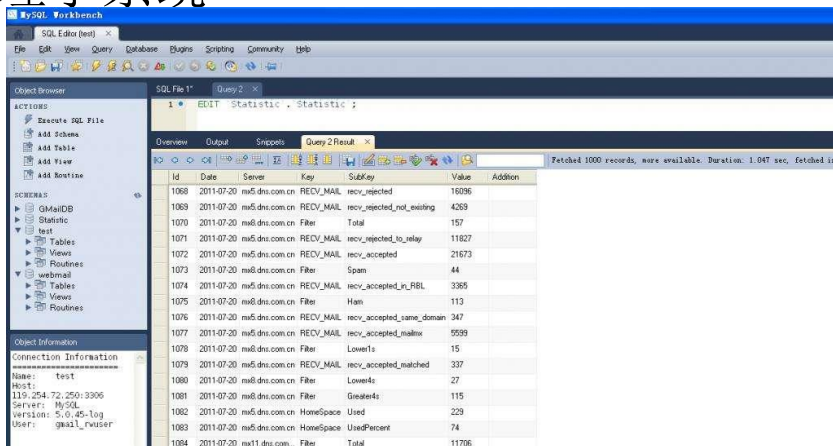
课程名	课程号 (PK)	学分
大数据分析基础	10010	2

课程号 (PK/FK)	学号 (PK/FK)	成绩
10010	35620191150001	99



- 数据库管理系统（Database Management Systems, DBMS）工具帮助指定数据库逻辑需求，在数据库中访问和使用信息
- DBMS有5个重要的组成部分

- DBMS引擎
- 数据定义子系统
- 数据操作子系统
- 应用程序生成子系统
- 数据管理子系统





❖ 数据库引擎是用于存储、处理和保护数据的核心服务

- ❖ 接受来自其他各DBMS子系统的逻辑查询请求，并将逻辑查询请求转换成其对应的物理形式
- ❖ 数据库的引擎决定了数据库系统的性能
- ❖ 不同的引擎有着不同的特点和适用环境
 - MySQL引擎：MyISAM、InnoDB、MERGE、MEMORY(HEAP)、BDB(BerkeleyDB)、EXAMPLE、FEDERATED、ARCHIVE、CSV、BLACKHOLE

❖ 物理视图：信息在存储设备上怎样进行物理排列、存储和读取。

❖ 逻辑视图：关注知识工作者要如何排列和存取信息，以满足其特定的业务需求。

❖ DBMS引擎将这两者进行了隔离



- ❖ 数据定义子系统—帮助人们在数据库中建立并维护数据字典，以及定义数据库中的文件结构
- ❖ 数据字典帮助定义
 - ❖ 字段名称
 - ❖ 数据类型（数字型等）
 - ❖ 格式（电话号码前是否有区号）
 - ❖ 缺省值
 - ❖ 有效范围
 - ❖ 输入约束等



- ✦ 数据操作子系统—帮助用户对数据库中的信息进行增加、修改和删除，并帮助用户在数据库中查询有价值的信息
- ✦ 是最主要的交互界面
- ✦ 数据操作工具包括：
 - ❑ 视图 (view)
 - ❑ 报表生成器 (report generators)
 - ❑ 范例查询工具 (Query-by-example tools)
 - ❑ 结构化查询语言 (SQL) 等



- ❖ 视图允许用户查看数据库文件的内容，对其进行必要的修改，完成简单的分类，并通过查找操作得到具体信息的位置
 - ❖ 视图实际上是基于查询的一个虚拟表
 - ❖ 它并不会存储在数据库中，只会在需要的时候查询并显示

The screenshot shows the Microsoft Access interface with a view named 'Order' open. The view displays a table with the following data:

