

数据库考点汇总

基本概念：三级模式-两级映像、数据库设计

数据库模型：E-R模型、关系模型、关系代数（结合SQL语言）

规范化：函数依赖、键与约束、范式、模式分解

事务并发：并发三种问题、三级封锁协议

数据库新技术：数据库安全与备份、反规范化、分布式数据库、缓存数据库、数据库集群

案例考点：ER图+关系模式设计、反规范化技术、缓存数据库等新技术。

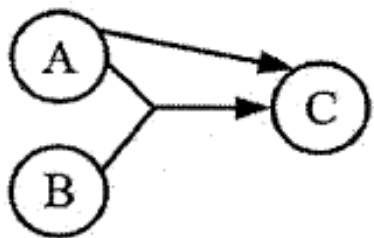
函数依赖

◆给定一个X，能唯一确定一个Y，就称X确定Y，或者说Y依赖于X，例如 $Y=X*X$ 函数。

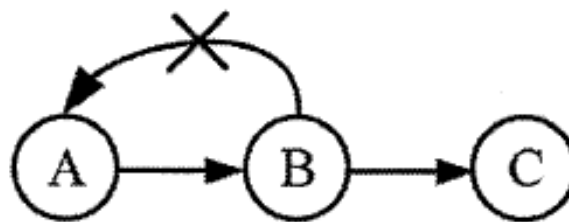
函数依赖又可扩展以下两种规则：

◆部分函数依赖：A可确定C，(A, B)也可确定C，(A, B)中的一部分（即A）可以确定C，称为部分函数依赖。

◆传递函数依赖：当A和B不等价时，A可确定B，B可确定C，则A可确定C，是传递函数依赖；若A和B等价，则不存在传递，直接就可确定C。



部分函数依赖



传递函数依赖

◆超键：能唯一标识此表的属性的组合。

◆候选键：超键中去掉冗余的属性，剩余的属性就是候选键。

◆主键：任选一个候选键，即可作为主键。

◆外键：其他表中的主键。

◆主属性：候选键内的属性为主属性，其他属性为非主属性。

函数依赖

◆函数依赖的公理系统(Armstrong)

设关系模式 $R\langle U, F \rangle$ ， U 是关系模式 R 的属性全集， F 是关系模式 R 的一个函数依赖集。对于 $R\langle U, F \rangle$ 来说有以下的：

◆自反律：若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 为 F 所逻辑蕴含

◆增广律：若 $X \rightarrow Y$ 为 F 所逻辑蕴含，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 为 F 所逻辑蕴含

◆传递律：若 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 为 F 所逻辑蕴含，则 $X \rightarrow Z$ 为 F 所逻辑蕴含

◆合并规则：若 $X \rightarrow Y$ ， $X \rightarrow Z$ ，则 $X \rightarrow YZ$ 为 F 所蕴涵

◆伪传递率：若 $X \rightarrow Y$ ， $WY \rightarrow Z$ ，则 $XW \rightarrow Z$ 为 F 所蕴涵

◆分解规则：若 $X \rightarrow Y$ ， $Z \subseteq Y$ ，则 $X \rightarrow Z$ 为 F 所蕴涵

范式

◆**第一范式1NF**：关系中的**每一个分量必须是一个不可分的数据项**

◆**实例**：用一个单一的关系模式学生来描述学校的教务系统：学生(学号, 学生姓名, 系号, 系主任姓名, 课程号, 成绩)

◆**依赖关系** (学号→学生姓名, 学号→系名, 系名→系主任姓名, (学号, 课程号)→成绩)

学号↵	学生姓名↵	所在系↵	系主任姓名↵	课程号↵	成绩↵
201102↵	张明↵	计算机系↵	章三↵	04↵	70↵
201103↵	王红↵	计算机系↵	章三↵	05↵	60↵
201103↵	王红↵	计算机系↵	章三↵	04↵	80↵
201103↵	王红↵	计算机系↵	章三↵	06↵	87↵
201104↵	李青↵	机械系↵	王五↵	09↵	79↵
... ↵	... ↵	... ↵	... ↵	... ↵	... ↵

范式

◆第二范式：如果关系R属于1NF，且每一个非主属性完全函数依赖于任何一个候选码，则R属于2NF。通俗地说，2NF就是在1NF的基础上，表中的每一个非主属性不会依赖复合主键中的某一个列。按照定义，上面的学生表就不满足2NF，因为学号不能完全确定课程号和成绩(每个学生可以选多门课)。将学生表分解为：

