

《MySQL 数据库 - 就业技能》

| | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch01 - 初识 MySQL |
| 题目 1) | 数据库的作用 |
| | 存储数据、检索数据、生成新的数据 |
| 题目 2) | 数据库第一范式,及其理解 |
| | <p>第一范式 (1NF)</p> <p>在任何一个关系数据库中, 第一范式 (1NF) 是对关系模式的基本要求, 不满足第一范式 (1NF) 的数据库就不是关系数据库。</p> <p>所谓第一范式 (1NF) 是指数据库表的每一列都是不可分割的基本数据项, 同一列中不能有多多个值, 即实体中的某个属性不能有多个值或者不能有重复的属性。如果出现重复的属性, 就可能需要定义一个新的实体, 新的实体由重复的属性构成, 新实体与原实体之间为一对多关系。在第一范式 (1NF) 中表的每一行只包含一个实例的信息。简而言之, 第一范式就是无重复的列。</p> |
| 题目 3) | 数据库第二范式,及其理解 |
| | <p>第二范式 (2NF)</p> <p>第二范式 (2NF) 是在第一范式 (1NF) 的基础上建立起来的, 即满足第二范式 (2NF) 必须先满足第一范式 (1NF)。第二范式 (2NF) 要求数据库表中的每个实例或行必须可以被惟一地区分。为实现区分通常需要为表加上一个列, 以存储各个实例的惟一标识。这个惟一属性列被称为主关键字或主键、主码。</p> <p>第二范式 (2NF) 要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性, 如果存在, 那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体, 新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列, 以存储各个实例的惟一标识。简而言之, 第二范式就是非主属性非部分依赖于主关键字。</p> |
| 题目 4) | 数据库第三范式,及其理解 |
| | <p>第三范式 (3NF)</p> <p>满足第三范式 (3NF) 必须先满足第二范式 (2NF)。简而言之, 第三范式 (3NF) 要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。例如, 存在一个部门信息表, 其中每个部门有部门编号 (dept_id)、部门名称、部门简介等信息。那么在员工信息表中列出部门编号后就不能再将部门名称、部门简介等与部门有关的信息再加入员工信息表中。如果不存在部门信息表, 则根据第三范式 (3NF) 也应该构建它, 否则就会有大量的数据冗余。简而言之, 第三范式就是属性不依赖于其它非主属性。(消除冗余)</p> |
| 题目 5) | 关系型数据库和非关系型数据库的区别 |
| | <p>1.实质:非关系型数据库的实质: 非关系型数据库产品是传统关系型数据库的功能阉割版本, 通过减少用不到或很少用的功能, 来大幅度提高产品性能。</p> <p>2.价格:目前基本上大部分主流的非关系型数据库都是免费的。而比较有名气的关系型数据库, 比如 Oracle、DB2、MSSQL 是收费的。虽然 Mysql 免费, 但它需要做很多工作才能正式用于生产。</p> <p>3.功能:实际开发中, 有很多业务需求, 其实并不需要完整的关系型数据库功能, 非关系型数据库的功能就足够使用了。这种情况下, 使用性能更高、成本更低的非关系型数据库当然是更明智的选择。</p> |

| | |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch02 - MySQL 数据库数据管理 |
| 题目 1) | 时下流行的关系型数据库 |
| | <p>Oracle</p> <p>Oracle 公司的产品、产品免费、服务收费</p> <p>SQL Server</p> <p>针对不同用户群体的多个版本、易用性好</p> <p>MySQL</p> <p>开放源代码、网站应用广泛</p> |
| 题目 2) | Mysql 数据库的优势 |
| | <p>运行速度快</p> <p>使用成本低</p> <p>可移植性强</p> <p>适用用户广</p> |
| 题目 3) | 简述存储引擎，MyISAM 和 InnoDB 的应用场合 |
| | <p>使用 MyISAM: 不需事务，空间小，以查询访问为主</p> <p>使用 InnoDB: 多删除、更新操作，安全性高，事务处理及并发控制</p> |
| 题目 4) | drop,delete 与 truncate 的区别 |
| | <p>drop 直接删掉表</p> <p>truncate 删除表中数据，再插入时自增长 id 又从 1 开始 delete 删除表中数据，可以加 where 字句，再插入时自增长 id 不会重置为 1</p> |
| 题目 5) | MySQL 中 varchar 与 char 的区别 |
| | <p>在单字节字符集下，char (N) 在内部存储的时候总是定长，而且没有变长字段长度列表中。在多字节字符集下面，char(N)如果存储的字节数超过 N,那么 char (N)将和 varchar (N) 没有区别。在多字节字符集下面，如果存储的字节数少于 N，那么存储 N 个字节，后面补空格，补到 N 字节长度。都存储变长的数据和变长字段长度列表。varchar(N)无论是什么字节字符集，都是变长的，即都存储变长数据和变长字段长度列表。</p> |

