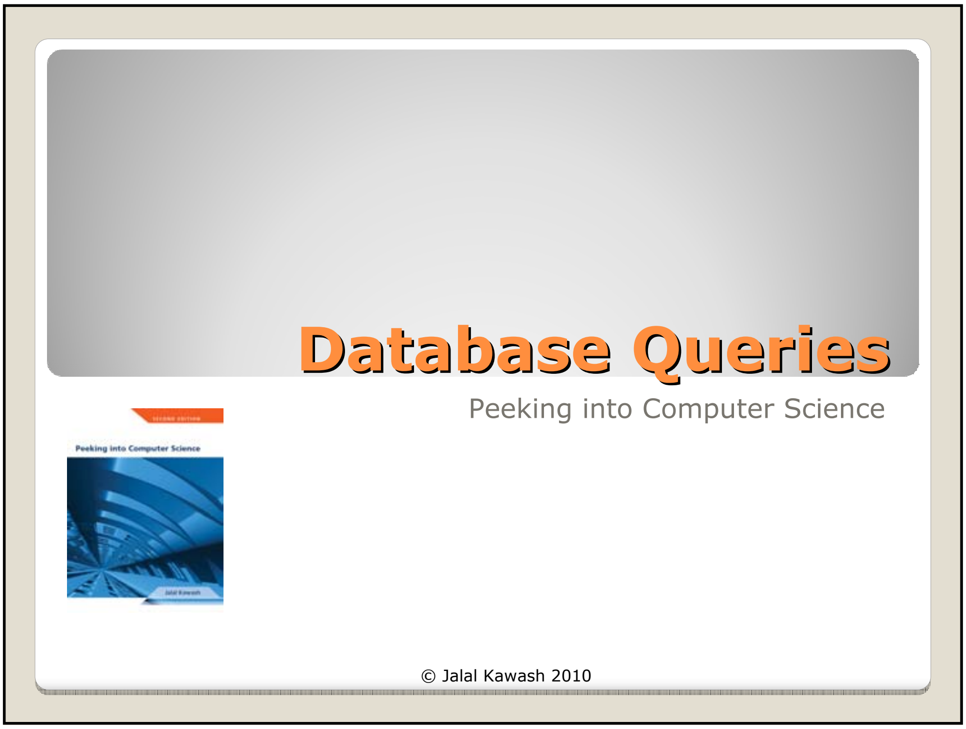
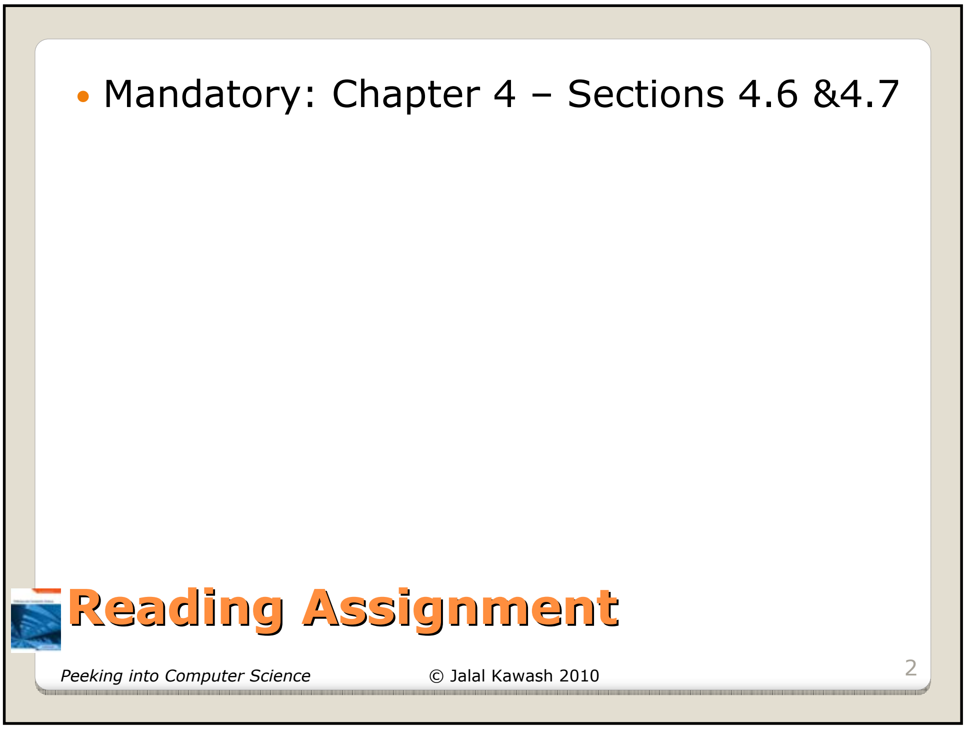