◆学生(学号, 学生姓名, 系名, 系主任)

◆选课(学号, 课程号, 成绩)。

每张表均属于2NF。

◆第三范式：在满足1NF的基础上，表中不存在非主属性对码的传递依赖。

◆继续上面的实例，学生关系模式就不属于3NF，因为学生无法直接决定系主任，是由学号→系名，再由系名→系主任，因此存在非主属性对主属性的传递依赖，

◆将学生表进一步分解为：

学生(学号, 学生姓名, 系编号)

系(系编号, 系名, 系主任)

选课(学号, 课程号, 成绩)

每张表都属于3NF。

考试真题

5&6. 某企业开发信息管理系统平台进行 E-R 图设计, 人力部门定义的是员工实体具有属性: 员工号、姓名、性别、出生日期、联系方式和部门, 培训部门定义的培训师实体具有属性: 培训师号, 姓名和职称, 其中职称={初级培训师, 中级培训师, 高级培训师}, 这种情况属于(5), 在合并E-R图时, 解决这一冲突的方法是(6)

A. 属性冲突 B. 结构冲突 C. 命名冲突 D. 实体冲突

- A. 员工实体和培训师实体均保持不变
- B. 保留员工实体、删除培训师实体
- C. 员工实体中加入职称属性, 删除培训师实体
- D. 将培训师实体所有属性并入员工实体, 删除培训师实体

试题(5) 通常在设计关系模式时, 派生属性不会作为关系中的属性来存储。按照这个原则, 原设计的学生关系模式为Students (学号, 姓名, 性别, 出生日期, 年龄, 家庭地址), 那么该关系模式正确的设计应为(5)。

- A. Students (学号, 性别, 出生日期, 年龄, 家庭地址)
- B. Students (学号, 姓名, 性别, 出生日期, 年龄)
- C. Students (学号, 姓名, 性别, 出生日期, 家庭地址)
- D. Students (学号, 姓名, 出生日期, 年龄, 家庭地址)

考试真题

试题 (6)、(7)

给出关系 $R(U, F)$, $U = \{A, B, C, D, E\}$, $F = \{A \rightarrow B, D \rightarrow C, BC \rightarrow E, AC \rightarrow B\}$, 求属性闭包的等式成立的是 (6)。 R 的候选关键字为 (7)。

(6) A. $(A)_F^+ = U$ B. $(B)_F^+ = U$ C. $(AC)_F^+ = U$ D. $(AD)_F^+ = U$

(7) A. AD B. AB C. AC D. BC

试题 (8) 在分布式数据库中有分片透明、复制透明、位置透明和逻辑透明等基本概念。其中, (8) 是指用户无需知道数据存放的物理位置。

(8) A. 分片透明 B. 逻辑透明 C. 位置透明 D. 复制透明

5. 数据库的安全机制中, 通过提供 (5) 供第三方开发人员调用进行数据更新, 从而保证数据库的关系模式不被第三方所获取。

A. 索引 B. 视图 C. 存储过程 D. 触发器

关系代数

- ◆笛卡尔积： $S1 \times S2$ ，产生的结果包括 $S1$ 和 $S2$ 的所有属性列，并且 $S1$ 中每条记录依次和 $S2$ 中所有记录组合成一条记录，最终属性列为 $S1+S2$ 属性列，记录数为 $S1 \times S2$ 记录数。
- ◆投影：实际是按条件选择某关系模式中的某列，列也可以用数字表示。
- ◆选择：实际是按条件选择某关系模式中的某条记录。

关系S1		
Sno	Sname	Sdept
No0001	Mary	IS
No0003	Candy	IS
No0004	Jam	IS

关系S2		
Sno	Sname	Sdept
No0001	Mary	IS
No0008	Katter	IS
No0021	Tom	IS

$S1 \times S2$ (笛卡尔积)					
Sno	Sname	Sdept	Sno	Sname	Sdept
No0001	Mary	IS	No0001	Mary	IS
No0001	Mary	IS	No0008	Katter	IS
No0001	Mary	IS	No0021	Tom	IS
No0003	Candy	IS	No0001	Mary	IS
No0003	Candy	IS	No0008	Katter	IS
No0003	Candy	IS	No0021	Tom	IS
No0004	Jam	IS	No0001	Mary	IS
No0004	Jam	IS	No0008	Katter	IS
No0004	Jam	IS	No0021	Tom	IS

