ECS 165A

Database Systems

Discussion Session 3

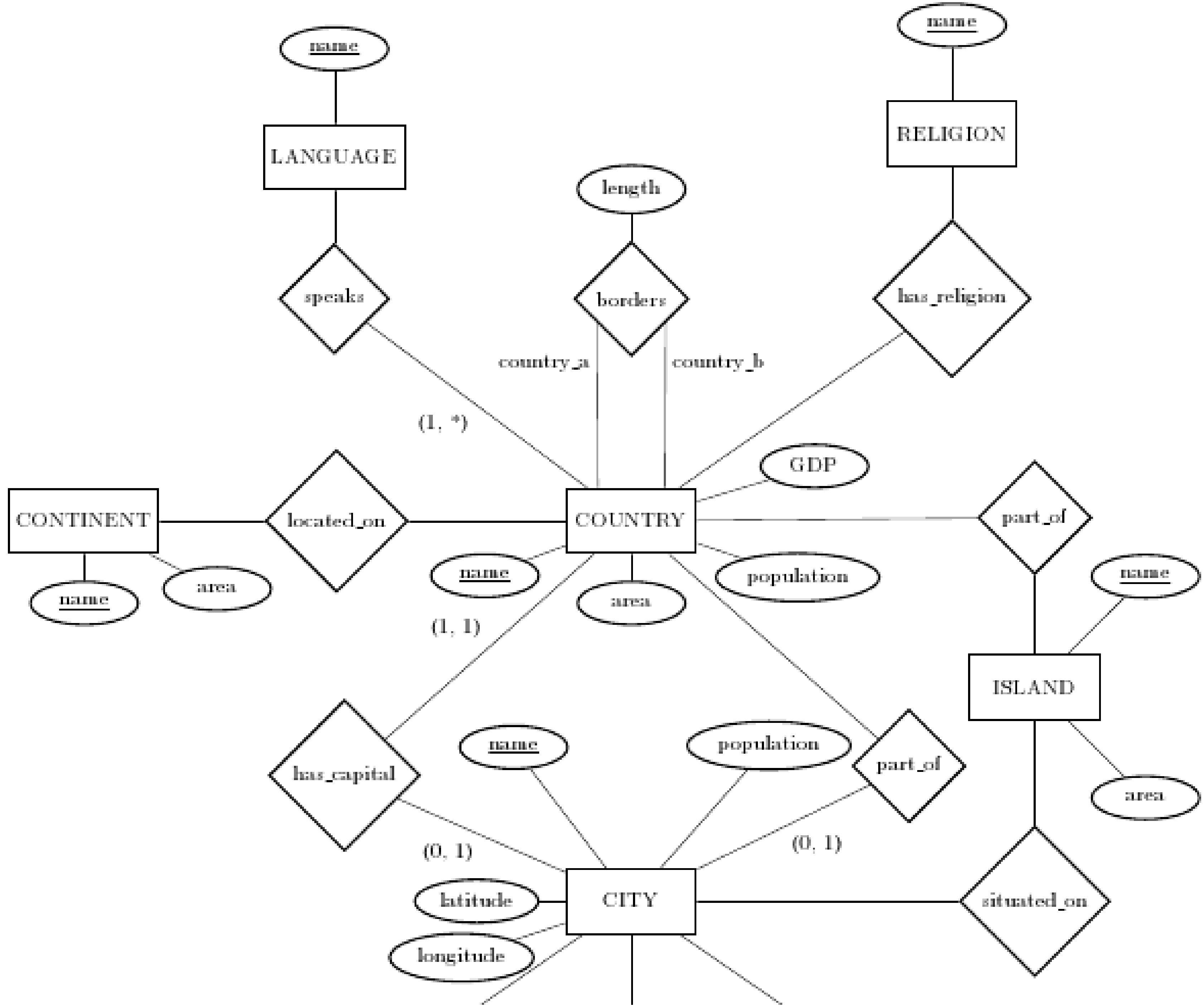
The Relational Algebra (Chap. 6)

