

数据库系统 | Database Systems

第二课:关系型模型与关系代数

Lecture2: relation model and relational algebra

Lecturer: Harbour

Date: 2021.11.15



Scan CC WeChat to Join the Community 添加CC好友,接受进群邀请 Welcome to follow the GitHub repo

欢迎关注我们的代码仓库

https://github.com/cnosdatabase/cnosdb



计划教学内容



必讲

DB/DBMS

关系模型与关系代数

数据库存储

散列索引 (哈希)

B+树

查询处理

并发控制

如果有兴趣

SQL

查询优化

恢复系统

分布式OLTP/OLAP

关系型模型



1970 Ted Codd

- ▶ 三大特点:
- 1. 以简单的数据结构(关系)存储数据库。
- **2. 通**过高级语言访问数据。
- 3. 物理存储留给实施,对上层应用透明。

Information Retrieval

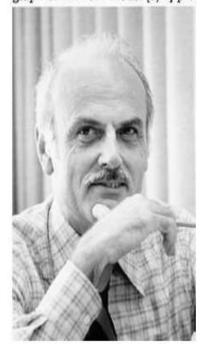
A Relational Model of Data for Large Shared Data Banks

E. F. Codd IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network

The relational view (or model Section 1 appears to be superior in graph or network model [3, 4] pres



数据模型



- 数据模型 (data model) 是用于描述数据库中数据的概念的集合。
- 模式(schema)是使用给定数据模型对特定数据集合的描述。
- Relational 关系型
- Key/value键值
- Graph 图
- Document 文本
- Column-family 列式
- Array/Matrix 数组矩阵

关系模型



结构 (Structure)

▶ 关系(relations)及其内容(contents)的定义。

完整性(Integrity)

▶ 确保数据库的内容满足约束条件。

操作(Manipulation)

▶ 确保数据库的内容满足约束条件。

关系型模型



关系

关系是一个无序集合, **它包含表示**实体的属性的关系。

笛卡尔积D1×D2×...×Dn的有限子集。

一个表及表中的数据。

元组

元组是关系中的一组属性值(也称为域)。

元组:笛卡尔积的每个元素(d1 , d2 , ... , dn)称作一个 n-元组(n-tuple),简称元组(Tuple)。

表中的一行及其数据。

歌手Artist (名字name, 出道时间year, 国家 country)

名字	出道时间	国家
王菲	1985	中国
陈奕迅	1996	中国
周杰伦	2000	中国

主键Primary Key



• **若关系中的某一属性**组的值能够唯一地标识一个元组,则称该属性为关系的主键。

歌手Artist

SEQUENCE (SQL:2003)

(名字name, 出道时间year, 国家country)

AUTO_INCREMENT (MySQL)

编号	名字	出道时间	国家
101	王菲	1985	中国
102	陈奕迅	1996	中国
103	周杰伦	2000	中国

外键 Foreign Keys



• 外键指定一个关系中的属性必须映射到另一个关系中的元组。

歌手专辑 ArtistAlbum(<u>artist_id</u>,<u>ablum_i</u>

Artist_id	Ablum_id
101	11
101	12
102	12
103	13

歌手Artist (<u>id</u>, 名字name, 出道时间year, 国家country)

编号	名字	出道时间	国家
101	王菲	1985	中国
102	陈奕迅	1996	中国
103	周杰伦	2000	中国

专辑Album (<u>id</u>, 名字name, 歌手artist, 年份year)

编号	名字	歌手	年份
11	我愿意	101	1997
12	Stranger Under Skin	??	2011
13	Jay	103	2000

DML数据操纵语言



▶ DML是用于数据库操作,对数据库其中的对象和数据运行访问工作的编程语句。

程序性

查询指定 DBMS 应使用的(高级)策略来查找所需结果。

非程序性

仅指定需要什么数据,而不指定如何找到它。

查询



• 关系模型独立于任何查询语言实现。

for line in file:

record = parse(line)

if "周杰伦" == record[0]:

print int(record[1])

