

作业2: SQL (2020春)

主讲教师: 邹兆年(znzou@hit.edu.cn)

姓名: _____ 学号: _____

题目	1	2	3	4	5	6	7	8	9	10	总分
得分											

1. (10分, 每题2分) 判断对错

- (a) SQL语句DELETE FROM TABLE R从数据库中删除关系R。
- (b) 将属性声明为PRIMARY KEY和UNIQUE NOT NULL作用是一样的。
- (c) ORDER BY A, B DESC将查询结果按照属性A和B的值降序排列。
- (d) SQL语句SELECT A FROM R与关系代数表达式 $\Pi_A(R)$ 的结果相同。
- (e) 若关系R的属性A被声明为UNIQUE, 则SQL语句SELECT COUNT(A) FROM R的结果是|R|。

2. (85分, 每题5分) 在MySQL上创建Product数据库(Database Systems The Complete Book - Exercise 2.4.1), 然后使用SQL表达下列数据库查询与更新, 并在MySQL上验证。

- (a) Find the manufacturers that sell laptops but not PC's. (使用集合差运算)
- (b) Find the manufacturers that sell laptops but not PC's. (使用含有IN的嵌套查询)
- (c) Find the manufacturers that sell laptops but not PC's. (使用含有EXISTS的嵌套查询)
- (d) Find the model numbers of all printers that are cheaper than the printer model 3002. (使用内连接查询)
- (e) Find the model numbers of all printers that are cheaper than the printer model 3002. (使用含有比较运算符的嵌套查询)
- (f) Find the model numbers of all printers that are cheaper than the printer model 3002. (使用含有EXISTS的嵌套查询)
- (g) Find the PC model with the highest available speed. (使用外连接查询)
- (h) Find the PC model with the highest available speed. (使用含有IN的嵌套查询)
- (i) Find the PC model with the highest available speed. (使用含有=的嵌套查询)
- (j) Find the PC model with the highest available speed. (使用含有>=的嵌套查询)
- (k) Find the PC model with the highest available speed. (使用含有EXISTS的嵌套查询)
- (l) Find the manufacturers of PC's with at least three different speeds. (使用内连接查询)
- (m) Find the manufacturers of PC's with at least three different speeds. (使用分组查询)
- (n) Find the manufacturers of PC's with at least three different speeds. (使用派生关系)
- (o) Decrease the price of all PC's made by maker A by 10%. (使用含有=的更新条件)
- (p) Decrease the price of all PC's made by maker A by 10%. (使用含有IN的更新条件)
- (q) Decrease the price of all PC's made by maker A by 10%. (使用含有EXISTS的更新条件)

3. (5分) 第2题(g)中的查询可以用多种SQL语句表示。尝试从语句的易读性和执行效率两方面对2(g)-2(k)的SQL语句进行分析和比较。在做效率分析时, 我们假定每个关系上只有主索引, 而没有其他索引(请自学索引的概念)。

1. (a) 错
(b) 错
(c) 错
(d) 错
(e) 错
2. (a) (SELECT DISTINCT maker FROM Product WHERE type = 'laptop') EXCEPT
(SELECT DISTINCT maker FROM Product WHERE type = 'pc');
在MySQL上写成

SELECT DISTINCT M1.maker
FROM (SELECT maker FROM Product WHERE type = 'laptop') M1
NATURAL LEFT OUTER JOIN
(SELECT maker FROM Product WHERE type = 'pc') M2
WHERE M2.maker IS NULL;

(b) SELECT DISTINCT maker FROM Product
WHERE type = 'laptop' AND maker NOT IN (
SELECT maker FROM Product WHERE type = 'pc');

(c) SELECT DISTINCT maker FROM Product
WHERE type = 'laptop' AND NOT EXISTS (
SELECT * FROM Product P WHERE P.maker = Product.maker AND type = 'pc');

(d) SELECT P1.model
FROM Printer P1 JOIN Printer P2 ON (P1.price < P2.price)
WHERE P2.model = '3002';

