# MySQL从入门到精通











**Uber** 





MySQL 是一款非常流行的数据库管理系统













### 软件开发岗位要求





### 职位描述

- 1. 能够熟练使用主流开发框架(SpringMVC/Spring/MyBatis/Hibernate/Struts2等);
- 2. 熟悉Spring Boot、RPC框架;
- 3. 熟悉数据库基本原理, 熟悉MySQL, 了解SQL的性能优化;
- 4. 熟悉html5/javascript/css等前端技术;
- 5. 自我驱动的学习和工作习惯,对未知技术和领域能快速掌握并实践;





#### 职位描述

#### 技能要求:

Java, MySQL, 数据库, SQL, Javascript, jQuery

#### 岗位要求:

- 1.大学本科及以上学历, 计算机相关专业;
- 2.2年以上Java技术开发经验,有带人经验更佳,具备一定架构设计经验,至少2个大型软件设计和研发经验
- 3.精通JAVA相关技术、J2EE架构和设计模式
- 4.熟练使用Oracle、SqlServer、MySQL数据库、具备一定的数据库优化经验
- 5.熟悉常用的缓存机制(memcached, Redis等)和消息队列技术



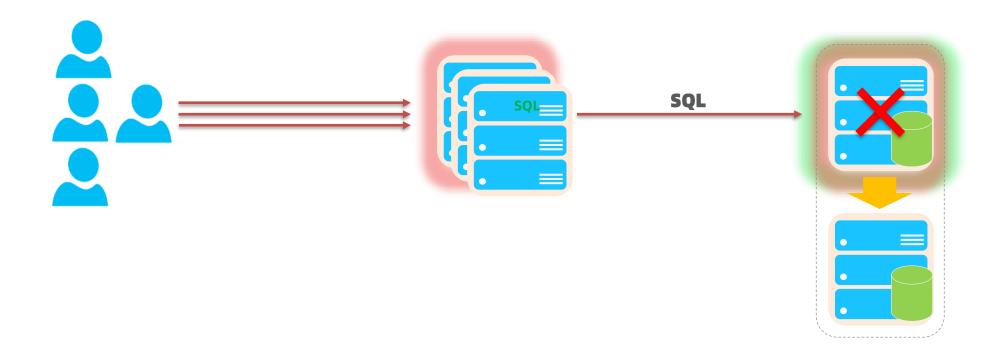
### 面试题

- 什么是事务,以及事务的四大特性?
- 事务的隔离级别有哪些,MySQL默认是哪个?
- 内连接与左外连接的区别是什么?
- 常用的存储引擎? InnoDB与MyISAM的区别?
- MySQL默认InnoDB引擎的索引是什么数据结构?
- 如何查看MySQL的执行计划?
- 索引失效的情况有哪些?
- 什么是回表查询?
- 什么是MVCC?
- MySQL主从复制的原理是什么?
- 主从复制之后的读写分离如何实现?
- 数据库的分库分表如何实现?



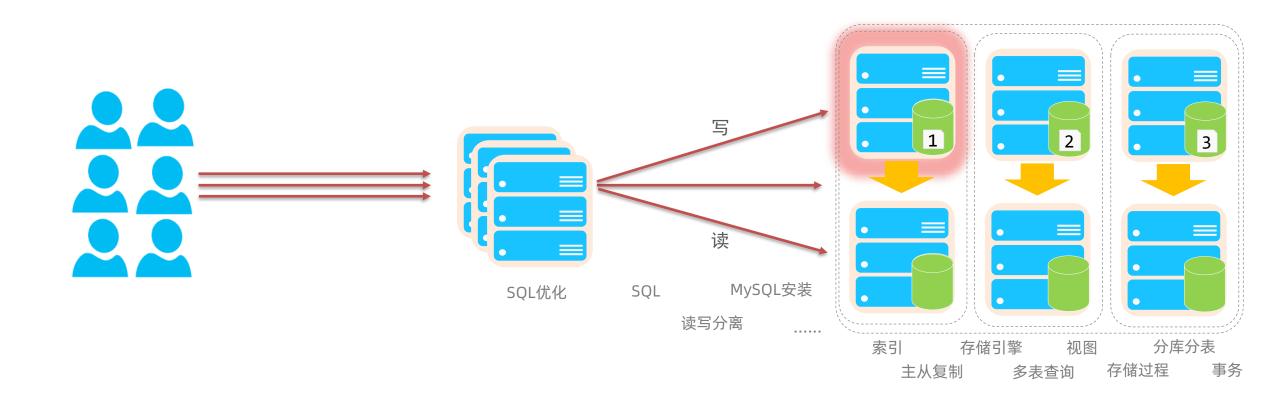


### 课程规划





### 课程规划





### 课程规划

### 基础篇

- 1. MySQL概述
- 2. SQL-DDL
- 3. SQL-DML
- 4. SQL-DQL
- 5. SQL-DCL
- 6. MySQL函数
- 7. 约束
- 8. 多表查询
- 9. 事务

### 进阶篇

- 1. MySQL体系结构
- 2. MySQL存储引擎
- 3. 索引
- 4. SQL优化
- 5. 视图
- 6. 存储过程/函数
- 7. 触发器
- 8. 锁
- 9. InnoDB存储引擎
- 10. MySQL管理

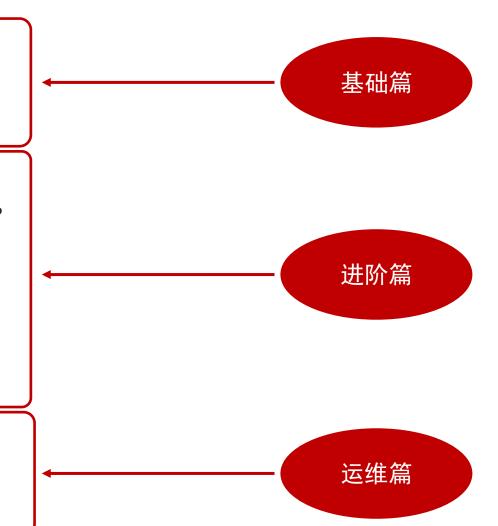
### 运维篇

- 1. MySQL日志
- 2. MySQL复制
- 3. MyCat概述
- 4. 分片相关概念
- 5. 分片配置
- 6. 分片规则
- 7. 读写分离



### 数据库相关面试题

- 什么是事务,以及事务的四大特性?
- 事务的隔离级别有哪些,MySQL默认是哪个?
- 内连接与左外连接的区别是什么?
- 常用的存储引擎? InnoDB与MyISAM的区别?
- MySQL默认InnoDB引擎的索引是什么数据结构?
- 如何查看MySQL的执行计划?
- | 索引失效的情况有哪些?
- 什么是回表查询?
- 什么是MVCC?
- MySQL主从复制的原理是什么?
- 主从复制之后的读写分离如何实现?
- 数据库的分库分表如何实现?









