DBMS 2016.10月考试真题

**填空（1'\*15）**

1. 数据仓库的四个基本特征  面面向主题 ， 不可更新， 集成， 随时间不断变化的    。

2. 数据仓库的设计中存在三级数据模型，分别是 概念模型  ，逻辑模型 ，物理模型 。

3. 从操作型数据库到数据仓库的过程中，要进行的步骤有数据抽取，   转换    ， 装载       ， 清洗      。

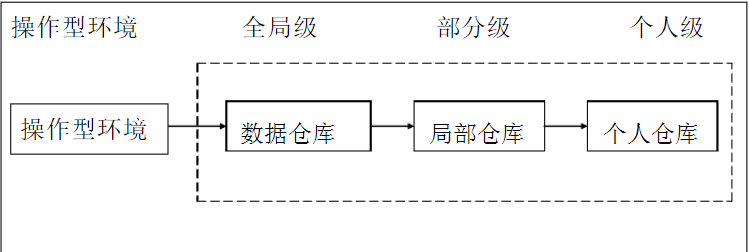
4. 粒度越小，细节成都越    越高    ，查询的种类越   越多      。

5. OLAP服务器的实现方法有ROLAP， MOLAP ， HOLAP 。

**名词解释（4'\*5）**

1，分割：指将数据分散到各自的物理单元中去以便能分别独立处理。

2. 数据仓库体系化环境



3. 粒度：对数据仓库中的数据的综合程度高低的一个度量,粒度越小，细节程度越高，综合程度越低,粒度大小影响数据仓库效率、能回答询问的种类,数据仓库是多粒度的，不同的粒度回答不同的查询

4. 分类：按照分析对象的属性，特征建立不同的组类来描述事务。例如银行部门根据以前的数据将客户分为不同的类别，现在就可以根据这些来区分新申请贷款的客户，以采取相应的贷款方案。

5. 聚类分析：根据数据的特征找出数据间的相似性，将相似的数据分成一个类。

**简答题（5'\*6）**

1. 简要说明数据仓库中元数据的内容

元数据是关于数据的数据。在数据仓库中，元数据是定义仓库对象的数据。对于给定数据仓库的数据名和定义，创建元数据。其它元数据包括对提取数据添加的时间标签、提取数据的源、被数据清理或集成处理添加的字段等。

2. 企业的数据仓库体系化结构的四个层次是什么，它们之间的关系是什么

层次的体系化环境四个层次分别为：操作型环境、全局级数据仓库、部门级的局部仓库、个人级数据仓库。

操作型环境存放：细节的操作型数据，服务于高性能事务处理

全局级数据仓库：存放细节数据、导出数据

部门级局部仓库：一般存放导出数据

个人级数据仓库：数据一般是暂时存放，用于启发式分析

1. 简要说明数据仓库的设计步骤

1.系统分析，确定主题

2.选择满足数据仓库系统要求的软件平台

3.建立数据仓库的逻辑模型

4.逻辑数据模型转化为数据仓库数据模型

5.数据仓库数据模型优化

6.数据清洗转换和传输

7.开发数据仓库的分析应用

8.数据仓库的管理

4. 你是如何理解数据仓库的数据是不可更新，又随时间不断变化的

不可更新，数据的主要操作是查询，原始数据一旦进入数据仓库后，一般不允许再修改，且会被长期保留。只需要定期的加载和刷新。

随时间不断变化，已有数据纪录不可更新，随时间变化不断增加新的数据，将不需要的数据从数据仓库中卸出，转储到其他设备，随着时间的变化不断地重新综合数据。

5. 举例说明数据仓库有哪三类聚集函数

分布型聚集函数，代数型聚集函数，整体型聚集函数

分布型聚集函数：将数据分成n份，对其中每一份应用该函数，可以得到n个聚集值，对这n个聚集值进行计算得到的结果和整个数据（不划分）应用该函数得出结果一致。具有可累计的特性。

代数型聚集函数：一个函数可以由若干个分布型函数进行代数运算得出。

整体型聚集函数：一个函数不能由其他函数进行代数运算得出。

（1）分布的：一个聚集函数是分布的，如果它能以如下分布方式进行计算：设数据被划分为n 个集合，函数在每一部分上的计算得到一个聚集值。如果将函数用于n 个聚集值得到的结果，与将函数用于所有数据得到的结果一样，则该函数可以用分布方式计算。例如，count()可以这样计算：首先将数据方分割成子方的集合，对每个子方计算count()，然后对这些子方得到的计数求和。因此，count()是分布聚集函数。同理，sum(), min()和max()是分布聚集函数。一个度量是分布的，如果它可以用分布聚集函数得到。

