数据库

1. 结构数据模型的种类
    a. 层次模型
    b. 网状模型
    c. 关系模型
    d. 面对对象模型
2. 数据库的三种模式
    a. 模式。模式是全体数据的逻辑结构和特征描述，是用户的公共数据视图
    b. 外模式。是数据库用户能够看到的局部数据结构和特征描述
    c. 内模式。是数据库物理结构和存储方式。
3. 两级映像
    a. 外模式/模式映像。存在于外模式和模式之间，描述在外模式中，当模式的描述改变的时候，通过映射，不影响外模式的结构，实现逻辑独立性
    b. 模式/内模式映像。在内模式中定义，当数据的物理存储结构发生变化时，对映像进行相应的改变就可以保证物理独立性。
4. 什么是数据模型

    数据模型由数据结构、数据操作和完整性约束构成。分为三类：概念数据模型（E-R模型）、结构数据模型、物理数据模型（索引、聚集）
5. DBMS有哪些控制功能
    a. 数据安全性保护
    b. 数据的完整性
    c. 并发控制
    d. 数据库的恢复
6. 什么是存取过程

存取过程可以看成记录集，是编译后的sql语句，所以他的执行效率更高，在网络通信中，采用存取过程可以减少通信量，提高通信效率，同时，采用存取过程可以让没有权限的用户执行语句，提高数据库的安全。

7.

关系型数据库

1. 关系型数据库的完整性约束
    a. 实体完整性（主键的每一分量不能为空）
    b. 参照完整性（外键要么为空，要么是另外一个表的主键）
    c. 用户定义的完整性（给属性添加的完整性规则，数据必须满足这个规则）
2. 数据操作有哪些

    选择、投影、连接、交、并、差、除的查询操作，增删改查的更新操作
3. 查询优化策略
    a. 选择投影尽早做
    b. 选择投影如果对一个关系进行操作，就一起做
    c. 投影操作可以和前后的双目运算符一起操作
    d. 执行连接的之前可以预处理，比如建立索引或者排序
    e. 将笛卡尔积和选择操作合并成连接操作
    f. 存储公共子表达式
4. 常见的数据依赖有哪些？
    a. 函数依赖

        平凡函数依赖， 当A包含B的时候，B就平凡函数依赖于A

        完全函数依赖，当A的所有的子集都不确定B，但是A确定B，那么B就完全依赖于A

        传递函数依赖：A→B,B→C, B不决定A那么是C传递函数依赖于A
    b. 多值依赖

        当X，Y确定，Z有多个值对应,并且Y只与X有关，那么X→→Y

5. 函数依赖的逻辑蕴涵，求闭包
    a. ArmStrong 公理
        i. 自反律 Y属于X, F⇒X→Y
        ii. 增广律 X→Y, F⇒ XZ→YZ
        iii. 传递率 X→Y, Y→Z , F⇒X→Z
    b. 推理
        i. 合并律 X→Y, X→Z, F⇒X→YZ
        ii. 伪传递率 X→Y, YW→Z, F⇒XW→Z
        iii. 分解率 X→Y, Z属于Y, F⇒X→Y
6. 范式
    a. 第一范式
    b. 第二范式，所有非主属性完全函数依赖于每个候选键
    c. 第三范式，非主属性不传递函数依赖候选键
    d. BC范式（定义在第一范式上）:每个属性都不传递函数依赖于任意一个候选键（所有函数依赖，X→Y,X必为候选键7）

       如果满足第三范式，并且候选键唯一那么就满足BC范式
    e. 第四范式（定义在第一范式上）：每个非平凡多值依赖X→→Y,X含有候选键。

       满足第四范式一定满足BC范式
7. 无损连接的判断算法

   构建一张横坐标是分解的关系，纵坐标是属性的表，初始化的规则是，如果属性出现在关系中，那么填a_j，否则填b_ij。然后遍历所有函数依赖，X→Y,更新表，规则是，对于相同的X属性的记录，将其Y更新，更新的规则是，当有一个分量是a，那么另一个修改为a，否则用i值较小的b来更新。遍历结束后，如果出现某行全是a_j，那么此次分解是无损连接分解。