SECOND EDITION
Peeking into Computer Science
Jalal Kawash

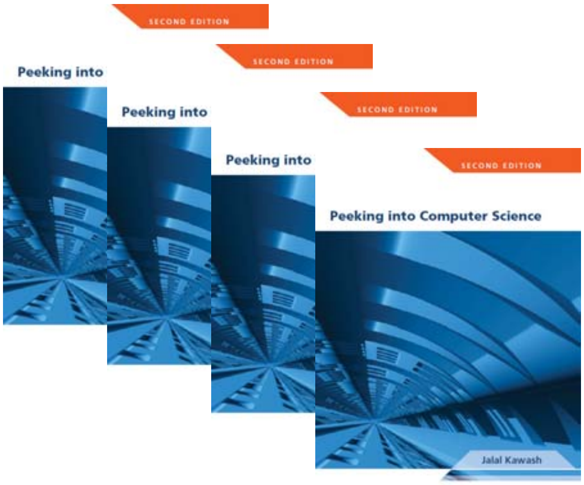
Database Queries

Peeking into Computer Science



- Mandatory: Chapter 4 – Sections 4.6 &4.7

Reading Assignment



Join Queries

By the end of this section, you will be able to:

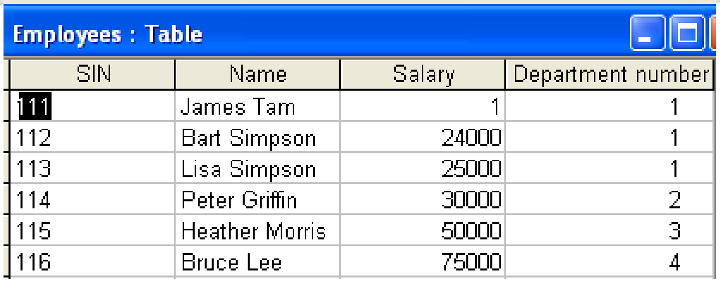
1. Formulate queries on multiple tables
2. Understand how natural joins work
3. Determine the result of a multi-table query

Objectives

JT’s Extra: Set Multiplication

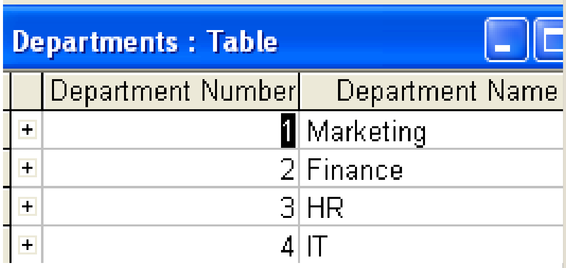
- Recall: set multiplication ‘determines all possible combinations of elements from each set’.
- In actual databases not all combinations may occur.
  - E.g., an employee typically is just a member of one department not all departments.
- The database implementation of a set multiplication is ‘join’.

