Relational Model and Relational Algebra

Dr Na Yao

Objectives

- Terminology of relational model.
- How tables are used to represent data.
- Properties of database relations.
- How to identify candidate, primary, alternate, and foreign keys.
- Meaning of entity integrity and referential integrity.
- How to form queries in relational algebra.

Relational model

- E. F. Codd proposed relational data model in 1970.
- In the relational model, all data is logically structured within **relation**s (tables).
- Each **relation** is made up of **attributes** (columns) of data.
- Each **tuple** (row) contains one value per attribute.

Terminology

- Relation: A relation is a table with columns and rows.
- Attribute: An attribute is a named column of a relation.
- Domain: the set of allowable values for one or more attributes.

Terminology

- Tuple: A tuple is row of a relation.
- Degree: the number of attributes in a relation.
- Cardinality: the number of tuples in a relation.
- Relational database: A collection of normalized relations with distinct relation names.

Attributes

Branch

Relation

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Cardinality

Degree

Primary key

Foreign key

Staff

Relation

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Examples of Attribute Domains

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

Creating relations (tables) in SQL

```
Create Table Branch(branchNo, street, city,
  postcode)

Create Table Branch(branchNo integer, street
  string, city string, postcode char(6))
```

Properties of relations

- Relation name is distinct from all other relation names in relational schema.
- Each cell of relation contains exactly one atomic (single) value. (*First normal form*)
- Each attribute has a distinct name.
- Values of an attribute are all from the same domain.
- Each tuple is distinct; there are no duplicate tuples.
- Order of attributes has no significance.
- Order of tuples has no significance, theoretically.

Relational keys

- Candidate Key
  - A set of attributes that uniquely identifies a tuple within a relation.
  - Uniqueness : In each tuple, candidate key uniquely identify that tuple.
  - Irreducibility: No proper subset of the candidate key has the uniqueness property.
- Primary Key
  - Candidate key selected to identify tuples uniquely within relation.
- Foreign Key
  - Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

Representing Relational Schema

- Relation name (attribute 1, attribute 2, … attribute n)
  - Branch (branchNo, street, city, postcode)
  - Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

- Exercise: do the relational schema for all the relations in *DreamHome* database.

Integrity Constraints

- Null
  - Represents value for an attribute that is currently unknown or not applicable for tuple.
  - Deals with incomplete or exceptional data.
  - Represents the absence of a value and is not the same as zero or spaces, which are values.

Integrity Constraints

- Entity Integrity

  In a base relation, no attribute of a primary key can be null.
- Referential Integrity

  If foreign key exists in a relation, either foreign key value must match a candidate key value of some tuple in its home relation or foreign key value must be null.
- General Constraints

Glossary

- A “crowd funded” glossary – every student can add entry and comment on an entry.

Students with good contribution will be rewarded ☺

Relational Algebra

- Relational algebra is formal language associated with the relational model.
- Relational algebra operations work on one or more relations to define another relation without changing the original relations.
- Allows expressions to be nested, just as in arithmetic. This property is called *closure*.

“Thames Barrier Closure” by Chris Wheal, flickr

Relational Algebra

- Basic operations:
  - Selection
  - Projection
  - Cartesian product
  - Union
  - Set difference
- Additional operations:
  - Join
  - Intersection
  - Division

Selection

- $\sigma_{condition}$ (R)

  Works on a single relation R and defines a relation that contains only those tuples (rows) of R that satisfy the specified condition (predicate).

Example

- List all staff with a salary greater than £10,000.

  $\sigma_{salary > 10000}$ (Staff)

- List all female staff with a salary greater than £10,000

  $\sigma_{salary > 10000 \wedge sex = \text{“F”}}$ (Staff)

- List all branches in London.