- ◆ MySQL概述
- ♦ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



- ◆ MySQL概述
- ♦ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



## MySQL概述

- 数据库相关概念
- MySQL数据库



## MySQL概述

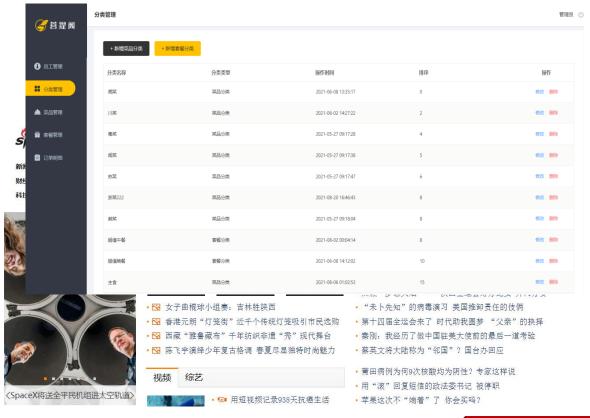
- 数据库相关概念
- MySQL数据库



### ● 数据库

Database, 简称DB。按照一定的数据结构来组织、存储和管理数据的仓库。







### 数据库管理系统

Database Management System,一种操纵和管理数据库的大型软件,用于创建、使用和维护数据库,简称DBMS。





### • 数据库管理系统

▶ 关系型数据库(RDBMS)

概念: 关系型数据库, 是建立在关系模型基础上, 由多张相互连接的二维表组成的数据库。

特点:

1. 使用表存储数据,格式统一,便于维护

2. 使用SQL语句操作,标准统一,使用方便

3. 数据存储在磁盘中,安全

概念: Not-Only SQL,泛指非关系型数据库,是对关系型数据库的补充。

特点:

1. 数据结构灵活

2. 伸缩性强

id	name	price	type
1	MySQL	100	数据库
2	Oracle	188	数据库
3	Spring	88	Java
4	Mybatis	58	Java



### 数据库管理系统

	Rank						ore	
Sep 2021	Aug 2021	Sep 2020	DBMS	Database Model	Sep 2021	Aug 2021	Sep 2020	
1.	1.	1.	Oracle 🚹	Relational, Multi-model 👔	1271.55	+2.29	-97.82	
2.	2.	2.	MySQL 🚹	Relational, Multi-model 👔	1212.52	-25.69	-51.72	
3.	3.	3.	Microsoft SQL Server 🖽	Relational, Multi-model 👔	970.85	-2.50	-91.91	
4.	4.	4.	PostgreSQL []	Relational, Multi-model 👔	577.50	+0.45	+35.22	
5.	5.	5.	IBM Db2	Relational, Multi-model 👔	166.56	+1.09	+5.32	
6.	6.	6.	SQLite 🚻	Relational	128.65	-1.16	+1.98	
7.	7.	7.	Microsoft Access	Relational	116.94	+2.10	-1.51	
8.	8.	8.	MariaDB 🚹	Relational, Multi-model 👔	100.70	+1.72	+9.09	
9.	9.	<b>1</b> 0.	Hive 😷	Relational	85.58	+1.64	+14.41	
10.	10.	<b>1</b> 11.	Microsoft Azure SQL Database	Relational, Multi-model 🛐	78.26	+3.11	+17.81	











- SQL
  - 结构化查询语言(Structured Query Language),简称SQL。
  - 是一种操作关系型数据库的编程语言,定义了一套操作关系型数据库统一标准。

SQL













## MySQL概述

- 数据库相关概念
- MySQL数据库



● 版本

MySQL官方提供了两种不同的版本:

社区版(MySQL Community Server)
 免费, MySQL不提供任何技术支持

商业版(MySQL Enterprise Edition)
 收费,可以试用30天,官方提供技术支持

本课程采用的是MySQL的最新社区版(MySQL Community Server 8.0.26)。

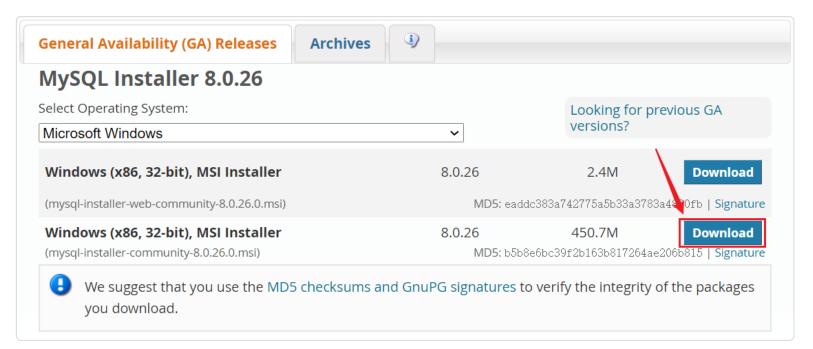


下载

下载地址: <u>https://dev.mysql.com/downloads/windows/installer/8.0.html</u>

### MySQL Community Downloads

MySQL Installer







● 安装



参考资料中提供的MySQL安装文档。



- 启动与停止
  - 启动

net start(mysql80

停止

net stop (mysql80)

MySQL Installer

Windows Service

MySQL Server 8.0.26

✓ Configure MySQL Server as a Windows Service

Windows Service Details

Please specify a Windows Service name to be used for this MySQL Server instance.

A unique name is required to seach instance.

Windows Service Name: MySQL80

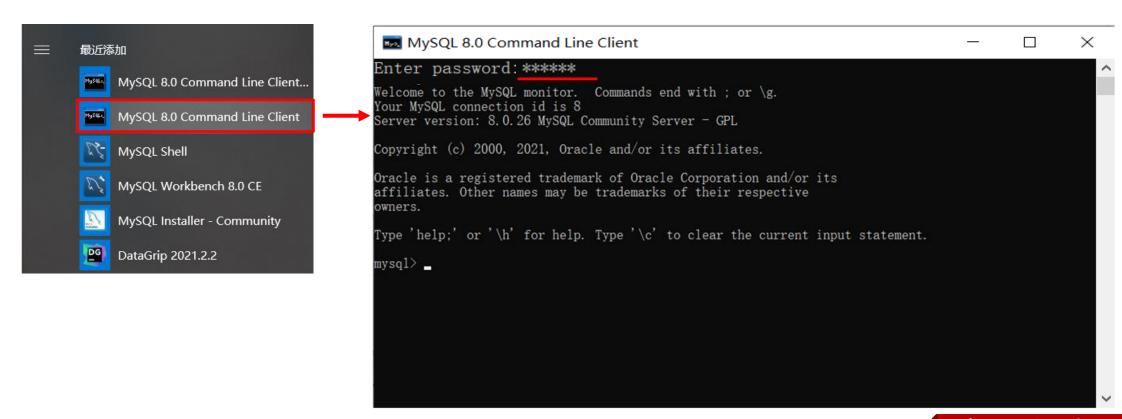
✓ Start the MySQL Server at System Startup