(e) SELECT model FROM Printer
WHERE price < (SELECT P.price FROM Printer P WHERE P.model = '3002');

(f) SELECT model FROM Printer WHERE NOT EXISTS (
SELECT * FROM Printer P WHERE P.model = '3002' AND P.price <= Printer.price);

(g) SELECT PC1.model
FROM PC PC1 LEFT OUTER JOIN PC PC2 ON (PC1.speed < PC2.speed)
WHERE PC2.model IS NULL;

(h) SELECT model FROM PC WHERE speed IN (SELECT MAX(speed) FROM PC);

(i) SELECT model FROM PC WHERE speed = (SELECT MAX(speed) FROM PC);

(j) SELECT model FROM PC WHERE speed >= ALL (SELECT speed FROM PC);

(k) SELECT model FROM PC
WHERE NOT EXISTS (SELECT * FROM PC PC2 WHERE PC2.speed > PC.speed);

(l) SELECT DISTINCT P1.maker
FROM (Product P1 NATURAL JOIN PC PC1)
JOIN (Product P2 NATURAL JOIN PC PC2) ON (P1.maker = P2.maker)
JOIN (Product P3 NATURAL JOIN PC PC3) ON (P1.maker = P3.maker)
WHERE PC1.speed != PC2.speed AND PC2.speed != PC3.speed AND PC1.speed != PC3.speed;

(m) SELECT maker FROM Product NATURAL JOIN PC
GROUP BY maker HAVING COUNT(DISTINCT speed) >= 3;

(n) SELECT R.maker
FROM (SELECT maker, COUNT(DISTINCT speed) cnt
FROM Product NATURAL JOIN PC GROUP BY maker) R
WHERE R.cnt >= 3;

(o) UPDATE PC SET price = price * 0.9
WHERE 'A' = (SELECT maker FROM Product WHERE Product.model = PC.model);

- (p) UPDATE PC SET price = price * 0.9
WHERE model IN (SELECT model FROM Product WHERE maker = 'A');
- (q) UPDATE PC SET price = price * 0.9
WHERE EXISTS (SELECT * FROM Product WHERE maker = 'A' AND Product.model = PC.model);
3. 在SQL语句易读性方面, (i)最好, (h)和(j)次之, (g)和(k)最差。在效率方面, 由于在speed上没有索引, (g), (j), (k)的效率最差; (h)和(i)需要对PC进行2遍扫描, 效率高一些。

作业3：数据库设计(2020春)

主讲教师：邹兆年(znzou@hit.edu.cn)

姓名：_____ 学号：_____

题目	1	2	3	4	5	6	7	8	9	10	总分
得分											

- (10分) 举例说明3个实体型之间的三元联系型与这3个实体型之间的3个二元联系型表示的意义通常是不同的。在什么情况下3个实体型之间的三元联系型与这3个实体型之间的3个二元联系型表示的意义会是相同的？
- (60分) 根据下面的大学数据库信息，用ER图表示该数据库的概念模型，明确标出键、基数比、参与度约束、角色。
 - Professors have an SSN, a name, an age, a rank, and a research specialty.
 - Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.
 - Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.).
 - Each project is managed by one professor (known as the project's principal investigator).
 - Each project is worked on by one or more professors (known as the project's co-investigators).
 - Professors can manage and/or work on multiple projects.
 - Each project is worked on by one or more graduate students (known as the project's research assistants).
 - When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
 - Departments have a department number, a department name, and a main office.
 - Departments have a professor (known as the chairman) who runs the department.
 - Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
 - Graduate students have one major department in which they are working on their degree.
 - Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.
- (20分) 将上题中的ER模型转换成关系数据库模式，明确给出关系的主键和外键。
- (10分) 该大学数据库上经常执行下面的查询：
 - 人事处经常按项目的负责人(principal investigator)和起止日期来查询项目信息；
 - 科研处经常按项目的资助方(sponsor)和起止日期来查询项目信息；

