

General form of Grouping and Aggregation

Evaluation steps:

```
SELECT  S  
FROM    R1,...,Rn  
WHERE   C1  
GROUP BY a1,...,ak  
HAVING  C2
```

Evaluate FROM-WHERE,
apply condition C1

Group by the attributes a₁,...,a_k

Apply condition C2 to each group
(may have aggregates)

Compute aggregates in S and return
the result

Example

- List the employees name and the department name that they manage.
- Temp \leftarrow (Employee $\bowtie_{\text{Ssn=Mgr_Ssn}}$ Department)
- Result $\leftarrow \pi_{\text{Fname, Minit, Lname, Dname}}(\text{Temp})$

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Left Outer Join

- List the employees name and the department name that they manage. **If they don't manage one, then indicate this with a null value.**
- Temp \leftarrow (Employee $\bowtie_{Ssn=Mgr_Ssn}$ Department)
- Result $\leftarrow \pi_{Fname, Minit, Lname, Dname}(Temp)$

DEPARTMENT			
Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Left Outer Join

- List the employees name and the department name that they manage. **If they don't manage one, then indicate this with a null value.**
- $\text{Temp} \leftarrow (\text{Employee} \bowtie_{\text{Ssn}=\text{Mgr_Ssn}} \text{Department})$
- $\text{Result} \leftarrow \pi_{\text{Fname, Minit, Lname, Dname}}(\text{Temp})$

RESULT

Fname	Minit	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

Right Outer Join

- List the employees name and the department name that they manage. If they don't manage one, then indicate this with a null value.

- Temp \leftarrow (Department $\bowtie_{\text{Mgr_Ssn} = \text{Ssn}}$ Employee)
- Result $\leftarrow \pi_{\text{Fname, Minit, Lname, Dname}}(\text{Temp})$

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Full Outer Join

List the employees name and the department name that they manage. If they don't manage one or the department have no manager, then indicate this with a null value.

Temp \leftarrow Employee \bowtie Department
Result $\leftarrow \pi_{Fname, Lname, Dname}(Temp)$

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19
CS	6		

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Full Outer Join vs Cartesian Product

What is the difference ?
OR ...
are they same ... ?

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19
CS	6		

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Outer Join Operation

- In INNER JOIN, tuples without a *matching* are eliminated from the join result
 - Tuples with null are also eliminated
 - This amounts to loss of information.