| | |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch03 - 使用 DQL 查询数据 (一) |
| 题目 1) | 外联接的种类 |
| | <p>1.左外联接 left join</p> <p>如果左表的某行在右表中没有匹配行，则在相关联的结果集行中右表的所有选择列表列均为空值(null)</p> <p>2.右外联接 right join</p> <p>如果右表的某行在左表中没有匹配行，则将为左表返回空值</p> <p>3.完整外联接 full join</p> <p>当某行在另一个表中没有匹配行时，则另一个表的选择列表列包含空值。如果表之间有匹配行，则整个结果集行包含基表的数据值。</p> |
| 题目 2) | 内联接的种类 |
| | <p>1.内联接 inner join</p> <p>内联接是用比较运算符比较要联接列的值的联接</p> <p>2.交叉联接 cross join</p> |

| | |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | 没有 WHERE 子句的交叉联接将产生联接所涉及的表的笛卡尔积。第一个表的行数乘以第二个表的行数等于笛卡尔积结果集的大小。 |
| 题目 3) | SQL 语句优化 |
| | <p>1.应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。</p> <p>2.应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描。</p> <p>3.很多时候用 exists 代替 in 是一个好的选择</p> <p>4.用 Where 子句替换 HAVING 子句 因为 HAVING 只会检索出所有记录之后才对结果集进行过滤</p> |
| 题目 4) | 什么是子查询以及应用场景 |
| | <p>当一个查询是另一个查询的条件时，称之为子查询。</p> <p>子查询可以使用几个简单命令构造功能强大的复合命令。</p> <p>子查询最常用于 SELECT-SQL 命令的 WHERE 子句中。</p> <p>子查询是一个 SELECT 语句，它嵌套在一个 SELECT、SELECT...INTO 语句、INSERT...INTO 语句、DELETE 语句、或 UPDATE 语句或嵌套在另一子查询中。</p> |
| 题目 5) | 子查询注意事项 |
| | 子查询语句可以嵌套在 SQL 语句中任何表达式出现的位置，只出现在子查询中而没有出现在父查询中的表不能包含在输出列中 |

| | |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch04 - 使用 DQL 查询数据 (二) |
| 题目 1) | 数据类型选择 |
| | <p>能够用数字类型的字段尽量选择数字类型而不用字符串类型的</p> <p>a) 数字类型 Float 和 double 选择 (尽量选择 float)</p> <p>b) 字符类型 char,varchar,text 的选择: 非万不得已不要使用 TEXT 数据类型, 定长字段, 建议使用 CHAR 类型 (填充空格), 不定长字段尽量使用 VARCHAR (自动适应长度, 超过阶段), 且仅仅设定适当的最大长度</p> <p>c) 时间类型 按选择优先级排序 DATE (精确到天)、TIMESTAMP、DATETIME (精确到时间)</p> <p>d) 避免使用 NULL 字段, 很难查询优化且占用额外索引空间</p> |
| 题目 2) | 什么是索引 |
| | 索引是对数据库表中一列或多列的值进行排序的一种结构 |
| 题目 3) | 索引的优缺点 |
| | <p>优点:</p> <p>a. 大大加快数据的检索速度</p> <p>b. 创建唯一性索引, 保证数据库表中每一行数据的唯一性</p> <p>c. 可以加速表与表之间的连接</p> <p>缺点:</p> <p>a. 索引需要占物理空间。</p> <p>b. 当对表中的数据进行增加、删除和修改的时候, 索引也要动态的维护,</p> <p>c. 降低了数据的维护速度。</p> |