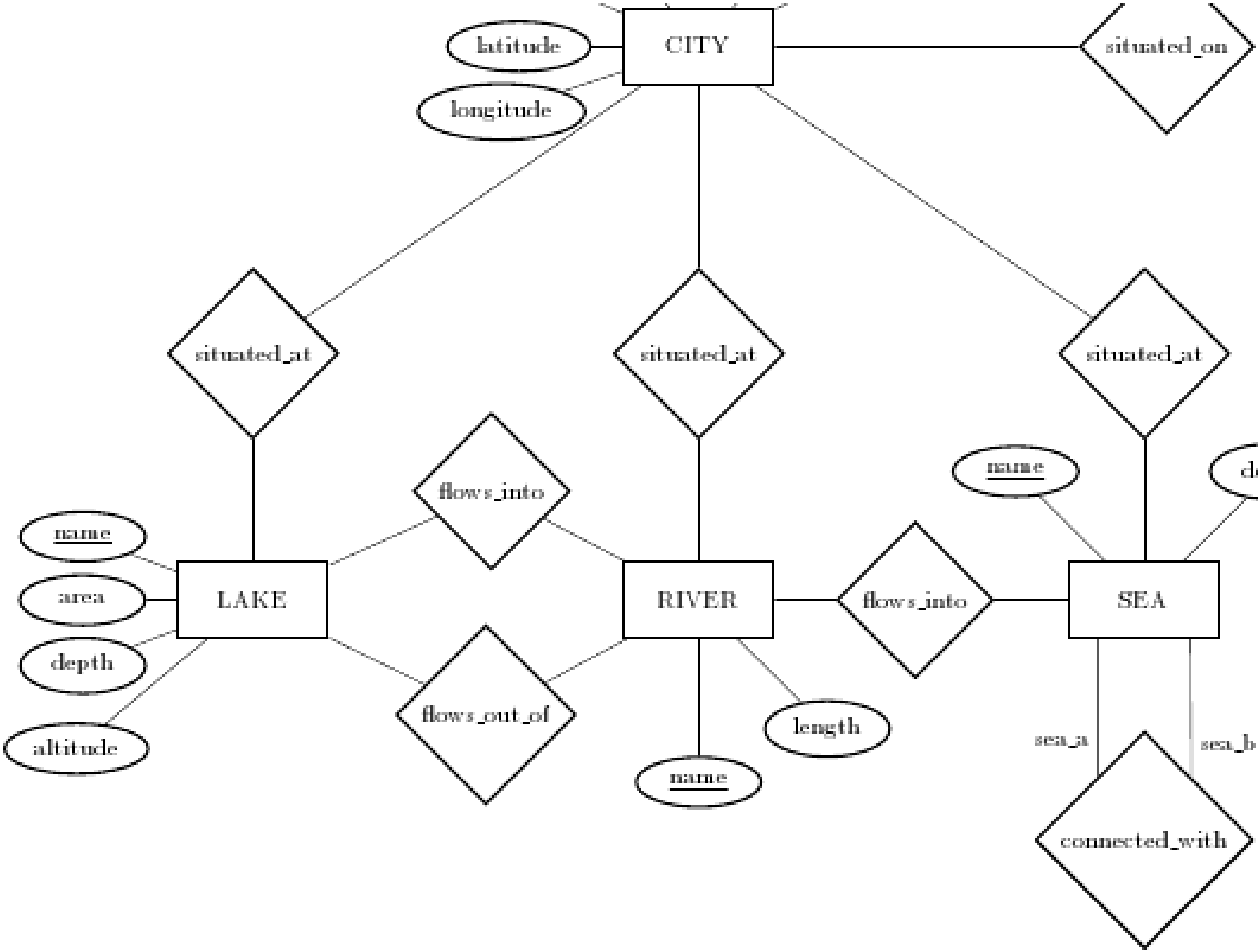
Outline

- Review Homework1 Solutions
- Review the Relational Algebra
- Examples

ER Modeling

- A country has a number of attributes, including a name, (land) area, population, and GDP. A country has also a capital city, which itself has a name, population, and geographical coordinates (latitude, longitude). With each country there are also a number of spoken languages and predominant religions associated.
- Countries are located on continents, the latter having a name and area. Any two countries may have a border between them of a particular length.
- A city can be part of country, and can also be situated on an island, or at a lake, river, or sea. A lake has a name, area, maximal depth, and altitude; a river has a name, length, and can flow in or out of a lake, or into a sea; a sea has a name, (maximal) depth, and may be connected with another sea. An island has a name, area, and be


