

数据库期末复习试题大纲

一、选择题（题目含标红答案）

1. 数据库安全性问题与计算机系统的安全性紧密联系，计算机系统的安全性问题可分为哪三大类（A. 技术安全类、管理安全类和政策法规类）
2. TCSEC/TDI 安全级别划分中，C2 级的特点是（B. 提供受控的存取保护，将 C1 级的 DAC 进一步细化，以个人身份注册负责，并实施审计和资源隔离）
3. 关于自主存取控制（DAC）和强制存取控制（MAC），以下说法正确的是（B. DAC 仅通过数据的存取权限进行控制，数据本身无安全性标记，可能存在数据的“无意泄露”）
4. 在 SQL 中，以下关于 GRANT 和 REVOKE 语句的说法错误的是（D. REVOKE 语句只能由数据库管理员执行，用户不能收回自己授予他人的权限）
5. 关于数据库角色，以下说法正确的是（C. 一个角色的权限包括直接授予这个角色的全部权限加上其他角色授予这个角色的全部权限）
6. 在 SQL 授权语句中，WITH GRANT OPTION 子句的作用是（B. 允许用户将获得的权限再授予其他用户）
7. REVOKE 语句中，CASCADE 和 RESTRICT 选项的区别在于（B. CASCADE 级联收回转授的权限，RESTRICT 在存在转授时阻止收回）
8. 关于数据库审计功能，以下说法错误的是（D. 审计功能默认是开启的以保障安全）
9. 透明存储加密的主要优点是（C. 对用户和应用程序完全透明）
10. 在 SSL 协议的数据库传输加密中，会话密钥的生命周期是（B. 仅限于本次通信会话）
11. 以下哪个选项不是创建数据库用户时的有效选项（C. CREATE ROLE）
12. 关于数据库存储加密，说法正确的是（B. 透明加密在数据写到磁盘时自动加密）
13. 在强制存取控制中，推理控制主要用于（B. 避免用户通过可访问数据推导出高敏感级数据）
14. 以下关于数据库角色的描述，正确的是（C. CREATE ROLE 用于创建角色）
15. “三权分立”的安全管理机制主要解决（B. DBA 权限过于集中的问题）
16. 下列关于数据库完整性的描述，错误的是（C. 完整性防范非法用户的非法操作）
17. 实体完整性中，主码的定义方式不包括（C. 多属性列级约束）
18. 参照完整性中，当删除被参照表的元组时，不包括的违约处理方式是（C. 报错并终止程序）
19. 下列关于用户定义完整性的 CHECK 约束，说法正确的是（C. 可定义在属性或元组上）
20. 触发器的触发时机不包括（B. DURING）
21. 若需先跳过结果集前 5 行，再取接下来的 4 行，标准 SQL 语法应写作（B. LIMIT 4 OFFSET 5）
22. 关于 GROUP BY 与 HAVING，下列说法正确的是（B. HAVING 子句必须配合聚集函数使用）
23. 字符匹配中，能正确找出“DB_开头且倒数第 3 字符为 i”的课程名的 LIKE 模式是（B. 'DB_%i_'）
24. 下列哪种数据类型最适合保存“课程学分”这一取值范围 1-10 且保留 1 位小数的数值？（C. NUMERIC(2,1)）

二、判断题（题目含标红答案）

1. 数据库的安全性是指保护数据库以防止不合法使用所造成的数据泄露、篡改或破坏。（√）
2. TCSEC/TDI 安全级别划分中，A1 级是验证设计，提供 B3 级保护的同时给出系统的形式化设计说明和验证，以确信各安全保护真正实现。（√）
3. 自主存取控制（DAC）和强制存取控制（MAC）是两种互斥的安全控制方法，不能同时使用。（×）
4. 在 SQL 中，GRANT 语句可以将对表的权限授予用户，而 REVOKE 语句可以收回授予的权限，但收回权限时不能指定 CASCADE 或 RESTRICT。（×）
5. 数据库角色可以简化授权的过程，通过角色可以将一组权限授予多个用户。（√）
6. GRANT 语句可以同时授予多个权限给多个用户。（√）