(投影)	
Sno	Sname
No0001	Mary
No0003	Candy
No0004	Jam

(选择)		
Sno	Sname	Sdept
No0003	Candy	IS

关系代数

◆自然连接的结果显示全部的属性列，但是相同属性列只显示一次，显示两个关系模式中属性相同且值相同的记录。

设有关系R、S如下左图所示，自然连接结果如下右图所示：

<i>A</i>	<i>B</i>	<i>C</i>
a	b	c
b	a	d
c	d	e
d	f	g

(a) 关系 *R*

<i>A</i>	<i>C</i>	<i>D</i>
a	c	d
d	f	g
b	d	g

(b) 关系 *S*

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
a	b	c	d
b	a	d	g

$R \bowtie S$

SQL语句关键字

◆数据库查询select...from...where;

◆分组查询group by, 分组时要注意select后的列名要适应分组, having为分组查询附加条件:

select sno, avg(score) from student group by sno having (avg(score)>60)

◆更名运算as: select sno as “学号” from t1

◆字符串匹配like, %匹配多个字符串, _匹配任意一个字符串: select * from t1 where sname like 'a_'

◆数据库插入insert into...values(): insert into t1 values('a', 66)

◆数据库删除delete from...where: delete t1 where sno=4

◆数据库修改update...set...where: update t1 set sname='aa' where sno=3

◆排序order by, 默认为升序, 降序要加关键字DESC: select * from t1 order by sno desc

◆ DISTINCT: 过滤重复的选项, 只保留一条记录。

◆ UNION : 出现在两个SQL语句之间, 将两个SQL语句的查询结果取或运算, 即值存在于第一句或第二句都会被选出。

◆ INTERSECT : 对两个SQL语句的查询结果做与运算, 即值同时存在于两个语句才被选出。

◆ MIN、AVG、MAX: 分组查询时的聚合函数

考试真题

8. 分布式数据库系统除了包含集中式数据库系统的模式结构之外，还增加了几个模式级别，其中
(8) 定义分布式数据库中数据的整体逻辑结构，使得数据使用方便，如同没有分布一样。

- A. 分片模式
- B. 全局外模式
- C. 分布模式
- D. 全局概念模式

给定关系 $R(A, B, C, D, E)$ 与 $S(A, B, C, F, G)$ ，那么与表达式 $\pi_{1, 2, 4, 6, 7}(\sigma_{1 < 6}(R \bowtie S))$ 等价的 SQL 语句 如下：

SELECT (5) FROM R, S WHERE (6) ;

- (5) A. R. A, R. B, R. E, S. C, G B. R. A, R. B, D, F, G
 C. R. A, R. B, R. D, S. C, F D. R. A, R. B, R. D, S. C, G
- (6) A. R. A=S. A OR R. B=S. B OR R. C=S. C OR R. A<S. F
 B. R. A=S. A OR R. B=S. B OR R. C=S. C OR R. A<S. B
 C. R. A=S. A AND R. B=S. B AND R. C=S. C AND R. A<S. F
 D. R. A=S. A AND R. B=S. B AND R. C=S. C AND R. A<S. B

考试真题

数据仓库中，数据（8）是指数据一旦进入数据仓库后，将被长期保留并定期加载和刷新，可以进行各种查询操作，但很少对数据进行修改和删除操作。

（8） A. 面向主题 B. 集成性 C. 相对稳定性 D. 反映历史变化

给定关系模式 $R(U, F)$ ，其中：属性集 $U = \{A1, A2, A3, A4, A5, A6\}$ ，函数依赖集 $F = \{A1 \rightarrow A2, A1 \rightarrow A3, A3 \rightarrow A4, A1A5 \rightarrow A6\}$ 。关系模式 R 的候选码为（ ），由于 R 存在非主属性对码的部分函数依赖，所以 R 属于（ ）。

A. $A1A3$ B. $A1A4$ C. $A1 A5$ D. $A1A6$
A. 1NF B. 2NF C. 3NF D. BCNF

分布式数据库两阶段提交协议中的两个阶段是指（12）。

A. 加锁阶段、解锁阶段 B. 获取阶段、运行阶段
C. 表决阶段、执行阶段 D. 扩展阶段、收缩阶段

考试真题

给定元组演算表达式 $R^* = \{t \mid (\exists u) (R(t) \wedge S(u) \wedge t[3] < u[2])\}$ ，若关系 R、S 如下图所示，则（）。

