一、单选题（20题）

1. 事务的本质是？（ ）

A. 单条SQL语句 B. 一组不可分割的SQL操作 C. 数据库备份 D. 索引更新

2. 事务的ACID特性中，“原子性”指的是？（ ）

A. 事务执行前后数据状态一致 B. 事务不可分割，要么全成要么全滚

C. 并发事务互不干扰 D. 事务提交后修改永久生效

3. MySQL中默认的事务存储引擎是？（ ）

A. MyISAM B. InnoDB C. Memory D. Archive

4. 显式开启事务的语句是？（ ）

A. BEGIN TRANSACTION B. START TRANSACTION C. OPEN TRANSACTION D. BEGIN WORK

5. 提交事务的语句是？（ ）

A. SUBMIT B. COMMIT C. SAVE D. FINISH

6. 回滚事务的语句是？（ ）

A. ROLLBACK B. CANCEL C. UNDO D. REVERT

7. 事务的自动提交由哪个变量控制？（ ）

A. AUTOCOMMIT B. AUTO\_SAVE C. TRANSACTION\_AUTO D. AUTO\_ROLLBACK

8. MySQL默认的事务隔离级别是？（ ）

A. READ UNCOMMITTED B. READ COMMITTED C. REPEATABLE READ D. SERIALIZABLE

9. 脏读可能发生在哪个隔离级别？（ ）

A. READ UNCOMMITTED B. READ COMMITTED C. REPEATABLE READ D. SERIALIZABLE

10. 可重复读解决的问题是？（ ）

A. 脏读 B. 不可重复读 C. 幻读 D. 死锁

11. 事务保存点的作用是？（ ）

A. 提交事务 B. 回滚到指定位置 C. 开启事务 D. 锁定数据

12. 创建保存点的语句是？（ ）

A. CREATE SAVEPOINT B. SET SAVEPOINT C. SAVEPOINT D. MARK SAVEPOINT

13. 回滚到保存点的语句是？（ ）

A. ROLLBACK TO SAVEPOINT B. REVERT TO SAVEPOINT C. UNDO TO SAVEPOINT D. BACK TO SAVEPOINT

14. 最高的事务隔离级别是？（ ）

A. READ UNCOMMITTED B. READ COMMITTED C. REPEATABLE READ D. SERIALIZABLE

15. 下列哪种隔离级别可能导致幻读？（ ）

A. READ UNCOMMITTED B. READ COMMITTED C. REPEATABLE READ D. SERIALIZABLE

16. 事务隔离级别中，性能最低的是？（ ）

A. READ UNCOMMITTED B. READ COMMITTED C. REPEATABLE READ D. SERIALIZABLE

17. 下列哪种操作会隐式提交事务？（ ）

A. SELECT B. INSERT C. CREATE TABLE D. UPDATE

18. MySQL中，事务是否支持嵌套？（ ）

A. 支持 B. 不支持 C. 部分支持 D. 取决于存储引擎

19. 只读事务的设置语句是？（ ）

A. SET TRANSACTION READ ONLY B. SET TRANSACTION READ ONLY MODE

C. BEGIN TRANSACTION READ ONLY D. START TRANSACTION READ ONLY

20. 事务的持久性是通过什么机制实现的？（ ）

A. 索引 B. 日志 C. 缓存 D. 锁

二、多选题（15题）

1. 事务的ACID特性包括？（ ）

A. 原子性 B. 一致性 C. 隔离性 D. 持久性

2. 下列支持事务的存储引擎有？（ ）

A. InnoDB B. MyISAM C. Memory D. NDB Cluster

3. 事务的基本操作包括？（ ）

A. 开启 B. 提交 C. 回滚 D. 保存

4. 下列关于自动提交的说法正确的有？（ ）

A. 默认开启 B. 可通过AUTOCOMMIT变量控制 C. 关闭后需手动提交 D. 单条SQL语句不受影响

5. 事务隔离级别的作用是？（ ）

A. 控制并发访问 B. 防止数据不一致 C. 提高查询效率 D. 简化事务操作

6. 可能出现脏读的隔离级别有？（ ）

A. READ UNCOMMITTED B. READ COMMITTED C. REPEATABLE READ D. 以上都不会

7. 可重复读级别解决的问题包括？（ ）

A. 脏读 B. 不可重复读 C. 幻读 D. 死锁

8. 保存点的相关操作包括？（ ）

A. 创建 B. 回滚到保存点 C. 删除 D. 提交

9. 下列操作会隐式提交事务的有？（ ）