注意: 默认mysql是开机自动启动的。



● 客户端连接

方式一: MySQL提供的客户端命令行工具





### ● 客户端连接

方式二:系统自带的命令行工具执行指令

mysql [-h 127.0.0.1] [-P 3306] -u root -p

```
Microsoft Windows [版本 10.0.17763.1217]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\My>mysql -u root -p
Enter password: *******
Welcome to the MySQL monitor. Commands end with; or \g.
Your MySQL connection id is 9
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

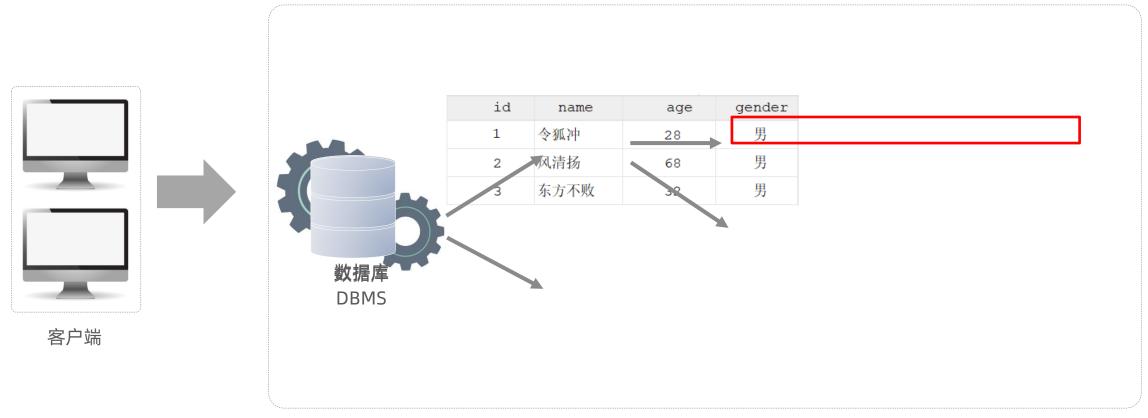
Type 'help:' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

注意: 使用这种方式时,需要配置PATH环境变量。



● 数据模型





- ◆ MySQL概述
- ♦ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



### ● SQL通用语法

- 1. SQL语句可以单行或多行书写,以分号结尾。
- 2. SQL语句可以使用空格/缩进来增强语句的可读性。
- 3. MySQL数据库的SQL语句不区分大小写,关键字建议使用大写。
- 4. 注释:
  - 单行注释: -- 注释内容 或 # 注释内容(MySQL特有)
  - 多行注释: /\* 注释内容 \*/



- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



### SQL分类

分类	全称	说明
DDL	Data Definition Language	数据定义语言,用来定义数据库对象(数据库,表,字段)
DML	Data Manipulation Language	数据操作语言,用来对数据库表中的数据进行增删改
DQL	Data Query Language	数据查询语言,用来查询数据库中表的记录
DCL	Data Control Language	数据控制语言,用来创建数据库用户、控制数据库的访问权限





id	name	age	gender
1	令狐冲	28	男
2	风清扬	68	男
3	东方不败	32	男



- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



- DDL-数据库操作
  - ▶ 查询

查询所有数据库

SHOW DATABASES;

查询当前数据库

SELECT DATABASE();

▶ 创建

CREATE DATABASE [IF NOT EXISTS] 数据库名 [DEFAULT CHARSET 字符集] [COLLATE 排序规则];

▶ 删除

DROP DATABASE [ IF EXISTS ] 数据库名;

▶ 使用

USE 数据库名;



- DDL-表操作-查询
  - 查询当前数据库所有表

SHOW TABLES;

> 查询表结构

DESC 表名;

> 查询指定表的建表语句

SHOW CREATE TABLE 表名;



● DDL-表操作-创建

```
      CREATE TABLE 表名(

      字段1 字段1类型[COMMENT 字段1注释],

      字段2 字段2类型[COMMENT 字段2注释],

      字段3 字段3类型[COMMENT 字段3注释],

      ......

      字段n 字段n类型[COMMENT 字段n注释]

      )[COMMENT 表注释];
```

id	name	age	gender
1	令狐冲	28	男
2	风清扬	68	男
3	东方不败	32	男

注意:[...]为可选参数,最后一个字段后面没有逗号



● DDL-表操作-数据类型

MySQL中的数据类型有很多,主要分为三类:数值类型、字符串类型、日期时间类型。

参照 《MySQL数据类型》



# 国 案例

### 根据需求创建表(设计合理的数据类型、长度)

#### 设计一张员工信息表,要求如下:

- 1. 编号 (纯数字)
- 2. 员工工号 (字符串类型,长度不超过10位)
- 3. 员工姓名(字符串类型,长度不超过10位)
- 4. 性别(男/女,存储一个汉字)
- 5. 年龄(正常人年龄,不可能存储负数)
- 6. 身份证号(二代身份证号均为18位,身份证中有X这样的字符)
- 7. 入职时间(取值年月日即可)



● DDL-表操作-修改

添加字段

ALTER TABLE 表名 ADD 字段名 类型(长度)[COMMENT 注释][约束];

#### 案例:

为emp表增加一个新的字段"昵称"为nickname, 类型为varchar(20)

ALTER TABLE emp ADD nickname varchar(20) COMMENT '昵称';



● DDL-表操作-修改

修改数据类型

ALTER TABLE 表名 MODIFY 字段名 新数据类型(长度);

#### 修改字段名和字段类型

ALTER TABLE 表名 CHANGE 旧字段名 新字段名 类型(长度)[COMMENT 注释][约束];

#### 案例:

将emp表的nickname字段修改为username, 类型为varchar(30)

ALTER TABLE emp CHANGE nickname username varchar(30) COMMENT '昵称';



● DDL-表操作-修改

删除字段

ALTER TABLE 表名 DROP 字段名;

#### 案例:

将emp表的字段username删除

ALTER TABLE emp DROP username;



● DDL-表操作-修改

修改表名

ALTER TABLE 表名 RENAME TO 新表名;

#### 案例:

将emp表的表名修改为 employee

ALTER TABLE emp RENAME TO employee;



● DDL-表操作-删除

删除表

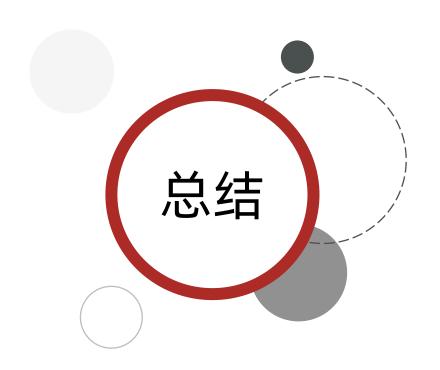
DROP TABLE [IF EXISTS] 表名;

删除指定表,并重新创建该表

TRUNCATE TABLE 表名;

注意: 在删除表时, 表中的全部数据也会被删除。





#### 1. DDL-数据库操作

```
SHOW DATABASES;

CREATE DATABASE 数据库名;

USE 数据库名;

SELECT DATABASE();

DROP DATABASE 数据库名;
```

#### 2. DDL-表操作

```
SHOW TABLES;

CREATE TABLE 表名(字段字段类型,字段字段类型);

DESC 表名;

SHOW CREATE TABLE 表名;

ALTER TABLE 表名 ADD/MODIFY/CHANGE/DROP/RENAME TO ...;

