Module 4

Updates and Retrievals in Relational Model

Objectives

- Learn update operations in relational model
- Learn Relational Algebra model for retrievals in Relational model
- Reference:
  Elmasri & Navathe, Chapter 6

Topics

- **Updating and Defining Relations**
- **Relational Algebra**

Update Operations on Relations

- **Two kinds of operations**
- **Updates**
  - -insert
  - -delete
  - -modify
  - -should not violate integrity constraints
- **Retrievals**
  - -querying the data
  - -modelled with relational algebra

Insert Operation

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

```
insert
<‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’,‘05-Apr-
50’,’6357 Windy Lane’, F, 28000, NULL, 4 >
into EMPLOYEE
```

- **Inserts a new tuple t into a relation r(R)**

...Insert Operation

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

```
insert
<‘Cecilia’, ‘F’, ‘Kolonsky’, ‘999887777’,‘05-Apr-
50’,’6357 Windy Lane’, F, 28000, NULL, 4 >
into EMPLOYEE
```

- **Violates Key Constraint**

...Insert Operation

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

```
insert
<‘Cecilia’, ‘F’, ‘Kolonsky’, NULL,‘05-Apr-50’,’6357
Windy Lane’, F, 28000, NULL, 4 >
into EMPLOYEE
```

- **Violates Entity Integrity**

...Insert Operation

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

```
insert
<‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’,‘05-Apr-
50’,’6357 Windy Lane’, F, 28000, NULL, 7 >
into EMPLOYEE
```

- **Violates Referential Integrity**
- **(Department with d_number = 7 does not exist yet)**

...Insert Operation

- Can violate
  -key constraint
  -entity integrity
  -referential integrity
- Remedies
  -reject insertion
  -try to correct by prompting user for more info.

Delete Operation

- Delete a tuple for a relation
- Can only violate referential integrity (but easily)
- Remedies
  -reject the deletion
  -cascade (propagate) the deletion
  -set referencing attribute value to NULL

Modify Operation

- Change the value of one or more attributes in a tuple
- Modifying a non-key or non-foreign-key attribute is usually OK
- Modifying a key or foreign-key attribute can violate all of the integrity constraints

Topics

- Updating and Defining Relations
- Relational Algebra

Relational Algebra

- Operations for manipulating entire relations
- Pure query language
- Procedural -says how data should be manipulated
- minimal set -all that's needed
- good mathematical properties
- not user-friendly
- low-level abstraction of query process

Relational Algebra Operations

- operations to select tuples from relations or combine related tuples from different relations
- Result of each operation is a new relation

- Set theoretic operations
  - -UNION
  - -INTERSECTION
  - -DIFFERENCE
  - -CARTESIAN PRODUCT

- Relational database specific
  - -SELECT
  - -PROJECT
  - -JOIN

Select Operation

σ <selection condition> (<relation name>)

- Select a subset of tuples in a relation that satisfy a selection condition
- Creates a new relation with same attributes as the source relation
- E.g.

σ (DNO=4 AND SALARY > 30000) (EMPLOYEE)

Select Operation -example

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

σ (DNO=4 AND SALARY > 25000)OR(DNO=5 AND SALARY > 30000) (EMPLOYEE)

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5

Selection Conditions

$\sigma_{\text{<selection condition>}}$ **(**<relation name>**)**

- **attribute_name operation constant**
  DNO = 4
  SALARY > 30000
- **attribute_name operation attribute_name**
- **operations:** =, <, ≤, >, ≥, ≠
- **constant value is from attribute domain**
- **clauses can be combined with** AND, OR, NOT

$\sigma_{\text{(DNO=4 AND SALARY > 30000) OR (DNO = 5)}}$ **(**EMPLOYEE**)**

...Select Operation

$\sigma_{\text{<selection condition>}}$ **(**<relation name>**)**

- **Select operation is unary -applies to only one relation**
- **only applies to one tuple at a time**
- **resulting relation has same degree (attribute set) as source relation**
- **select is commutative**
  (can be applied in any order)

$\sigma_{\text{(cond1)}} (\sigma_{\text{(cond2)}} (R)) = \sigma_{\text{(cond2)}} (\sigma_{\text{(cond1)}} (R))$

Project Operation

π <attribute list> (<relation name>)

- "select" the columns in the attribute list from the source relation
- Creates a new relation with same attributes as the attribute list
- E.g.