Order Number	Order Date	Customer Number	Delivery Address	Concrete	Amount	Truck Number	Driver ID	Click
100000	9/1/2004	1234	55 Smith Lane	1	8	111	123456789	
100001	9/1/2004	3456	2122 E. Biscayne	1	3	222	785934444	
100002	9/2/2004	1234	55 Smith Lane	5	6	222	435296657	
100003	9/3/2004	4567	1333 Burr Ridge	2	4	333	435296657	
100004	9/4/2004	4567	1333 Burr Ridge	2	8	222	785934444	
100005	9/4/2004	5678	1222 Westminster	1	4	222	785934444	
100006	9/5/2004	1234	222 East Hampton	1	4	111	123456789	
100007	9/6/2004	2345	9 W. Palm Beach	2	5	333	785934444	
100008	9/6/2004	6789	4532 Lane Circle	1	8	222	785934444	
100009	9/7/2004	1234	987 Furlong	3	8	111	123456789	
100010	9/9/2004	6789	4532 Lane Circle	2	7	222	435296657	
100011	9/9/2004	4567	3500 Tomahawk	5	6	222	785934444	
*	0	0	0	0	0	0	0	0



- ❖ 结构化查询语言(Structured Query Language, SQL)，是一种特殊目的的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统
- ❖ 不过各种通行的数据库系统在其实践过程中都对SQL规范作了某些编改和扩充。所以，实际上不同数据库系统之间的SQL不能完全相互通用



❖ 数据定义（DDL）

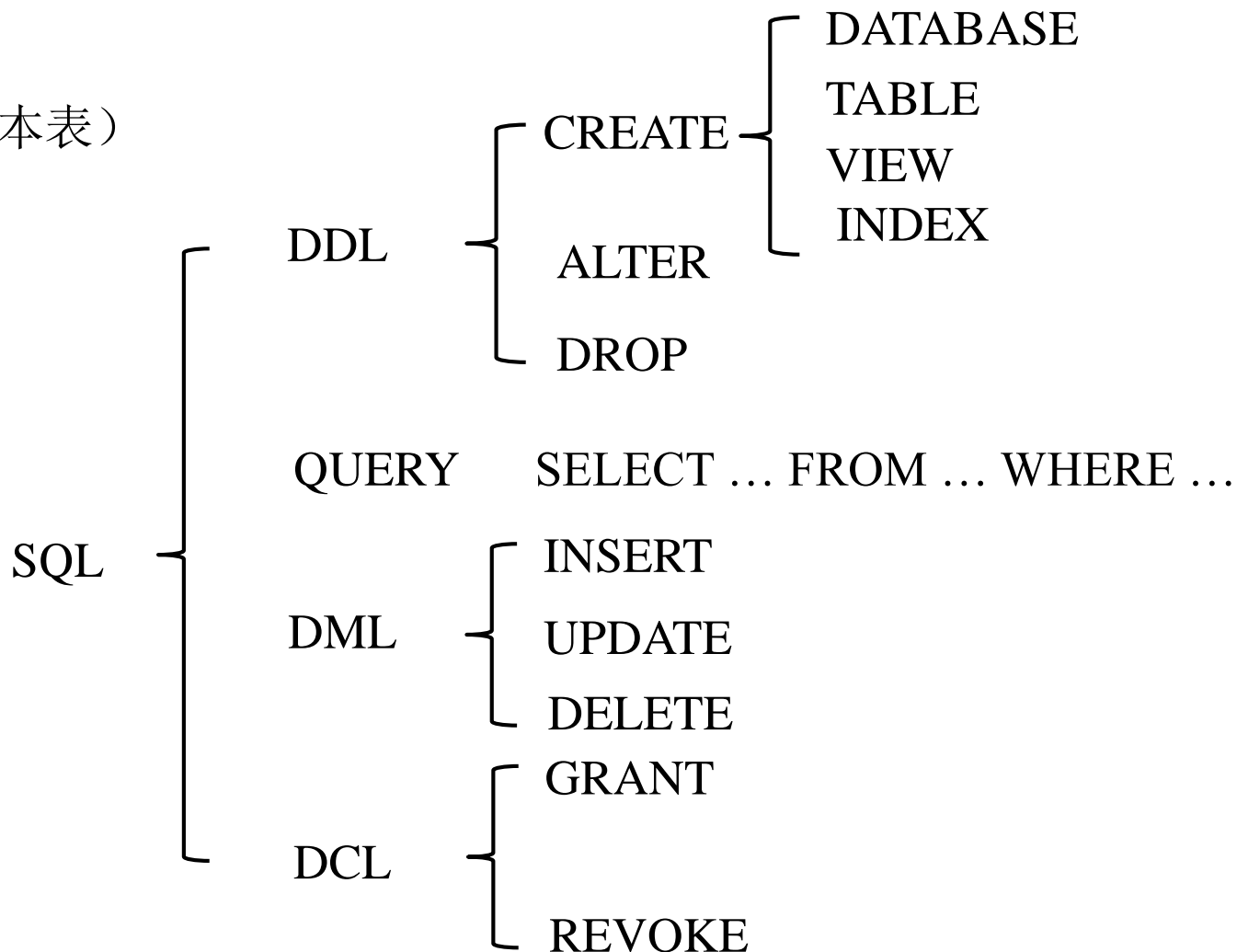
- ❖ 定义、删除、修改关系模式（基本表）
- ❖ 定义、删除视图（View）
- ❖ 定义、删除索引（Index）