OUTER joins operations are used when we want to keep

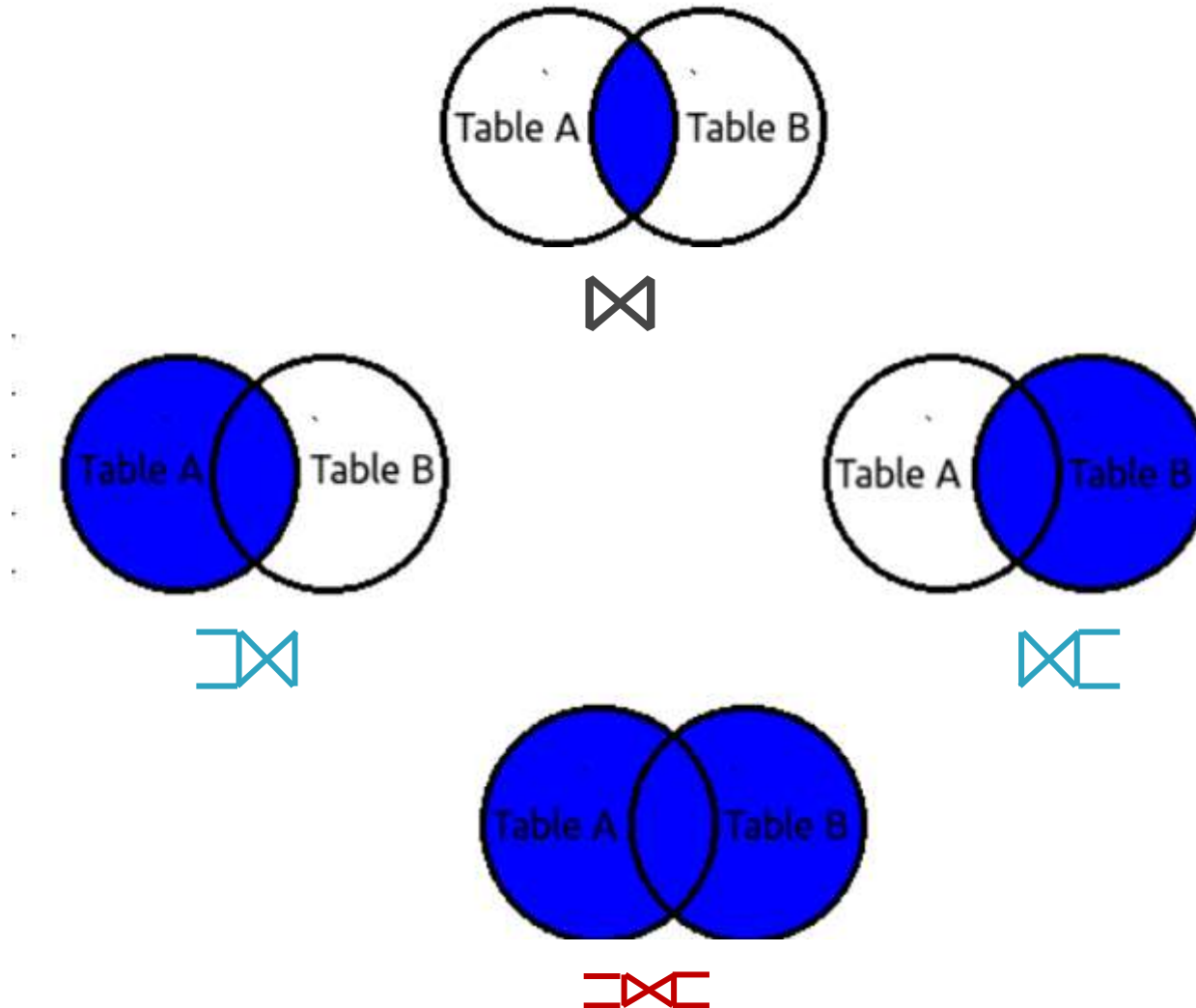
- **all the tuples in R** in the join result , or
- **all the tuples in S** in the join result, or
- **all tuples in both relations R and S** in the join result

Outer Join Operation

- **Left outer join:** keeps every tuple in R, denoted as $R \bowtie S$
 - if no matching tuple is found in S, then the attributes of S in the join result are filled with null values.
- **Right outer join:** keeps every tuple in S in the result of R S.
- **Full outer join:** keeps all tuples in both the left and the right relations. It is denoted by



Inner and Outer Joins



Class Exercise - Another Example Outer Join

List the employees name and the Project name that they work on. If they don't work on any project or a project have no employee working on it, then indicate this with a null value.

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

DIFFERENT JOINS IN SQL

```
SELECT  E.FNAME, E.LNAME, S.FNAME, S.LNAME  
FROM    EMPLOYEE AS E , EMPLOYEE AS S  
WHERE   E.SUPERSSN=S.SSN
```

```
SELECT  E.FNAME, E.LNAME, S.FNAME, S.LNAME  
FROM    (EMPLOYEE E JOIN EMPLOYEE S)  
        ON E.SUPERSSN=S.SSN
```

```
SELECT  E.FNAME, E.LNAME, S.FNAME, S.LNAME  
FROM    (EMPLOYEE E LEFT OUTER JOIN EMPLOYEE S  
        ON E.SUPERSSN=S.SSN)
```

NATURAL JOIN IN SQL

No Natural Join in Transact-SQL

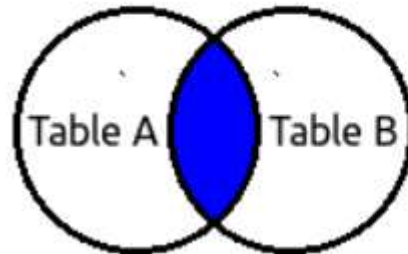
The keyword **OUTER** is marked as optional that is