A. CREATE DATABASE B. ALTER TABLE C. DROP TABLE D. SELECT

10. 事务隔离级别的设置语句中，作用范围包括？（ ）

A. SESSION B. GLOBAL C. LOCAL D. TRANSACTION

11. 下列关于SERIALIZABLE级别的说法正确的有？（ ）

A. 最高隔离级别 B. 解决所有并发问题 C. 可能导致锁等待 D. 性能最低

12. 不可重复读与幻读的区别是？（ ）

A. 不可重复读是数据值变化 B. 幻读是记录数变化

C. 不可重复读发生在更新时 D. 幻读发生在插入时

13. 事务回滚的情况包括？（ ）

A. 执行ROLLBACK语句 B. 事务执行失败 C. 连接中断 D. 提交事务后

14. 下列关于事务日志的说法正确的有？（ ）

A. 保证持久性 B. 记录数据变化 C. 用于事务恢复 D. 与索引无关

15. 只读事务的特点是？（ ）

A. 不能执行写操作 B. 提高查询效率 C. 可提交 D. 可回滚

三、判断题（20题）

1. 事务是一组不可分割的SQL操作。（ ）

2. MyISAM存储引擎支持事务。（ ）

3. 事务的原子性指事务执行前后数据状态一致。（ ）

4. 显式开启事务后，SQL语句不会自动提交。（ ）

5. ROLLBACK可回滚已提交的事务。（ ）

6. 保存点在事务提交后仍然有效。（ ）

7. MySQL默认隔离级别是READ COMMITTED。（ ）

8. 脏读是指读取到未提交的数据。（ ）

9. REPEATABLE READ可解决幻读问题。（ ）

10. 事务可以嵌套。（ ）

11. AUTOCOMMIT默认值为1（开启）。（ ）

12. CREATE TABLE会隐式提交事务。（ ）

13. SERIALIZABLE级别可能导致锁等待。（ ）

14. 不可重复读是指两次查询记录数不同。（ ）

15. 事务的持久性是绝对的，不受硬件影响。（ ）

16. SAVEPOINT可用于部分回滚。（ ）

17. 只读事务不能执行UPDATE操作。（ ）

18. 隔离级别越高，并发性能越好。（ ）

19. 日志机制保证了事务的一致性。（ ）

20. 回滚到保存点后，该保存点依然存在。（ ）

四、填空题（20题）

1. 事务的四大特性是原子性、一致性、\_\_\_\_\_\_和持久性。

2. 开启事务的语句是\_\_\_\_\_\_。

3. 提交事务的语句是\_\_\_\_\_\_。

4. 回滚事务的语句是\_\_\_\_\_\_。

5. 控制事务自动提交的变量是\_\_\_\_\_\_。

6. MySQL默认的事务隔离级别是\_\_\_\_\_\_。

7. 脏读可能发生在\_\_\_\_\_\_隔离级别。

8. 创建保存点的语句是\_\_\_\_\_\_。

9. 回滚到保存点的语句是\_\_\_\_\_\_。

10. 最高的事务隔离级别是\_\_\_\_\_\_。

11. 不可重复读是指\_\_\_\_\_\_。

12. 事务的持久性通过\_\_\_\_\_\_机制实现。

13. 支持事务的存储引擎是\_\_\_\_\_\_。

14. 关闭自动提交的语句是\_\_\_\_\_\_。

15. SERIALIZABLE级别可能导致\_\_\_\_\_\_和锁竞争。

16. 幻读是指\_\_\_\_\_\_。

17. 事务提交后，修改是\_\_\_\_\_\_的。

18. 只读事务的设置语句是\_\_\_\_\_\_。

19. 隔离级别\_\_\_\_\_\_可避免不可重复读。

20. 事务日志记录\_\_\_\_\_\_。

五、简答题（10题）

1. 简述事务的概念及ACID特性。

2. 事务的基本操作有哪些？分别用什么语句实现？

3. 什么是脏读、不可重复读和幻读？各发生在哪个隔离级别？

4. 简述MySQL的四种事务隔离级别及特点。

5. 保存点的作用是什么？如何创建和使用保存点？

6. 自动提交（AUTOCOMMIT）的作用是什么？如何开启和关闭？

7. 哪些操作会隐式提交事务？

8. 事务回滚的场景有哪些？

9. 为什么InnoDB支持事务而MyISAM不支持？

10. 简述只读事务的特点及使用场景。

六、SQL实操题（15题）

1. 开启一个事务，向student表插入一条记录（name='张三', money=1000），然后提交事务。

2. 开启事务，更新student表中name='张三'的money为2000，不提交，查询结果后回滚，再查询验证。

3. 开启事务，创建保存点s1，更新记录后回滚到s1，验证数据状态。

4. 查看当前会话的事务隔离级别。

5. 将当前会话的隔离级别设置为READ COMMITTED。

6. 关闭当前会话的自动提交，执行一条INSERT语句，不提交，查看数据是否可见。

7. 开启事务，执行两条UPDATE语句（转账操作），提交事务，验证数据一致性。

