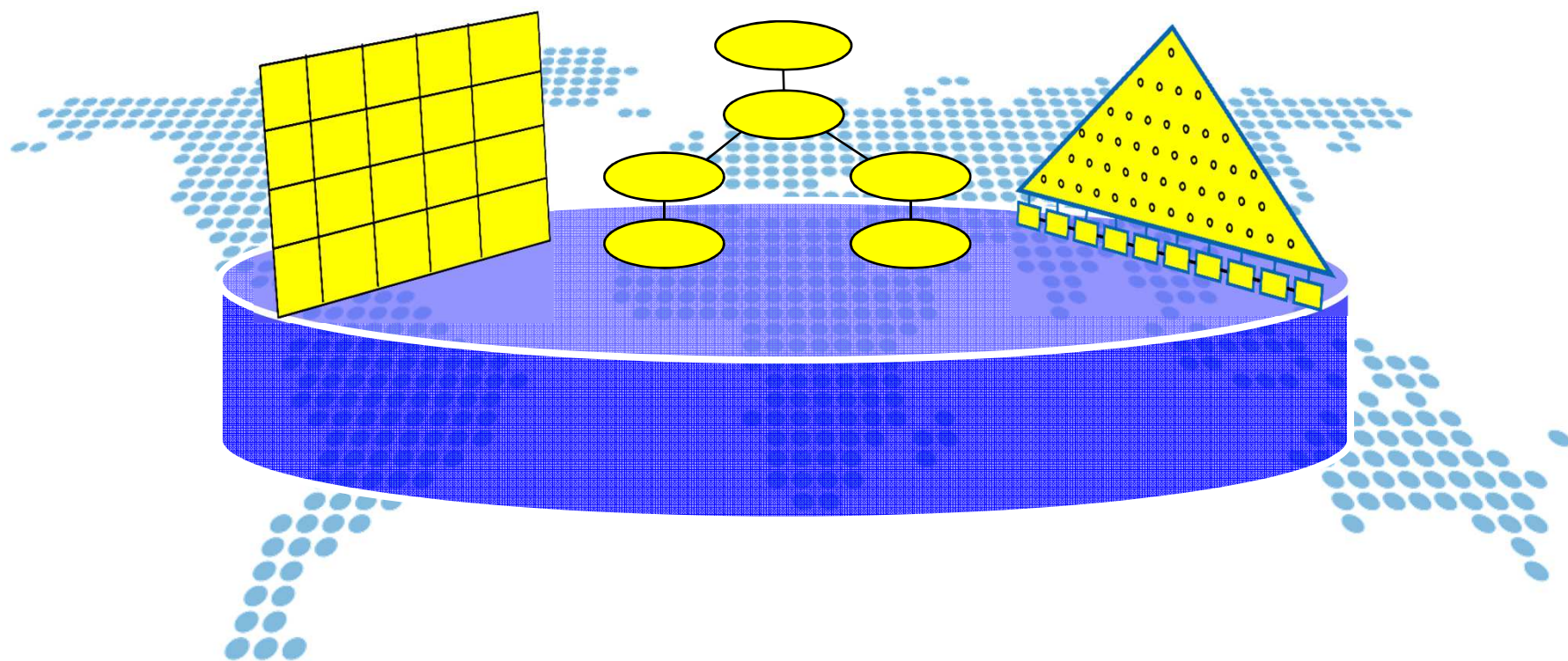


数据库系统

LAB3: 数据库原型实现

陈世敏

(中科院计算所)



实验3安排

- 内容：开发一个简化的数据库原型系统
- 掌握
 - 数据存储与访问路径
 - 查询处理
- 本堂课：12月1日
 - 实验3具体要求
- 1月5日：验收实验3
- 1月12日：期末考试

实验3：简化的数据库原型系统

• 简化

□ 两种数据类型：int和varchar

□ 简化的语句

- create/drop: table

- insert: 没有delete, update

- select

- 选择：在一个关系表上最多只有一个过滤条件

- 投影

- 连接：两个表，等值连接，可以用simple hash join

- 分组聚集：可以用一遍算法

运行方式

- 编译成为一个可执行文件

- 例如: ucasdb

- 命令行调用

- \$ ucasdb <包含sql语句的文件名>

- 打开文件, 依次执行文件中的SQL语句

- 每个语句以分号结束, 语句可以跨多行

- 以--两个减号开头的行为注释

- 语句内部没有注释, 注释一定在语句之间

举例

<simple.sql>

-- 1. create a table

create table student (id int, name varchar, grade int);

-- 2. insert a record

insert into student

values (123, 'Zhang Fei', 80);

-- 3. select

select name, grade

from student

where id=123;

\$ ucasdb simple.sql

说明：页的大小

- 长度4096字节
- 每个记录都可以在一页中放下，没有跨页的问题
- 数据表都采用4KB的页

说明：文件位置

- 所有文件都在当前目录下的db子目录下
- 每个表都存为一个文件
 - db/表名.tbl
- 系统的元信息也存储在db/目录下，具体名称格式自定义，没有统一要求

要求1：两种数据类型

- int

- 定长，32位整数

- varchar

- 变长，字符串

- 注意不是varchar(n)，我们省略了n

要求2：数据表的创建和删除

```
create table 表名 (  
    列名 类型,  
    ...  
    列名 类型  
);
```

```
drop table 表名;
```