name
LANGUAGE
speaks
(1, *)
length
borders
country_a
country_b
name
RELIGION
has_religion
GDP
CONTINENT
located_on
COUNTRY
part_of
name
area
name
area
population
name
(1, 1)
ISLAND
name
population
has_capital
part_of
area
(0, 1)
(0, 1)
latitude
CITY
situated_on
longitude


CITY
latitude
longitude
situated_on
situated_at
situated_at
situated_at
flows_into
name
area
depth
altitude
LAKE
RIVER
flows_into
SEA
name
depth
flows_out_of
length
name
sea_a
sea_b
connected_with

Constraints that cannot be expressed in ER Model

- The population of a country should be greater than or equal to the sum of the populations of the cities in that country.

- The area of a country should be less than or equal to the land it is located on (continent(s) and island(s)).

- The capital city of a country is part of that same country.

And, …

Relations

- COUNTRY(name, area, population, GDP, capital_city_name → CITY)
- CITY(name, population, latitude, longitude, country_name → COUNTRY)
- country_speaks_language(cname → COUNTRY, lname → LANGUAGE)
- country_has_religion(cname → COUNTRY, rname → RELIGION)
- LANGUAGE(name)
- RELIGION(name)
- country_locatated_on_continent(country_name → COUNTRY, continent_name → CONTINENT)
- CONTINENT(name, area)
- country_borders_country(cname_a → COUNTRY, cname_b → COUNTRY, length)

Relations (cont'd.)

- city_situated_on_island(cname → CITY, iname → ISLAND)
- city_situated_at_lake(cname → CITY, lname → LAKE)
- city_situated_at_river(cname → CITY, rname → RIVER)
- city_situated_at_sea(cname → CITY, sname → SEA)
- LAKE(name, area, depth, altitude)
- RIVER(name, length)
- river_flows_into_lake(rname → RIVER, lname → LAKE)
- river_flows_out_of_lake(rname → RIVER, lname → LAKE)
- river_flows_into_sea(rname → RIVER, sname → SEA)
- SEA(name, depth)
- sea_connected_with_sea(sea_name_a → SEA, sea_name_b → SEA)
- ISLAND(name, area)
- island_part_of_country(iname → ISLAND, cname → COUNTRY)

Relations (cont'd.)

- The relationships country has capital city and city part of country do not need tables, since
  - we assumed that a country can have only one capital (which is actually not true in the real world) and that a city is a part of only one country.
- Therefore, we can just place foreign keys in the COUNTRY and CITY tables, respectively, as shown in the above relational schema.

Changing Attributes

- A country’s population and GDP.
- A city’s population.
- A lake’s area and depth.
- A river’s length.
- A sea’s depth.
- , ….

Changing Attributes Example: GDP

- Remove the GDP attribute from Country. Add a TIME entity with a primary key of t. Add a relationship between TIME and Country called country gdp at time with an attribute named GDP.

- The relational schema changes and additions would be:

- COUNTRY(name, area, population, capital_city_name → CITY)
- TIME(t)
- country_gdp_at_time(cname → COUNTRY, t → TIME, GDP)

Relation Instance

- The current values (*relation instance*) of a relation are specified by a table.
- An element *t* of *R* is a *tuple*, represented by a *row* in a table

attributes
(or columns)

customer_name	*customer_street*	customer_city
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

tuples
(or rows)

customer

What is “algebra”

- Mathematical model consisting of:
  - *Operands ---* Variables or values;
  - *Operators ---* Symbols denoting procedures that construct new values from a given values
- **Relational Algebra** is algebra whose operands are relations and operators are designed to do the most commons things that we need to do with relations.