编号	名字	出道时间	国家
101	王菲	1985	中国
102	陈奕迅	1996	中国
103	周杰伦	2000	中国

SELECT year **FROM** artists

WHERE name = "周杰伦"

09 关系代数



- 检索和操作关系中的元组的基本操作。
- 每个操作将一个或多个关系作为其输入并输出一个新关系。

σ Select Sigma

Π Projection Pi

U Union

∩ Intersection

- Difference
- × Product
- ⋈ Join Bow tie

Select



• $\sigma_{predicate}(R)$

 $\sigma_{a_id=`a2}$, (R)

a_id	b_id
a2	102
a2	103

 $\sigma_{a_id='a2'\land\ b_id>102}(R)$

a_id	b_id
a2	103

R(a_id,b_id)

a_id	b_id
a1	101
a2	102
a2	103
a3	104

SELECT * FROM R **WHERE** a_id='a2' AND b_id>102;

Projection



• $\Pi_{A1,A2,...,An}(R)$

$$\prod_{b_id,a_id} (\sigma_{a_id='a2'}(R))$$

b_id	a_id
102	a2
103	a2

R(a_id,b_id)

a_id	b_id
a1	101
a2	102
a2	103
a3	104

SELECT b_id, a_id FROM R **WHERE** a_id = 'a2';

Union



- 生成一个关系, 其中包含仅出现在一个或两个输入关系中的所有元组。
- (R U S)

a_id	b_id
a1	101
a2	102
a3	103
a4	104
a5	105

R(a_id,b_id)

S(a_	_id,	b_i	ld)
------	------	-----	-----

a_id	b_id
a1	101
a2	102
a3	103

a_id	b_id
a3	103
a4	104
a5	105

(SELECT * FROM R)

UNION ALL

(SELECT * FROM S);

Intersection



- 生成仅包含出现在两个输入关系中的元组的关系。
- $(R \cap S)$

a_id	b_id	
a3	103	

a_id	b_id
a1	101
a2	102
a3	103

a_id	b_id
a3	103
a4	104
a5	105

(SELECT * FROM R) INTERSECT (SELECT * FROM S);

Difference



- 生成只包含出现在第一个而不是第二个输入关系中的元组的关系。
- (R-S) (SELECT * FROM R) EXCEPT (SELECT * FROM S);

a_id	b_id
a1	101
a2	102

(S-R) (SELECT * FROM S) EXCEPT (SELECT * FROM R);

a_id	b_id
a4	104
a5	105

R(a_id,b_id)	$S(a_id,b_id)$
11(4_14,0_14)	$\mathcal{S}(\mathbf{a}_{-1}\mathbf{a}, \mathbf{b}_{-1}\mathbf{a})$

a_id	b_id	a_id	b_id
a1	101	a3	103
a2	102	a4	104
a3	103	a5	105

Product



- 从输入关系生成包含所有可能的元组组合的关系。
- (R × S)

R.a_id	R.b_id	S.a_id	S.b_id
a1	101	a3	103
a1	101	a4	104
a1	101	a5	105
a2	102	a3	103
a2	102	a4	104
a2	102	a5	105
a3	103	a3	103
a3	103	a4	104
a3	103	a5	105

$$R(a_id,b_id)$$

S(a_id,b_id)

a_id	b_id	a_id	b_id
a1	101	a3	103
a2	102	a4	104
a3	103	a5	105

SELECT * FROM R CROSS JOIN S;

SELECT * FROM R, S;

Join



- 生成一个包含所有元组的关系,这些元组是两个元组(每个输入关系一个)的组合,具有一个或多个属性的公共值。
- $(R \bowtie S)$

a_id	b_id	
a3	103	

 $R(a_id,b_id)$ $S(a_id,b_id)$

a_id	b_id	a_id	b_id
a1	101	a3	103
a2	102	a4	104
a3	103	a5	105

SELECT * FROM R NATURAL JOIN S;



Q&A



Scan CC WeChat to Join the Community 添加CC好友,接受进群邀请

Welcome to follow the GitHub repo 欢迎关注我们的代码仓库

https://github.com/cnosdatabase/cnosdb