π LNAME, FNAME, SALARY (EMPLOYEE)

Project Operation -example

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

π LNAME, FNAME, SALARY (EMPLOYEE)

LNAME	FNAME	SALARY
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabber	Ahmad	25000
Borg	James	55000

NOTE ORDER

WHAT HAPPENS WHEN WE DON'T SELECT A KEY?

Project Operation

$\pi_{\text{<attribute list>}}(\text{<relation name>})$

- attributes appear in order of attribute list
- if attribute does not contain a key, duplicates will be removed in resulting relation
- project is not commutative

$\pi_{\text{LNAME, FNAME, SALARY}}(\text{EMPLOYEE})$

Combining Operations

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

$\pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO = 5}}(\text{EMPLOYEE}))$

FNAME	LNAME	SALARY
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Renaming and Sequencing Operations

Temp <- $\sigma_{\text{DNO = 5}}$ (EMPLOYEE)

RESULT <- $\pi_{\text{FNAME, LNAME, SALARY}}$ (Temp)

RESULT		
FNAME	LNAME	SALARY
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Renaming Attributes

Temp <- $\sigma_{\text{DNO = 5}}$ (EMPLOYEE)

R(FNAME,LNAME,SALARY) <- $\pi_{\text{FNAME, LNAME, SALARY}}$ (Temp)

TEMP									
FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5

R		
FirstName	LastName	SALARY
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Set-Based Operations

- **UNION**
  $R \cup S$ is a relation containing all tuples in R or S, or both. Duplicates are eliminated
- **INTERSECTION**
  $R \cap S$ is a relation containing all tuples in both R and S
- **DIFFERENCE**
  R - S is a relation containing all that are in R but not in S
- **These operations only defined if R and S are over the same attribute types (domains)**

Union Example

Example: Find the social insurance number of all employees who work in department 5, or who supervise someone in department 5

DN5EMP <- $\sigma_{\text{DNO = 5}}$ (EMPLOYEE)

Result1 <- π_{SSN} (DN5EMP)

Result2(SSN) <- π_{SUPERSSN} (DN5EMP)

Result <- Result1 $\cup$ Result2

RESULT1

ssn
123456789
333445555
666884444
453453453

RESULT2

ssn
333445555
888665555

RESULT

ssn
123456789
333445555
666884444
453453453
888665555

More Examples

STUDENT

FName	LName
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FName	LName
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

FName	LName
Susan	Yao
Ramesh	Shah

STUDENT ∩ INSTRUCTOR

FName	Lname
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

STUDENT ∪ INSTRUCTOR

Still More Examples

STUDENT

FName	LName
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FName	LName
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