为该数据库设计一个索引来提高上述查询的处理效率，写出创建该索引的SQL语句，说明你用到了哪些索引技术。

答案

1. 如果满足下面的条件，则3个实体型之间的三元联系型与这3个实体型之间的3个二元联系型表示的含义相同：对于任意两个三元联系 (a, b, c) 和 (a', b', c') ，
 - 如果 $a = a'$ 且 $b = b'$ ，则 $c = c'$ 。
 - 如果 $a = a'$ 且 $c = c'$ ，则 $b = b'$ 。
 - 如果 $c = c'$ 且 $b = b'$ ，则 $a = a'$ 。
- 2.
3.
 - Professor(ssn, name, age, rank, speicality)
 - Project(id, sponsor, start_date, end_date, budget, pi)
 - Graduate(ssn, name, age, degree, department, advisor)
 - Department(id, name, office, chair)
 - CoI(id, ssn)
 - RA(id, ssn, supervisor)
 - Affiliation(id, ssn, time)

 - Project.pi参照Professor.ssn
 - Graduate.department参照Department.id
 - Graduate.advisor参照Graduate.ssn
 - Department.chair参照Professor.ssn
 - CoI.id参照Project.id
 - CoI.ssn参照Professor.ssn
 - RA.id参照Project.id
 - RA.ssn参照Graduate.ssn
 - RA.supervisor参照Professor.ssn
 - Affiliation.id参照Department.id
 - Affiliation.ssn参照Professor.ssn
4. CREATE INDEX idx_date ON Project(start_date, end_date, PI_SSN);
该索引不仅可以支持第一个查询，而且其前缀索引可以支持第二个查询。

作业4: 关系数据库规范化(2020春)

主讲教师: 邹兆年(znzou@hit.edu.cn)

姓名: _____ 学号: _____

题目	1	2	3	4	5	6	7	8	9	10	总分
得分											

1. (5分) 已知关系模式 R 的一个实例如下:

A	B	C
x_1	y_1	z_1
x_1	y_1	z_1
x_2	y_1	z_1
x_2	y_1	z_3

R 上可能存在哪些非平凡函数依赖?

2. (5分) 已知关系模式 $R(A, B, C)$ 表示两个实体型之间的联系, 这两个实体型的主键分别为 A 和 B 。设 R 上存在函数依赖 $A \rightarrow B$ 和 $B \rightarrow A$, 那么这两个实体型之间存在什么联系型?
3. (5分) 已知关系模式 $R(A, B, C)$ 的主键是 A , 且满足函数依赖 $B \rightarrow C$, 那么 R 是否可能属于BCNF? 如果是, 需要满足什么条件? 如果不是, 请说明原因。
4. (25分, 每题5分) 判断下列命题是否正确。对于正确的命题, 用Armstrong公理证明其正确性。对于错误的命题, 给出一个关系实例, 使该实例满足推理规则左侧的函数依赖集合, 但不满足右侧的函数依赖集合。
- (a) $\{W \rightarrow Y, X \rightarrow Z\} \models \{WX \rightarrow Y\}$
 - (b) $\{X \rightarrow Y, X \rightarrow W, WY \rightarrow Z\} \models \{X \rightarrow Z\}$
 - (c) $\{X \rightarrow Z, Y \rightarrow Z\} \models \{X \rightarrow Y\}$
 - (d) $\{X \rightarrow Y, Z \rightarrow W\} \models \{XZ \rightarrow YW\}$
 - (e) $\{X \rightarrow Y, Y \rightarrow Z\} \models \{X \rightarrow YZ\}$

5. (60分, 每题10分) 已知关系模式 $R(A, B, C, D, E)$ 上的函数依赖集合 F 如下:

$$A \rightarrow B \quad A \rightarrow C \quad BC \rightarrow A \quad CD \rightarrow E \quad B \rightarrow D \quad E \rightarrow A$$

回答下列问题:

- (a) 计算属性集合 BC 关于 F 的闭包 $(BC)_F^+$ 。
- (b) 找出 R 的全部候选键。
- (c) 判断 R 属于第几范式。
- (d) 计算 F 的极小覆盖。
- (e) 假设 R 被分解为两个关系模式 $R_1(A, B, C)$ 和 $R_2(C, D, E)$, 证明该分解不是无损连接分解。提示: 给出 R 的一个实例 r , 使 $\Pi_{A, B, C}(r) \bowtie \Pi_{C, D, E}(r) \neq r$ 。
- (f) 给出 R 的一个既满足无损连接性, 又满足函数依赖保持性的BCNF分解。

答案

1. R 上可能存在的非平凡函数依赖有 $A \rightarrow B$ 和 $AC \rightarrow B$ 。
2. 两个实体型之间存在1:1联系型。
3. R 可能属于BCNF。如果 $B \rightarrow A$ 是 R 上的函数依赖，则 $R \in BCNF$ 。
4. (a) 正确。
 - 根据自反律，有 $WX \rightarrow W$ 。
 - 根据传递律， $\{WX \rightarrow W, W \rightarrow Y\} \models \{WX \rightarrow Y\}$ 。
- (b) 正确。
 - 根据合并规则， $\{X \rightarrow Y, X \rightarrow W\} \models \{X \rightarrow WY\}$ 。
 - 根据传递律， $\{X \rightarrow WY, WY \rightarrow Z\} \models \{X \rightarrow Z\}$ 。
- (c) 错误。假设关系 $R(X, Y, Z)$ 的实例中包含2个元组 (x, y, z) 和 (x, y', z) ，其中 $y \neq y'$ 。
- (d) 正确。
 - 根据增广律， $\{X \rightarrow Y\} \models \{XZ \rightarrow YZ\}$ 。
 - 根据增广律， $\{Z \rightarrow W\} \models \{YZ \rightarrow YW\}$ 。
 - 根据传递律， $\{XZ \rightarrow YZ, YZ \rightarrow YW\} \models \{XZ \rightarrow YW\}$ 。
- (e) 正确。
 - 根据增广律， $\{Y \rightarrow Z\} \models \{Y \rightarrow YZ\}$ 。
 - 根据传递律， $\{X \rightarrow Y, Y \rightarrow YZ\} \models \{X \rightarrow YZ\}$ 。
5. (a) i. $X^{(0)} = BC$.
 ii. $X^{(1)} = X^{(0)} \cup AD = ABCD$.
 iii. $X^{(2)} = X^{(1)} \cup BCE = ABCDE$. 因为 $X^{(2)}$ 中已包含 R 中全部属性，算法终止。因此， $(BC)_F^+ = ABCDE$ 。
- (b)
 - A 是 R 的候选键，因为 $A \xrightarrow{f} ABCDE$ 。
 - BC 是 R 的候选键，因为 $BC \xrightarrow{f} ABCDE$ 。
 - CD 是 R 的候选键，因为 $CD \xrightarrow{f} ABCDE$ 。
 - E 是 R 的候选键，因为 $E \xrightarrow{f} ABCDE$ 。
- (c) R 的所有属性都是主属性，因此 $R \in 3NF$ 。另外，因为 $B \rightarrow D$ ，所以主属性部分函数依赖于候选键 BC ，故 $R \notin BCNF$ 。
- (d)
 - $BC \rightarrow A$ 是冗余的，因为 $F - \{BC \rightarrow A\} \models \{BC \rightarrow A\}$ 。因此， $F = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ 。
 - F 中任意函数依赖的左部没有冗余属性。
 - $A \rightarrow B$ 和 $A \rightarrow C$ 具有相同的左部，合并为 $A \rightarrow BC$ ，因此 $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ 。
 - 此时， F 中不存在冗余函数依赖，任意函数依赖的左部不存在冗余属性，不存在可合并的函数依赖，所以 F 的极小覆盖是 $\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ 。
- (e) 设 $r = \{(a, b, c, d, e), (a', b', c, d', e')\}$ 。我们有

$$\Pi_{A,B,C}(r) \bowtie \Pi_{C,D,E}(r) = \{(a, b, c, d, e), (a', b', c, d', e'), (a, b, c, d', e'), (a', b', c, d, e)\} \neq r.$$
- (f) $\{R_1(A, B, C, E), R_2(C, D, E), R_3(B, D)\}$ 是 R 的分解，其中 $R_1 \in BCNF$ ， $R_2 \in BCNF$ ， $R_3 \in BCNF$ 。该分解既满足无损连接性，又满足函数依赖保持性。