JT’s Extra: Join Example



Employees : Table

SIN	Name	Salary	Department number
111	James Tam	1	1
112	Bart Simpson	24000	1
113	Lisa Simpson	25000	1
114	Peter Griffin	30000	2
115	Heather Morris	50000	3
116	Bruce Lee	75000	4

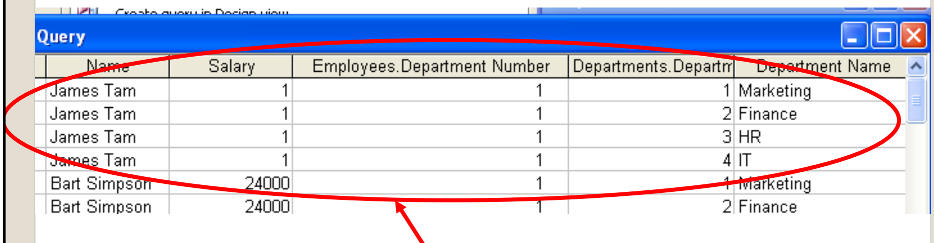


Departments : Table

	Department Number	Department Name
+	1	Marketing
+	2	Finance
+	3	HR
+	4	IT

- SELECT *
- FROM Employees, Departments;

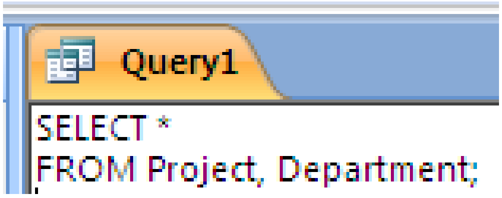
JT's Extra: Query For Join Example



Name	Salary	Employees.Department Number	Departments.Departm	Department Name
James Tam	1	1	1	Marketing
James Tam	1	1	2	Finance
James Tam	1	1	3	HR
James Tam	1	1	4	IT
Bart Simpson	24000	1	1	Marketing
Bart Simpson	24000	1	2	Finance

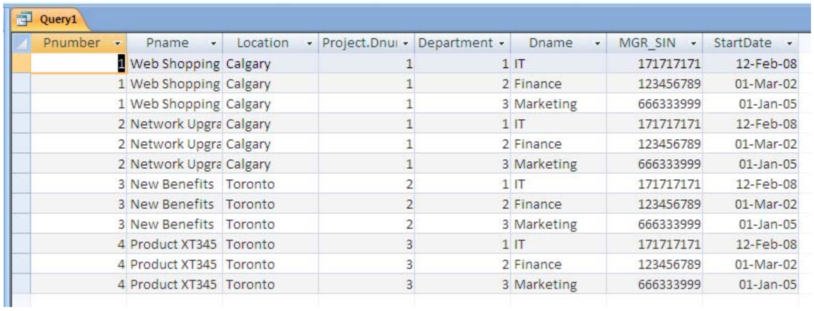
James Tam is only in Department 1 but the query yields all possible combinations (whether they actually occur in the data or not)

JT's Extra: Result Of The Query



Query1

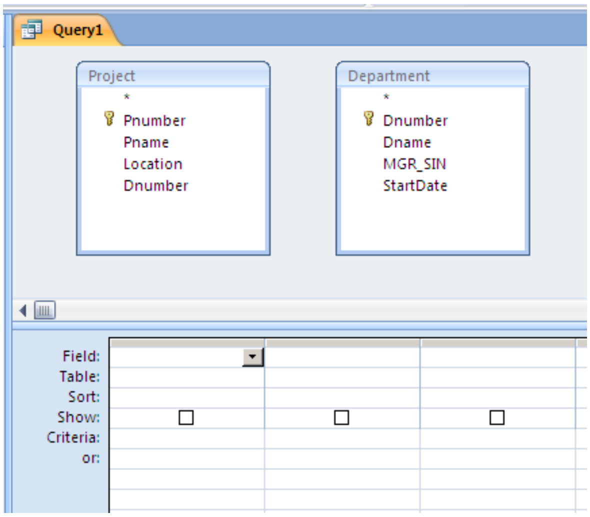
```
SELECT *
FROM Project, Department;
```