- 类型为int 或varchar
- 省略：主键、外键、not null、unique、check等
- 不考虑属性值为null的情况

结果：数据表的创建和删除

- 语法错误：关键字错了，类型错了，没有分号，以及其它不符合前述语法的错误
 - 显示：“Syntax error”
- create table
 - 成功时显示 “Successfully created table 表名”
 - 失败时显示 “Can’t create table 表名”
 - 当表名已经存在时失败
- drop table
 - 成功时显示 “Successfully dropped table 表名”
 - 失败时显示 “Can’t drop table 表名”
 - 当表名不存在时失败

要求3：插入新记录

insert into 表名 **values** (值以逗号隔开);

- 必须插入一个记录所有的列
- 不支持null
- 允许记录重复
- 字符串值
 - 只包含ASCII可显式字符
 - 以'单引号括起来

结果：插入新记录

- 语法错误

- 显示: "Syntax error"

- 类型不匹配：列的类型不匹配

- 显示: "Value and column type mismatch"

- 列数不对时

- 显示: "Wrong number of columns"

- 正确：不显示任何信息

要求4: Select语句选择和投影

select 列名,..., 列名

from 表名

where 列名 op 常量;

- select子句

- 可以为*, 那么选择所有列
- 或者是一个或多个列

- where子句

- 可以省略, 这时输出所有行
- 如果存在, 那么在一个表上最多有一个过滤条件

说明：过滤条件

列名 op 常量

- 对于int类型， op可为6种， 常量为整数

- ☐ <
- ☐ <=
- ☐ >
- ☐ >=
- ☐ =
- ☐ !=

- 对于varchar类型， op可为4种， 常量为一个'字符串'

- ☐ = 相等
- ☐ != 不等
- ☐ like 包含字符串为子串
- ☐ not like 不包含字符串为子串

↑
注意：单引号

结果：select语句

- 语法错误

- 显示: "Syntax error"

- 表不存在

- 显示: "Table 表名 doesn't exist"

- 列不存在

- 显示: "Column 列名 doesn't exist"

- 类型和比较操作或者常量不匹配

- 显示: "Predicate 列 比较 常量 error"

- 正确情况见下页

select语句输出结果

- 仿照postgres格式
 - 第一行输出列名
 - 下面每一行是一条结果记录
 - 列之间使用竖线|隔开

列1名|列2名|...|列k名

列1值|列2值|...|列k值

列1值|列2值|...|列k值

列1值|列2值|...|列k值

.....

要求5: Select语句等值连接

select 列名,..., 列名

from 表名1, 表名2

where 连接条件 **and** 过滤条件 **and** 过滤条件;

- **select子句**

- 可以为*, 那么选择所有列
- 或者是一个或多个列

- **where子句**

- 必须有连接条件: 列1=列2
- 过滤条件如果存在, 那么在一个表上最多有一个过滤条件
- 所以where子句可以有1至3个条件

说明：列名的写法

- 简化写法
 - 列名
- 完整写法
 - 表名.列名
- 当简化写法有歧义时，必须采用完整写法，否则报错

结果：select语句

- 语法错误、表不存在、列不存在、类型和比较操作或者常量不匹配：如前面报错
- 列存在两个表，有歧义
 - 显示：“Ambiguous column 列名”
- 连接条件类型不匹配
 - 显示：“Join predicate error”
- 正确情况如前显示结果记录

可以使用一遍算法

- 默认两个表中的小表可以放入内存
- 于是可以采用一遍算法，例如
 - simple hash join

要求6: Select语句分组聚集

select 分组列, 聚集(列名),...

from 表名

where 列名 op 常量

group by 列名;

select 分组列, 聚集(列名),...

from 表名1, 表名2

where 连接条件 **and** 过滤条件 **and** 过滤条件;

group by 列名;

- 在前面规定的两种select语句用法的基础上
- 增加group by子句: 单个列
- select子句可以有分组的列, 和其它列的聚集

说明：聚集的种类

- SUM
 - COUNT
 - AVG
 - MIN
 - MAX
-
- COUNT括号中的列可以是*
 - 其它聚集的括号中必须指明是哪个列
 - AVG的结果四舍五入取整

结果：select语句

- 语法错误、表不存在、列不存在、类型和比较操作或者常量不匹配、列存在两个表中有歧义、连接条件类型不匹配：如前面报错
- 聚集的列不是int类型
 - 显示：“Column 列名 is not int and can't be used in aggregation”
- select子句出现非分组的列，又不是聚集
 - 显示：“Non-group-by column 列名 in select list”
- 正确情况如前显示结果记录
 - 如果包括聚集，那么表头部分显示“聚集(列)”

可以使用一遍算法

- 默认分组的数量比较小，使分组聚集的计算可以一遍实现
- 于是可以采用一遍算法，例如
 - hash based group by

评分标准

- 总分：25分

- 数据表的创建和删除：2分
- 数据记录的插入：5分
- select语句：18分
 - 实现至少一个select：3分
 - select语句选择投影：5分
 - select语句连接：5分
 - select语句分组聚集：5分
- 提交代码，进行抄袭相似性检查
 - 如果发现抄袭，那么0分，并通知教务处
- 我们验收时会给定输入，检查输出是否正确