Unary Relational Operations: SELECT

- The SELECT Operation
  - Subset of the tuples from a relation that satisfies a selection condition:

$$\sigma_{<\text{selection condition}>}(R)$$

Unary Relational Operations: SELECT (cont'd.)

- Example:

$$\sigma_{(\text{Dno}=4 \textbf{ AND } \text{Salary}>25000) \textbf{ OR } (\text{Dno}=5 \textbf{ AND } \text{Salary}>30000)}(\text{EMPLOYEE})$$

- <selection condition> applied independently to each individual tuple *t* in *R*
  - If condition evaluates to TRUE, tuple selected
- Boolean conditions **AND**, **OR**, and **NOT**

SELECT Operation: Example

Relation *R*

A	*B*	*C*	*D*
α	α	*1*	*7*
α	β	*5*	*7*
β	β	*12*	*3*
β	β	*23*	*10*

$\sigma_{A=B \wedge D > 5}(R)$

A	*B*	*C*	*D*
α	α	*1*	*7*
β	β	*23*	*10*

The PROJECT Operation

- Selects columns from table and discards the other columns:
  $\pi_{\text{<attribute list>}}(R)$
- **Degree**
  - Number of attributes in <attribute list>
- **Duplicate elimination**
  - Result of PROJECT operation is a set of distinct tuples

The PROJECT Operation (cont'd.)

- account (branch-name, account-number, balance)

- E.g. to eliminate the *branch-name* attribute of *account*

$$\Pi_{\textit{account-number, balance}}\ (\textit{account})$$

- If relation Account contains 50 tuples, how many tuples are in $\Pi_{\textit{account-number, balance}}$ $(\textit{account})$?

- If relation Account contains 50 tuples, how many tuples are in $\Pi_{\textit{balance}}$ $(\textit{account})$?

Sequences of Operations

- **In-line** expression:

$$\pi_{\text{Fname, Lname, Salary}}(\sigma_{\text{Dno}=5}(\text{EMPLOYEE}))$$

- Sequence of operations:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEE})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Salary}}(\text{DEP5\_EMPS})$$

Relational Algebra Operations from Set Theory

- **UNION**, **INTERSECTION**, and **MINUS**
  - Merge the elements of two sets in various ways
  - Binary operations
  - Relations must have the same type of tuples
- UNION
  - $R \cup S$
  - Includes all tuples that are either in R or in S or in both R and S
  - Duplicate tuples eliminated

UNION Operation: Example

- Relations *R*,

A	*B*
α	*1*
α	*2*
β	*1*

R

A	*B*
α	*2*
β	*3*

S

R $\cup$ S:

A	*B*
α	*1*
α	*2*
β	*1*
β	*3*

Relational Algebra Operations from Set Theory (cont'd.)

- SET DIFFERENCE (or MINUS)
  - $R - S$
  - Includes all tuples that are in R but not in S

Set Difference Operation – Example

- Relations *R, S*:

A	B
α	1
α	2
β	1

R

A	B
α	2
β	3

S

$R - S$:

A	B
α	1
β	1

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

- **CARTESIAN PRODUCT**
  - **CROSS PRODUCT** or **CROSS JOIN**
  - Denoted by $\times$
  - Binary set operation

CARTESIAN-PRODUCT: Example

Relations *R, S*:

A	*B*
α	1
β	2

R

C	*D*	*E*
α	10	a
β	10	a
β	20	b
γ	10	b

S

R x *S*:

A	*B*	*C*	*D*	*E*
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

RENAME Operation

- Allows us to name, and therefore to refer to, the results of relational-algebra expressions.
- Allows us to refer to a relation by more than one name.

Example:

$$\rho_X(E)$$

returns the expression *E* under the name *X.*

If a relational-algebra expression *E* has arity *n*, then

$$\rho_{(A1,\ A2,\ \ldots,\ An)}(E)$$

returns the result of expression *E* under the name *X*, and with the attributes renamed to *A1, A2, …., An*.