（2）代数的：一个聚集函数是代数的，如果它能够由一个具有M（其中，M 是一个整数界）个参数的代数函数计算，而每个参数都可以用一个分布聚集函数求得。例如，avg()可以由sum()/count()计算，其中sum()和count()是分布聚集函数。类似地，可以表明min\_N(), max\_N()和standard\_deviation()是代数聚集函数。一个度量是代数的，如果它可以用代数聚集函数得到。

（3）整体的：一个聚集函数是整体的，如果描述它的子聚集所需的存储没有一个常数界。即，不存在一个具有M 个（其中，M 是常数）参数的代数函数进行这一计算。整体函数的常见例子包括median(),mode()（即，最常出现的项），和rank()。一个度量是整体的，如果它可以用整体聚集函数得到。大部分数据方应用需要有效地计算分布的和代数的度量。对于这些，存在许多有效的技术。相比之下，有效地计算整体度量是很困难的。然而，对于有些整体函数的近似计算，有效的技术是存在的。

6. 为什么说naive Bayesian分类法是naive的

朴素贝叶斯分类假定一个属性值对给定类的影响独立于其它属性的值。该假定称作类条件独立。做此假定是为了简化所需计算，并在此意义下称为“朴素的”。

**问答题（10'\*2）**

1. 在数据仓库追加数据时，捕捉数据变化的常用途径有哪些

数据追加：数据仓库的数据初装完成后, 再向数据仓输入数据的过程

追加内容：上次数据追加后在OLTP数据库中变化了的数据

变化数据的捕捉途径:

1）时标方法　（如果数据含有时标，对新插入或更新的数据记录,加更新时的时标）

2）DELTA文件 （由应用生成DELTA文件，记录应用所改变的所有内容）

3）前后映象文件（抽取数据到数据仓库之后, 本次将抽取数据之前，对数据库分别作一次快照，比较两幅快照的不同，确定追加的数据）

4）日志文件（利用DB的固有机制，数据只限于日志文件，不用扫描整个数据库）

2. 试述数据仓库的设计中提高数据仓库性能的方法和技术

由于数据仓库的数据极少甚至不再更新，可采取如下技术来提高数据仓库的性能：

1)合并表

2)建立数据序列

3)引入冗余

4)进一步细分数据

5)生成导出数据

6)建立广义索引

7)粒度划分

8)分割

**计算题（15'）**

请根据能找出预先指定的参数，cluster的形状，存在的缺陷三个方面对如下聚类方法进行评价K-Means，BIRCH，DBSCAN，OPTICS

K-means算法是很典型的基于距离的[聚类](https://baike.baidu.com/item/%E8%81%9A%E7%B1%BB)算法，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法认为簇是由距离靠近的对象组成的，因此把得到紧凑且独立的簇作为最终目标。

k个初始类聚类中心点的选取对聚类结果具有较大的

[式](https://baike.baidu.com/pic/K-means/4934806/0/263e802f6a5bbe681f308981?fr=lemma&ct=single)公式

影响，因为在该[算法](https://baike.baidu.com/item/%E7%AE%97%E6%B3%95)第一步中是随机的选取任意k个对象作为初始聚类的中心，初始地代表一个簇。该算法在每次迭代中对数据集中剩余的每个对象，根据其与各个簇中心的距离将每个对象重新赋给最近的簇。当考察完所有数据对象后，一次迭代运算完成，新的聚类中心被计算出来。如果在一次迭代前后，J的值没有发生变化，说明算法已经收敛。

Birch 算法[1]  的主要思想是：通过扫描数据库，建立一个初始存放于内存中的聚类特征树， 然后对聚类特征树的叶结点进行聚类。它的核心是聚类特征（CF）和聚类特征树（CF Tree）。 2.1.1 CF CF 是指三元组CF=（N，LS，SS），用来概括子簇信息，而不是存储所有的数据点。 其中：N：簇中d 维点的数目； LS：N 个点的线性和；SS：N 个点的平方和。比如给定一 个由二维点组成的集合{(3,4),(2,6),(4,5)}，那么： CF 结构概括了簇的基本信息，并且是高度压缩的，它存储了小于实际数据点的聚类信 息。[3]同时CF 的三元结构设置使得计算簇的半径、簇的直径、簇与簇之间的距离等非常容 易。