Query1

Pnumber	Pname	Location	Project.Dnu	Department	Dname	MGR_SIN	StartDate
1	Web Shopping	Calgary	1	1	IT	171717171	12-Feb-08
1	Web Shopping	Calgary	1	2	Finance	123456789	01-Mar-02
1	Web Shopping	Calgary	1	3	Marketing	666333999	01-Jan-05
2	Network Upgra	Calgary	1	1	IT	171717171	12-Feb-08
2	Network Upgra	Calgary	1	2	Finance	123456789	01-Mar-02
2	Network Upgra	Calgary	1	3	Marketing	666333999	01-Jan-05
3	New Benefits	Toronto	2	1	IT	171717171	12-Feb-08
3	New Benefits	Toronto	2	2	Finance	123456789	01-Mar-02
3	New Benefits	Toronto	2	3	Marketing	666333999	01-Jan-05
4	Product XT345	Toronto	3	1	IT	171717171	12-Feb-08
4	Product XT345	Toronto	3	2	Finance	123456789	01-Mar-02
4	Product XT345	Toronto	3	3	Marketing	666333999	01-Jan-05

PROJECT x DEPARTMENT



QBE

JT’s Extra: Join Of Actual Cases From The Database

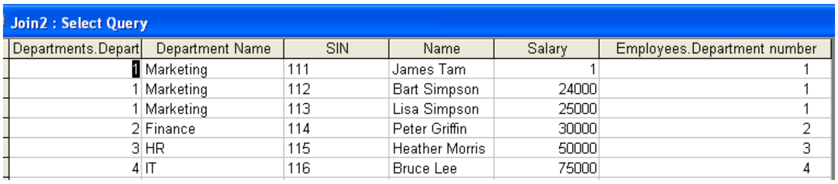
- In the previous example this would only include the cases where the department number of the Employees table matched the department number of the Departments table.
  - (It should exclude non-existent combinations of employees and departments.

JT’s Extra: MS-Access SQL Query

- SELECT Departments.*, Employees.*
- FROM Departments INNER JOIN Employees ON
  - Departments.[Department Number] = Employees.[Department number];

JT's Extra: Query Results

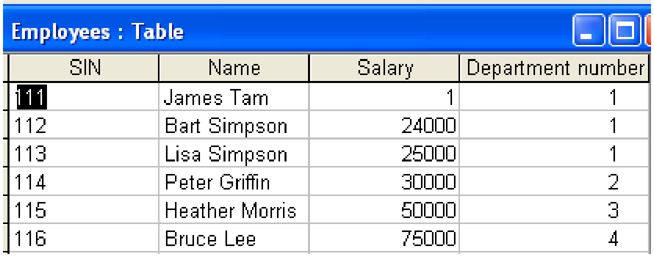
Results



Join2 : Select Query

Departments.Depart	Department Name	SIN	Name	Salary	Employees.Department number
1	Marketing	111	James Tam	1	1
1	Marketing	112	Bart Simpson	24000	1
1	Marketing	113	Lisa Simpson	25000	1
2	Finance	114	Peter Griffin	30000	2
3	HR	115	Heather Morris	50000	3
4	IT	116	Bruce Lee	75000	4

Employees table

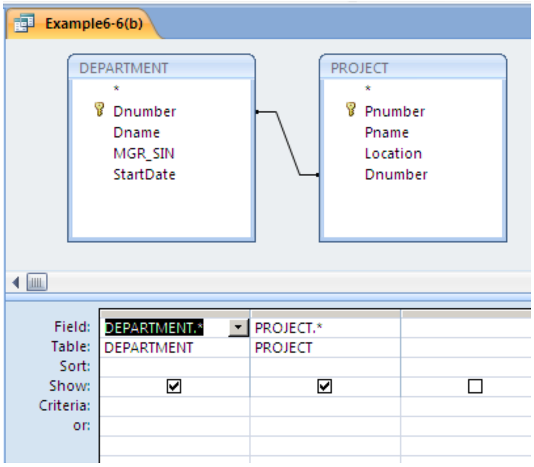


Employees : Table

SIN	Name	Salary	Department number
111	James Tam	1	1
112	Bart Simpson	24000	1
113	Lisa Simpson	25000	1
114	Peter Griffin	30000	2
115	Heather Morris	50000	3
116	Bruce Lee	75000	4