FName	LName
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

STUDENT – INSTRUCTOR

FName	LName
John	Smith
Ricardo	Browne
Francis	Johnson

INSTRUCTOR – STUDENT

Commuting and Associating

- $R \cup S = S \cup R$
  $R \cup (S \cup T) = (S \cup R) \cup T$

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

- $R - S \neq S - R$

Cartesian Product

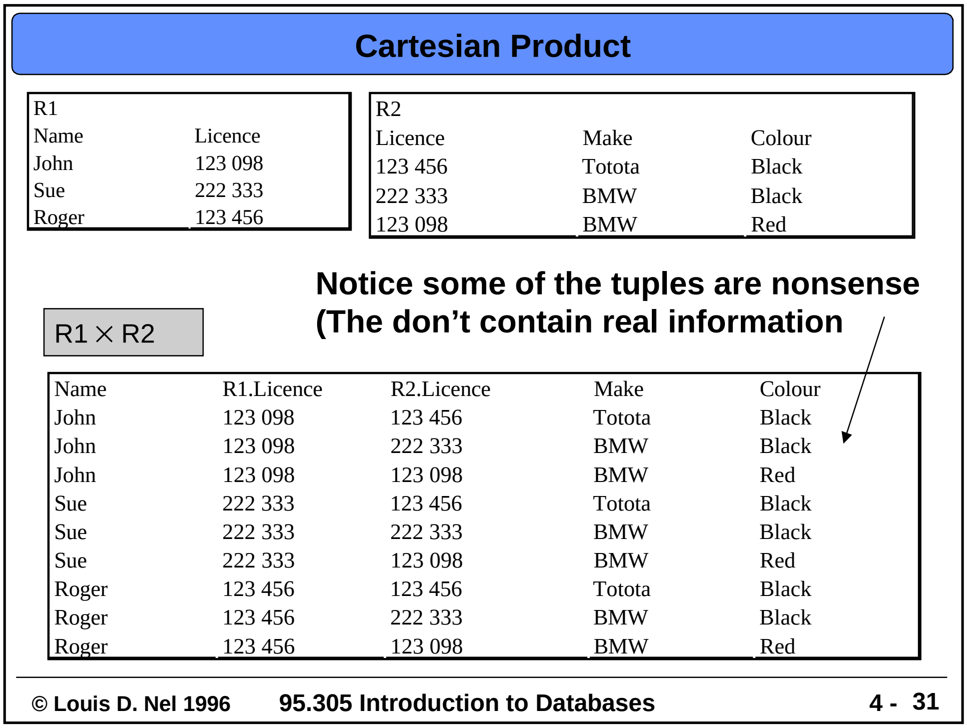
R1

Name	Licence
John	123 098
Sue	222 333
Roger	123 456

R2

Licence	Make	Colour
123 456	Totota	Black
222 333	BMW	Black
123 098	BMW	Red

- **Find everyone who drives a black car**
- **Cannot do this with simple select or project**
- **Set operations cannot help, at first, because relations are over different attribute sets**



Cartesian Product

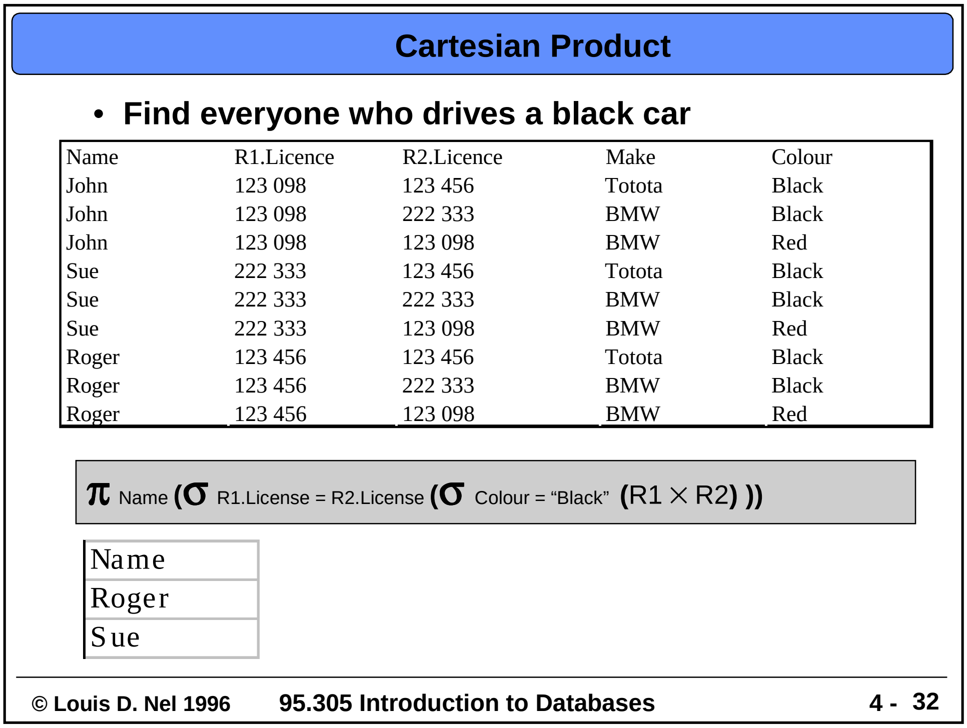
R1	
Name	Licence
John	123 098
Sue	222 333
Roger	123 456

