

Relational model

重要程度: ★★★★
难易程度: ★★★★☆

- Relational Model 是数据库管理的一个模型
- Consist of two parts (Heading & Body)
 - Relation heading (a.k.a. relational schema)
 - RelationName(attribute1, attribute2, attribute3...) 表格列的名称
 - STUDENT(stud_id, stud_name, stud_address...)
 - EACH attribute(stud_id) corresponds to one underlying domain
 - $\text{dom}(\text{stud_id}) \rightarrow \text{student_id}$
 - Relation Body (a.k.a. relation instance) - 表格里的具体数据
 - Relation degree: 表格中attribute的数量
 - Relational cardinality: number of tuples 表格中数据的行数

custno	custname	custadd	credlimit
SMI13	SMITH	Wide Rd, Clayton, 3168	2000
JON44	JONES	Narrow St, Clayton, 3168	10000
BRO23	BROWN	Here Rd, Clayton, 3168	10000

Note: 注意区别 relation(关系) 和 relational schema(关系模式)

- relation 是动态的，随着时间可能会发生生变化
(添加删除信息等)
- relational schema是静态的，关系模式是不会发生生改变的

Relational Model -- Property

重要程度: ★★
难易程度: ★★

- **NO duplicate tuples** 没有完全一样的两行数据
- **Tuples have no order within a relation** 表格中的行是无序的
- **Attributes have no order** 表格中的列是无序的
- **Tuple values are atomic (multi-valued attributes are NOT allowed in the table)**

custno	custname	custadd	credlimit
SMI13	SMITH	Wide Rd, Clayton, 3168	2000
JON44	JONES	Narrow St, Clayton, 3168	10000
BRO23	BROWN	Here Rd, Clayton, 3168	10000

Relational Model – KEY

重要程度: ★★
难易程度: ★★

❖ Four types: Superkey, Candidate Key, Primary Key, Foreign Key

- **Superkey:** 能标识唯一tuple的属性的集合
 - $\{stud_id, stud_name\} \rightarrow$ 唯一一行信息, 所以 $\{stud_id, stud_name\}$ 是superkey
 - $\{stud_id\} \rightarrow$ 唯一一行信息, 所以 $\{stud_id\}$ 是superkey
 $\{stud_name\} \rightarrow$ 可能多于一行信息, 所以 $\{stud_name\}$ 不是superkey
- **Candidate key:** 不含多余attribute的superkey
 - $\{stud_id, stud_name\}$ 是superkey, 但是输入 $stud_name$ 自身实际上并不能保证只能得到唯一一行信息, 所以 $stud_name$ 属于 superkey中多余的attribute, $stud_id$ 是unique的, 可以保证只得到唯一一行信息, 所以 $stud_id$ 是 candidate key
 - 同理, $stud_oshc$ is candidate key (Every student has a unique $stud_oshc$)

stud_id	stud_name	stud_dob	stud_oshc
1110	John	02/01/1996	mon2000
1111	Amy	05/05/1998	mon2001
1112	Ben	02/01/1996	mon2002
1113	John	02/11/1996	mon2003

Relational Model – KEY

重要程度: ★★
难易程度: ★★

❖ Four types: Superkey, Candidate Key, Primary Key, Foreign Key

- **Primary key:** 从candidate key中选一个，每个表中有且仅有一个pk (但有可能由多个attribute组成 --> composite pk)
 - better choose the one with numeric data type
 - Natural pk & Surrogate pk (logical model) - Natural: case 中本身带有的
 - surrogate: 根据自己的需求添加的 (conceptual model 中不允许出现)
- **Foreign key :** 某个表的fk是另一表的pk (两表有关联的前提下) 或为NULL
 - 目的:描述两个表的关系

stud_id	stud_name	stud_dob	stud_oshc
1110	John	02/01/1996	mon2000
1111	Amy	05/05/1998	mon2001
1112	Ben	02/01/1996	mon2002
1113	John	02/11/1996	mon2003