作业5: 关系数据库存储与查询执行(2020春)

主讲教师: 邹兆年(znzou@hit.edu.cn)

姓名: _____ 学号: _____

题目	1	2	3	4	5	6	7	8	9	10	总分
得分											

1. (10分) 在PostgreSQL上首先使用下面的SQL语句创建关系 t :

```
CREATE TABLE t (  
  id INT PRIMARY KEY,  
  val CHAR(1024) NOT NULL DEFAULT 'val');
```

然后使用下面的SQL语句插入元组:

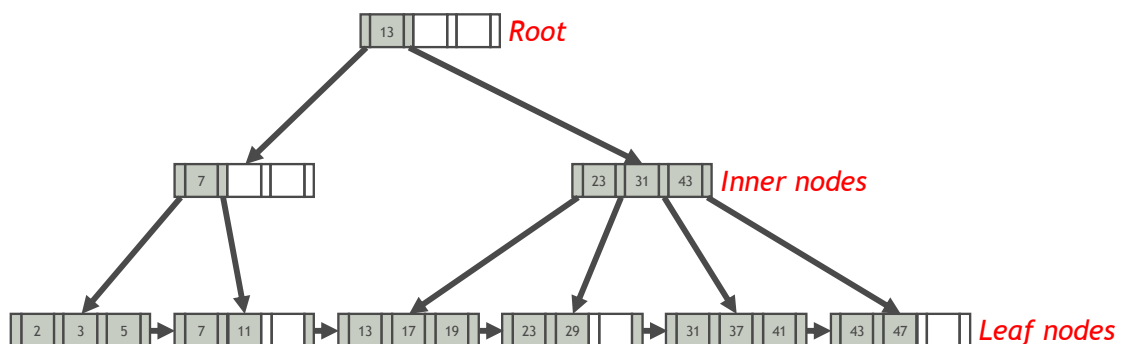
```
INSERT INTO t(id) VALUES (1), (2), (3), (4), (5), (6), (7), (8), (9), (10);
```

执行SQL查询“SELECT ctid, id FROM t;”, 我们得到以下查询结果:

ctid	id
(0,1)	1
(0,2)	2
(0,3)	3
(0,4)	4
(0,5)	5
(0,6)	6
(0,7)	7
(1,1)	8
(1,2)	9
(1,3)	10

其中ctid表示元组的记录号(record ID), 即元组所在页号(page ID)和槽号(slot number)构成的对。请分析该DBMS使用的页的大小是4KB、8KB、16KB中的哪一个? 说明理由。