R2		
Licence	Make	Colour
123 456	Totota	Black
222 333	BMW	Black
123 098	BMW	Red

R1 × R2

Notice some of the tuples are nonsense (The don't contain real information

Name	R1.Licence	R2.Licence	Make	Colour
John	123 098	123 456	Totota	Black
John	123 098	222 333	BMW	Black
John	123 098	123 098	BMW	Red
Sue	222 333	123 456	Totota	Black
Sue	222 333	222 333	BMW	Black
Sue	222 333	123 098	BMW	Red
Roger	123 456	123 456	Totota	Black
Roger	123 456	222 333	BMW	Black
Roger	123 456	123 098	BMW	Red



Cartesian Product

- **Find everyone who drives a black car**

Name	R1.Licence	R2.Licence	Make	Colour
John	123 098	123 456	Totota	Black
John	123 098	222 333	BMW	Black
John	123 098	123 098	BMW	Red
Sue	222 333	123 456	Totota	Black
Sue	222 333	222 333	BMW	Black
Sue	222 333	123 098	BMW	Red
Roger	123 456	123 456	Totota	Black
Roger	123 456	222 333	BMW	Black
Roger	123 456	123 098	BMW	Red

$$\pi_{\text{Name}} (\sigma_{\text{R1.License = R2.License}} (\sigma_{\text{Colour = “Black”}} (R1 \times R2)))$$

Name
Roger
Sue

Cartesian Product

- More than anything the Cartesian product is a way to combine two relations which range over different attribute sets
- R(A1,...,An) × S(B1,...,Bm) is a relation over the attributes (A1,...,An,B1,...,Bm)
- R × S has a tuple for each combination of a tuple from R matched with a tuple from S
- If R has nr tuples and S has ns tuples,
  R × S will have ns*nr tuples. -quite big

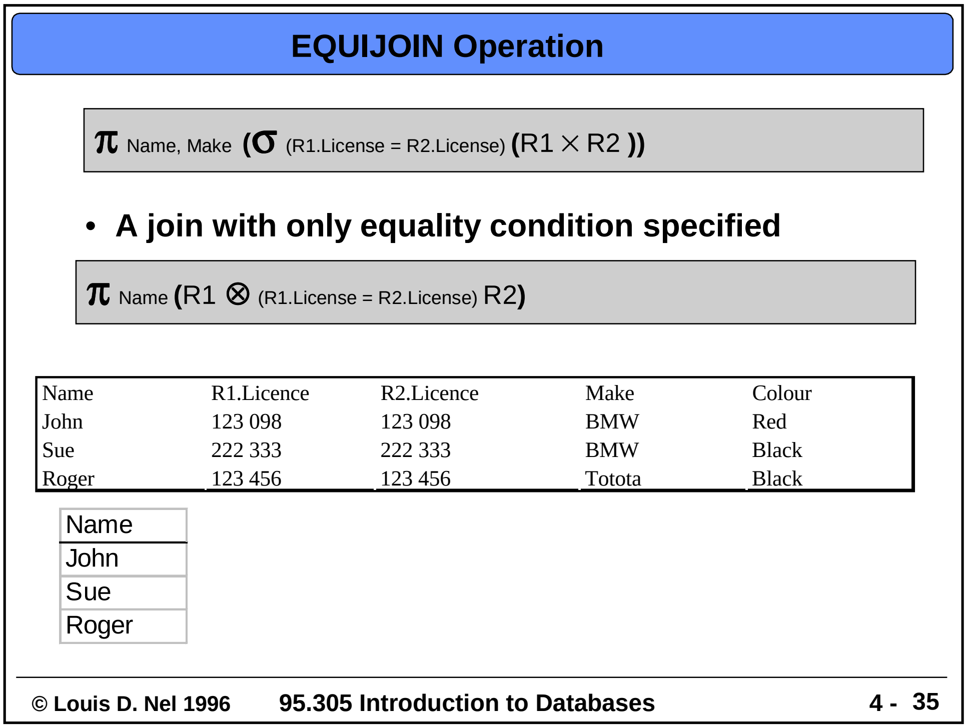
(Theta) Join Operation

π Name (σ (R1.License = R2.License) AND (Colour = “Black”) (R1 × R2))

- Cartesian Product followed by Select is very common and so is combined in a Join operation

π Name (R1 ⊗ (R1.License = R2.License) AND (Colour = “Black”) R2)