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一个比较有代表性的基于密度的[聚类算法](https://baike.baidu.com/item/%E8%81%9A%E7%B1%BB%E7%AE%97%E6%B3%95)。与划分和层次聚类方法不同，它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的[空间数据库](https://baike.baidu.com/item/%E7%A9%BA%E9%97%B4%E6%95%B0%E6%8D%AE%E5%BA%93)中发现任意形状的聚类。

**Ordering points to identify the clustering structure** (**OPTICS**) OPTICS聚类算法是基于密度的聚类算法，全称是Ordering points to identify the clustering structure，目标是将空间中的数据按照密度分布进行聚类，其思想和DBSCAN非常类似，但是和DBSCAN不同的是，OPTICS算法可以获得不同密度的聚类，直接说就是经过OPTICS算法的处理，理论上可以获得任意密度的聚类。因为OPTICS算法输出的是样本的一个有序队列，从这个队列里面可以获得任意密度的聚类。

1.预先指定的参数，

K-Means，随机的选取任意k个对象作为初始聚类的中心，初始地代表一个簇

BIRCH，无预先指定的参数，通过扫描数据库，建立一个初始存放于内存中的聚类特征树， 然后对聚类特征树的叶结点进行聚类。

DBSCAN，扫描半径 (eps)和最小包含点数(minPts)

OPTICS，类似DBSCAN，后续算法不同

2.cluster的形状

K-Means， 质心为中心的类圆，各聚类本身尽可能的紧凑，而各聚类之间尽可能的分开

BIRCH， 树结构

DBSCAN，任意形状

OPTICS， 有序队列

3.存在的缺陷

K-Means，

① 在 K-means 算法中 K 是事先给定的，这个 K 值的选定是非常难以估计的。很多时候，事先并不知道给定的数据集应该分成多少个类别才最合适。这也是 K-means 算法的一个不足。有的算法是通过类的自动合并和分裂，得到较为合理的类型数目 K，例如 ISODATA 算法。关于 K-means 算法中聚类数目K 值的确定在文献中，是根据[方差分析](https://baike.baidu.com/item/%E6%96%B9%E5%B7%AE%E5%88%86%E6%9E%90)理论，应用混合 F[统计量](https://baike.baidu.com/item/%E7%BB%9F%E8%AE%A1%E9%87%8F)来确定最佳分类数，并应用了模糊划分熵来验证最佳分类数的正确性。在文献中，使用了一种结合全[协方差矩阵](https://baike.baidu.com/item/%E5%8D%8F%E6%96%B9%E5%B7%AE%E7%9F%A9%E9%98%B5)的 RPCL 算法，并逐步删除那些只包含少量训练数据的类。而文献中使用的是一种称为次胜者受罚的[竞争学习](https://baike.baidu.com/item/%E7%AB%9E%E4%BA%89%E5%AD%A6%E4%B9%A0)规则，来自动决定类的适当数目。它的思想是：对每个输入而言，不仅竞争获胜单元的权值被修正以适应输入值，而且对次胜单元采用惩罚的方法使之远离输入值。

② 在 K-means 算法中，首先需要根据初始聚类中心来确定一个初始划分，然后对初始划分进行优化。这个初始聚类中心的选择对聚类结果有较大的影响，一旦初始值选择的不好，可能无法得到有效的聚类结果，这也成为 K-means算法的一个主要问题。对于该问题的解决，许多算法采用[遗传算法](https://baike.baidu.com/item/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95)（GA），例如文献 中采用遗传算法（GA）进行初始化，以内部聚类准则作为评价指标。

③ 从 K-means 算法框架可以看出，该算法需要不断地进行样本分类调整，不断地计算调整后的新的聚类中心，因此当数据量非常大时，算法的时间开销是非常大的。所以需要对算法的时间复杂度进行分析、改进，提高算法应用范围。在文献中从该算法的时间复杂度进行分析考虑，通过一定的相似性准则来去掉聚类中心的侯选集。而在文献中，使用的 K-means 算法是对样本数据进行聚类，无论是初始点的选择还是一次迭代完成时对数据的调整，都是建立在随机选取的样本数据的基础之上，这样可以提高算法的收敛速度。

BIRCH，

1) 由于CF Tree对每个节点的CF个数有限制，导致聚类的结果可能和真实的类别分布不同.

2) 对高维特征的数据聚类效果不好。此时可以选择Mini Batch K-Means

3) 如果数据集的分布簇不是类似于超球体，或者说不是凸的，则聚类效果不好。

DBSCAN，

1. DBScan不能很好反映高维数据。

2. DBScan不能很好反映数据集以变化的密度。

OPTICS，

么有显示的生成产生数据簇，需要额外的存储空间