❖ 数据操纵（DML）

- ❖ 数据查询
- ❖ 数据增、删、改

❖ 数据控制（DCL）

- ❖ 用户访问权限的授予、收回





✦ 高度非过程化的语言

- ✦ 用户只需提出“干什么”，至于“怎么干”由DBMS解决。

✦ 面向集合的语言

- ✦ 每一个SQL的操作对象是一个或多个关系，操作的结果也是一个关系。

✦ 一种语法结构，两种使用方式

- ✦ 即可独立使用，又可嵌入到宿主语言中使用，具有自主型和宿主型两种特点。

✦ 具有查询、操作、定义和控制四种语言一体化的特点。

✦ 语言简洁、易学易用

- ✦ 核心功能只有9个动词：**SELECT、CREATE、DROP、ALTER、INSERT、UPDATE、DELETE、GRANT、REVOKE**，语法简单，接近英语。



应用程序生成子系统& 数据管理子系统

❖ 应用程序生成子系统—包含帮助用户建立面向事务处理的应用程序

- ❖ 建立数据输入屏幕功能
- ❖ 为特定的DBMS选定程序设计语言
- ❖ 建立公共的操作交互界面

❖ 数据管理子系统—帮助管理整个数据库环境通过提供

- ❖ 备份与恢复
- ❖ 安全管理
- ❖ 查询优化
- ❖ 重新组织
- ❖ 并发控制功能
- ❖ 变动管理



最新数据库流行度排名

357 systems in ranking, November 2019

Rank			DBMS	Database Model	Score		
Nov 2019	Oct 2019	Nov 2018			Nov 2019	Oct 2019	Nov 2018
1.	1.	1.	Oracle +	Relational, Multi-model i	1336.07	-19.81	+34.96
2.	2.	2.	MySQL +	Relational, Multi-model i	1266.28	-16.78	+106.39
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	1081.91	-12.81	+30.36
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	491.07	+7.16	+50.83
5.	5.	5.	MongoDB +	Document, Multi-model i	413.18	+1.09	+43.70
6.	6.	6.	IBM Db2 +	Relational, Multi-model i	172.60	+1.83	-7.27
7.	7.	↑ 8.	Elasticsearch +	Search engine, Multi-model i	148.40	-1.77	+4.94
8.	8.	↓ 7.	Redis +	Key-value, Multi-model i	145.24	+2.32	+1.06
9.	9.	9.	Microsoft Access	Relational	130.07	-1.10	-8.36
10.	10.	↑ 11.	Cassandra +	Wide column	123.23	+0.01	+1.48
11.	11.	↓ 10.	SQLite +	Relational	121.03	-1.60	-1.68
12.	12.	12.	Splunk	Search engine	89.06	+2.23	+8.69
13.	13.	↑ 14.	MariaDB +	Relational, Multi-model i	85.57	-1.20	+12.32
14.	14.	↑ 15.	Hive +	Relational	84.22	-0.52	+19.65
15.	15.	↓ 13.	Teradata +	Relational, Multi-model i	80.35	+1.61	+1.04
16.	16.	↑ 21.	Amazon DynamoDB +	Multi-model i	61.37	+1.19	+7.56
17.	17.	↓ 16.	Solr	Search engine	57.78	+0.21	-3.10
18.	18.	↑ 20.	FileMaker	Relational	55.73	-0.94	-0.02
19.	19.	↓ 18.	SAP Adaptive Server	Relational	55.29	-0.54	-1.27
20.	20.	↓ 19.	SAP HANA +	Relational, Multi-model i	55.11	-0.24	-0.76