7. REVOKE 语句默认使用 CASCADE 选项级联收回权限。(x) X
8. 只有数据库管理员才能设置系统级审计。(v) ✓
9. 链路加密方式会同时加密报头和报文内容。(v) ✓
10. 端到端加密在中间节点需要解密数据才能转发。(x) X
11. 审计功能开启后会对数据库性能产生一定影响。(v) ✓
12. 创建用户时 SUPERUSER 和 CREATEDB 选项作用完全相同。(x) ✓ X
13. NOAUDIT 语句用于取消已经设置的审计功能。(v) X
14. 视图机制可以作为数据库安全性控制的一种手段。(v) ✓
15. 数据加密技术可以有效防止所有类型的数据库安全威胁。(x) X
16. 数据库的完整性和安全性是同一概念，均用于保护数据不被破坏。(x) X
17. 实体完整性检查时，主码的任一属性都不允许取空值。(v) ✓
18. 参照完整性中，外码列必须允许取空值。(x) ✓
19. 完整性约束命名子句可以为约束指定名称，方便后续修改或删除。(v) ✓
20. 语句级触发器中，触发动作体可以使用 NEW 或 OLD 引用数据变化。(x) X
21. SQL92 开始把标准拆成多个部分，例如增加了调用接口和永久存储模块。(v) X
22. 在 SQL 中，DROP TABLE 缺省行为是 CASCADE。(x) ✓
23. 索引一旦建立，就不能再修改名字，只能先删除再重建。(x) X
24. SELECT DISTINCT Sno FROM SC 与 SELECT Sno FROM SC GROUP BY Sno 在任何情况下结果都完全相同。(v) ✓
25. 使用 LEFT OUTER JOIN 时，左表中不满足连接条件的元组会被丢弃。(x) X
26. 在 WHERE 子句里可以使用列别名，例如 SELECT Sname AS sn FROM Student WHERE sn='李勇'。(x) X

三、大题（含考察点、题目及答案）

（一）表创建与数据操作类

考察点：表创建（含约束、外键、联合主键）、数据插入、更新、删除

1. 题目：假设其他表格已创建，编写 SQL 语句创建销售明细清单 sales_item。答案：
sql

```
CREATE TABLE sales_item (

    soid INT NOT NULL,

    cid INT NOT NULL,

    squANTITY INT NOT NULL CHECK (squANTITY > 0),

    sdiscount DECIMAL(5,2) NOT NULL DEFAULT 0.00 CHECK (sdiscount BETWEEN 0
AND 1),

    sactual_price DECIMAL(8,2) NOT NULL,

    PRIMARY KEY (soid, cid),

    FOREIGN KEY (soid) REFERENCES sales_order(soid),

    FOREIGN KEY (cid) REFERENCES creative(cid));
```

1. 题目：向文创产品表插入以下数据|cname|ccategory|cprice|cstock|---|---|---|---| 校园 logo 笔记本 | 文具 | 19.90|200| 校训马克杯 | 文具 | 29.90|150| 校园风景钥匙扣 | 摆件 | 15.80|300| 定制校服外套 | 服饰 | 199.00|80| 图书馆主题书签 | 文具 | 9.90|500|答案：

sql

```
INSERT INTO creative (cname, ccategory, cprice, cstock)VALUES ('校园 logo 笔记本', '文具', 19.90, 200), ('校训马克杯', '文具', 29.90, 150), ('校园风景钥匙扣', '摆件', 15.80, 300), ('定制校服外套', '服饰', 199.00, 80), ('图书馆主题书签', '文具', 9.90, 500);
```

1. 题目：因原材料成本上涨，将所有“文具”类产品的销售单价上调 12%，同时更新库存（每种文具库存增加 50 件）。答案：

sql

```
UPDATE creativeSET cprice = cprice * 1.12, cstock = cstock + 50WHERE ccategory = '文具';
```

1. 题目：向 Student 表中插入一个新学生，学号为“20220001”，姓名为“李华”，性别为“女”，出生日期为“2002-07-15”，主修专业为“计算机科学与技术”。答案：

sql

```
INSERT INTO Student (Sno, Sname, Ssex, Sbirthdate, Smajor)VALUES ('20220001', '李华', '女', '2002-07-15', '计算机科学与技术');
```