INTERSECTION Operation

- Notation: $R \cap S$
- Defined as:
- $R \cap S = \{ t \mid t \in R \textbf{ and } t \in S \}$
- Assume:
  - R, S have the *same arity*
  - attributes of R and S are compatible
- Note:
- $R \cap S = R - (R - S)$
- $R \cap S = ((R \cup S) - (R - S))-(S-R)$

INTERSECTION Operation: Example

- Relation R, S

A	B
α	1
α	2
β	1

R

A	B
α	2
β	3

S

- $R \cap S$

A	B
α	2

Binary Relational Operations: JOIN and DIVISION

- The **JOIN** Operation
  - Denoted b $\bowtie$
  - Combine related tuples from two relations into single “longer” tuples
  - General join condition of the form <condition> **AND** <condition> **AND**...**AND** <condition>
  - $$\text{DEPT\_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr\_ssn=Ssn}} \text{EMPLOYEE}$$
    $$\text{RESULT} \leftarrow \pi_{\text{Dname, Lname, Fname}}(\text{DEPT\_MGR})$$

NATURAL JOIN: Example

- Relations R, S:

R

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

S

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

$R \bowtie S$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Natural Join: Example

<u>sid</u>	<u>bid</u>	<u>day</u>
22	101	10/10/96
58	103	11/12/96

R1

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

R1 $\bowtie$ S1 =

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

Binary Relational Operations: JOIN (cont'd.)

- **THETA** JOIN
  - Each <condition> of the form A_i θ B_j
  - A_i is an attribute of R
  - B_j is an attribute of S
  - A_i and B_j have the same domain
  - θ (theta) is one of the comparison operators:
    - $\{=, <, \leq, >, \geq, \neq\}$

“THETA JOIN: Example

sid	bid	day
22	101	10/10/96
58	103	11/12/96

R1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

$$S1 \bowtie_{S1.sid < R1.sid} R1 =$$

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	58	103	11/12/96

A Complete Set of Relational Algebra Operations

- Set of relational algebra operations $\{\sigma, \pi, \cup, \rho, -, \times\}$ is a **complete set**
  - Any relational algebra operation can be expressed as a sequence of operations from this set

The DIVISION Operation

- Denoted by ÷
- Example: retrieve the names of employees who work on all the projects that ‘John Smith’ works on
- Apply to relations $R(Z) \div S(X)$
  - Attributes of R are a subset of the attributes of S

Division: Example

Relations *R*, *S*:

A	*B*
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
$\in$	6
$\in$	1
β	2

R

B
1
2

S

$R \div S$:

A
α
β

Another Division Example

Relations
R, S:

A	B	C	D	E
α	a	α	a	*1*
α	a	γ	a	*1*
α	a	γ	b	*1*
β	a	γ	a	*1*
β	a	γ	b	*3*
γ	a	γ	a	*1*
γ	a	γ	b	*1*
γ	a	β	b	*1*

R

D	E
a	*1*
b	*1*

S

$R \div S$:

A	B	C
α	a	γ
γ	a	γ

Aggregate Functions and Operations

- **Aggregation function** takes a collection of values and returns a single value as a result.

  avg: average value
  min: minimum value
  max: maximum value
  sum: sum of values
  count: number of values

- **Aggregate operation** in relational algebra

$$_{G1, G2, \ldots, Gn}\ \mathcal{G}_{\ F1(A1), F2(A2), \ldots, Fn(An)}\ (E)$$

  – E is any relational-algebra expression
  – $G_1, G_2 \ldots, G_n$ is a list of attributes on which to group (can be empty)
  – Each F_i is an aggregate function
  – Each A_i is an attribute name

Aggregate Operation: Example

Relation *R*:

A	*B*	*C*
α	α	7
α	β	7
β	β	3
β	β	10

$\mathcal{g}_{\text{sum(c)}}(\text{R})$

sum-C
27

Aggregate Operation: Example

- Relation *account* grouped by *branch-name*:

branch-name	*account-number*	*balance*
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

$${}_{\textit{branch-name}}\, \mathcal{G}\, {}_{\textit{sum(balance)}}\, (\textit{account})$$

branch-name	*balance*
Perryridge	1300
Brighton	1500
Redwood	700

OUTER JOIN Operations

- **Outer joins**
  - Keep all tuples in *R*, or all those in *S*, or all those in both relations regardless of whether or not they have matching tuples in the other relation
  - **Types**
    - **LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN**
  - Example

$$\text{TEMP} \leftarrow (\text{EMPLOYEE} ⟕_{\text{Ssn=Mgr\_ssn}} \text{DEPARTMENT})$$

$$\text{RESULT} \leftarrow \pi_{\text{Fname, Minit, Lname, Dname}}(\text{TEMP})$$

$⟕$

The OUTER UNION Operation

- Take union of tuples from two relations that have some common attributes
  - All tuples from both relations included in the result
  - But tuples with the same value combination will appear only once

Outer Join – Example

Relation *loan*

loan-number	*branch-name*	*amount*
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

Relation *borrower*

customer-name	*loan-number*
Jones	L-170
Smith	L-230
Hayes	L-155

Left Outer Join

- **Join**

loan $\bowtie$ *Borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

- **Left Outer Join**

loan ⟕ *Borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	NULL

Right Outer Join, Full Outer Join

- **Right Outer Join**

loan ⟖ *Borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	NULL	NULL	Hayes

- **Full Outer Join**

loan ⟗ *Borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	NULL
L-155	NULL	NULL	Hayes

EXAMPLE

Consider the following tables:

- B(<u>bid</u>, bname, color)
- R (<u>sid</u>, <u>bid</u>, <u>day</u>)
- S(<u>sid</u>, sname, rating, age)

- B(<u>*bid*</u>, *bname*, *color*)
- R(<u>*sid*</u>, <u>*bid*</u>, <u>*day*</u>)
- S(<u>*sid*</u>, *sname*, *rating*, *age*)

- Find the names of sailors who've reserved a red or a green boat.

$$\pi_{sname}(\sigma_{color='red' \vee color='green'}(B) \bowtie R \bowtie S)$$

- Find the names of sailors who've reserved a red and a green boat.

$$\rho_{T1(sid)}(\pi_{sid}(\sigma_{color='red'}(B) \bowtie R))$$

$$\rho_{T2(sid)}(\pi_{sid}(\sigma_{color='green'}(B) \bowtie R))$$

$$\pi_{sname}((T1 \cap T2) \bowtie S)$$

- B(<u>*bid*</u>, *bname*, *color*)
- R(<u>*sid*</u>, <u>*bid*</u>, <u>*day*</u>)
- S(<u>*sid*</u>, *sname*, *rating*, *age*)

- Find the names of sailors who've reserved boat #103.

$$\pi_{sname}(\sigma_{bid=103 \wedge R.sid=S.sid}(R \times S))$$

$$\pi_{sname}(\sigma_{bid=103}(R) \bowtie S)$$

$$\pi_{sname}(\sigma_{bid=103}(R \bowtie S))$$

- Find the names of sailors who've reserved a red boat.

$$\pi_{sname}(\sigma_{color='red'}(B) \bowtie R \bowtie S)$$

- B(<u>*bid*</u>, *bname*, *color*)
- R(<u>*sid*</u>, <u>*bid*</u>, <u>*day*</u>)
- S(<u>*sid*</u>, *sname*, *rating*, *age*)

- Find the names of sailors who've reserved at least two boats.

$$\rho_{R1(sid1, bid1, day1)}(R)$$
$$\rho_{R2(sid2, bid2, day2)}(R)$$
$$\pi_{sname}(\sigma_{(sid1=sid2)\wedge(bid1\neq bid2)}(R1\times R2\times S))$$

- Find the names of sailors who've reserved all boats.

$$\pi_{sname}((\pi_{sid,bid}(R\bowtie S)\div\pi_{bid}(B))\bowtie S)$$
$$\pi_{sname}(\pi_{sid,sname,bid}(R\bowtie S)\div\pi_{bid}(B))$$

References

- Chapter 6 Slides, Fundamental of Database Systems, 6e, Elamasri and Navathe,
- Slides from Database Management Systems, Ramakishnan and Gehrke.