Where Dnumbers Match

Pnumber	Pname	Location	Dnumber	Dnumber	Dname	MGR_SIN	StartDate
1	Web Shopping	Calgary	1	1	IT	171717171	12-Feb-2008
1	Web Shopping	Calgary	1	2	Finance	123456789	1-Mar-2002
1	Web Shopping	Calgary	1	3	Marketing	666333999	1-Jan-2005
2	Network Upgrade	Calgary	1	1	IT	171717171	12-Feb-2008
2	Network Upgrade	Calgary	1	2	Finance	123456789	1-Mar-2002
2	Network Upgrade	Calgary	1	3	Marketing	666333999	1-Jan-2005
3	New Benefits	Toronto	2	1	IT	171717171	12-Feb-2008
3	New Benefits	Toronto	2	2	Finance	123456789	1-Mar-2002
3	New Benefits	Toronto	2	3	Marketing	666333999	1-Jan-2005
4	Product XT345	Toronto	3	1	IT	171717171	12-Feb-2008
4	Product XT345	Toronto	3	2	Finance	123456789	1-Mar-2002
4	Product XT345	Toronto	3	3	Marketing	666333999	1-Jan-2005

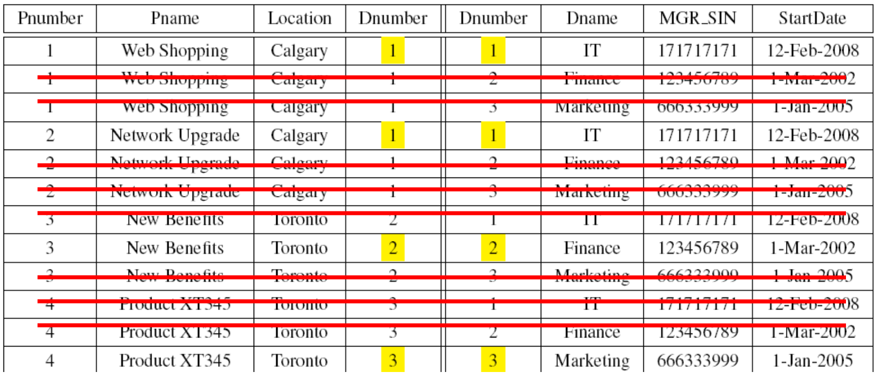


Field:	DEPARTMENT.*	PROJECT.*	
Table:	DEPARTMENT	PROJECT	
Sort:			
Show:	☑	☑	□
Criteria:			
or:			

Natural Join

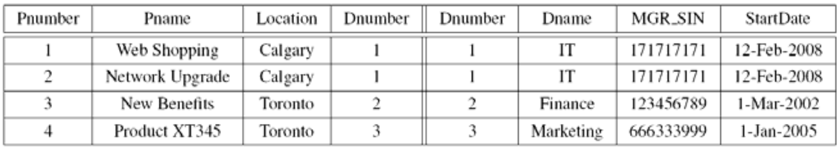
```
SELECT  *
FROM    PROJECT, DEPARTMENT
WHERE   PROJECT.Dnumber = DEPARTMENT.Dnumber
```