1. 题目：将所有“信息管理与信息系统”专业学生的选课成绩置为 0。答案：

sql

```
UPDATE SCSET Grade = 0WHERE Sno IN (SELECT Sno FROM Student WHERE Smajor = '信息管理与信息系统');
```

1. 题目：删除所有未选课的学生（即在 SC 表中没有记录的学生）。答案：

sql

```
DELETE FROM StudentWHERE Sno NOT IN (SELECT DISTINCT Sno FROM SC);
```

（二）数据查询类

考察点：单表查询、多表连接查询、分组统计查询、条件筛选、排序

1. 题目：查询所有“文具”类且名称包含“校园”字样的文创产品信息，包括产品名称、类别、单价和库存。答案：

sql

```
SELECT cname AS 产品名称, ccategory AS 类别, cprice AS 单价, cstock AS 库存 FROM creativeWHERE ccategory = '文具' AND cname LIKE '%校园%';
```

1. 题目：查询销售点“宿舍区店”2025 年 1 月 1 日之后的所有销售记录，包括购买人、产品名称、实际成交单价、销售数量和销售日期。答案：

sql

```
SELECT so.scustomer AS 购买人, c.cname AS 产品名称, si.sactual_price AS 实际成交单价, si.squantity AS 销售数量, so.sdate AS 销售日期 FROM campus_store csJOIN sales_order so ON cs.csid = so.csidJOIN sales_item si ON so.soid = si.soidJOIN
```

```
creative c ON si.cid = c.cidWHERE cs.csname = '宿舍区店' AND so.sdate >
'2025-01-01';
```

1. 题目：查询各类别文创产品的总销售数量和总销售额（总销售额 = 实际成交单价 × 销售数量），按总销售额降序排列。答案：

sql

```
SELECT c.ccategory AS 产品类别, SUM(si.squantity) AS 总销售数量,
SUM(si.sactual_price * si.squantity) AS 总销售额FROM creative cJOIN
sales_item si ON c.cid = si.cidGROUP BY c.ccategoryORDER BY 总销售额 DESC;
```

1. 题目：查询“信息安全”专业 2000 年及以后出生的学生姓名与年龄（当年年份 - 出生年份），按年龄降序、姓名升序排列。答案：

sql

```
SELECT Sname,
EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM Sbirthdate) AS
ageFROM StudentWHERE Smajor = '信息安全'
AND EXTRACT(YEAR FROM Sbirthdate) >= 2000ORDER BY age DESC, Sname ASC;
```

1. 题目：列出选修了 3 门及以上课程的学生学号及选课门数，仅保留选课门数≥3 的记录。答案：

sql

```
SELECT Sno, COUNT(*) AS cntFROM SCGROUP BY SnoHAVING COUNT(*) >= 3;
```

1. 题目：输出每门课的间接先修课（先修课的先修课）课程号、间接先修课号及间接先修课学分，要求间接先修课存在。答案：

sql

```
SELECT F.Cno AS course_no,
S.Cpno AS indirect_cpno,
S.Ccredit AS indirect_creditFROM Course FJOIN Course S ON F.Cpno =
S.CnoWHERE S.Cpno IS NOT NULL;
```

1. 题目：使用 LEFT JOIN 列出所有学生的学号、姓名、已选课程号及成绩，无选课者课程号与成绩显示为 NULL，结果按学号升序、课程号降序排列。答案：

sql

```
SELECT St.Sno, St.Sname, SC.Cno, SC.GradeFROM Student StLEFT JOIN SC ON St.Sno
= SC.SnoORDER BY St.Sno ASC, SC.Cno DESC;
```