A	B	C
1	2	3
4	5	6
7	8	9
10	11	12

R

A	B	C
3	7	11
4	5	6
5	9	13
6	10	14

S

- A. $R^* = \{(3, 7, 11), (5, 9, 13), (6, 10, 14)\}$
- B. $R^* = \{(3, 7, 11), (4, 5, 6), (5, 9, 13), (6, 10, 14)\}$
- C. $R^* = \{(1, 2, 3), (4, 5, 6), (7, 8, 9)\}$
- D. $R^* = \{(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12)\}$

案例真题1

阅读以下关于数据库设计的叙述，在答题纸上回答问题1至问题3。

【说明】

某医药销售企业因业务发展，需要建立线上药品销售系统，为用户提供便捷的互联网药品销售服务、该系统除了常规药品展示、订单、用户交流与反馈功能外，还需要提供当前热销产品排名、评价分类管理等功能。

通过对需求的分析，在数据管理上初步决定采用关系数据库（MySQL）和数据库缓存（Redis）的混合架构实现。

经过规范化设计之后，该系统的部分数据库表结构如下所示。

供应商（供应商ID，供应商名称，联系方式，供应商地址）；

药品（药品ID，药品名称，药品型号，药品价格，供应商ID）；

药品库存（药品ID，当前库存数量）；

订单（订单号码，药品ID，供应商ID，药品数量，订单金额）；

案例真题1

【问题1】（9分）

在系统初步运行后，发现系统数据访问性能较差。经过分析，刘工认为原来数据库规范化设计后，关系表过于细分，造成了大量的多表关联查询，影响了性能。例如当用户查询商品信息时，需要同时显示该药品的信息、供应商的信息、当前库存等信息。

为此，刘工认为可以采用反规范化设计来改造药品关系的结构，以提高查询性能。修改后的药品关系结构为：

药品（药品ID，药品名称，药品型号，药品价格，供应商ID，供应商名称，当前库存数量）；

请用200字以内的文字说明常见的反规范化设计方法，并说明用户查询商品信息应该采用哪种反规范化设计方法。

【问题2】（9分）

王工认为，反规范化设计可提高查询的性能，但必然会带来数据的不一致性问题。请用200字以内的文字说明在反规范化设计中，解决数据不一致性问题的三种常见方法，并说明该系统应该采用哪种方法。

【问题3】（7分）

该系统采用了Redis来实现某些特定功能（如当前热销药品排名等），同时将药品关系数据放到内存以提高商品查询的性能，但必然会造成Redis和MySQL的数据实时同步问题。

（1）Redis的数据类型包括String、Hash、List、Set和ZSet等，请说明实现当前热销药品排名的功能应该选择使用哪种数据类型。

（2）请用200字以内的文字解释说明解决Redis和MySQL数据实时同步问题的常见方案。

案例真题1

答案：

【问题1】

常见反规范化技术如下：

- （1）增加冗余列：在多个表中保留相同的列，通过增加数据冗余减少或避免查询时的连接操作。
 - （2）增加派生列：在表中增加可以由本表或其它表中数据计算生成的列，减少查询时的连接操作并避免计算或使用集合函数。
 - （3）重新组表：如果许多用户需要查看两个表连接出来的结果数据，则把这两个表重新组成一个表来减少连接而提高性能。
 - （4）水平分割表：根据一行或多列数据的值，把数据放到多个独立的表中，主要用于表数据规模很大、表中数据相对独立或数据需要存放到多个介质上时使用。
 - （5）垂直分割表：对表进行分割，将主键与部分列放到一个表中，主键与其它列放到另一个表中，在查询时减少I/O次数。
- 用户查询商品信息采用的是增加冗余列的方式。

案例真题1

【问题2】

批处理维护、应用逻辑和触发器。

(1) 批处理维护：指对复制列或派生列的修改积累一定的时间后，运行一批处理作业或存储过程对复制或派生列进行修改，这只能在对实时性要求不高的情况下使用。

(2) 应用逻辑：要求必须在同一事务中对所有涉及的表进行增、删、改操作。用应用逻辑来实现数据的完整性风险较大，因为同一逻辑必须在所有的应用中使用和维护，容易遗漏，特别是在需求变化时，不易于维护。

(3) 触发器：对数据的任何修改立即触发对复制列或派生列的相应修改。触发器是实时的，而且相应的处理逻辑只在一个地方出现，易于维护。一般来说，是解决这类问题比较好的办法。