8. 模拟事务失败：开启事务，执行错误SQL（如更新不存在的表），查看是否自动回滚。

9. 创建只读事务，尝试执行UPDATE操作，观察结果。

10. 查看AUTOCOMMIT变量的当前值。

11. 开启事务，设置两个保存点s1、s2，更新数据后回滚到s1，删除保存点s2。

12. 将全局事务隔离级别设置为REPEATABLE READ。

13. 开启事务，执行多条INSERT语句，部分回滚到保存点后提交剩余操作。

14. 验证SERIALIZABLE级别下的锁等待：两个会话同时操作同一行，观察等待现象。

15. 开启事务，执行DELETE操作后回滚，验证数据是否恢复。

**答案解释**

一、单选题（20题）

1. B（解析：定义事务为“针对数据库的一组操作，由一条或多条SQL语句组成，不可分割”）

2. B（解析：原子性指“事务不可分割，所有操作要么全成功，要么全回滚”）

3. B（解析：明确“InnoDB存储引擎支持事务，MyISAM不支持”）

4. B（解析：显式开启事务的语句为`START TRANSACTION`）

5. B（解析：提交事务的语句为`COMMIT`）

6. A（解析：回滚事务的语句为`ROLLBACK`）

7. A（解析：“事务的自动提交通过`AUTOCOMMIT`变量控制”）

8. C（解析：“MySQL默认隔离级别为`REPEATABLE READ`”）

9. A（解析：“`READ UNCOMMITTED`级别可能出现脏读”）

10. B（解析：“`REPEATABLE READ`解决不可重复读问题”）

11. B（解析：保存点用于“回滚到事务中的指定位置”）

12. C（解析：创建保存点的语句为`SAVEPOINT 保存点名`）

13. A（解析：回滚到保存点的语句为`ROLLBACK TO SAVEPOINT 保存点名`）

14. D（解析：“`SERIALIZABLE`是最高隔离级别”）

15. B（解析：“`READ COMMITTED`可能出现幻读”）

16. D（解析：“`SERIALIZABLE`级别性能最低”）

17. C（解析：“`CREATE TABLE`等DDL操作会隐式提交事务”）

18. B（解析：“MySQL事务不支持嵌套”）

19. A（解析：“只读事务设置语句为`SET TRANSACTION READ ONLY`”）

20. B（解析：“事务的持久性通过日志机制实现”）

二、多选题（15题）

1. ABCD（解析：事务的ACID特性包括原子性、一致性、隔离性、持久性）

2. A（解析：仅InnoDB支持事务）

3. ABC（解析：事务基本操作包括开启、提交、回滚）

4. ABC（解析：“`AUTOCOMMIT`默认开启，关闭后需手动提交，单条SQL也受影响”）

5. AB（解析：隔离级别用于“控制并发访问，防止数据不一致”）

6. A（解析：仅`READ UNCOMMITTED`可能出现脏读）

7. ABC（解析：“InnoDB的`REPEATABLE READ`解决脏读、不可重复读、幻读”）

8. ABC（解析：保存点操作包括创建、回滚到保存点、删除保存点）

9. ABC（解析：“`CREATE DATABASE`、`ALTER TABLE`、`DROP TABLE`等DDL操作隐式提交事务”）

10. AB（解析：隔离级别作用范围为`SESSION`（当前会话）和`GLOBAL`（全局））

11. ABCD（解析：“`SERIALIZABLE`是最高级别，解决所有并发问题，但可能导致锁等待，性能最低”）

12. ABCD（解析：“不可重复读是数据值变化（更新导致），幻读是记录数变化（插入导致）”）

13. ABC（解析：“事务回滚场景包括执行`ROLLBACK`、执行失败、连接中断”，提交后无法回滚）

14. ABC（解析：“事务日志记录数据变化，保证持久性，用于事务恢复”）

15. ACD（解析：“只读事务不能执行写操作，可提交和回滚，与查询效率无关”）

三、判断题（20题）

1. √（解析：事务是“一组不可分割的SQL操作”）

2. ×（解析：“MyISAM不支持事务”）

3. ×（解析：原子性指“不可分割”，一致性指“事务前后数据状态一致”）

4. √（解析：“显式开启事务后，SQL语句需手动提交”）

5. ×（解析：“已提交的事务无法回滚”）

6. ×（解析：“事务提交后，保存点失效”）

7. ×（解析：“MySQL默认隔离级别是`REPEATABLE READ`”）

8. √（解析：“脏读指读取到其他事务未提交的数据”）

9. √（解析：“InnoDB的`REPEATABLE READ`通过多版本并发控制解决幻读”）

10. ×（解析：“MySQL事务不支持嵌套”）

11. √（解析：“`AUTOCOMMIT`默认值为1（开启）”）

12. √（解析：“`CREATE TABLE`是DDL操作，会隐式提交事务”）

