第二章作业

1.

- (1)
  - 共同点:
    - 两者都使用了`FOR XML`子句，查询结果将以XML格式返回,列值作为元素的属性
  - 区别:
    - RAW模式可将结果将元素命名为自定义的名称
    - AUTO模式以表名为元素的名称，可以形成简单的层次关系，表名作为节点，多表连接时从左至右的表依次形成父子关系的嵌套结构
- (2)
  - (a)

```
<row tid="1" tname="张一" sid="101" sname="李思" />
<row tid="1" tname="张一" sid="102" sname="王学" />
<row tid="2" tname="赵二" sid="103" sname="孙行" />
```

  - (b)

```
<teacher tid="1" tname="张一">
    <student sid="101" sname="李思" />
    <student sid="102" sname="王学" />
</teacher>
<teacher tid="2" tname="赵二">
    <student sid="103" sname="孙行" />
</teacher>
```

2.

1. **通过类似SQL的语法为用户提供Hadoop上数据管理与查询处理能力**：**在SELECT语句中增加**DISTRIBUTR BY、CLUSTER BY子句，类似MapReduce中partition进行分区
2. **直接嵌入MapRedece程序**

3.

- 执行流程：
    1. 创建决策树模型
        1. 定义客户会员卡类型 `member_card` 和其他属性之间的关系
        2. 将 `customer` 与 `sales_fact` 连接表用作训练模型的数据
    2. 创建决策树模型存储表
    3. 保存决策树模型存储表
    4. 使用模型对输入数据进行预测

4.

- (1)

```
INSERT INTO R1 (sno, cno, eval)
VALUES (
    1,
    3,
    '{"evaluation": [{"stage": 1, "result": "ok"}, {"stage": 2, "result": "good"},
{"stage": 3, "result": "excellent"}]}'
);
```

- (2)

```
SELECT sno, cno, stage, result
FROM R1,
JSON_TABLE(
    eval->'$.evaluation',
    '$[*]'
    COLUMNS(
        stage INT PATH '$.stage',
        result VARCHAR(50) PATH '$.result'
    )
) AS json_table
WHERE stage<3;
```