Relational Model – Data Integrity

重要程度: ★★
难易程度: ★★

- **Entity Integrity** 实体完整性
 - Primary key MUST NOT be NULL - Primary key MUST be UNIQUE
- **Referential Integrity** 参照完整性
 - The values of FK must either MATCH a value of full PK in the related relation or to be NULL
 - FK 必须来自于某个有关联的表的PK, 不然就为空
- **Column/Domain Integrity**
 - All values in a given column must come from the same domain (the same data type and range) –
 - 同一列的数据需要为同一数据类型以及范围

Relational Algebra -- Overview

重要程度: ★★★★
难易程度: ★★★★

❖ Select, Project, Join

- SELECT 选择运算

- $\sigma_{expr}(Rel)$ <- Standard Notation
- **Select rows** (select one or more rows from [TABLE] where [CONDITION])
- Our notation: Sel [expr] (Rel)
- e.g. Sel[student_mark > 70] (MARK) : select rows from table MARK where student_mark > 70

- PROJECT 投影运算

- $\pi_{A,B,C}(Rel)$ <- Standard Notation
- **Select columns**
- Our notation: Proj [A,B,C] (Rel)
- e.g. Proj [stud_id] (STUDENT): select whole column named stud_id

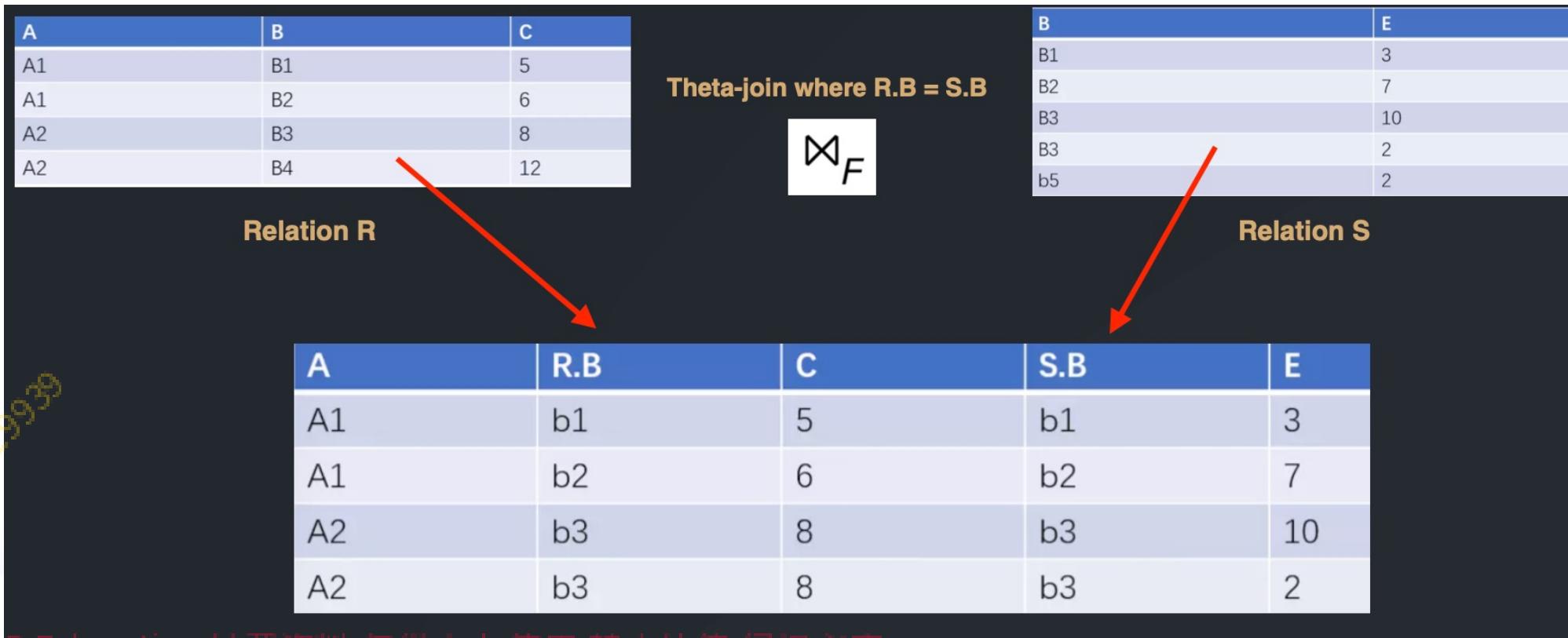
- JOIN 连接运算

- Four methods: theta-join, equi-join, natural join & outer join(full outer, left outer, right outer)