13. √（解析：“`SERIALIZABLE`级别可能导致锁等待和超时”）

14. ×（解析：“不可重复读是两次查询数据值不同，幻读是记录数不同”）

15. ×（解析：“事务持久性是‘相对’的，硬件故障可能导致数据丢失”）

16. √（解析：“保存点可实现部分回滚”）

17. √（解析：“只读事务禁止写操作（如`UPDATE`）”）

18. ×（解析：“隔离级别越高，并发性能越低”）

19. √（解析：“一致性主要由日志机制实现”）

20. ×（解析：“回滚到保存点后，该保存点之后的保存点会消失”）

四、填空题（20题）

1. 隔离性（解析：事务的ACID特性）

2. `START TRANSACTION`（解析：开启事务的语句）

3. `COMMIT`（解析：提交事务的语句）

4. `ROLLBACK`（解析：回滚事务的语句）

5. `AUTOCOMMIT`（解析：控制自动提交的变量）

6. `REPEATABLE READ`（解析：默认隔离级别）

7. `READ UNCOMMITTED`（解析：脏读发生的级别）

8. `SAVEPOINT 保存点名`（解析：创建保存点的语句）

9. `ROLLBACK TO SAVEPOINT 保存点名`（解析：回滚到保存点的语句）

10. `SERIALIZABLE`（解析：最高隔离级别）

11. 同一事务中两次查询的结果不一致（解析：不可重复读的定义）

12. 日志（解析：持久性通过日志实现）

13. InnoDB（解析：支持事务的存储引擎）

14. `SET AUTOCOMMIT = 0`（解析：关闭自动提交的语句）

15. 超时（解析：`SERIALIZABLE`可能导致超时）

16. 同一事务中两次查询的记录数不一致（解析：幻读的定义）

17. 永久（解析：事务提交后修改永久生效）

18. `SET TRANSACTION READ ONLY`（解析：设置只读事务的语句）

19. `REPEATABLE READ`（解析：可避免不可重复读的级别）

20. 数据库的所有变化（解析：日志记录数据库变化）

五、简答题（10题）

1. 事务的概念及ACID特性：

事务是针对数据库的一组不可分割的操作，由一条或多条SQL语句组成，所有操作要么全成功，要么全回滚。

ACID特性：

原子性：事务不可分割，要么全执行，要么全回滚；

一致性：事务执行前后，数据库状态保持一致；

隔离性：并发事务互不干扰，未提交事务的结果不被其他事务可见；

持久性：事务提交后，修改永久生效（）。

2. 事务的基本操作及语句：

开启事务：`START TRANSACTION`（或`BEGIN`）；

提交事务：`COMMIT`（使操作生效）；

回滚事务：`ROLLBACK`（撤销未提交的操作）（）。

3. 脏读、不可重复读、幻读的定义及发生级别：

脏读：读取到其他事务未提交的数据，发生在`READ UNCOMMITTED`级别；

不可重复读：同一事务中两次查询结果不一致（因其他事务更新），发生在`READ COMMITTED`级别；

幻读：同一事务中两次查询记录数不一致（因其他事务插入），发生在`READ COMMITTED`级别（）。

4. MySQL的四种隔离级别及特点：

`READ UNCOMMITTED`：最低级别，允许读取未提交数据，可能出现脏读、不可重复读、幻读；

`READ COMMITTED`：仅读取已提交数据，避免脏读，但可能出现不可重复读、幻读；

`REPEATABLE READ`（默认）：保证同一事务中查询结果一致，避免脏读、不可重复读，InnoDB中可避免幻读；

`SERIALIZABLE`：最高级别，通过加锁避免所有并发问题，但性能最低（）。

5. 保存点的作用及操作：

作用：允许事务部分回滚（仅撤销保存点之后的操作）。

操作：

创建：`SAVEPOINT 保存点名`；

回滚到保存点：`ROLLBACK TO SAVEPOINT 保存点名`；

删除：`RELEASE SAVEPOINT 保存点名`（）。

6. 自动提交（AUTOCOMMIT）的作用及开关：

作用：控制SQL语句是否自动提交（默认开启，每条SQL自动作为独立事务提交）。

开启：`SET AUTOCOMMIT = 1`；

关闭：`SET AUTOCOMMIT = 0`（需手动`COMMIT`提交事务）（）。

7. 隐式提交事务的操作：

包括创建/删除数据库（`CREATE/DROP DATABASE`）、创建/修改/删除表（`CREATE/ALTER/DROP TABLE`）等DDL操作（）。

8. 事务回滚的场景：

执行`ROLLBACK`语句；

事务中存在错误SQL（如语法错误）；

数据库连接中断或服务器崩溃（）。

9. InnoDB支持事务而MyISAM不支持的原因：

InnoDB支持事务是因为其实现了ACID特性所需的日志机制（如重做日志、回滚日志）和锁机制；而MyISAM设计为轻量级存储引擎，不支持日志和锁机制，因此不支持事务（）。