2. (20分) 已知如下B+树



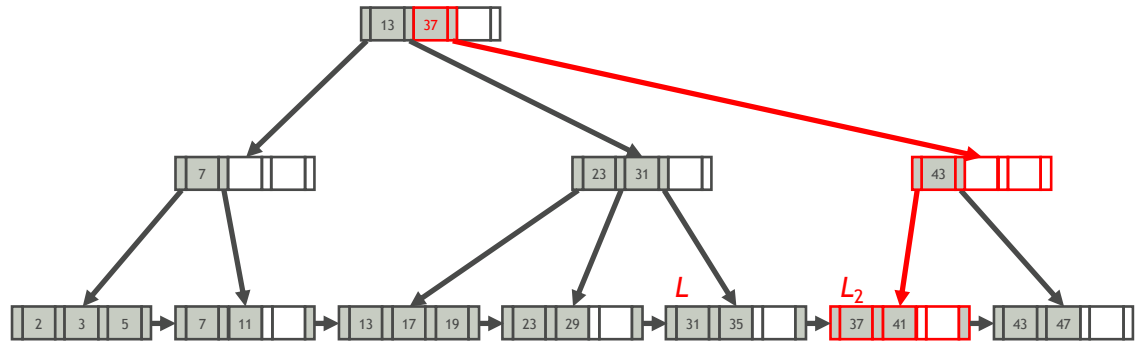
回答下列问题。

- (a) (10分) 插入键值为35的索引项(index entry)后, 该B+树变成什么样? 请绘制出来。
 - (b) (10分) 删除键值为13的索引项(index entry)后, 该B+树变成什么样? 请绘制出来。
3. (70分) 已知关系 $R(w, x), S(x, y), T(y, z)$ 的块数分别为5000, 10000, 10000。我们准备执行关系代数查询 $(R \bowtie S) \bowtie T$ 。假设缓冲池中有 $M = 101$ 个页可用, R, S, T 上均无索引且未按连接属性排序。请回答下列问题。
- (a) (10分) 使用什么算法执行 $R \bowtie S$ 最适合? 说明理由。
 - (b) (10分) 使用(a)中选择的算法执行 $R \bowtie S$ 的I/O代价是多少?
 - (c) (10分) 如果 $R \bowtie S$ 的结果不超过49块, 那么在使用(a)中选择的算法执行 $R \bowtie S$ 时, $R \bowtie S$ 的结果是否需要物化(materialize)到文件中? 说明理由。
 - (d) (10分) 如果 $R \bowtie S$ 的结果不超过49块, 那么使用什么算法将 $R \bowtie S$ 的结果与 T 进行自然连接最合适? 说明理由。
 - (e) (10分) 使用(d)中选择的算法计算连接结果的I/O代价是多少?
 - (f) (10分) 如果 $R \bowtie S$ 的结果大于49块, 那么使用什么算法将 $R \bowtie S$ 的结果与 T 进行自然连接最合适? 说明理由。
 - (g) (10分) 使用(f)中选择的算法计算连接结果的I/O代价是多少?

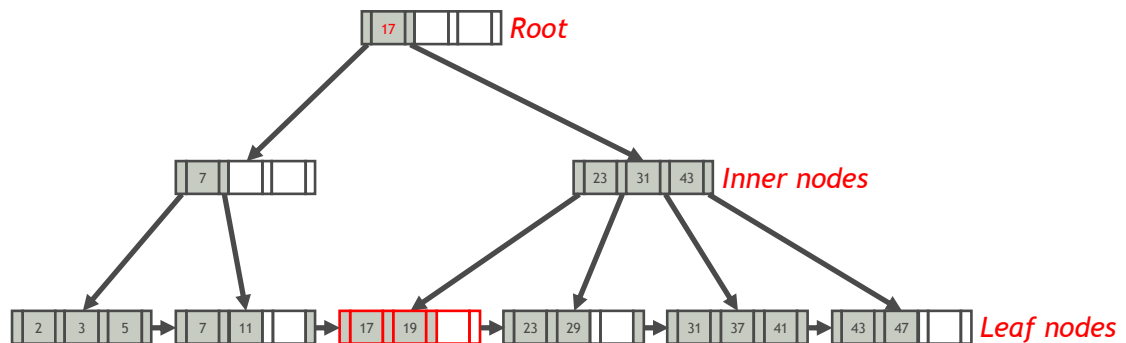
答案

1. 8KB。根据关系模式的定义，一个元组的大小至少为1KB。根据查询结果，一个页面最多存放7个元组，因此页面至少是7KB。所以，在4KB、8KB、16KB中，最合理的选择是8KB。

2. (a)



(b)