Relational Algebra -- Join

重要程度: ★★★★☆
难易程度: ★★★★☆

- Theta-Join 等值连接
 - 选取属性间满足某条件的tuples进行连接



Relational Algebra -- Join

重要程度: ★★★★☆
难易程度: ★★★★☆

- Natural Join 等值连接

- 选取带有相同意义属性间相同值的tuples进行连接(然后删除其中一个重复列)
- 无法合并的行会被删除



Relational Algebra -- Join

重要程度: ★★★★☆
难易程度: ★★★★☆

- FULL Outer Join

- 根据选定的列进行合并，无论两个表的那一列是否存在相同的值

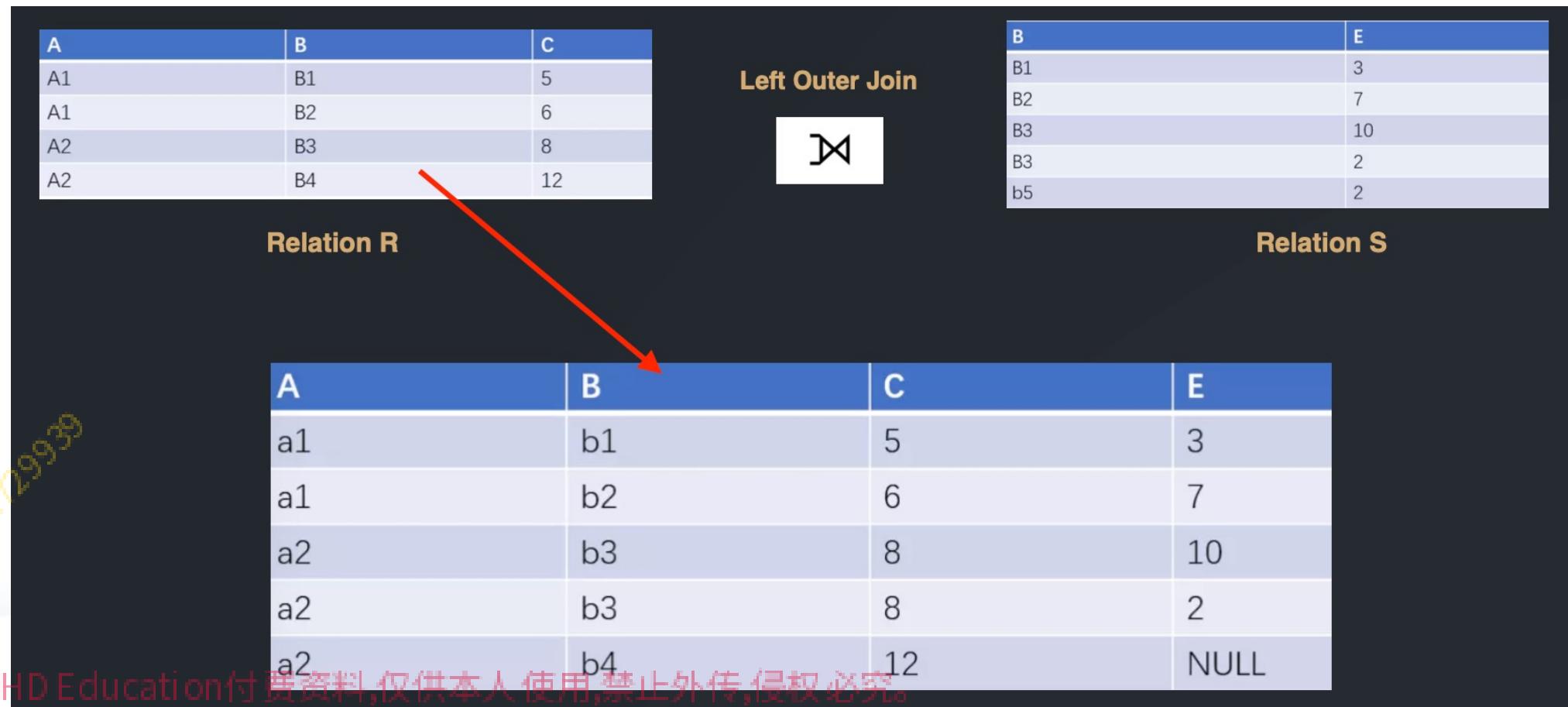


Relational Algebra -- Join

重要程度: ★★★★☆
难易程度: ★★★★☆

- LEFT Outer Join

- 保留左边选定列的所有数据，与右边进行连接



Relational Algebra -- Join

重要程度: ★★★★☆
难易程度: ★★★★☆

- RIGHT Outer Join

- 保留右边选定列的所有数据，与左边进行连接

A	B	C
A1	B1	5
A1	B2	6
A2	B3	8
A2	B4	12

Relation R

Right Outer Join



B	E
B1	3
B2	7
B3	10
B3	2
b5	2

Relation S

A	B	C	E
A1	b1	5	3
A1	b2	6	7
A2	b3	8	10
A2	b3	8	2
NULL	b5	NULL	2

HD@千里眼412729933

例题讲解

重要程度: ★★★★
难易程度: ★★★

HOTEL (HOTEL_NO, HOTEL_NAME, HOTEL_CITY)
ROOM (ROOM_NO, HOTEL_NO, ROOM_TYPE, ROOM_PRICE)
BOOKING (HOTEL_NO, GUEST_NO, BDATE_FROM, BDATE_TO, ROOM_NO)
GUEST (GUEST_NO, GUEST_NAME, GUEST_ADDRESS)

List the names and cities of all hotels

List all single rooms with a price below \$50

List the price and type of all rooms at the Grosvenor Hotel

例题讲解

重要程度: ★★★★☆
难易程度: ★★★★☆

HOTEL (HOTEL_NO, HOTEL_NAME, HOTEL_CITY)
ROOM (ROOM_NO, HOTEL_NO, ROOM_TYPE, ROOM_PRICE)