该系统应该采用触发器。

【问题3】

(1) ZSet

(2) 一、对强一致要求比较高的，应采用实时同步方案，即查询缓存查询不到再从DB查询，保存到缓存；更新缓存时，先更新数据库，再将缓存的设置过期(建议不要去更新缓存内容，直接设置缓存过期)。

二、对于并发程度较高的，可采用异步队列的方式同步，可采用kafka等消息中间件处理消息生产和消费。

三、使用阿里的同步工具canal，canal实现方式是模拟mysql slave和master的同步机制，监控DB binlog的日志更新来触发缓存的更新，此种方法可以解放程序员双手，减少工作量，但在使用时有些局限性。

四、采用UDF自定义函数的方式，面对mysql的API进行编程，利用触发器进行缓存同步。

案例真题2

某互联网文化发展公司因业务发展，需要建立网上社区平台，为用户提供一个对网络文化产品(如互联网小说、电影、漫画等)进行评论、交流的平台。该平台的部分功能如下：

- (a) 用户帖子的评论计数器；
- (b) 支持粉丝列表功能；
- (c) 支持标签管理；
- (d) 支持共同好友功能等；
- (e) 提供排名功能，如当天最热前10名帖子排名、热搜榜前5排名等；
- (f) 用户信息的结构化存储；
- (g) 提供好友信息的发布/订阅功能。

该系统在性能上需要考虑高性能、高并发，以支持大量用户的同时访问。开发团队经过综合考虑，在数据管理上决定采用Redis+数据库(缓存+数据库)的解决方案。

【问题1】(10分)
Redis支持丰富的数据类型，并能够提供一些常见功能需求的解决方案。请选择题干描述的(a)~(g)功能选项，填入表4-1中(1)~(5)的空白处。

表 4-1 Redis 数据类型与业务功能对照表

数据类型	存储的值	可实现的业务功能
STRING	字符串、整数或浮点数	(1)
LIST	列表	(2)
SET	无序集合	(3)
HASH	包括键值对的无序散列表	(4)
ZSET	有序集合	(5)

案例真题2

【问题2】(7分)

该网上社区平台需要为用户提供7*24小时的不间断服务。同时在系统出现宕机等故障时，能在最短时间内通过重启等方式重新建立服务。为此，开发团队选择了 Redis持久化支持。Redis有两种持久化方式，分别是RDB(Redis DataBase)持久化方式和AOF(Append Only File)持久化方式。开发团队最终选择了 RDB方式。

请用200字以内的文字，从磁盘更新频率、数据安全、数据一致性、重启性能和数据文件大小五个方面比较两种方式，并简要说明开发团队选择RDB的原因。

【问题3】(8分)

缓存中存储当前的热点数据，Redis为每个KEY值都设置了过期时间，以提高缓存命中率。为了清除非热点数据，Redis选择“定期删除+惰性删除”策略。如果该策略失效，Redis 内存使用率会越来越高，一般应采用内存淘汰机制来解决。

请用100字以内的文字简要描述该策略的失效场景，并给出三种内存淘汰机制。

案例真题2

参考答案

【问题1】 (1) (a) (2) (b)、(g) (3) (c)、(d) (4) (f) (5) (e)

【问题2】 磁盘更新频率：AOF比RDB文件更新频率高。

数据安全：AOF比RDB更安全。

数据一致性：RDB间隔一段时间存储，可能发生数据丢失和不一致；AOF通过append 模式写文件，即使发生服务器宕机，也可通过redis-check-aof工具解决数据一致性问题。

重启性能：RDB性能比AOF好。

数据文件大小：AOF文件比RDB文件大。

综合上述五个方面的比较，考虑在系统出现宕机等故障时，需要在最短时间内通过重启等方式重新建立服务，因此开发团队最终选择了RDB方式。

【问题3】 失效场景：如果“定期删除”没删除KEY，也没即时去请求KEY，也就是说“惰性删除”也没生效。这样，Redis默认的“定期删除+惰性删除”策略就失效了。

对此，可采用内存淘汰机制解决：

- (1) 从已设置过期时间的数据集最近最少使用的数据淘汰。
- (2) 从已设置过期时间•间的数据集将要过期的数据淘汰。
- (3) 从已设置过期时间的数据集任意选择数据淘汰。
- (4) 从数据集最近最少使用的数据淘汰。
- (5) 从数据集任意选择数据淘汰。



谢谢！