- The join operation R ⊗ <join condition> S is a relation over the attributes of R and S in which contains all matchings of a tuple from R and a tuple from S which satisfy the join condition.
- Join conditions can be specified like select conditions



EQUIJOIN Operation

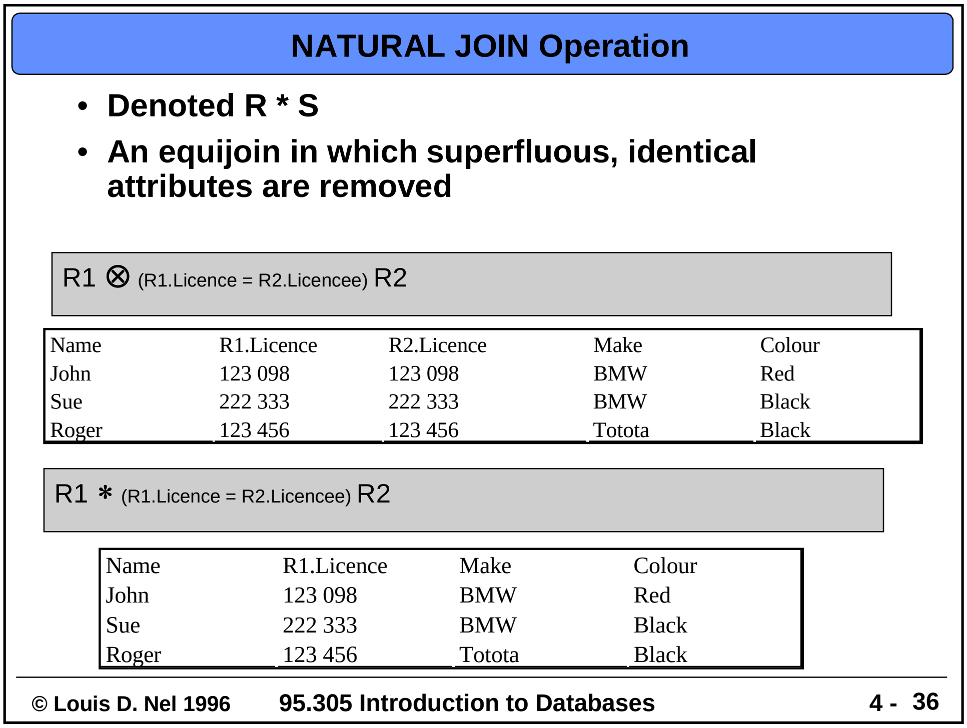
$\pi_{\text{Name, Make}}\ (\sigma_{(\text{R1.License = R2.License})}\ (\text{R1} \times \text{R2}))$

- A join with only equality condition specified

$\pi_{\text{Name}}\ (\text{R1} \otimes_{(\text{R1.License = R2.License})} \text{R2})$

Name	R1.Licence	R2.Licence	Make	Colour
John	123 098	123 098	BMW	Red
Sue	222 333	222 333	BMW	Black
Roger	123 456	123 456	Totota	Black

Name
John
Sue
Roger



NATURAL JOIN Operation

- Denoted R * S
- An equijoin in which superfluous, identical attributes are removed

$\text{R1} \otimes_{(\text{R1.Licence = R2.Licencee})} \text{R2}$

Name	R1.Licence	R2.Licence	Make	Colour
John	123 098	123 098	BMW	Red
Sue	222 333	222 333	BMW	Black
Roger	123 456	123 456	Totota	Black

$\text{R1} *_{(\text{R1.Licence = R2.Licencee})} \text{R2}$

Name	R1.Licence	Make	Colour
John	123 098	BMW	Red
Sue	222 333	BMW	Black
Roger	123 456	Totota	Black

NATURAL JOIN Operation

- **Denoted R * S**
- **An equijoin in which superfluous, identical attributes are removed**
- **Based on join attributes -which have the same name in each relation**

join attribute

R1

Name	Licence
John	123 098
Sue	222 333
Roger	123 456

R2

Licence	Make	Colour
123 456	Totota	Black
222 333	BMW	Black
123 098	BMW	Red

R1 ∗ R2

Name	Licence	Make	Colour
John	123 098	BMW	Red
Sue	222 333	BMW	Black
Roger	123 456	Totota	Black

NATURAL JOIN Operation

- **Forms a cartesian product of two relations but keeps only those tuples for which common attributes are equal**
- **Multiple common attributes are compared pairwise and then ANDed**
- **Natural Join is associative**

(R1 * R2) * R3 = R1 * (R2 * R3)

Division Operation

- Defined for R(A), S(B) where B is a subset of A
- R ÷ S is a relation on A-B and contains a tuple t if for every ts in S(B) there is a tuple tr in R(A) such that tr[B] = ts and t[A-B] = tr[A-B].