A LEFT JOIN B is same as A LEFT OUTER JOIN B

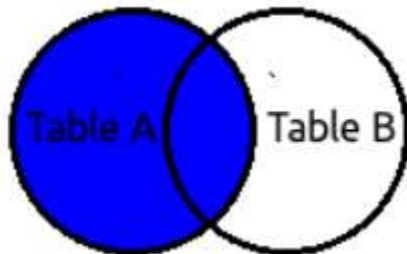
CROSS JOIN is for cartesian product



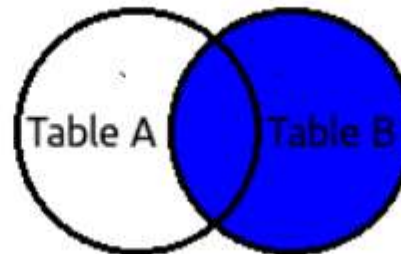
INNER AND OUTER JOINS



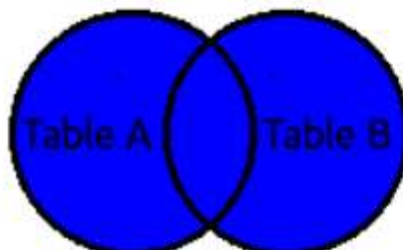
```
SELECT [list] FROM  
  [Table A] A  
  INNER JOIN  
  [Table B] B  
  ON A.Value = B.Value
```



```
SELECT [list] FROM  
  [Table A] A  
  LEFT JOIN  
  [Table B] B  
  ON A.Value = B.Value
```



```
SELECT [list] FROM  
  [Table A] A  
  RIGHT JOIN  
  [Table B] B  
  ON A.Value = B.Value
```



```
SELECT [list] FROM  
  [Table A] A  
  FULL OUTER JOIN  
  [Table B] B  
  ON A.Value = B.Value
```


General form of Grouping and Aggregation

Evaluation steps:

```
SELECT  S  
FROM    R1,...,Rn  
WHERE   C1  
GROUP BY a1,...,ak  
HAVING  C2
```

