20个选择，20分

填空10空，20分

判断题10题，10分

编写SQL语句11条， 30分

编程3题 20分

模块1 数据库规划与设计

1. 数据库：有组织、可共享
2. 数据库管理系统DBMS，五大功能（定义、操纵、查询、管理、维护）
3. 数据库系统构成：软件、硬件、操作系统、数据库、DBMS、程序、DBA、用户
4. DBA（数据库管理员）管理、维护数据库系统的人员
5. ER图：绘制ER图 实体矩形、联系菱形 、属性椭圆、联系类型1:1 1:n m:n

* 1:n联系🡪主外键，多的一方的外键参照一的一方的主键
* 1:1联系🡪合并为一个关系
* n:m联系🡪独立的关系，新关系的两个外键分别参照原来的关系

1. （关系）数据模型三要素：数据结构、数据操作、完整性数据约束
2. 层次、网状、关系
3. 关系模型的数据结构——二维表 数据操作——关系代数、关系演算、SQL
4. 键——候选键（多个）、主键（有且仅有1个）、外键（定义）
   * 能够唯一标识一个元组（实体）的属性或属性集称为候选键，这些是主属性
   * 在候选键中指定其中一个作为主要候选键，即主键
   * 关系中的某个属性或属性集，虽然不是该关系的主键，但却是另外一个关系的主键，则称为外键。
5. 关系完整性约束：实体（主键）、参照（外键）、用户自定义
6. 关系异常：数据冗余、插入、删除、更新异常
7. 函数依赖：完全（定义）、部分（定义）、传递（定义）
8. 范式分析：1NF（原子不可再分） 2NF（消除部分依赖） 3NF（消除传递依赖）
9. 数据库设计的6个阶段以及每个阶段简述

* 需求分析（撰写SRS）
* 概念结构设计（ER图）
* 逻辑结构设计（转换成关系）
* 物理结构设计（表结构、存储结构设计）
* 数据库实施（建库建表建对象）
* 数据库运行维护（备份还原、安全设置）

1. 数据库模式：内模式（**存储**）、模式（包括表的结构（列名、数据类型、约束条件等）、表之间的关系（如主键 - 外键关系）以及数据的完整性规则等）、外模式（**视图**）

模块2 数据库安装与实施

1. MySQL数据库的配置文件my.ini、数据库文件存储（C:\ProgramData\MySQL\MySQL Server 8.0\Data\文件夹中（Mysql80），每个数据库一个同名的文件夹）、8.0默认的存储引擎InnoDB
2. 系统数据库：Information\_schem(数据库服务器的信息)、mysql(MySQL服务器的用户权限)、performance\_schema(监视服务器在运行时的性能)、sys(MySQL服务器的性能诊断)。使用net命令启动和停止MySQL服务

* net start mysql80
* net stop mysql80

1. 数据库创建语句 create database score
2. 修改、删除数据库、表、对象 Alter Drop
3. 约束关键字 null/not null PRIMARY KEY (student\_id, course\_id)

FOREIGN KEY (course\_id) REFERENCES course(course\_id)

unique check default

1. 写出创建数据表的语句（编程）

CREATE TABLE score\_bk (

term VARCHAR(10) NOT NULL,

student\_id VARCHAR(36) NOT NULL,

PRIMARY KEY (student\_id, course\_id),

FOREIGN KEY (student\_id) REFERENCES student(student\_id),

);

1. 当构成主键的字段超过1个时，只能使用表级约束进行定义

create table score

(

student\_id varchar(36) not null,

course\_id varchar(36) not null ,

score int，

constraint pk\_score primary key(student\_id,course\_id)

)

1. 索引的作用和类型(普通/唯一性、单列/复合)

模块3 数据查询与维护

1. Insert语句格式 INSERT INTO table\_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...);

(value1\_2, value2\_2, value3\_2,...),

Update语句格式: UPDATE table\_name

SET column1 = value1,

column2 = value2,-- 可以继续设置更多要更新的列和对应的值

WHERE condition;