   特别的当分解后的关系只有两个，那么如果(U1 ∩ U2)→(U1 - U2)或者(U1∩U2)→(U2 - U1)有一个成立，那么分解是无损的。

8. 保持函数依赖的判断算法
    a. 先计算函数依赖闭包在分解上的投影G，然后求出原函数依赖除了G以外的函数依赖， 对于其中所有函数依赖X→Y, Y如果都在X_{G}^{+}，那么分解保持函数依赖
9. 分解为3NF
10. DBMS考虑提供的数据库安全和保护功能
    a. 安全性保护
        i. 用户鉴别
        ii. 存取权限控制
        iii. 视图机制
        iv. 跟踪审查
        v. 数据加密存储
    b. 完整性保护
        i. 删除元素
            1. 级联删除
            2. 受限删除
            3. 置空值删除
        ii. 插入元素
            1. 受限插入
            2. 递归插入
        iii. 修改元素
            1. 不可修改
            2. 可以修改
                a. 级联修改
                b. 受限修改
                c. 置空值修改

c. 并发控制

采用并发锁、共享锁实现三级封锁协议、两段封锁协议

三级封锁协议：

1. 一级封锁协议，在修改之前加X锁
2. 二级封锁协议，基于1，在读之前加S锁，读完释放S锁
3. 三级封锁协议，基于2，在事务结束之后才释放

产生死锁的解决办法：

1. 预防法（一次封锁法、顺序封锁法）
2. 诊断解除法

两段封锁协议：加锁动作在释放锁之前。使得事务具有可串行性，避免死锁发生。

d. 数据库恢复

i. 故障的分类

1. 事务故障
2. 系统故障
3. 介质故障
4. 病毒破坏

ii. 技术

1. 数据转储

   按是否能修改分为

   1. 静态转储
   2. 动态转储

   按转储的数据分为：

   1. 海量转储
      a. 增量转储
2. 什么是事务，有哪些特点

事务是数据恢复、并发控制的基本单位。原子性、一致性、隔离性、持续性

3. 日志文件

11. 并发引起的问题
    a. 丢失修改
    b. 脏读
    c. 不能重读
12. 什么是索引

    索引是一种数据结构，常见的有B+树、哈希索引。索引的主要目的是加快查询速度。
13. mysql语句
    a. 创建数据库 create database name;
    b. 删除数据库 drop database name;
    c. 选择某个数据库 use name;
    d. 数据类型 int、float、double、decimal、char、varchar、blob、text、date、time
    e. 创建表 create table tablename(column name column type, ...)engine=innodb default charset=utf8;
    f. 删除表 drop table name;
    g. 插入语句 insert into tablename(column name, ...) vaules(value1, ...);
    h. 查询 select column name from tablename where ... limit N offset N
    i. 更新 update table_name set fields=value1, ... where ...
    j. 删除 delete from table name where ...
    k. Like : where field like condition
    l. Union: select * from tablename1 Union all/distinct select * from tablename2
    m. Order by: select column name from tablename order by column name ASC/DESC
    n. group by: select function(column name from) from table group by column name

o. 等值连接：select table1.column table2.column from table1 Inner/left/right join table2 on table1.column = table2.column;

p. 正则表达式 select column from table1 where column1 regexp '...';

q. 事务 begin .... rollback/commit 可以设置set autocommit = 0/1

r. 修改表头：

    i. alter table tablename drop columnname

    ii. alter table tablename add columnname type [after columnname/first]

    iii. alter table tablename modify columnname newtype;

    iv. alter table tablename change columnname newcolumnname newtype;

    v. alter table tablename alter column set default value / drop default

    vi. alter table tablename rename to newtablename

s. 常见索引：

    i. 普通索引：create index indexname on tablename(columnname);drop index indexname on tablename

    ii. 唯一索引： CREATE UNIQUE INDEX indexName ON mytable(username(length))