| | |
|--------------|-----------------------------------------------------------------------|
| 题目 4) | 索引分类 |
| | a.普通索引 b.唯一索引 c.主键索引 d.组合索引 |
| 题目 5) | 什么是数据库事务 |
| | 数据库事务(Database Transaction) , 是指作为单个逻辑工作单元执行的一系列操作, 要么完全地执行, 要么完全不执行。 |

| | |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch05 - 使用 JDBC 操作数据库 |
| 题目 1) | jdbc 是什么 (Java Database Connectivity) |
| | JDBC 是 Java 数据库连接技术的简称, 提供连接各种常用数据库的能力 |
| 题目 2) | 纯 Java 方式连接数据库的特征 |
| | 1.由 JDBC 驱动直接访问数据库 2.优点: 完全 Java 代码, 快速、跨平台 3.缺点: 访问不同的数据库需要下载专用的 |
| 题目 3) | PreparedStatement 和 Statement 的特点 |
| | 1.PreparedStatement 继承自 Statement 2.PreparedStatement 使用起来更加灵活, 更加高效 |
| 题目 4) | jdbc 的工作原理以及工作过程 |
| | JDBC 只定义接口, 具体实现由各个数据库厂商负责。 程序员使用时只需要调用接口, 实际调用的是底层数据库厂商的实现部分。 加载驱动, 建立连接 创建语句对象 执行 SQL 语句 处理结果集 关闭连接 |
| 题目 5) | DriverManager、Connection、Statement、ResultSet 各自的作用 |
| | DriverManager : 依据数据库的不同, 管理 JDBC 驱动 Connection : 负责连接数据库并担任传送数据的任务 Statement : 由 Connection 产生、负责执行 SQL 语句 ResultSet : 负责保存 Statement 执行后所产生的查询结果 |

| | |
|--------------|------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch06 - DAO 模式及单例模式 |
| 题目 1) | 什么是 dao 设计模式 |
| | DAO (数据访问对象) 的主要功能是数据操作; |
| 题目 2) | DAO 由几个部分组成 |
| | DatabaseConnection : 专门负责数据的打开与关闭操作的类; VO: 主要由属性、setter、getter 方法组成, VO 类中属性与表中的字段对应, 每一个 VO 类的对象都表示表中的每一条记录; |

| | |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>DAO: 主要定义操作的接口, 定义一系列数据库的原子性操作标准, 如增加、修改、删除、按 ID 查询等;</p> <p>Impl: DAO 接口的真实实现类, 完成具体的数据库操作, 但是不负责数据库的打开和关闭;</p> <p>Proxy: 代理实现类, 主要完成数据库的打开和关闭, 并且调用真实的实现类对象的操作;</p> <p>Factory: 工厂类, 通过工厂取得一个 DAO 的实例化操作;</p> |
| 题目 3) | 程序开发的标准架构, 以及作用 |
| | <p>客户层: 现在都采用 B/S 开发架构, 一般用户都使用浏览器进行访问;</p> <p>显示层: 使用 JSP/Servlet 进行页面效果的显示;</p> <p>业务层 (Business Object, 业务对象): 会将多个原子性的 DAO 操作进行组合, 组合成一个完整的业务逻辑;</p> <p>数据层 (DAO): 提供多个原子性的 DAO 操作, 如增加、修改、删除等, 都属于原子性操作;</p> |
| 题目 4) | DAO 的封装的原则 |
| | <p>a) 一个表对应一个 DAO, 相应地封装一个 DAO 类</p> <p>b) 对于 DAO 接口, 必须由具体的类型定义。这样可以避免被错误地调用</p> |
| 题目 5) | 为什么要使用 Properties 类读取配置文件 |
| | <p>能够脱离程序本身去修改相关的变量设置</p> <p>Properties 配置文件: 由键-值对组成</p> <p>读取配置文件: 使用 Properties 类相应方法</p> |