Selection

Projection

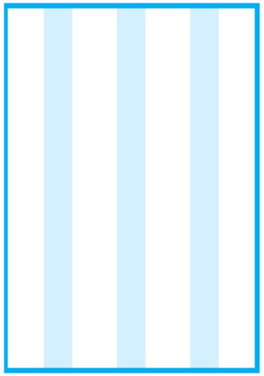
- $\Pi_{\text{col1, . . . , coln}}$(R)

  Works on a single relation R and defines a relation that contains a vertical subset of R, extracting the values of specified attributes and ***eliminating duplicates***.

Example

- Produce a list of salaries for all staff, showing only staffNo, fName, lName, and salary details.

$$\Pi_{\text{staffNo, fName, lName, salary}}(\text{Staff})$$



Projection

Combine selection and projection

Example

- List staff number, position and salary of any staff whose salary is greater than £20,000.

Answer 1: $\sigma_{\text{salary} > 20000}(\Pi_{\text{staffNo, position, salary}}(\text{Staff}))$

Answer 2: $\Pi_{\text{staffNo, position, salary}}(\sigma_{\text{salary} > 20000}(\text{Staff}))$

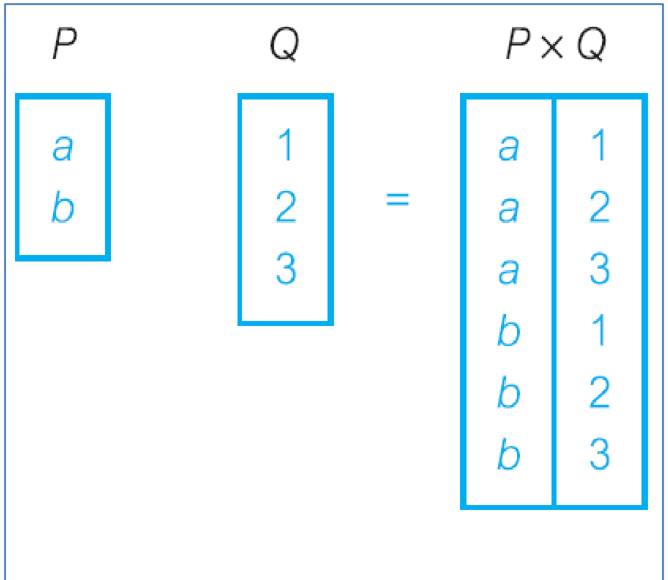
- List the branchNo of all branches in London.

Cartesian product

- R X S

Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.

R S R X S



Cartesian product

Cartesian product

- R X S

  Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.

  R S R X S

  Example

- List the names and comments of all clients who have viewed a property for rent.

  $(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \text{ X } (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$

  $\sigma_{\text{Client.clientNo = Viewing.clientNo}}(((\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \text{ X } (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing})))$

Join

- Derivative of Cartesian product, equivalent to performing a Selection operation, using the join condition as the selection formula, over the Cartesian product of the two operand relations.
- Natural Join
- Theta join

Natural Join

- R ⋈ S
  - The result of the natural join is the set of all combinations of tuples in *R* and *S* that are equal on their common attribute names.

T

A	*B*
a	1
b	2

U

B	*C*
1	*x*
1	*y*
3	*z*

T ⋈ *U*

A	*B*	*C*
a	1	*x*
a	1	*y*

Natural join

Natural Join

- R $\bowtie$ S
  - The result of the natural join is the set of all combinations of tuples in *R* and *S* that are equal on their common attribute names.

Example

- List the names and comments of all clients who have viewed a property for rent.

$(\Pi_{\text{clientNo, fName, lName}}(\text{Client}) \bowtie$

$(\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$

$\Pi_{\text{clientNo, fName, lName, propertyNo, comment}}\ ((\text{Client}) \bowtie\ (\text{Viewing}))$

Theta join

- $R \bowtie_F S$
  - Defines a relation that contains tuples satisfying the condition F from the Cartesian product of R and S.
- Can rewrite Theta join using basic Selection and Cartesian product operations.