DROP TABLE 表名;
```



MySQL图形化界面



Sqlyog



Navicat



DataGrip



- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



#### ● DML-介绍

DML英文全称是Data Manipulation Language(数据操作语言),用来对数据库中表的数据记录进行增删改操作。

- ➤ 添加数据 (INSERT)
- ▶ 修改数据 (UPDATE)
- ▶ 删除数据 (DELETE)

id	name	age	gender
1	令狐冲	23	男



- DML-添加数据
  - 1. 给指定字段添加数据

INSERT INTO 表名(字段名1,字段名2,...) VALUES(值1,值2,...);

2. 给全部字段添加数据

INSERT INTO 表名 VALUES (值1, 值2, ...);

3. 批量添加数据

INSERT INTO 表名(字段名1,字段名2,...) VALUES(值1,值2,...),(值1,值2,...),(值1,值2,...);

INSERT INTO 表名 VALUES (值1,值2,…),(值1,值2,…),(值1,值2,…);

#### 注意:

- 插入数据时, 指定的字段顺序需要与值的顺序是一一对应的。
- 字符串和日期型数据应该包含在引号中。
- 插入的数据大小,应该在字段的规定范围内。



● DML-修改数据

UPDATE 表名 SET 字段名1 = 值1,字段名2 = 值2,....[WHERE 条件];

注意:修改语句的条件可以有,也可以没有,如果没有条件,则会修改整张表的所有数据。



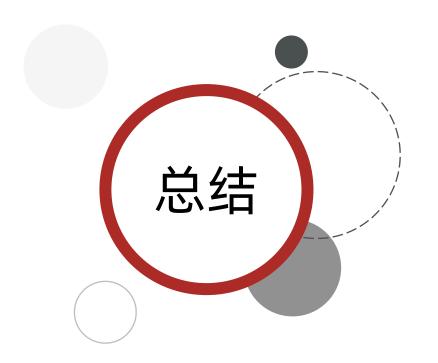
● DML-删除数据

DELETE FROM 表名 [WHERE 条件]

#### 注意:

- DELETE 语句的条件可以有,也可以没有,如果没有条件,则会删除整张表的所有数据。
- DELETE 语句不能删除某一个字段的值(可以使用UPDATE)。





#### 1. 添加数据

INSERT INTO 表名 (字段1,字段2,...) VALUES(值1,值2,...)[,(值1,值2,...)...];

#### 2. 修改数据

UPDATE 表名 SET 字段1 = 值1,字段2 = 值2 [WHERE 条件];

#### 3. 删除数据

DELETE FROM 表名[WHERE 条件];



- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



● DQL-介绍

DQL英文全称是Data Query Language(数据查询语言),数据查询语言,用来查询数据库中表的记录。

查询关键字: SELECT



● DQL-语法

SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



● DQL-语法

SELECT 字段列表 FROM 表名列表

WHERE

条件列表

GROUP BY

分组字段列表

HAVING

分组后条件列表

ORDER BY

排序字段列表

LIMIT

分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



- DQL-基本查询
  - 1. 查询多个字段

SELECT 字段1,字段2,字段3 ... FROM 表名;

SELECT \* FROM 表名;

#### 2. 设置别名

SELECT 字段1 [AS 别名1],字段2 [AS 别名2] ... FROM 表名;

3. 去除重复记录

SELECT DISTINCT 字段列表 FROM 表名;

注意:\*号代表查询所有字段,在实际开发中尽量少用(不直观、影响效率)。



● DQL-语法

SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



- DQL-条件查询
  - 1. 语法

SELECT 字段列表 FROM 表名 WHERE 条件列表;

#### 2. 条件

比较运算符	功能
>	大于
>=	大于等于
<	小于
<=	小于等于
=	等于
<> 或!=	不等于
BETWEEN AND	在某个范围之内(含最小、最大值)
IN()	在in之后的列表中的值,多选一
LIKE 占位符	模糊匹配(_匹配单个字符,%匹配任意个字符)
IS NULL	是NULL

逻辑运算符	功能
AND 或 &&	并且 (多个条件同时成立)
OR 或 ‖	或者 (多个条件任意一个成立)
NOT 或!	非,不是



● DQL-语法

SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



- DQL-聚合函数
  - 1. 介绍

将一列数据作为一个整体,进行纵向计算。

2. 常见聚合函数

函数	功能
count	统计数量
max	最大值
min	最小值
avg	平均值
sum	求和

#### 3. 语法

SELECT 聚合函数(字段列表) FROM 表名;

注意: null值不参与所有聚合函数运算。



● DQL-语法

SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



- DQL-分组查询
  - 1. 语法

SELECT 字段列表 FROM 表名「WHERE 条件] GROUP BY 分组字段名[HAVING 分组后过滤条件];

- 2. where与having区别
- ▶ 执行时机不同:where是分组之前进行过滤,不满足where条件,不参与分组;而having是分组之后对结果进行过滤。
- 》 判断条件不同: where不能对聚合函数进行判断, 而having可以。

#### 注意

- 执行顺序: where > 聚合函数 > having 。
- 分组之后,查询的字段一般为聚合函数和分组字段,查询其他字段无任何意义。



● DQL-语法

SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



- DQL-排序查询
  - 1. 语法

SELECT 字段列表 FROM 表名 ORDER BY 字段1 排序方式1,字段2 排序方式2;

- 2. 排序方式
- > ASC: 升序 (默认值)
- > DESC: 降序

注意:如果是多字段排序,当第一个字段值相同时,才会根据第二个字段进行排序。



● DQL-语法

SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数

- 基本查询
- 条件查询 (WHERE)
- 聚合函数 (count、max、min、avg、sum)
- 分组查询 (GROUP BY)
- 排序查询 (ORDER BY)
- 分页查询 (LIMIT)



- DQL-分页查询
  - 1. 语法

SELECT 字段列表 FROM 表名 LIMIT 起始索引, 查询记录数;

#### 注意

- 起始索引从0开始, 起始索引 = (查询页码 1) \* 每页显示记录数。
- 分页查询是数据库的方言,不同的数据库有不同的实现,MySQL中是LIMIT。
- 如果查询的是第一页数据, 起始索引可以省略, 直接简写为 limit 10。





#### 按照需求完成如下DQL语句编写

- 1. 查询年龄为20,21,22,23岁的员工信息。
- 2. 查询性别为 男 , 并且年龄在 20-40 岁(含)以内的姓名为三个字的员工。
- 3. 统计员工表中, 年龄小于60岁的, 男性员工和女性员工的人数。
- 4. 查询所有年龄小于等于35岁员工的姓名和年龄,并对查询结果按年龄升序排序,如果年龄相同按入职时间降序排序。
- 5. 查询性别为男,且年龄在20-40岁(含)以内的前5个员工信息,对查询的结果按年龄升序排序,年龄相同按入职时间升序排序。



● DQL-执行顺序





FROM

表名列表

WHERE

条件列表

GROUP BY

分组字段列表

HAVING

分组后条件列表

SELECT

字段列表

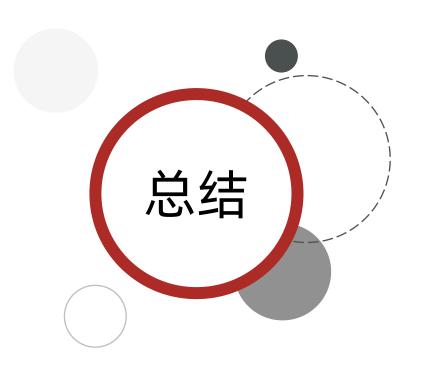
ORDER BY

排序字段列表

LIMIT

分页参数





## 1. DQL语句

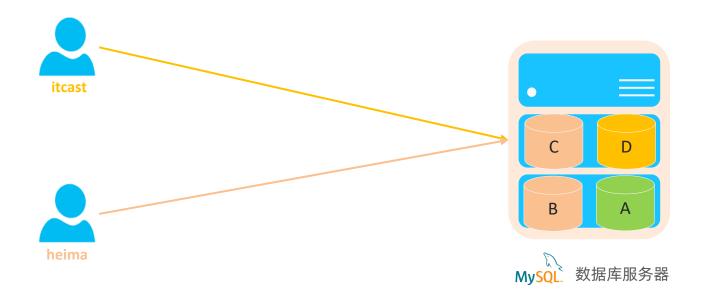


- SQL通用语法
- SQL分类
- DDL
- DML
- DQL
- DCL



#### • DCL-介绍

DCL英文全称是Data Control Language(数据控制语言),用来管理数据库用户、控制数据库的访问权限。





- DCL-管理用户
  - 1. 查询用户