1. 题目：查询所有成绩为空的选课记录，显示学号、课程号和成绩。答案：

sql

```
SELECT Sno, Cno, GradeFROM SCWHERE Grade IS NULL;
```

（三）视图与角色权限类

考察点：视图创建、角色创建与授权、权限收回

1. 题目：创建视图 `v_campus_sales`，显示销售点名称、产品类别、产品名称、销售数量、折扣率和销售日期，仅包含折扣率大于 0 的销售记录。答案：

sql

```
CREATE VIEW v_campus_sales AS SELECT cs.csname AS 销售点名称, c.ccategory AS 产品类别, c.cname AS 产品名称, si.squantity AS 销售数量, si.sdiscount AS 折扣率, so.sdate AS 销售日期 FROM campus_store cs JOIN sales_order so ON cs.csid = so.csid JOIN sales_item si ON so.soid = si.soid JOIN creative c ON si.cid = c.cid WHERE si.sdiscount > 0;
```

1. 题目：创建一个角色 `R2`，并使角色 `R2` 拥有 `Course` 表的 `SELECT`、`INSERT`、`UPDATE` 权限，然后将这个角色授予用户李明和刘强。答案：

sql

```
-- 创建角色 R2
CREATE ROLE R2;
-- 授予角色 R2 对 Course 表的权限
GRANT SELECT, INSERT, UPDATE ON Course TO R2;
-- 将角色授予李明和刘强
GRANT R2 TO 李明, 刘强;
```

1. 题目：假设存在一个学生选课数据库，其中包含学生表 `Student` (`Sno`, `Sname`, `Ssex`, `Sage`, `Sdept`) 和选课表 `SC` (`Sno`, `Cno`, `Grade`)。请编写 SQL 语句，创建一个视图 `V1`，显示计算机科学与技术专业 (`Sdept='计算机科学与技术'`) 学生的选课信息 (包括学生姓名、课程号和成绩)，并将对该视图的 `SELECT` 权限授予教师王老师。答案：

sql

```
-- 创建视图 V1
CREATE VIEW V1 AS SELECT s.Sname AS 学生姓名, sc.Cno AS 课程号, sc.Grade AS 成绩 FROM Student s JOIN SC sc ON s.Sno = sc.Sno WHERE s.Sdept = '计算机科学与技术';
-- 授予王老师对视图 V1 的 SELECT 权限
GRANT SELECT ON V1 TO 王老师;
```

1. 题目：假设用户 `U8` 拥有对表 `T1` 的 `SELECT`、`INSERT`、`UPDATE` 权限，现在需要收回 `U8` 对表 `T1` 的 `UPDATE` 权限，并且级联收回所有从 `U8` 处获得该权限的用户的 `UPDATE` 权限，请写出相应的 SQL 语句。答案：

sql

```
REVOKE UPDATE ON T1 FROM U8 CASCADE;
```

1. 题目：以超级用户 `system` 登录，创建一个具有 `CREATEDB` 权限的用户 `U9`，并以 `U9` 用户登录创建一个名为 `U9DB` 的数据库。答案：

sql

```
-- 以 system 身份创建用户 U9 并授予 CREATEDB 权限
CREATE USER U9 WITH CREATEDB PASSWORD '用户密码';
-- 以 U9 身份登录后创建数据库
CREATE DATABASE U9DB;
```

1. 题目：假设存在一个员工表 `Employee` (`Eid`, `Ename`, `Eage`, `Esex`, `Edept`)，现在需要创建一个角色 `R3`，使角色 `R3` 拥有 `Employee` 表的 `SELECT`、`UPDATE` 权限，并且可以将这些权限授予其他用户，然后将角色 `R3` 授予用户赵敏，并使赵敏具有将这些权限授予其他用户的权限。答案：

sql

```
-- 创建角色 R3
CREATE ROLE R3;
-- 授予角色 R3 对 Employee 表的权限并允许转授
GRANT SELECT, UPDATE ON Employee TO R3 WITH GRANT OPTION;
-- 将角色 R3 授予赵敏并允许其转授
GRANT R3 TO 赵敏 WITH GRANT OPTION;
```