| | |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch07 - 项目实战-嗖嗖移动业务大厅 (一) |
| 题目 1) | 数据库事务的四大特性 |
| | <p>(1) 原子性 事务必须是原子工作单元; 对于其数据修改, 要么全都执行, 要么全都不执行。</p> <p>(2) 一致性 事务的一致性指的是在一个事务执行之前和执行之后数据库都必须处于一致性状态。事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。</p> <p>(3) 隔离性 (关于事务的隔离性数据库提供了多种隔离级别) 一个事务的执行不能干扰其它事务。即一个事务内部的操作及使用的数据对其它并发事务是隔离的, 并发执行的各个事务之间不能互相干扰。</p> <p>(4) 持久性 事务完成之后, 它对于数据库中的数据改变是永久性的。该修改即使出现系统故障也将一直保持。</p> |
| 题目 2) | 列举几个数据库 sql 函数 |
| | <p>Avg() 平均值</p> <p>Count() 数量</p> <p>Max() 最大值</p> <p>Min() 最小值</p> <p>Sum() 总和</p> |
| 题目 3) | 描述 group by 的作用 |
| | <p>GROUP BY 语句用于结合聚合函数, 根据一个或多个列对结果集进行分组。如果查询列中有聚合函数, 那么其他列必须在聚合函数里面, 否则 mysql 数据库查询结果不正确, oracle 数据库直接报错</p> |
| 题目 4) | 主键 超键 候选键 外键 |

| | |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>1) 超键(super key)：在关系中能唯一标识元组的属性集称为关系模式的超键</p> <p>2) 候选键(candidate key)：不含有多余属性的超键称为候选键</p> <p>3) 主键(primary key)：用户选作元组标识的一个候选键称为主键</p> <p>4) 外键(foreign key)：如果关系模式 R1 中的某属性集不是 R1 的主键，而是另一个关系 R2 的主键则该属性集是关系模式 R1 的外键</p> |
| 题目 5) | 什么是数据库视图 |
| | 视图是虚拟的表，与包含数据的表不一样，视图只包含使用时动态检索数据的查询；不包含任何列或数据。使用视图可以简化复杂的 sql 操作，隐藏具体的细节，保护数据；视图创建后，可以使用与表相同的方式利用它们。 |

| | |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 章节 | Ch08 -项目实战-嗖嗖移动业务大厅（二） |
| 题目 1) | 数据库如何优化 |
| | <p>a.sql 优化</p> <p>b.索引优化</p> <p>c.数据库结构优化（数据是范式的运用）</p> <p>d.服务器硬件优化（多花钱）</p> |
| 题目 2) | 什么是数据库触发器 |
| | <p>触发器是一种特殊类型的存储过程，不由用户直接调用。创建触发器时会对其进行定义，以便在对特定表或列作特定类型的数据修改时执行。</p> <p>触发器可以查询其他表，而且可以包含复杂的 SQL 语句。它们主要用于强制服从复杂的业务规则或要求。例如，您可以根据客户当前的帐户状态，控制是否允许插入新订单。</p> <p>触发器也可用于强制引用完整性，以便在多个表中添加、更新或删除行时，保留在这些表之间所定义的关系。</p> |
| 题目 3) | 触发器和存储过程有什么区别 |
| | <p>触发器与存储过程非常相似，触发器也是 SQL 语句集，两者唯一的区别是触发器不能用 EXECUTE 语句调用，而是在用户执行 T-SQL 语句时自动触发（激活）执行。</p> <p>触发器是在一个修改了指定表中的数据时执行的存储过程。通常通过创建触发器来强制实现不同表中的逻辑相关数据的引用完整性和一致性。</p> <p>由于用户不能绕过触发器，所以可以用它来强制实施复杂的业务规则，以确保数据的完整性。</p> <p>触发器不同于存储过程，触发器主要是通过事件执行触发而被执行的，存储过程可以通过存储过程名称名字而直接调用。</p> |
| 题目 4) | |
| | |
| 题目 5) | |
| | |