```
USE mysql;
SELECT * FROM user;
```

2. 创建用户

CREATE USER '用户名'@'主机名' IDENTIFIED BY '密码';

3. 修改用户密码

ALTER USER '用户名'@'主机名' IDENTIFIED WITH mysql\_native\_password BY '新密码';

4. 删除用户

DROP USER '用户名'@'主机名';

#### 注意:

- 主机名可以使用%通配。
- 这类SQL开发人员操作的比较少,主要是DBA( Database Administrator 数据库管理员)使用。



## **SQL**

#### ● DCL-权限控制

MySQL中定义了很多种权限,但是常用的就以下几种:

权限	说明
ALL, ALL PRIVILEGES	所有权限
SELECT	查询数据
INSERT	插入数据
UPDATE	修改数据
DELETE	删除数据
ALTER	修改表
DROP	删除数据库/表/视图
CREATE	创建数据库/表

其他权限描述及含义,可以直接参考官方文档。



### SQL

- DCL-权限控制
  - 1. 查询权限

SHOW GRANTS FOR '用户名'@'主机名';

2. 授予权限

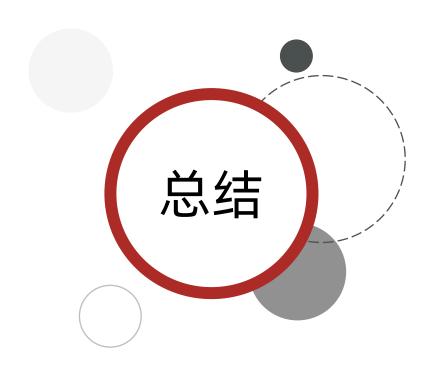
GRANT 权限列表 ON 数据库名.表名 TO '用户名'@'主机名';

3. 撤销权限

REVOKE 权限列表 ON 数据库名.表名 FROM '用户名'@'主机名';

#### 注意:

- 多个权限之间,使用逗号分隔
- 授权时, 数据库名和表名可以使用\*进行通配, 代表所有。



#### 1. 用户管理

CREATE USER '用户名'@'主机名' IDENTIFIED BY '密码';

ALTER USER '用户名'@'主机名' IDENTIFIED WITH mysql\_native\_password BY '密码';

DROP USER '用户名'@'主机名'

### 2. 权限控制

GRANT 权限列表 ON 数据库名.表名 TO '用户名'@'主机名';

REVOKE 权限列表 ON 数据库名.表名 FROM '用户名'@'主机名';



- ◆ MySQL概述
- ◆ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



函数 是指一段可以直接被另一段程序调用的程序或代码。



数据库表中,存储的是入职日期,如 2000-11-12,如何快速计算入职天数???



数据库表中,存储的是学生的分数值,如98、75,如何快速判定分数的等级呢??





## 函数

- 字符串函数
- 数值函数
- 日期函数
- 流程函数



## 函数

- 字符串函数
- 数值函数
- 日期函数
- 流程函数



## 字符串函数

MySQL中内置了很多字符串函数,常用的几个如下:

<b>函数</b>	功能		
CONCAT(S1,S2,Sn)	字符串拼接,将S1, S2, Sn拼接成一个字符串		
LOWER(str)	将字符串str全部转为小写		
UPPER(str)	将字符串str全部转为大写		
LPAD(str,n,pad)	左填充,用字符串pad <b>对</b> str <b>的左边进行填充,达到</b> n个字符串长度		
RPAD(str,n,pad)	右填充,用字符串pad对str的右边进行填充,达到n个字符串长度		
TRIM(str)	去掉字符串头部和尾部的空格		
SUBSTRING(str,start,len)	返回从字符串str从start位置起的len个长度的字符串		

SELECT 函数(参数);





### · 根据需求完成以下SQL编写

由于业务需求变更,企业员工的工号,统一为5位数,目前不足5位数的全部在前面补0。比如:1号员工的工号应该为00001。



## 函数

- 字符串函数
- 数值函数
- 日期函数
- 流程函数



## 数值函数

#### 常见的数值函数如下:

<b>函数</b>	功能
CEIL(x)	向上取整
FLOOR(x)	向下取整
MOD(x,y)	返回×/y <b>的模</b>
RAND()	返回0~1内的随机数
ROUND(x,y)	求参数×的四舍五入的值,保留y位小数





## · 根据需求完成以下SQL编写

通过数据库的函数,生成一个六位数的随机验证码。



## 函数

- 字符串函数
- 数值函数
- 日期函数
- 流程函数



## 日期函数

#### 常见的日期函数如下:

<b>函数</b>	功能
CURDATE()	返回当前日期
CURTIME()	返回当前时间
NOW()	返回当前日期和时间
YEAR(date)	<b>获取指定</b> date <b>的年份</b>
MONTH(date)	<b>获取指定</b> date <b>的月份</b>
DAY(date)	<b>获取指定</b> date <b>的日期</b>
DATE_ADD(date, INTERVAL expr type)	返回一个日期/时间值加上一个时间间隔expr后的时间值
DATEDIFF(date1,date2)	返回起始时间date1 和 结束时间date2之间的天数





## · 根据需求完成以下SQL编写

查询所有员工的入职天数,并根据入职天数倒序排序。



## 函数

- 字符串函数
- 数值函数
- 日期函数
- 流程函数



## 流程函数

流程函数也是很常用的一类函数,可以在SQL语句中实现条件筛选,从而提高语句的效率。

函数	功能
IF(value, t, f)	如果value为true,则返回t,否则返回f
IFNULL(value1 , value2)	如果value1不为空,返回value1,否则返回value2
CASE WHEN [val1] THEN [res1] ELSE [default] END	如果val1为true,返回res1,…否则返回default默认值
CASE [expr] WHEN [val1] THEN [res1] ELSE [default] END	如果expr的值等于val1,返回res1,…否则返回default默认值



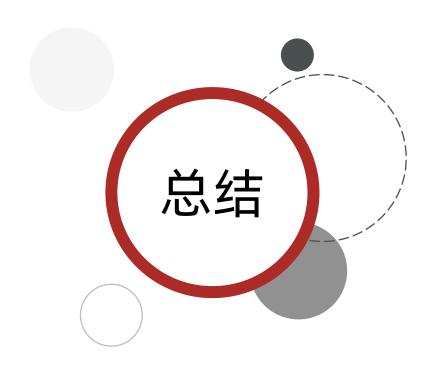


### · 根据需求完成以下SQL编写

统计班级各个学员的成绩,展示的规则如下:

- >= 85, 展示<mark>优秀</mark>
- >= 60, 展示<mark>及格</mark>
- 否则,展示不及格





#### 1. 字符串函数

CONCAT, LOWER, UPPER, LPAD, RPAD, TRIM, SUBSTRING

#### 2. 数值函数

CEIL, FLOOR, MOD, RAND, ROUND

#### 3. 日期函数

CURDATE, CURTIME, NOW, YEAR, MONTH, DAY, DATE ADD, DATEDIFF

#### 4. 流程函数

IF , IFNULL , CASE [...] WHEN ... THEN ... ELSE ... END



1495天



数据库表中,存储的是入职日期,如 2000-11-12,如何快速计算入职天数???



CASE ... WHEN ...

数据库表中,存储的是学生的分数值,如98、75,如何快速判定分数的等级呢??

?



- ◆ MySQL概述
- ♦ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



# 约束

- 概述
- 约束演示
- 外键约束



# 约束

- 概述
- 约束演示
- 外键约束



#### 概述

1. 概念:约束是作用于表中字段上的规则,用于限制存储在表中的数据。

2. 目的:保证数据库中数据的正确、有效性和完整性。

3. 分类:

约束	描述	关键字
非空约束	限制该字段的数据不能为null	NOT NULL
唯一约束	保证该字段的所有数据都是唯一、不重复的	UNIQUE
主键约束	主键是一行数据的唯一标识,要求非空且唯一	PRIMARY KEY
默认约束	保存数据时,如果未指定该字段的值,则采用默认值	DEFAULT
检查约束(8.0.16版本之后)	保证字段值满足某一个条件	CHECK
外键约束	用来让两张表的数据之间建立连接,保证数据的一致性和完整性	FOREIGN KEY

注意:约束是作用于表中字段上的,可以在创建表/修改表的时候添加约束。



# 约束

- 概述
- 约束演示
- 外键约束



## 国 案例

### 根据需求,完成表结构的创建

字段名	字段含义	字段类型	约束条件
id	ID唯一标识	int	主键,并且自动增长
name	姓名	varchar(10)	不为空,并且唯一
age	年龄	int	大于0,并且小于等于120
status	状态	char(1)	如果没有指定该值,默认为1
gender	性别	char(1)	无