1. 题目：现有学生 - 课程数据库，包含 Student、Course 和 SC 表。要求写出 SQL 语句：（1）授予用户 'ZhangSan' 查询 Student 表和 Course 表的权限。（2）授予用户 'LiSi' 对 SC 表的 INSERT 和 UPDATE 权限，并允许其将权限转授他人。（3）授予所有用户对 Course 表的课程名 (Cname) 列的 SELECT 权限。答案：

sql

```
-- (1) 授予 ZhangSan 查询权限 GRANT SELECT ON Student, Course TO 'ZhangSan';--  
(2) 授予 LiSi 对 SC 表的 INSERT 和 UPDATE 权限并允许转授 GRANT INSERT, UPDATE ON SC  
TO 'LiSi' WITH GRANT OPTION;-- (3) 授予所有用户对 Course 表 Cname 列的 SELECT 权  
限 GRANT SELECT (Cname) ON Course TO PUBLIC;
```

1. 题目：假设用户 'WangWu' 已将对 Student 表的 DELETE 权限转授给了用户 'ZhaoLiu'。要求写出 SQL 语句：（1）收回 'WangWu' 对 Student 表 DELETE 权限，要求同时级联收回其转授的权限。（2）仅收回 'ZhaoLiu' 对 Student 表 DELETE 权限（不影响 WangWu 的权限）。答案：

sql

```
-- (1) 级联收回 WangWu 的 DELETE 权限 REVOKE DELETE ON Student FROM 'WangWu'  
CASCADE;-- (2) 仅收回 ZhaoLiu 的 DELETE 权限 REVOKE DELETE ON Student FROM  
'ZhaoLiu';
```

1. 题目：创建一个名为 CS_Student 的视图，包含所有“计算机科学与技术”专业的学生信息（学号、姓名、性别、出生日期）。然后基于该视图查询年龄小于等于 21 岁的学生学号和出生日期。答案：

sql

```
-- 创建视图 CS_Student  
CREATE VIEW CS_Student AS  
SELECT Sno, Sname, Ssex,  
Sbirthdate  
FROM Student  
WHERE Smajor = '计算机科学与技术';-- 基于视图查询年龄小于  
等于 21 岁的学生信息  
SELECT Sno, Sbirthdate  
FROM CS_Student  
WHERE EXTRACT(YEAR  
FROM CURRENT_DATE) - EXTRACT(YEAR FROM Sbirthdate) <= 21;
```

（四）模式与索引操作类

考察点：模式创建与授权、索引创建、修改、删除

1. 题目：在“学生选课”数据库（模式名 S_C_SC）下完成下列要求，写出完整 SQL 语句。（1）创建模式并授权给用户 ZHANG。（2）在该模式下建表 Teacher (Tno,Tname,Title)，主码 Tno: Title 默认为‘讲师’。（3）为 Teacher (Tname) 建立唯一升序索引 Idx_Tname。（4）将索引改名为 Idx_TeacherName。（5）删除该索引。答案：

sql

```
-- (1) 创建模式并授权给 ZHANG  
CREATE SCHEMA S_C_SC AUTHORIZATION ZHANG;-- (2)  
创建 Teacher 表  
CREATE TABLE S_C_SC.Teacher(  
  
    Tno    CHAR(6) PRIMARY KEY,  
  
    Tname  VARCHAR(30) NOT NULL,  
  
    Ttitle VARCHAR(20) DEFAULT '讲师';-- (3) 创建唯一升序索引  
CREATE UNIQUE INDEX  
Idx_Tname  
  
ON S_C_SC.Teacher(Tname ASC);-- (4) 修改索引名称  
ALTER INDEX Idx_Tname  
RENAME TO Idx_TeacherName;-- (5) 删除索引  
DROP INDEX Idx_TeacherName;
```

（五）触发器类

考察点：触发器创建（触发事件、时机、条件、动作）

1. 题目：请为“学生表 Student”创建一个行级触发器 Stu_Birth_Trigger，要求如下：

- 触发事件为 INSERT 或 UPDATE；
- 触发时机为 BEFORE；
- 触发条件为学生出生日期（Sbirthdate）早于 1990 年 1 月 1 日；
- 触发动作：若满足触发条件，自动将出生日期改为 1990 年 1 月 1 日。答案：

sql

```
CREATE TRIGGER Stu_Birth_Trigger
```

```
BEFORE INSERT OR UPDATE ON Student FOR EACH ROW WHEN (NEW.Sbirthdate <  
'1990-01-01') BEGIN
```

```
    SET NEW.Sbirthdate = '1990-01-01';END;
```