10. 只读事务的特点及使用场景：

特点：禁止执行写操作（如`INSERT`、`UPDATE`、`DELETE`），仅允许查询；可提交或回滚。

场景：统计分析、报表生成等仅需读取数据的操作，避免误写（）。

六、SQL实操题（15题）

1. `START TRANSACTION; INSERT INTO student (name, money) VALUES ('张三', 1000); COMMIT;`（事务基本操作）

2. `START TRANSACTION; UPDATE student SET money = 2000 WHERE name = '张三'; SELECT \* FROM student WHERE name = '张三'; ROLLBACK; SELECT \* FROM student WHERE name = '张三';`（回滚示例）

3. `START TRANSACTION; UPDATE student SET money = 3000 WHERE name = '张三'; SAVEPOINT s1; UPDATE student SET money = 4000 WHERE name = '张三'; ROLLBACK TO SAVEPOINT s1; SELECT \* FROM student WHERE name = '张三';`（保存点示例）

4. `SELECT @@session.transaction\_isolation;`（查看隔离级别）

5. `SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;`（设置隔离级别）

6. `SET AUTOCOMMIT = 0; INSERT INTO student (name, money) VALUES ('李四', 1000); SELECT \* FROM student WHERE name = '李四';`（未提交，其他会话不可见）（自动提交）

7. `START TRANSACTION; UPDATE student SET money = money 100 WHERE name = '张三'; UPDATE student SET money = money + 100 WHERE name = '李四'; COMMIT; SELECT \* FROM student WHERE name IN ('张三', '李四');`（转账案例）

8. `START TRANSACTION; UPDATE non\_existent\_table SET col = 1;`（错误SQL，事务自动回滚）（事务失败回滚）

9. `SET TRANSACTION READ ONLY; UPDATE student SET money = 500 WHERE name = '张三';`（执行失败，提示只读事务禁止写操作）（只读事务）

10. `SELECT @@autocommit;`（查看自动提交状态）

11. `START TRANSACTION; SAVEPOINT s1; UPDATE student SET money = 600 WHERE name = '张三'; SAVEPOINT s2; UPDATE student SET money = 700 WHERE name = '张三'; ROLLBACK TO SAVEPOINT s1; RELEASE SAVEPOINT s2;`（保存点操作）

12. `SET GLOBAL TRANSACTION ISOLATION LEVEL REPEATABLE READ;`（设置全局隔离级别）

13. `START TRANSACTION; INSERT INTO student VALUES ('王五', 1000); SAVEPOINT s1; INSERT INTO student VALUES ('赵六', 1000); ROLLBACK TO SAVEPOINT s1; COMMIT;`（仅提交“王五”的记录）（部分回滚）

14. 会话1：`SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE; START TRANSACTION; SELECT \* FROM student WHERE name = '张三' FOR UPDATE;`（加锁）

会话2：`UPDATE student SET money = 800 WHERE name = '张三';`（等待会话1提交后执行）（`SERIALIZABLE`锁等待）

15. `START TRANSACTION; DELETE FROM student WHERE name = '张三'; ROLLBACK; SELECT \* FROM student WHERE name = '张三';`（数据恢复）（删除回滚）