```
CREATE TABLE tb_user(
    id int AUTO_INCREMENT PRIMARY KEY COMMENT 'ID唯一标识',
    name varchar(10) NOT NULL UNIQUE COMMENT '姓名',
    age int check (age > 0 && age <= 120) COMMENT '年龄',
    status char(1) default '1' COMMENT '状态',
    gender char(1) COMMENT '性别'
);
```



# 约束

- 概述
- 约束演示
- 外键约束



#### 外键约束

#### ● 概念

外键用来让两张表的数据之间建立连接,从而保证数据的一致性和完整性。

<b>.</b> ₹id	.⊞ name	■ age	⊞job	I salary	<b>I</b> ■ entrydate	⊪ managerid	dept_id
1	金庸	66	总裁	20000	2000-01-01	<null></null>	5
2	张无忌	20	项目经理	12500	2005-12-05	1	1
3	杨逍	33	开发	8400	2000-11-03	2	1
4	韦一笑	48	开发	11000	2002-02-05	2	1
5	常遇春	43	开发	10500	2004-09-07	3	1
			员工	表 emp (子表)			

注意:目前上述的两张表,在数据库层面,并未建立外键关联,所以是无法保证数据的一致性和完整性的。



### 外键约束

- 语法
  - ▶ 添加外键

```
CREATE TABLE 表名(
字段名 数据类型,
...

[CONSTRAINT] [外键名称] FOREIGN KEY (外键字段名) REFERENCES 主表 (主表列名)
);
```

ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY(外键字段名) REFERENCES 主表(主表列名);

▶删除外键

ALTER TABLE 表名 DROP FOREIGN KEY 外键名称;

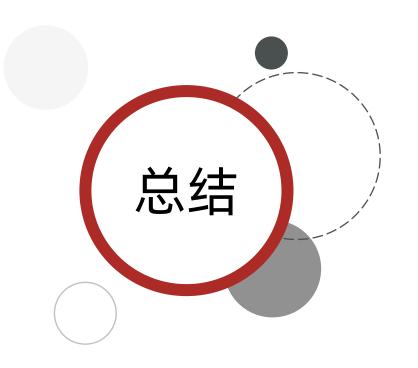


## 外键约束

● 删除/更新行为

行为	说明					
NO ACTION	当在父表中删除/更新对应记录时,首先检查该记录是否有对应外键,如果有则不允许删除/更新。(与 RESTRICT 一致)					
RESTRICT	当在父表中删除/更新对应记录时,首先检查该记录是否有对应外键,如果有则不允许删除/更新。(与 NO ACTION 一致)					
CASCADE	当在父表中删除/更新对应记录时,首先检查该记录是否有对应外键,如果有,则也删除/更新外键在子表中的记录。					
SET NULL	当在父表中删除对应记录时,首先检查该记录是否有对应外键,如果有则设置子表中该外键值为null(这就要求该外键允许取null)。					
SET DEFAULT	父表有变更时,子表将外键列设置成一个默认的值 (Innodb不支持)					

ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (外键字段) REFERENCES 主表名(主表字段名) ON UPDATE CASCADE ON DELETE CASCADE;



1. 非空约束: NOT NULL

2. 唯一约束: UNIQUE

3. 主键约束: PRIMARY KEY (自增: AUTO\_INCREMENT)

4. 默认约束: DEFAULT

5. 检查约束: CHECK

6. 外键约束: FOREIGN KEY



- ◆ MySQL概述
- ♦ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



## 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



## 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



## 多表关系

● 概述

项目开发中,在进行数据库表结构设计时,会根据业务需求及业务模块之间的关系,分析并设计表结构,由于业务之间相互关联,所以各个表结构之间也存在着各种联系,基本上分为三种:

- ▶ 一对多(多对一)
- > 多对多
- ➤ 一对一



## 多表关系

- 一对多(多对一)
  - ▶ 案例: 部门与员工的关系
  - ▶ 关系: 一个部门对应多个员工, 一个员工对应一个部门
  - > 实现: 在多的一方建立外键, 指向一的一方的主键

员工表(emp) N

<b>.</b> ₹id	.⊞ name	III age
1	张无忌	20
2	杨逍	33
3	赵敏	18
4	常遇春	43

1 部门表(dept)

laname

1 研发部

2 市场部

3 财务部

4 销售部



## 多表关系

#### ● 多对多

》 案例: 学生与课程的关系

▶ 关系: 一个学生可以选修多门课程, 一门课程也可以供多个学生选择

> 实现:建立第三张中间表,中间表至少包含两个外键,分别关联两方主键





## 多表关系

#### ● 一对一

▶ 案例: 用户与用户详情的关系

▶ 关系: 一对一关系, 多用于单表拆分, 将一张表的基础字段放在一张表中, 其他详情字段放在另一张表中, 以提升操作效率

▶ 实现: 在任意一方加入外键,关联另外一方的主键,并且设置外键为唯一的(UNIQUE)

. ↓												_
. id	III name	III age	■ gender	III phone		III d	<b>.</b> ₹id	II degree	⊞ major	I primaryschool	I≣ middleschool	■ university
1	黄渤	45	1	18800001111	l11	2	1	本科	舞蹈	静安区第一小学	静安区第一中学	北京舞蹈学院
2	冰冰	35	2	18800002222	222	Ą	2	硕士	表演	朝阳区第一小学	朝阳区第一中学	北京电影学院
3	码云	55	1	18800008888	388	2	3	本科	英语	杭州市第一小学	杭州市第一中学	杭州师范大学
4	李彦宏	50	1	18800009999	799	2	4	本科	应用数学	阳泉第一小学	阳泉区第一中学	清华大学

用户基本信息表(tb\_user)

用户教育信息表(tb\_user\_edu)



# 多表查询

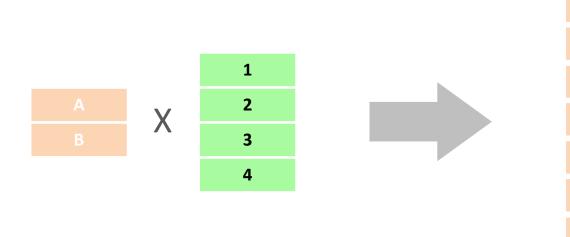
- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



## 多表查询概述

● 概述: 指从多张表中查询数据

● 笛卡尔积: 笛卡尔乘积是指在数学中,两个集合A集合 和 B集合的所有组合情况。(在多表查询时,需要消除无效的笛卡尔积)



Α	1
Α	2
А	3
А	4
В	1
В	2
В	3
В	4

笛卡尔积



## 多表查询概述

● 多表查询分类

▶ 连接查询

内连接:相当于查询A、B交集部分数据

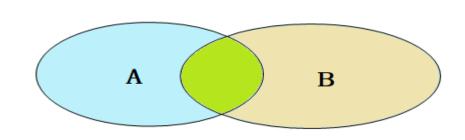
外连接:

左外连接: 查询左表所有数据, 以及两张表交集部分数据

右外连接:查询右表所有数据,以及两张表交集部分数据

自连接: 当前表与自身的连接查询, 自连接必须使用表别名

> 子查询





# 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



## 连接查询-内连接

内连接查询语法:

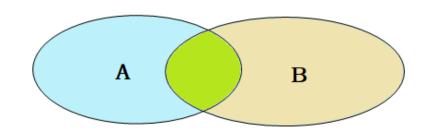
▶ 隐式内连接

SELECT 字段列表 FROM 表1,表2 WHERE 条件...;

▶ 显式内连接

SELECT 字段列表 FROM 表1 [INNER] JOIN 表2 ON 连接条件 ...;

内连接查询的是两张表交集的部分





# 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



## 连接查询-外连接

外连接查询语法:

▶ 左外连接

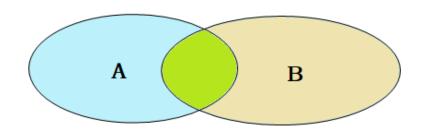
SELECT 字段列表 FROM 表1 LEFT [OUTER] JOIN 表2 ON 条件 ...;

相当于查询表1(左表)的所有数据包含表1和表2交集部分的数据

▶ 右外连接

SELECT 字段列表 FROM 表1 RIGHT [OUTER] JOIN 表2 ON 条件 ...;

相当于查询表2(右表)的所有数据包含表1和表2交集部分的数据





# 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



## 连接查询-自连接

自连接查询语法:

SELECT 字段列表 FROM 表A 别名A JOIN 表A 别名B ON 条件 ...;

自连接查询,可以是内连接查询,也可以是外连接查询。



## 联合查询-union , union all

对于union查询,就是把多次查询的结果合并起来,形成一个新的查询结果集。

SELECT 字段列表 FROM 表A ...

UNION [ ALL ]

SELECT 字段列表 FROM 表B ....;

对于联合查询的多张表的列数必须保持一致、字段类型也需要保持一致。

union all 会将全部的数据直接合并在一起, union 会对合并之后的数据去重。



# 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例



● 概念: SQL语句中嵌套SELECT语句, 称为嵌套查询, 又称子查询。