R

COLOUR	SIZE
red	small
yellow	small
green	small
blue	small
red	med
green	med
yellow	large
green	large
blue	large
red	extra large
yellow	extra large
green	extra large

÷

S

COLOUR
red
yellow
green

=

SIZE
small
extra large

Complete Algebra

- A minimal, but complete set of operations are
  - -Select
  - -Project
  - -Union
  - -Difference
  - -Cartesian Product
- The other operations (Intersections, Join, Division...) can be expressed as combinations of the above
- for example:

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

Additional Operations

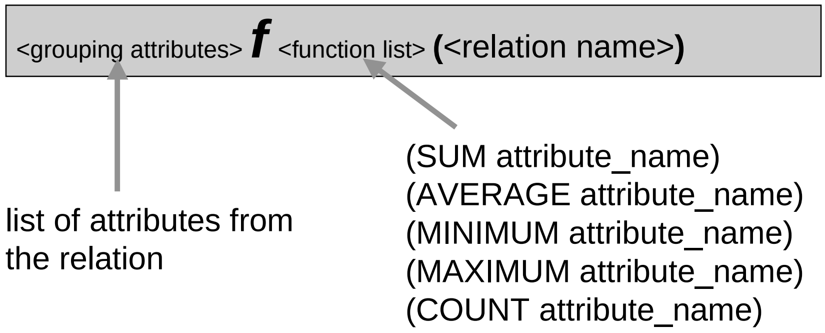
- Some database requests cannot be performed using relational algebra
- Commercial query languages extent the relational operations by providing operations for

  -aggregate functions
  -outer join
  -outer union

Aggregate Functions

- Mathematical computation over tuples in a relation
- e.g.
  -computing sum of attribute values
  -computing average value of an attribute
  -computing min. or max. value
  -counting tuples
- also
  -grouping tuples by some value and then computing aggregate functions for each group separately

...Aggregate Functions



<grouping attributes> ***f*** <function list> **(**<relation name>**)**

list of attributes from
the relation

(SUM attribute_name)
(AVERAGE attribute_name)
(MINIMUM attribute_name)
(MAXIMUM attribute_name)
(COUNT attribute_name)

- **Example retrieve the department number, number of employees in the department and average salary in the department from the employee relation**

R(DNO, NUM_EMP, AVG_SALARY) <-

DNO ***f*** COUNT SSN AVERAGE SALARY **(**EMPLOYEE**)**

Computing Functions

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

DNO ***f*** COUNT SSN AVERAGE SALARY (EMPLOYEE)

R

DNO	COUNT_SSN	AVERAGE_SALARY
5	4	33250
4	3	31000
1	1	55000

Computing Functions and Renaming Attributes

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

R(DNO, NUM_EMP, AVG_SALARY) <-

DNO ***f*** COUNT SSN AVERAGE SALARY (EMPLOYEE)

R

DNO	NUM_EMP	AVG_SALARY
5	4	33250
4	3	31000
1	1	55000

Computing Functions without Grouping

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

f COUNT SSN AVERAGE SALARY **(**EMPLOYEE**)**

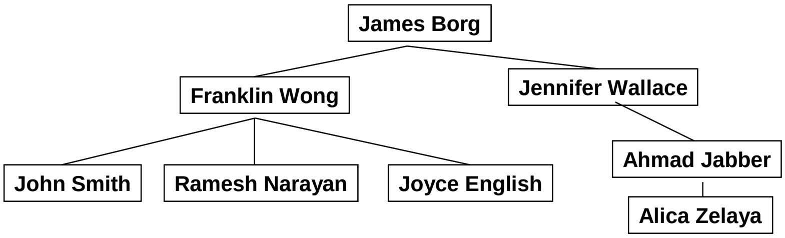
COUNT_SSN	AVERAGE_SALARY
8	35125

Recursive Retrievals

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry	F	43000	888665555	4
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak	M	38000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice	F	25000	333445555	5
Ahmad	V	Jabber	987987987	29-Mar-59	980 Dallas	M	25000	987654321	4
James	E	Borg	888665555	10-Nov-27	450 Stone	M	55000	NULL	1