BOOKING (HOTEL_NO, GUEST_NO, BDATE_FROM, BDATE_TO, ROOM_NO)
GUEST (GUEST_NO, GUEST_NAME, GUEST_ADDRESS)

List the names and cities of all hotels

Answer1 = $\pi_{\text{hotel_name}, \text{hotel_city}} \text{HOTEL}$

List all single rooms with a price below \$50

Answer2 = $\sigma_{\text{room_type}=\text{'single'} \text{ and } \text{room_price} < 50} \text{ROOM}$

List the price and type of all rooms at the Grosvenor Hotel

PSuiteNo = $\pi_{\text{hotel_no}} (\sigma_{\text{room_type} = \text{'presidential suite'}} \text{ROOM})$

Answer4 = $\pi_{\text{hotel_name}} (\text{PSuiteNo} \bowtie (\pi_{\text{hotel_no}, \text{hotel_name}} \text{HOTEL}))$

重难点总结

HD Education付费资料,仅供本人使用,禁止外传,侵权必究。

重难点总结

1、Relational Model Property

- NO duplicate tuples
- Tuples have no order within a relation
- Attributes have no order
- Tuple values are atomic

2、Keys

- **Superkey**: 能标识唯一tuple的属性的集合
- **Candidate key**: 不含多余attribute的superkey
- **Primary key**: 从candidate key中选一个，每个表中有且仅有一个pk (但有可能由多个attribute组成 --> composite pk)
- **Foreign key** : 某个表的fk是另一表的pk (两表有关联的前提下) 或为NULL

HD@易趣录41272993

重难点总结

1、Relational Algebra

- **SELECT** 选择运算
- 选行的时候用（筛选条件）
- **PROJECT** 投影运算
- 选列的时候用
- **JOIN** 连接运算
同时需要多个表的时候用

$$\sigma_{expr}(Rel)$$

$$\pi_{A,B,C}(Rel)$$