```
SELECT * FROM t1 WHERE column1 = (SELECT column1 FROM t2);
```

子查询外部的语句可以是INSERT / UPDATE / DELETE / SELECT 的任何一个。

- 根据子查询结果不同,分为:
  - ▶ 标量子查询(子查询结果为单个值)
  - ▶ 列子查询(子查询结果为一列)
  - ▶ 行子查询(子查询结果为一行)
  - ▶ 表子查询(子查询结果为多行多列)

● 根据子查询位置,分为:WHERE之后、FROM之后、SELECT之后。



● 标量子查询

子查询返回的结果是单个值(数字、字符串、日期等),最简单的形式,这种子查询成为标量子查询。

常用的操作符: = <> > >= < <=



● 列子查询

子查询返回的结果是一列(可以是多行),这种子查询称为列子查询。

常用的操作符: IN 、NOT IN 、ANY 、SOME 、ALL

操作符	描述 
IN	在指定的集合范围之内, 多选一
NOT IN	不在指定的集合范围之内
ANY	子查询返回列表中,有任意一个满足即可
SOME	与ANY等同,使用SOME <b>的地方都可以使用</b> ANY
ALL	子查询返回列表的所有值都必须满足



● 行子查询

子查询返回的结果是一行(可以是多列),这种子查询称为行子查询。

常用的操作符: = 、<> 、IN 、NOT IN



● 表子查询

子查询返回的结果是多行多列,这种子查询称为表子查询。

常用的操作符: IN



# 多表查询

- 多表关系
- 多表查询概述
- 内连接
- 外连接
- 自连接
- 子查询
- 多表查询案例

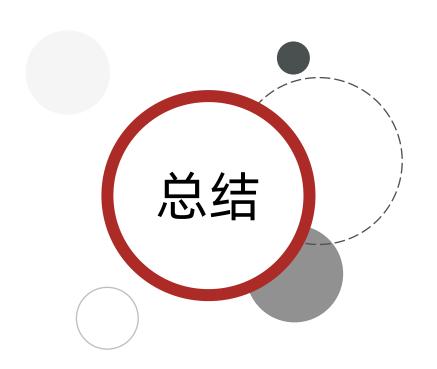




## 根据需求,完成SQL语句的编写

- 1. 查询员工的姓名、年龄、职位、部门信息。
- 2. 查询年龄小于30岁的员工姓名、年龄、职位、部门信息。
- 3. 查询拥有员工的部门ID、部门名称。
- 4. 查询所有年龄大于40岁的员工,及其归属的部门名称;如果员工没有分配部门,也需要展示出来。
- 5. 查询所有员工的工资等级。
- 6. 查询 "研发部" 所有员工的信息及工资等级。
- 7. 查询 "研发部" 员工的平均工资。
- 8. 查询工资比 "灭绝" 高的员工信息。
- 9. 查询比平均薪资高的员工信息。
- 10. 查询低于本部门平均工资的员工信息。
- 11. 查询所有的部门信息,并统计部门的员工人数。
- 12. 查询所有学生的选课情况, 展示出学生名称, 学号, 课程名称





## 1. 多表关系

一对多: 在多的一方设置外键,关联一的一方的主键

多对多: 建立中间表, 中间表包含两个外键, 关联两张表的主键

一对一:用于表结构拆分,在其中任何一方设置外键(UNIQUE),关联另一方的主键

### 2. 多表查询

#### 内连接

隐式: SELECT ... FROM 表A , 表B WHERE 条件 ...

显式: SELECT ... FROM 表A INNER JOIN 表B ON 条件 ...

#### 外连接

左外: SELECT ... FROM 表A LEFT JOIN 表B ON 条件 ...

右外: SELECT ... FROM 表A RIGHT JOIN 表B ON 条件 ...

自连接: SELECT ... FROM 表A 别名1,表A 别名2 WHERE 条件 ...

子查询:标量子查询、列子查询、行子查询、表子查询



- ◆ MySQL概述
- ♦ SQL
- ◆ 函数
- ◆ 约束
- ◆ 多表查询
- ◆ 事务



## 事务

- 事务简介
- 事务操作
- 事务四大特性
- 并发事务问题
- 事务隔离级别



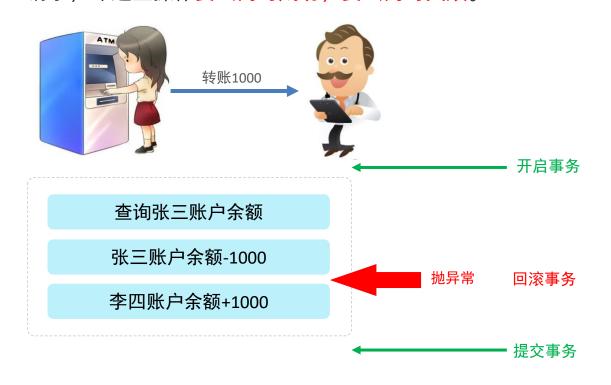
## 事务

- 事务简介
- 事务操作
- 事务四大特性
- 并发事务问题
- 事务隔离级别



## 事务简介

事务是一组操作的集合,它是一个不可分割的工作单位,事务会把所有的操作作为一个整体一起向系统提交或撤销操作请求,即这些操作要么同时成功,要么同时失败。







## 事务

- 事务简介
- 事务操作
- 事务四大特性
- 并发事务问题
- 事务隔离级别



## 事务操作

• 查看/设置事务提交方式

