## 一、判断题

请完成教材第三、四、五、六章习题的判断题

## 第三章

FFFFT

1. SQL语言是一种数据操作语言，不能用于应用程序编程。（🞨）
2. 除非select查询结果是单个数据，否者结果应是一个关系。（🞨）
3. 与表操作相同，视图也可以任意更新数据库的数据。（🞨）
4. 在SELECT查询语句中，可使用SORT子句来排序结果数据。（🞨）
5. 在多表关联查询中，如果最终的结果来自单表，可使用子查询实现。（✓）

## 第四章

TFTFT

1. E-R模型是一种实现现实世界概念数据模型的有效方法。（✓）
2. 弱实体的标识符中都含有他所依赖实体的标识符。（🞨）
3. 消除属性间传递依赖的关系满足3NF范式。（✓）
4. 关系数据库设计规范程度越高越好。（🞨）
5. 在逻辑数据库模型设计中可以加入视图对象元素。（✓）

## 第五章

TFFTT

1. 数据库调优是DBA进行数据库管理的工作之一。（✓）
2. 只要是事务程序，它就能保证数据一致性。（🞨）
3. 在数据库系统中，一旦用户登陆数据库后，就可以访问数据库。（🞨）
4. 若所有事务遵从两阶段加锁协议，则这些事务的任何并发调度都是可串行化调度。（✓）
5. 只要有数据库备份文件，就可以将数据库恢复到故障点状态。（✓）

## 第六章

FFTTT

1. 数据库的存储过程和触发器都可以有输入参数。（🞨）
2. JDBC和ODBC都可以在任何高级语言中建立与数据库的连接。（🞨）
3. Servlet程序是用java语言编写的。（✓）
4. 在ODBC数据库编程中，驱动程序的加载是由用户应用程序完成的。（✓）
5. 触发器可以用于实现数据库表的数据完整性约束。（✓）

## 二、简答题

#### 1．数据完整性包含什么，SQL中如何定义 ？

1. 实体完整性：1.主键不为空。2.主键应该唯一。
2. 参照完整性：1.作为外键的属性在该表中的取值集合应该属于它作为主键的表中取值集合。
3. 用户自定义完整性：1.用户自己对关系进行数据约束。

#### 2．在数据库设计过程中，各层次数据模型的用途是什么？

1. 概念数据模型：反映业务领域的数据对象的内在联系，确保数据模型满足用户数据需求。
2. 逻辑数据模型设计：使整个系统的实体联系更加完善和规范，以便于在特定类型数据库实现，同时又不依赖于具体的DBMS。
3. 物理数据模型：从系统设计实现角度描述数据模型在特定DBMS中的具体设计实现方案。

#### 3．在数据库系统中，如何预防事务死锁状况的出现？

1. 在并发事务执行时，预防死锁。如银行家算法
2. 当死锁出现后，解除死锁。在并发执行过程中，如果检测到死锁，以代价最小的方式让事务释放资源，解除死锁。除了互斥条件外，破坏另外三个引起死锁的三个必要条件，请求和保持，不可抢占，环路等待。

#### 4．JDBC连接数据库包括哪几个主要步骤？

1. 下载数据库对应的JDBC驱动。
2. 添加到相关库里面。
3. 加载java包，java.sql.\*，然后加载驱动程序（Class.forName()）。
4. 创建数据库连接对象，DriverManager.getConnection().
5. 用Connection对象的createStatement方法创建statement对象。
6. 调用statement对象的相关方法执行sql语句。
7. 处理数据库返回结果。
8. 关闭数据库连接。

#### 5．Mybatis访问数据库主要包括哪些基本步骤。

1. 读取配置文件SqlMapConfig.xml。
2. SqlSessionFactoryBuilder通过Configuration生成sqlSessionFactory对象。
3. 通过sqlSessionFactory打开一个数据库会话sqlSession,操作数据库需要通过sqlSession进行。
4. MyBatis底层自定义了Executor执行器接口操作数据库，Executor接口负责动态sql的生成和查询缓存的维护，将MappedStatement对象进行解析，sql参数转化，动态sql拼接，生成JDBC Statement对象。

#### 三、实践题

完成教材 P103 实践操作题。

请将各小题的SQL语句提交

|  |
| --- |
| 1. CREATE DATABASE EstateDB |
| create table Owner(  PersonID char(18) NOT NULL PRIMARY KEY ,  Name varchar(20) NOT NULL ,  Gender char(2) NOT NULL ,  Occupation varchar(20) NOT NULL ,  Addr varchar(50) NOT NULL ,  Tel varchar(11) not null  );  create table Estate(  EstateID char(15) NOT NULL PRIMARY KEY ,  EstateName varchar(50) NOT NULL ,  EstateBuildName varchar(50) NOT NULL ,  EstateAddr varchar(60) NOT NULL ,  EstateCity varchar(60) NOT NULL ,  EstateType char(4) NOT NULL check ( EstateType IN ('住宅','商铺','车位','别墅')),  PropertyArea numeric(5,2) NOT NULL,  UsableArea numeric(5,2) NOT NULL,  CompletedDate date NOT NULL ,  YearLength int NOT NULL DEFAULT 70,  Remark varchar(100)  );  create table Registration(  RegisterID int NOT NULL ,  PersonID char(18) NOT NULL ,  EstateID char(15) NOT NULL ,  CONSTRAINT RegisterID\_PK PRIMARY KEY (RegisterID),  CONSTRAINT PersonID\_FK FOREIGN KEY (PersonID) REFERENCES Owner(PersonID) on DELETE cascade,  CONSTRAINT EstateID\_FK FOREIGN KEY (EstateID) REFERENCES Estate(EstateID) on DELETE cascade ,  Price money NOT NULL ,  PurchasedDate date NOT NULL ,  DeliverDate date NOT NULL  ); |
| 1. insert into Owner values('412821200004297236', '周玉川', '男', '前端攻城狮', '河南驻马店', '13258266367');     insert into Estate values('123456789112345', '玉川地产', '玉川楼盘', '成都成华区', '成都', '住宅', '120', '110', '2020-6-15'  ,70,'无备注');  insert into Registration values(429, '412821200004297236', '123456789112345', 3500000, '2019-10-10','2020-10-1'); |
| 1. select \* from Estate   where EstateType = '商铺'; |
| 1. select \* Estate   where CompletedDate > ‘2018-12-1’ and PropertyArea > 90; |
| select \*  from Owner  where PersonID in  (  select PersonID  from Registration  GROUP BY Registration.PersonID  HAVING count(\*)>2  ); |
| select \*  from Owner  where PersonID in  (  select PersonID  from Registration，Estate  where Registration.EstateID = Estate.EstateID and EstateCity = ‘成都’  GROUP BY Registration.PersonID  HAVING count(\*)>0  ); |
| select sum(PropertyArea) as 销售面积  from Estate  where EstateCity = '成都'  GROUP BY EstateType; |
| select sum(Price) as 销售价格  from Estate,Registration  where EstateCity = '成都' and Estate.EsateID = Registration.EstateID  GROUP BY EstateType; |
| create view query1 as  select Estate.EstateID,EstateName,EstateType,PropertyArea,Price,EstateBuildName,EstateCity  from Estate,Registration  where PersonID = '412821200004297236' and Estate.EstateID = Registration.EstateID  order by PurchasedDate |
| select count(Estate.EstateID), sum(Price)  from Estate,Registration  where Estate.EstateID = Registration.EstateID  GROUP BY EstateCity |