- 3.
- Grace哈希连接算法最合适，因为 R 和 S 的块数都超过了 M ，一趟算法不可用； R 和 S 无索引，索引连接不可用； R 和 S 未排序，排序归并连接不可用。
 - $3B(R) + 3B(S) = 45000$ 。
 - 不需要。使用Grace哈希连接算法执行 $R \bowtie S$ ， R 和 S 都被分到100个桶中，因此 R 的每个桶大约50块， S 的每个桶大约100块。在执行 $R_i \bowtie S_i$ 时，可以使用一趟连接算法，需要使用内存缓冲区51个页面，还剩50个，能够存放 $R \bowtie S$ 的结果。
 - 如果 $R \bowtie S$ 的结果不超过49块，那么在执行 $R \bowtie S$ 时，结果可以存放在剩余可用缓冲区中，因此使用一趟连接算法执行 $(R \bowtie S) \bowtie T$ 最合适。
 - 参考第10章PPT最后一部分。
 - 策略如下：
 - 如果 $k \leq 49$, one-pass join + pipelining
 - 如果 $50 < k \leq 300$, nested-loop join + materialization
 - 如果 $300 < k \leq 5000$, Grace hash join + pipelining
 - 如果 $k > 5000$, Grace hash join + materialization

作业1：关系数据模型(2020春)

主讲教师：邹兆年(znzou@hit.edu.cn)

姓名：_____ 学号：_____

题目	1	2	3	4	5	6	7	8	9	10	总分
得分											

- (30分，每题3分) 判断下列命题是否成立。若不成立，请给出反例。
 - $\sigma_{\theta_1}(\sigma_{\theta_2}(R)) = \sigma_{\theta_2}(\sigma_{\theta_1}(R)) = \sigma_{\theta_1 \wedge \theta_2}(R)$
 - $\Pi_{L_1}(\Pi_{L_2}(R)) = \Pi_{L_2}(\Pi_{L_1}(R))$
 - $\Pi_L(\sigma_\theta(R)) = \sigma_\theta(\Pi_L(R))$
 - $\Pi_L(R \cup S) = \Pi_L(R) \cup \Pi_L(S)$ // 假定 R 和 S 满足求并的条件
 - $\Pi_L(R \cap S) = \Pi_L(R) \cap \Pi_L(S)$ // 假定 R 和 S 满足求交的条件
 - $\sigma_\theta(R \cap S) = \sigma_\theta(R) \cap S = R \cap \sigma_\theta(S)$
 - $\sigma_\theta(R - S) = \sigma_\theta(R) - S = R - \sigma_\theta(S)$
 - $(R \bowtie_{\theta_1} S) \bowtie_{\theta_2} T = R \bowtie_{\theta_1} (S \bowtie_{\theta_2} T)$
 - $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
 - $R \bowtie R = R \cap R$
- (10分) 设属性 K 是关系 R 的主键，写一个关系代数表达式来验证 R 的实例是否违反实体完整性约束，说明如何用该关系代数表达式的结果来完成验证。
- (10分) 设属性 K 是关系 R 的主键，关系 S 的外键 F 参照 $R.K$ ，写一个关系代数表达式来验证 R 和 S 的实例是否违反参照完整性约束，说明如何用该关系代数表达式的结果来完成验证。
- (50分) 在关系代数运算器(<https://dbis-uibk.github.io/relax/calc.htm>)上加载数据集“Database Systems The Complete Book - Exercise 2.4.1”，用关系代数表达式表示下列查询，并用关系代数运算器进行验证。要求：每个查询均使用两种不同类型方法做，给出关系代数表达式、表达式树和查询结果(关系代数表达式用数学公式写，表达式树和查询结果在关系代数运算器中截图)。
 - (15分) Find the PC model with the highest available speed.
 - (15分) Find those hard-disk sizes that occur in two or more PC's.
 - (15分) What manufacturers make all types of products (PC, laptop, and printer)?
 - (5分) $Product \bowtie Printer$ 的结果是什么？为什么会得到这样的结果？