Evaluate FROM-WHERE,
apply condition C1

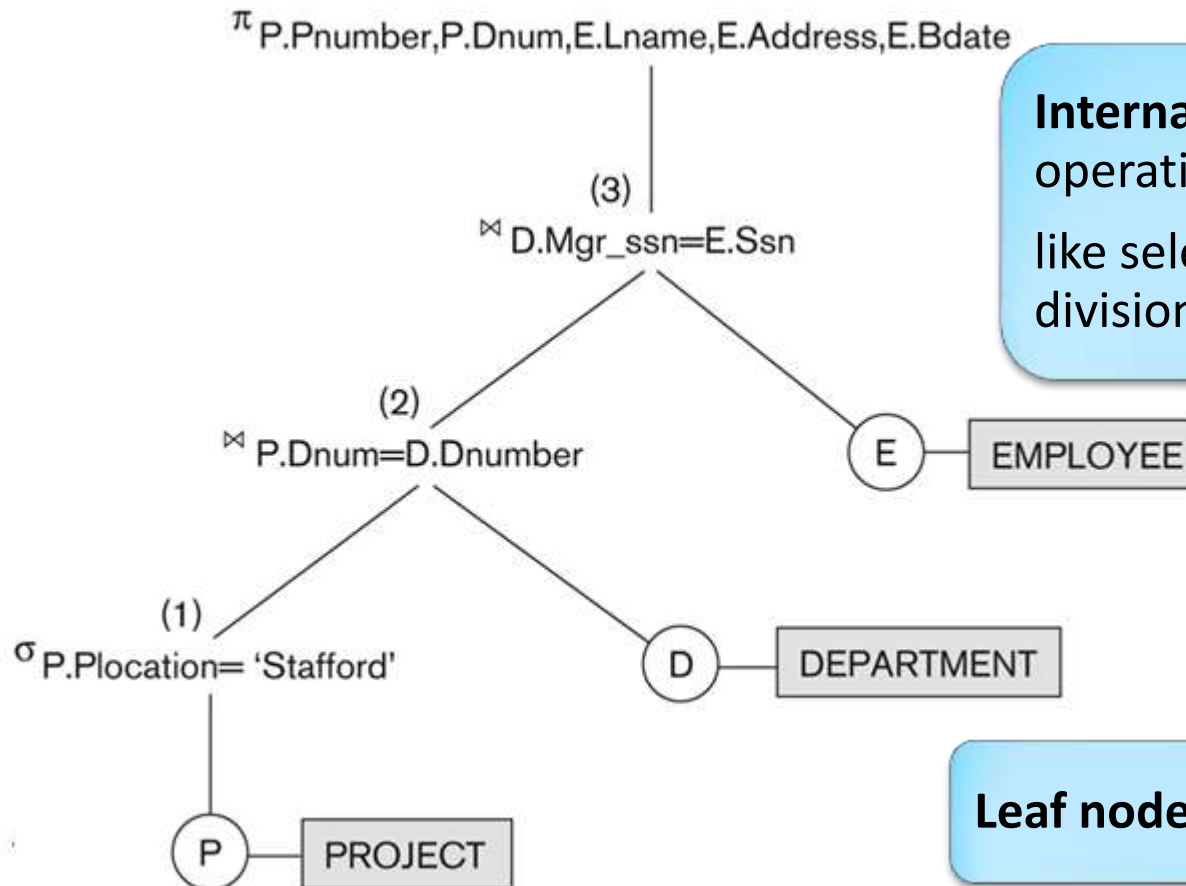
Group by the attributes a₁,...,a_k

Apply condition C2 to each group
(may have aggregates)

Compute aggregates in S and return
the result

Example of Query Tree

For every project located in 'Stafford', list the **project no**, the **controlling department no**, and the **department manager's last name, address, and birth date**.



Internal Nodes stand for operations

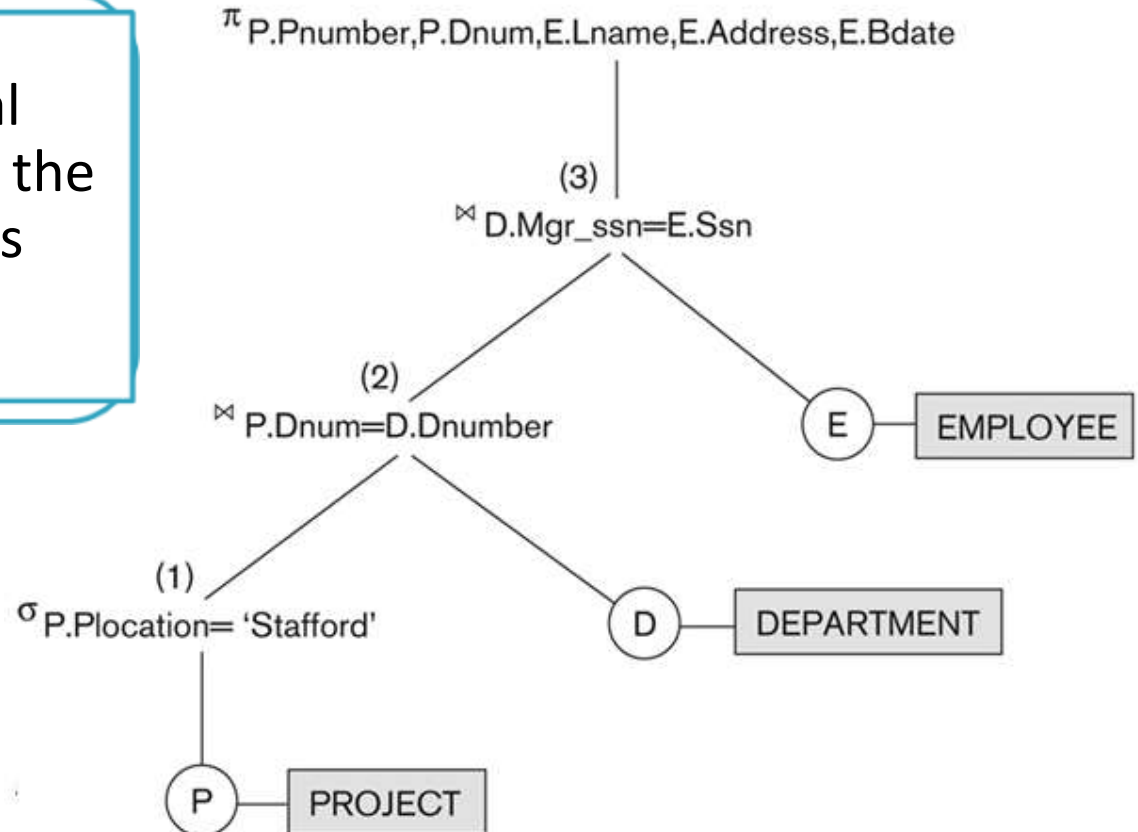
like selection, projection, join, division,