基于嵌套子查询的Update: UPDATE table\_name

SET column1 = (subquery1),--嵌套子查询语句

column2 = (subquery2),-- 可以继续设置更多基于子查询更新的列

WHERE condition;

例子：UPDATE employees

SET salary=(SELECT AVG(salary) FROM another\_table WHERE department = employees.department)

WHERE age > 30;

Delete语句格式，基于嵌套子查询的Delete

Delete from score

where student\_id=(select student\_id from student where name='汪铭')

1. 简单查询：分组group by、排序order by
2. 模糊查询：关键字 like 通配符 % \_(单个字符)
3. 内连接查询：4张表连接，用在视图
4. 嵌套子查询（**子查询可独立执行，**先执行子查询，将结果用于父查询，**共执行** 2次查询）

模块4 数据库编程

1. 全局变量SET GLOBAL 变量名 = 值; 用户变量@（赋值）SET @变量名 = 值;

局部变量（定义、赋值）:

DELIMITER //

CREATE PROCEDURE example\_procedure()

BEGIN

DECLARE local\_variable INT DEFAULT 0;-- 可以在这里对局部变量进行操作

END //

DELIMITER ;

1. SELECT student\_name,

CASE grade

WHEN 'A' THEN '优秀'

WHEN 'B' THEN '良好'

WHEN 'C' THEN '中等'

ELSE '待提高'

END AS grade\_description

FROM students;

DELIMITER //

CREATE PROCEDURE evaluate\_score(IN input\_score INT)

BEGIN

IF input\_score >= 90 THEN

SELECT '优秀';

ELSEIF input\_score >= 80 THEN

SELECT '良好';

ELSEIF input\_score >= 60 THEN

SELECT '中等';

ELSE

SELECT '待提高';

END IF;

END //

DELIMITER ;

1. 注释 -- # /\* \*/
2. 视图的概念：一种虚拟的表
3. 事务的四个特性：原子性、一致性、隔离性、持久性
4. 事务保证数据的一致性 要么不做要么一起做 START TRANSACTION COMMIT ROLLBACK
5. 第一类更新丢失的后果是极其严重的，MySQL 已将其完全避免；第二类更新丢失可以防止

视图的定义（编程）

CREATE VIEW view\_name AS

SELECT column1, column2,...

FROM table\_name

WHERE condition;

CREATE VIEW view\_name AS

SELECT column1, column2,...

FROM (

SELECT column3, aggregate\_function(column4) AS alias\_name,...

FROM table\_name

WHERE condition

GROUP BY column3

HAVING group\_condition

ORDER BY column5

) subquery\_alias;

1. 带输出参数的存储过程

Create Procedure p\_num(id varchar(36)，out num int)

select count(\*) into num

from major

where department\_id=id

-- 调用

Set @num= 0;

Call p\_num(‘10’,@num);

Select @num;

DELIMITER //

CREATE PROCEDURE get\_student\_avg\_score(

IN student\_id\_param VARCHAR(36),

OUT avg\_score\_param DECIMAL(5,2)

)

BEGIN

SELECT AVG(score.score) INTO avg\_score\_param

FROM score

WHERE score.student\_id = student\_id\_param;

END //

DELIMITER ;

模块5 数据库安全性

1. mysql数据库中user表记录了允许连接到服务器的账号信息和一些全局级的权限信息
2. 用户创建后为其分配权限
3. 角色创建后为其分配权限，再将角色授予用户
4. 拥有最高权限的超级用户: root(登录账号)
5. 授予GRANT授权权限和回收权限REVOKE

模块6 数据库维护

1. 完整备份（指备份整个数据库，其他所有备份类型都依赖于完整备份）、增量备份（数据库从上一次备份以来改变的内容的备份）、差异备份（从最近一次完整数据库备份以后发生改变的数据进行备份）
2. 如何用语句备份mysql数据库：mysqldump（命令行中执行）
3. 用语句实现数据导导入和导出的写法

* SELECT \* FROM course INTO OUTFILE 'D:\course.txt';
* LOAD DATA INFILE 'D:\course.txt' INTO TABLE scoresys.course

1. 日志就是将数据库中的每一个变化和操作时产生的信息记载到专用的文件中