$$R \bowtie_F S = \sigma_F (R \times S)$$

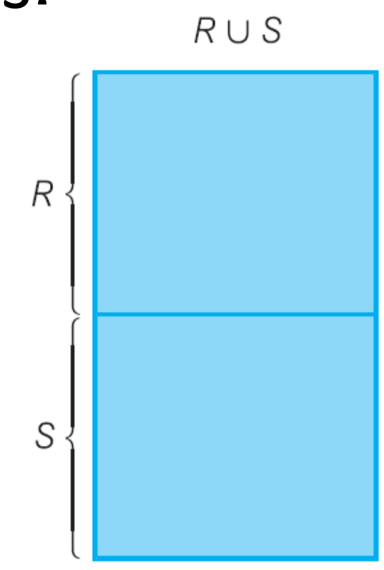
Union

- $R \cup S$
  - Union of two relations R and S defines a relation that contains all the tuples of R, or S, or both R and S, duplicate tuples being eliminated.
  - R and S must be union-compatible.
- If R and S have **I** and **J** tuples, respectively, union is obtained by concatenating them into one relation with a maximum of (***I*** + ***J***) tuples.

Example

- List all cities where there is either a branch office or a property for rent.

$$\Pi_{city}(\text{Branch}) \cup \Pi_{city}(\text{PropertyForRent})$$



Union

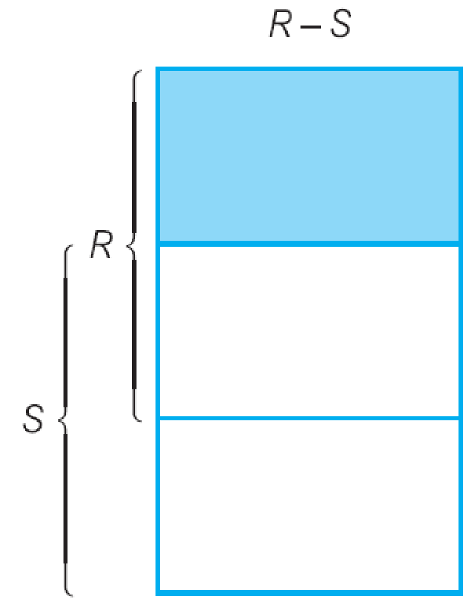
Set difference

- R – S
  - Defines a relation consisting of the tuples that are in relation R, but not in S.
  - R and S must be union-compatible.

example

- List all cities where there is a branch office but no properties for rent.

$$\Pi_{city}(Branch) - \Pi_{city}(PropertyForRent)$$



Set difference

Intersection

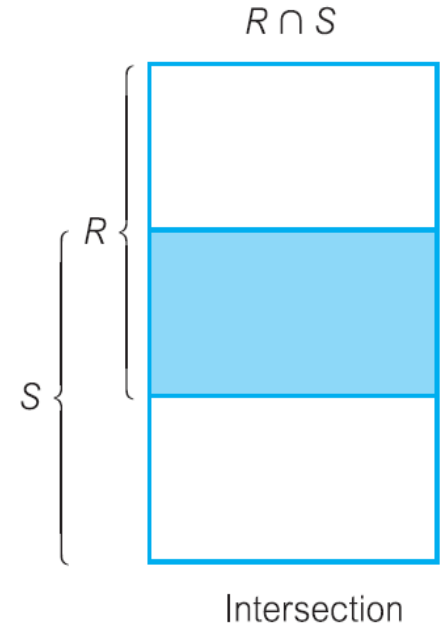
- $R \cap S$
  - Defines a relation consisting of the set of all tuples that are in both R and S.
  - R and S must be union-compatible.
- Expressed using basic operations:

  $R \cap S = R - (R - S)$

Example

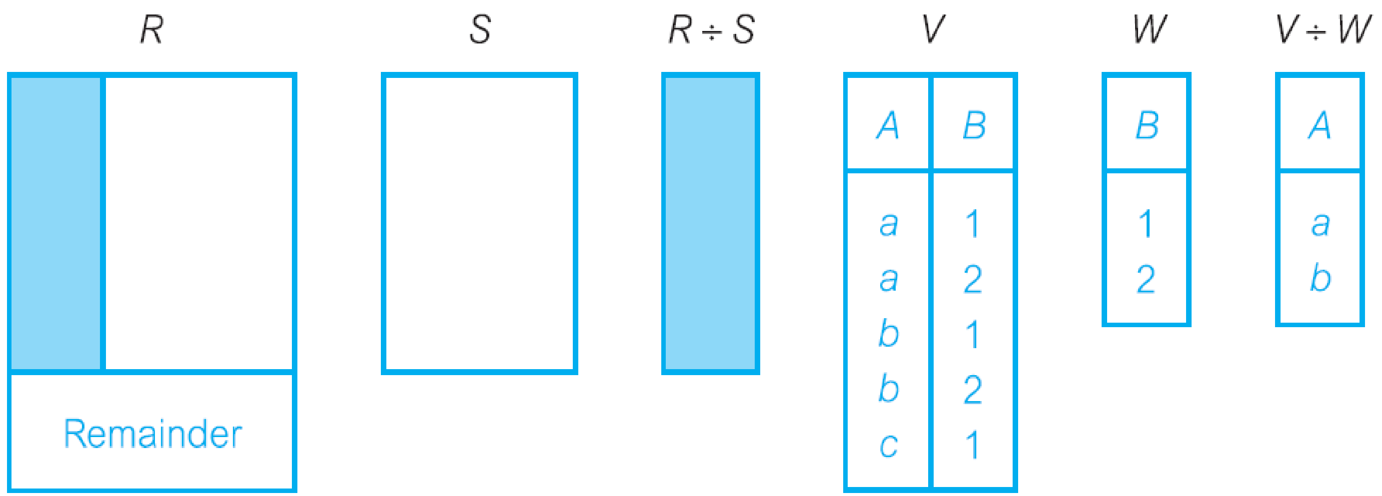
- List all cities where there is both a branch office and at least one property for rent.

  $\Pi_{city}(\text{Branch}) \cap \Pi_{city}(\text{PropertyForRent})$



Division

- Assume relation R is defined over the attribute set A and relation S is defined over the attribute set B such that B ⊆ A (B is a subset of A). Let C = A – B, that is, C is the set of attributes of R that are not attributes of S.
- R ÷ S defines a relation over the attributes C that consists of set of tuples from R that match combination of *every* tuple in S.



Division (shaded area)

Example of division

Example - Division

- Identify all clients who have viewed all properties with three rooms.

$$(\Pi_{\text{clientNo, propertyNo}}(\text{Viewing})) \div$$
$$(\Pi_{\text{propertyNo}}(\sigma_{\text{rooms = 3}}\ (\text{PropertyForRent})))$$

$\Pi_{\text{clientNo,propertyNo}}(\text{Viewing})$

clientNo	propertyNo
CR56	PA14
CR76	PG4
CR56	PG4
CR62	PA14
CR56	PG36

$\Pi_{\text{propertyNo}}(\sigma_{\text{rooms=3}}(\text{PropertyForRent}))$

propertyNo
PG4
PG36

RESULT

clientNo
CR56

Relational Algebra Exercise

Given the following relational schema:

```
Hotel(hotelNo, hotelName, city)
Room(roomNo, hotelNo, type, price)
Booking(hotelNo, guestNo, dateFrom, dateTo, roomNo)
Guest(guestNo, guestName, guestAddress)
```

Formulate the following queries in relational algebra:

1. Give the hotel number of those hotels with a room price greater than £50.
2. List the price and types of all rooms in Grosvenor Hotel.
3. Produce a relation containing all rooms at all hotels.
4. Give all hotel names with a room price above £50.
5. List the guest number of all guests currently staying at the Grosvenor Hotel. (Hint: use current_date() for today's date)
6. List guest number of all guests who have booked all hotels in Beijing.
7. List names of all guests who have booked all hotels in London.