Find Everyone who works directly or indirectly for James Borg



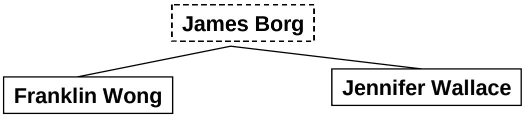
...Recursive Retrievals

BORG_SSN <- π SSN (σ FNAME = ‘James’ AND LNAME =‘Borg’ (EMPLOYEE))

SUPERVISION(SSN1,SSN2) <- π SSN, SUPERSSN (EMPLOYEE)

RESULT1(SSN) <- π SSN1 (SUPERVISION $\otimes$ (SSN2 = SSN) BORG_SSN)

Find Everyone who works directly for James Borg



...Recursive Retrievals

BORG_SSN <- π SSN (σ FNAME = ‘James’ AND LNAME =‘Borg’ (EMPLOYEE))

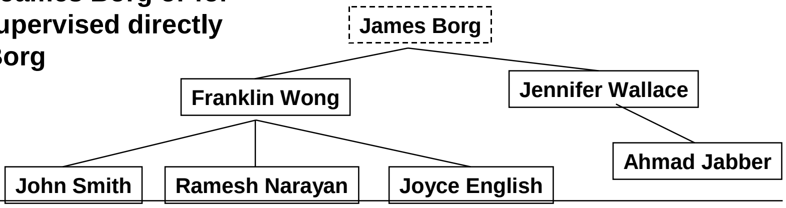
SUPERVISION(SSN1,SSN2) <- π SSN, SUPERSSN (EMPLOYEE)

RESULT1(SSN) <- π SSN1 (SUPERVISION $\otimes$ (SSN2 = SSN) BORG_SSN)

RESULT2(SSN) <- π SSN1 (SUPERVISION $\otimes$ (SSN2 = SSN) RESULT1)

RESULT3 <- RESULT1 $\cup$ RESULT2

Find Everyone who works directly for James Borg or for someone supervised directly by James Borg



Outer Joins

EMPLOYEE

FNAME	INIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	9-Jan-55	731 Fondern	M	30000	333445555	5
Franklin	T	Wong	333445555	8-Dec-45	638 Voss	M	40000	888665555	5
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle	F	25000	987987987	4
Jennifer	S	Wallace	98765						
Ramesh	K	Narayan	66688						
Joyce	A	English	45345						
Ahmad	V	Jabber	98798						
James	E	Borg	88866						

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
Research	5	333445555	22-May-78
Administration	4	987654321	1-Jan-85
Headquarters	1	888665555	19-Jun-71

List the names of all employees and the names of the departments they supervise -if they supervise any

Left Outer Join

- **Left outer join**

TEMP <- (EMPLOYEE ⟕ $_{\text{SSN = MGRSSN}}$ DEPARTMENT)

RESULT <- $\pi_{\text{FNAM,INIT, LNAME, DNAME}}$ (TEMP)

RESULT

FNAME	INIT	LNAME	DNAME
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

Outer Join

- Left outer join keeps all the tuples in the first relation, matches them with tuples from the second based on join condition -if no tuple matches the fields are padded with NULL
- Right outer join keeps all the tuples in the second relation, matches them with tuples from the first based on join condition -if no tuple matches the fields are padded with NULL
- Full outer join keeps all tuples from both relations and pads unmatched tuples with NULLs as needed

Outer Union

- Union of two relations which have only partially Union compatible schemas

  STUDENT(Name, SSN, Dept, Adviser)
  FACULTY(Name, SSN, Dept, Rank)

- Resulting schema will be over attributes

  (Name, SSN, Dept, Adviser, Rank)

- Both relations contribute all of their tuples, but Student tuples will have the Rank = NULL, and Faculty tuples will have the Adviser = NULL