下载

✚ <https://dev.mysql.com/downloads/>

✚ 安装过程基本上按照默认设置就可以

MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- **MySQL Installer for Windows**
- MySQL for Excel
- MySQL for Visual Studio
- MySQL Notifier
- Connector/C (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives

由于网络问题，
建议选择直接下
载离线安装包

General Availability (GA) Releases

MySQL Installer 8.0.18

Select Operating System:
Microsoft Windows

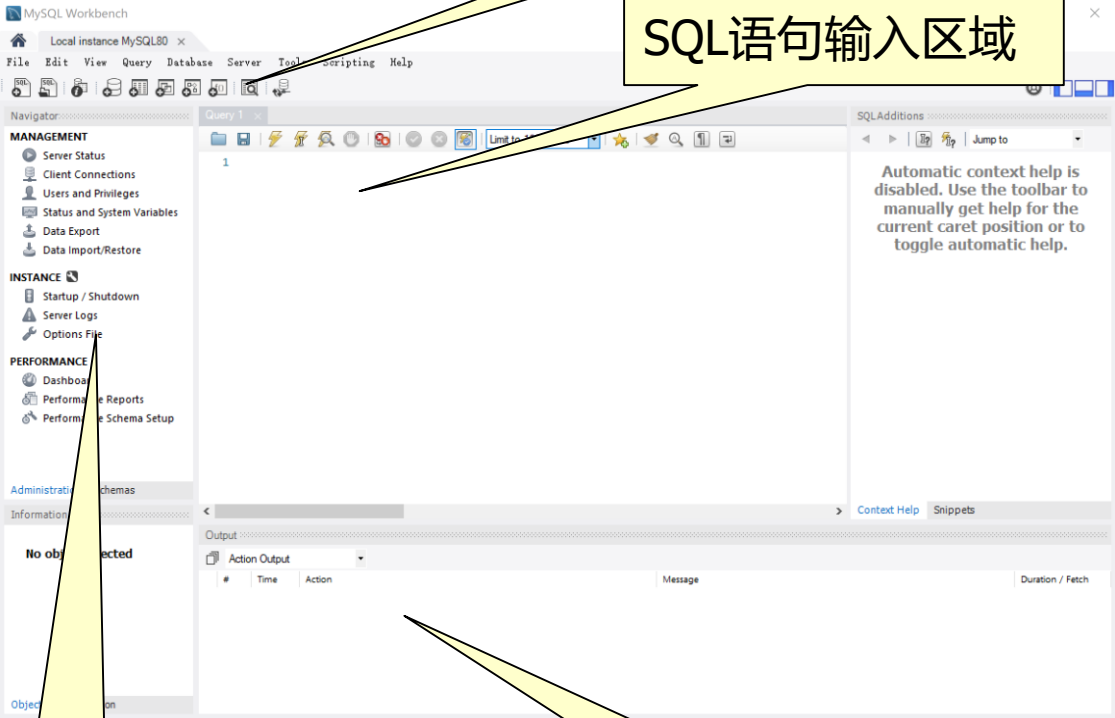
Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer	8.0.18	18.6M	Download
<small>(mysql-installer-web-community-8.0.18.0.msi) MD5: c509966c1033462027a009cc51a98c74 Signature</small>			
Windows (x86, 32-bit), MSI Installer	8.0.18	415.1M	Download
<small>(mysql-installer-community-8.0.18.0.msi) MD5: 906b5f84343d487f716f03b5925d8286 Signature</small>			

! We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.



操作界面



SQL语句输入区域
(完全依赖SQL语句
句进行操作)



数据库展示
、管理区域

结果输出区域