Leaf nodes represent **base relations**

Query Tree is an internal data structure to represent a query

Standard technique to estimate the work done in executing the query, and the ***optimization of execution***

A tree gives a good visual feel of the complexity of the query and the operations involved



Practice.. Practice.. Practice

Display the names of the departments that have no female employees.

Select Dname From Department

Except

Select Dname From Department **join** Employee on Dnumber = Dnum

Where Gender = 'F'

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Practice.. Practice.. Practice

Find the SSN of all employees who are older than their Department Manager.

Select ssn

From (Employee as E join Department as D on E.dno=D.dnumber) join

Employee as M on mgr_ssn=ssn

Where E.birthdate < M.birthdate

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Practice.. Practice.. Practice

Display the SSN and name of all employees who report to John. (in other words John is their immediate supervisor)

Select E.ssn

From Employee E join Employee S on E.Super_ssn=S.ssn and
Where S.Fname='John'

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

PROJECT

PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

WORKS_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

Practice.. Practice.. Practice

Find out how many managers there are without listing them.

```
Select count(distinct mgr_ssn)
From department
```

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

PROJECT

PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

WORKS_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

Practice.. Practice.. Practice

Find out the difference between highest and lowest salaries.

```
Select max(salary) - min(salary)
From department
```

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Practice.. Practice.. Practice

List SSN and Fname of all employees with more than 2 children

Select ssn, Fname

From employee join dependent on ssn=essn

Where Relationship = 'Son' or Relationship = 'daughter'

Groupby ssn

Having count(ssn)> 2

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Practice.. Practice.. Practice

- List the names of the departments, where all the employees have salary > 30000

```
SELECT Dname
FROM Employee, Department
WHERE dno= dnumber
GROUP BY Dnumber, Dname
HAVING 30000 < min(Salary)
```



Example: Boat Rental database

○ Consider the following Boat Rental database schema:

- SAILOR (SID, SName, Phone, City)
- BOAT (BName, BType, Price, OID)
- RESERVATION (SID, BName, Date, Duration)
- OWNER (OID, OName, Phone, Street, City, Country)

What does the query do?

```
SELECT Bname
FROM   (Boat b join Owner o on b.OID = o.OID)
       join Reservation r on r.BName = b.BName
WHERE  Country = 'Pakistan'
ORDER BY Price
```



Example: Boat Rental database

- Consider the following Boat Rental database schema:
 - SAILOR (SID, SName, Phone, City)
 - BOAT (BName, BType, Price, OID)
 - RESERVATION (SID, BName, Date, Duration)
 - OWNER (OID, OName, Phone, Street, City, Country)
- Select bname,count(*)
- From reservation r ,boat b,owner o
- Where b.bname=r.bname and b. oid=o.oid and country='USA'
- Group by bname
- Having count(*) > 10

What does the above query do?