```
SELECT @@autocommit ;
SET @@autocommit = 0 ;
```

• 提交事务

COMMIT;

• 回滚事务

ROLLBACK;



## 事务操作

• 开启事务

START TRANSACTION 或 BEGIN;

• 提交事务

COMMIT;

• 回滚事务

ROLLBACK;



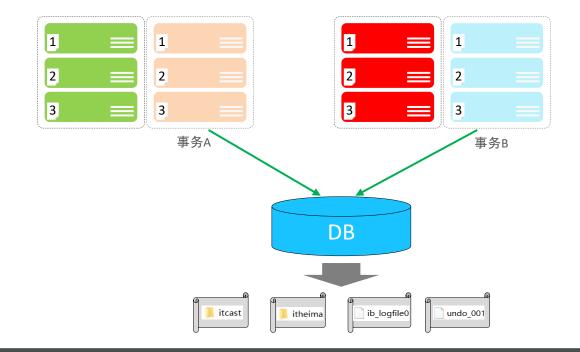
## 事务

- 事务简介
- 事务操作
- 事务四大特性
- 并发事务问题
- 事务隔离级别



## 事务四大特性(ACID)

- 原子性(Atomicity):事务是不可分割的最小操作单元,要么全部成功,要么全部失败。
- 一致性(Consistency):事务完成时,必须使所有的数据都保持一致状态。
- 隔离性(Isolation):数据库系统提供的隔离机制,保证事务在不受外部并发操作影响的独立环境下运行。
- 持久性(Durability):事务一旦提交或回滚,它对数据库中的数据的改变就是永久的。





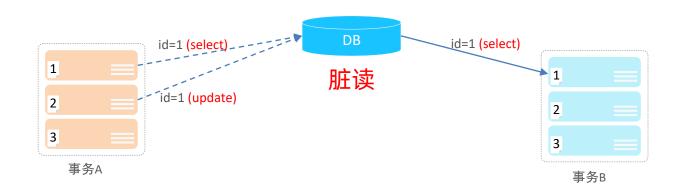
## 事务

- 事务简介
- 事务操作
- 事务四大特性
- 并发事务问题
- 事务隔离级别



## 并发事务问题

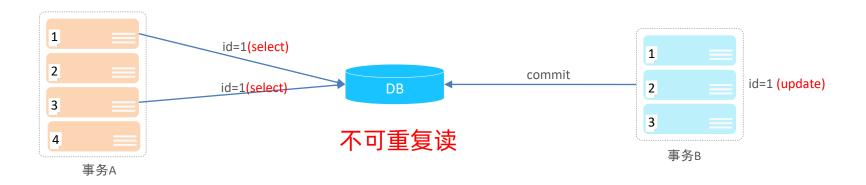
问题	描述						
脏读	一个事务读到另外一个事务还没有提交的数据。						
不可重复读	一个事务先后读取同一条记录,但两次读取的数据不同,称之为不可重复读。						
幻读	一个事务按照条件查询数据时,没有对应的数据行,但是在插入数据时,又发现这行数据已经存在,好像出现了"幻影"。						





## 并发事务问题

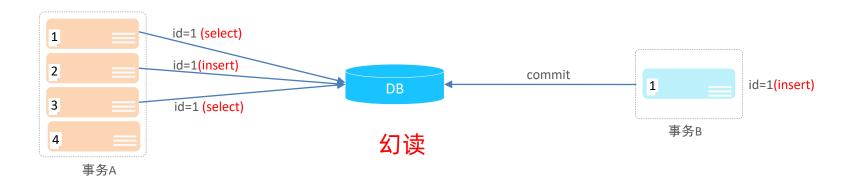
问题	描述						
脏读	一个事务读到另外一个事务还没有提交的数据。						
不可重复读	一个事务先后读取同一条记录,但两次读取的数据不同,称之为不可重复读。						
幻读	一个事务按照条件查询数据时,没有对应的数据行,但是在插入数据时,又发现这行数据已经存在,好像出现了"幻影"。						





## 并发事务问题

问题	描述				
脏读	一个事务读到另外一个事务还没有提交的数据。				
不可重复读	一个事务先后读取同一条记录,但两次读取的数据不同,称之为不可重复读。				
幻读	一个事务按照条件查询数据时,没有对应的数据行,但是在插入数据时,又发现这行数据已经存在,好像出现了"幻影"。				





## 事务

- 事务简介
- 事务操作
- 事务四大特性
- 并发事务问题
- 事务隔离级别



## 事务隔离级别

隔离级别	脏读	不可重复读	幻读
Read uncommitted	$\sqrt{}$	$\sqrt{}$	$\checkmark$
Read committed	×		
Repeatable Read( <b>默认</b> )	×	×	
Serializable	×	×	×

#### -- 查看事务隔离级别

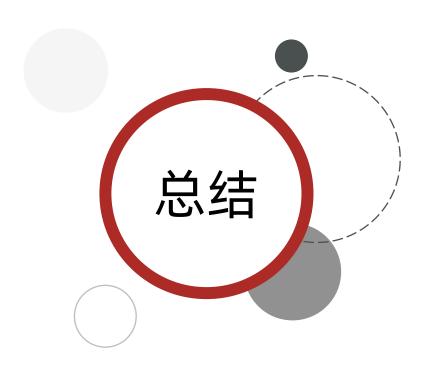
SELECT @@TRANSACTION\_ISOLATION;

#### -- 设置事务隔离级别

SET [SESSION|GLOBAL] TRANSACTION ISOLATION LEVEL {READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE}

注意:事务隔离级别越高,数据越安全,但是性能越低。





## 1. 事务简介

事务是一组操作的集合,这组操作,要么全部执行成功,要么全部执行失败。

## 2. 事务操作

START TRANSACTION; -- 开启事务

COMMIT / ROLLBACK; -- 提交/回滚事务

## 3. 事务四大特性

原子性( Atomicity )、一致性( Consistency ) 、隔离性( Isolation ) 、持久性( Durability )

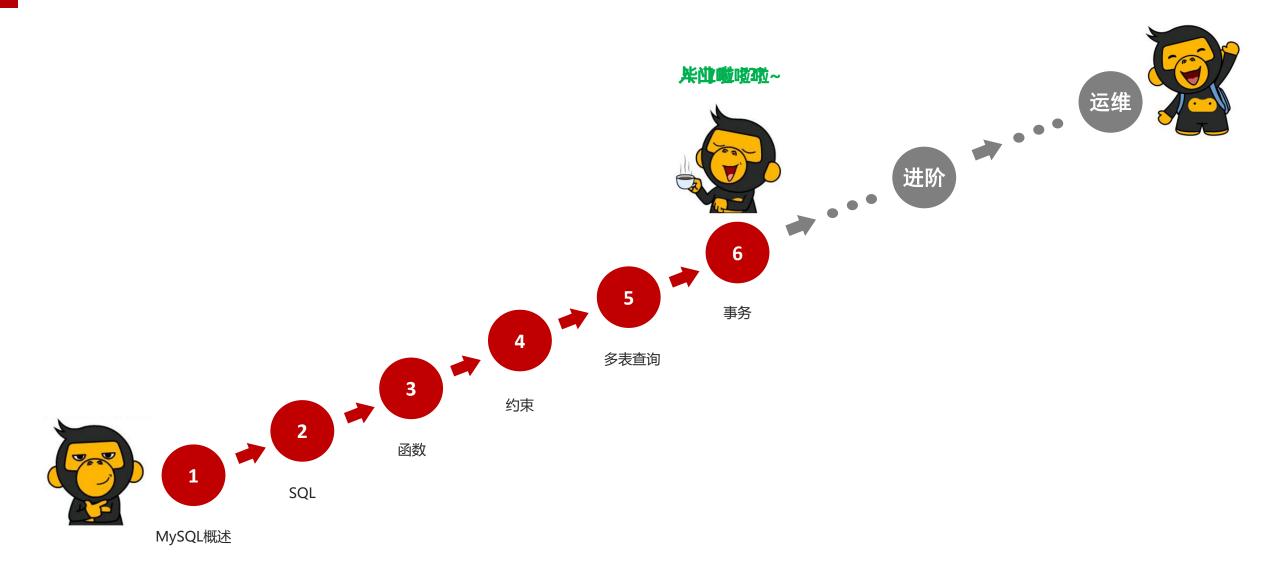
### 4. 并发事务问题

脏读、不可重复读、幻读

### 5. 事务隔离级别

READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE







传智教育旗下高端IT教育品牌