答案

1. (a) 成立。
 (b) 不成立。 $\Pi_a(\Pi_{a,b}(R)) \neq \Pi_{a,b}(\Pi_a(R))$
 (c) 不成立。 $\Pi_a(\sigma_{b>0}(R)) = \sigma_{b>0}(\Pi_a(R))$
 (d) 成立。
 (e) 不成立。设 $R(a,b) = \{(1,2)\}$, $S(a,b) = \{(1,3)\}$, 则 $\Pi_a(R \cap S) = \emptyset$, $\Pi_a(R) \cap \Pi_a(S) = \{(1)\}$
 (f) 成立。
 (g) 不成立。 $\sigma_\theta(R - S) = \sigma_\theta(R) - S$, 但 $\sigma_\theta(R) - S \neq R - \sigma_\theta(S)$ 。设 $R(a,b) = \{(1,2)\}$, $S(a,b) = \{(3,4)\}$, 则 $\sigma_{a=3}(R) - S = \emptyset$, $R - \sigma_{a=3}(S) = \{(1,2)\}$
 (h) 不成立。 $R(a,b), S(b,c), T(a,c), (R \bowtie_{R.b=S.b} S) \bowtie_{R.a=T.a} T \neq R \bowtie_{R.b=S.b} (S \bowtie_{R.a=T.a} T)$
 (i) 成立。
 (j) 成立。
2. 若 $\Pi_K(\sigma_{K=null}(R)) \cup \Pi_K(\sigma_{amt>1}(\gamma_{K;count(*) \rightarrow amt}(R)))$ 的结果不为空, 则 R 的实例违反实体完整性约束。
3. 关系代数表达式可以是 $\sigma_{R.K=null}(\sigma_{K \neq null}(S) \bowtie_{S.K=R.K} R)$ 或 $\Pi_K(\sigma_{K \neq null}(S)) - \Pi_K(R)$ 。若关系代数表达式的结果不为空, 则 R 和 S 的实例违反参照完整性约束。
4. (a) (用集合差) $\Pi_{model}(PC) - \Pi_{PC1.model}(\rho_{PC1}(PC) \bowtie_{PC1.speed < PC2.speed} \rho_{PC2}(PC))$
 (用外连接) $\Pi_{PC1.model}(\sigma_{PC2.model=null}(\rho_{PC1}(PC) \bowtie_{PC1.speed < PC2.speed} \rho_{PC2}(PC)))$
 (用聚集) $\Pi_{model}(PC \bowtie_{speed=max_spd} \gamma_{max(speed) \rightarrow max_spd}(PC))$
 (b) (用内连接) $\Pi_{PC1.hd}(\rho_{PC1}(PC) \bowtie_{PC1.hd=PC2.hd \wedge PC1.model \neq PC2.model} \rho_{PC2}(PC))$
 (用分组聚集) $\Pi_{hd}(\sigma_{amt \geq 2}(\gamma_{hd;count(*) \rightarrow amt}(PC)))$
 (c) (用集合交) $\Pi_{maker}(\sigma_{type='pc'}(Product)) \cap \Pi_{maker}(\sigma_{type='laptop'}(Product)) \cap \Pi_{maker}(\sigma_{type='printer'}(Product))$
 (用内连接) $\Pi_{P1.maker}(\rho_{P1}(Product) \bowtie_{P1.maker=P2.maker \wedge P1.type='pc' \wedge P2.type='laptop'} \rho_{P2}(Product) \bowtie_{P1.maker=P3.maker \wedge P3.type='printer'} \rho_{P3}(Product))$
 (用分组聚集) $\Pi_{maker}(\sigma_{amt=3}(\gamma_{maker;count(*) \rightarrow amt}(\Pi_{maker,type}(\sigma_{type='pc' \vee type='laptop' \vee type='printer'}(Product)))))$
 (用除法) $\Pi_{maker,type}(Product) \div \Pi_{type}(\sigma_{type='pc' \vee type='laptop' \vee type='printer'}(Product))$
 (d) $Product \bowtie Printer = \emptyset$, 因为 $Product$ 和 $Printer$ 的同名属性有 $maker$ 和 $type$, 而 $Product.type$ 和 $Printer.type$ 的取值域不同, 因此 $Product$ 和 $Printer$ 中没有元组满足自然连接条件 $Product.maker = Printer.maker \wedge Product.type = Printer.type$ 。