Example: Boat Rental database

- Consider the following schema
 - SAILOR (SID, SName, Phone, City)
 - BOAT (BName, BType, Price, OID)
 - RESERVATION (SID, BName, Date, Duration)
 - OWNER (OID, OName, Phone, Street, City, Country)
- **Find the names of boats that are reserved by at least ten different sailors.**
- - Select bname
 - From reservation r
 - Group by bname
 - Having count(DISTINCT SID) >9

Example: Boat Rental database

- Consider the following schema
 - SAILOR (SID, SName, Phone, City)
 - BOAT (BName, BType, Price, OID)
 - RESERVATION (SID, BName, Date, Duration)
 - OWNER (OID, OName, Phone, Street, City, Country)
- List name and price of the boats that were reserved in 2018 or in 2019.

Select distinct b.bname, b.price

From reservation r join boat b on r.bname = b.bname

Where r.date LIKE '%2018%' or r.date LIKE '%2019%'

Example: Boat Rental database

- Consider the following schema
 - SAILOR (SID, SName, Phone, City)
 - BOAT (BName, BType, Price, OID)
 - RESERVATION (SID, BName, Date, Duration)
 - OWNER (OID, OName, Phone, Street, City, Country)
- **List name and price of the boats that were reserved in 2018 and in 2019.**

```
Select distinct b.bname, b.price
From reservation r, boat b
Where r.bname = b.bname and r.date LIKE '%2018%'
INTERSECT
Select distinct b.bname, b.price
From reservation r, boat b
Where r.bname = b.bname and r.date LIKE '%2019%'
```

Example: Boat Rental database

- Consider the following schema
 - SAILOR (SID, SName, Phone, City)
 - BOAT (BName, BType, Price, OID)
 - RESERVATION (SID, BName, Date, Duration)
 - OWNER (OID, OName, Phone, Street, City, Country)
- List name, owner name, and price of the boats which were reserved in 2018 but not in 2019.

Select distinct b.bname, b.price, o.ename

From reservation r, boat b, owner o

Where r.bname = b.bname and b.oid=o.oid and r.date LIKE '%2018%'

EXCEPT

Select distinct b.bname, b.price, o.ename

From reservation r, boat b, owner o

Where r.bname = b.bname and b.oid=o.oid and r.date LIKE '%2019%'

The boat rental schema

- SAILOR (SID, SName, Phone, City)
- BOAT (BName, BType, Price, OID)
- RESERVATION (SID, BName, Date, Duration)
- OWNER (OID, OName, Phone, Street, City, Country)

Boat Rental

List name and price of the boats which were reserved in 2018 and 2019 but not in 2020.

```
(Select distinct b.bname, b.price
From reservation r join boat b on r.bname = b.bname
Where r.date LIKE '%2018%'
INTERSECT
Select distinct b.bname, b.price
From reservation r join boat b on r.bname = b.bname
Where r.date LIKE '%2019%')
EXCEPT
Select distinct b.bname, b.price
From reservation r join boat b on r.bname = b.bname
Where r.date LIKE '%2020%'
```

Example: Boat Rental database

- Consider the following schema
 - SAILOR (SID, SName, Phone, City)
 - BOAT (BName, BType, Price, OID)
 - RESERVATION (SID, BName, Date, Duration)
 - OWNER (OID, OName, Phone, Street, City, Country)
- Find ids of the sailors who **only** reserved a boat owned by Mr. Jonas with OID=12345
 - All sailors who reserved a boat – sailors who have reserved a boat not owned by MR Jonas
- Find ids of the sailors who have **never** reserved a boat owned by Mr. Jonas with OID=12345
 - All sailors – sailors who have reserved a boat owned by MR jonas

SQL SERVER (TSQL) Functions

- Visit following slides for details on SQL functions
 - Like aggregate
 - String Functions
 - Date Functions
 - Math Functions
- <https://docs.microsoft.com/en-us/sql/t-sql/functions/functions?view=sql-server-ver15>