Natural Join Condition



Pnumber	Pname	Location	Dnumber	Dnumber	Dname	MGR_SIN	StartDate
1	Web Shopping	Calgary	1	1	IT	171717171	12-Feb-2008
~~1~~	~~Web Shopping~~	~~Calgary~~	~~1~~	~~2~~	~~Finance~~	~~123456789~~	~~1-Mar-2002~~
~~1~~	~~Web Shopping~~	~~Calgary~~	~~1~~	~~3~~	~~Marketing~~	~~666333999~~	~~1-Jan-2005~~
2	Network Upgrade	Calgary	1	1	IT	171717171	12-Feb-2008
~~2~~	~~Network Upgrade~~	~~Calgary~~	~~1~~	~~2~~	~~Finance~~	~~123456789~~	~~1-Mar-2002~~
~~2~~	~~Network Upgrade~~	~~Calgary~~	~~1~~	~~3~~	~~Marketing~~	~~666333999~~	~~1-Jan-2005~~
~~3~~	~~New Benefits~~	~~Toronto~~	~~2~~	~~1~~	~~IT~~	~~171717171~~	~~12-Feb-2008~~
3	New Benefits	Toronto	2	2	Finance	123456789	1-Mar-2002
~~3~~	~~New Benefits~~	~~Toronto~~	~~2~~	~~3~~	~~Marketing~~	~~666333999~~	~~1-Jan-2005~~
~~4~~	~~Product XT345~~	~~Toronto~~	~~3~~	~~1~~	~~IT~~	~~171717171~~	~~12-Feb-2008~~
~~4~~	~~Product XT345~~	~~Toronto~~	~~3~~	~~2~~	~~Finance~~	~~123456789~~	~~1-Mar-2002~~
4	Product XT345	Toronto	3	3	Marketing	666333999	1-Jan-2005

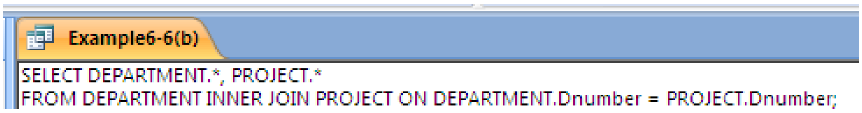
Natural Join Result



Pnumber	Pname	Location	Dnumber	Dnumber	Dname	MGR_SIN	StartDate
1	Web Shopping	Calgary	1	1	IT	171717171	12-Feb-2008
2	Network Upgrade	Calgary	1	1	IT	171717171	12-Feb-2008
3	New Benefits	Toronto	2	2	Finance	123456789	1-Mar-2002
4	Product XT345	Toronto	3	3	Marketing	666333999	1-Jan-2005

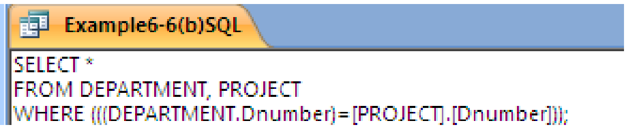
Natural Join Result

Default QBE Joins in ACCESS


Example6-6(b)

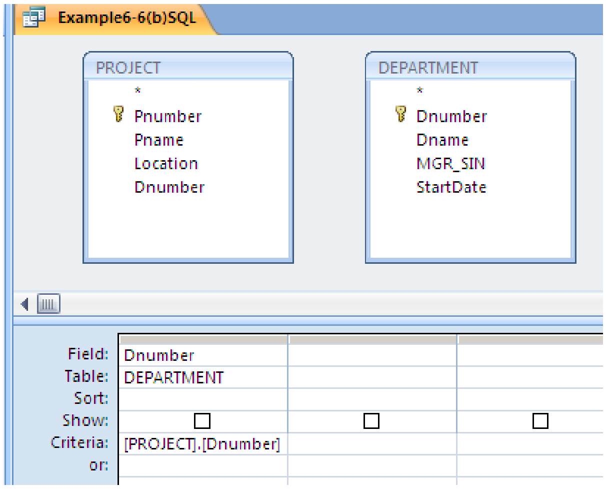
```
SELECT DEPARTMENT.*, PROJECT.*
FROM DEPARTMENT INNER JOIN PROJECT ON DEPARTMENT.Dnumber = PROJECT.Dnumber;
```

- Can be also done explicitly

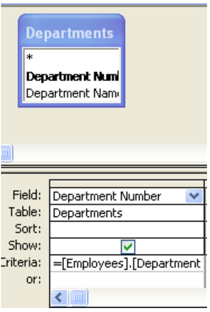

Example6-6(b)SQL

```
SELECT *
FROM DEPARTMENT, PROJECT
WHERE (((DEPARTMENT.Dnumber)=[PROJECT].[Dnumber]));
```

QBE with Explicit Join



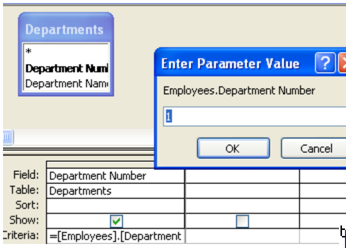
Field:	Dnumber		
Table:	DEPARTMENT		
Sort:			
Show:	☐	☐	☐
Criteria:	[PROJECT].[Dnumber]		
or:			



Field: Department Number
Table: Departments
Sort:
Show: ☑
Criteria: =[Employees].[Department
or:

QBE

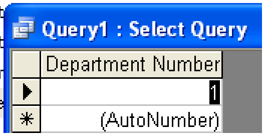
Query criteria



Employees.Department Number: 1

Field: Department Number
Table: Departments
Sort:
Show: ☑ □
Criteria: =[Employees].[Department

Query result



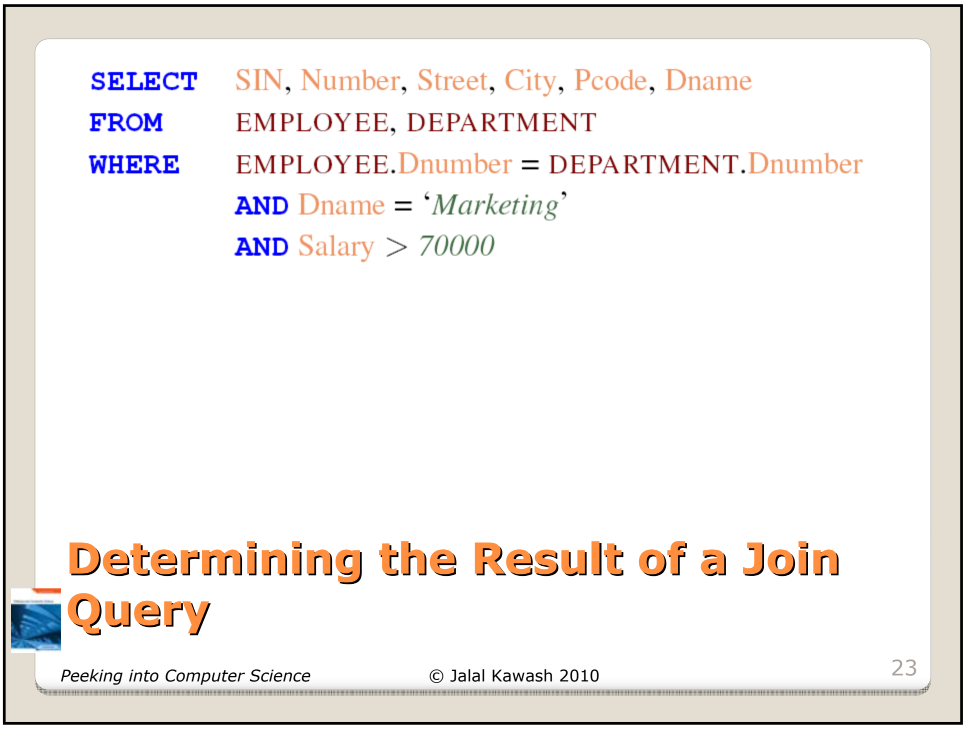
Department Number
1
(AutoNumber)

JT's Extra: Effect Of QBE With Explicit Join

Retrieve the address of each employee of the IT department

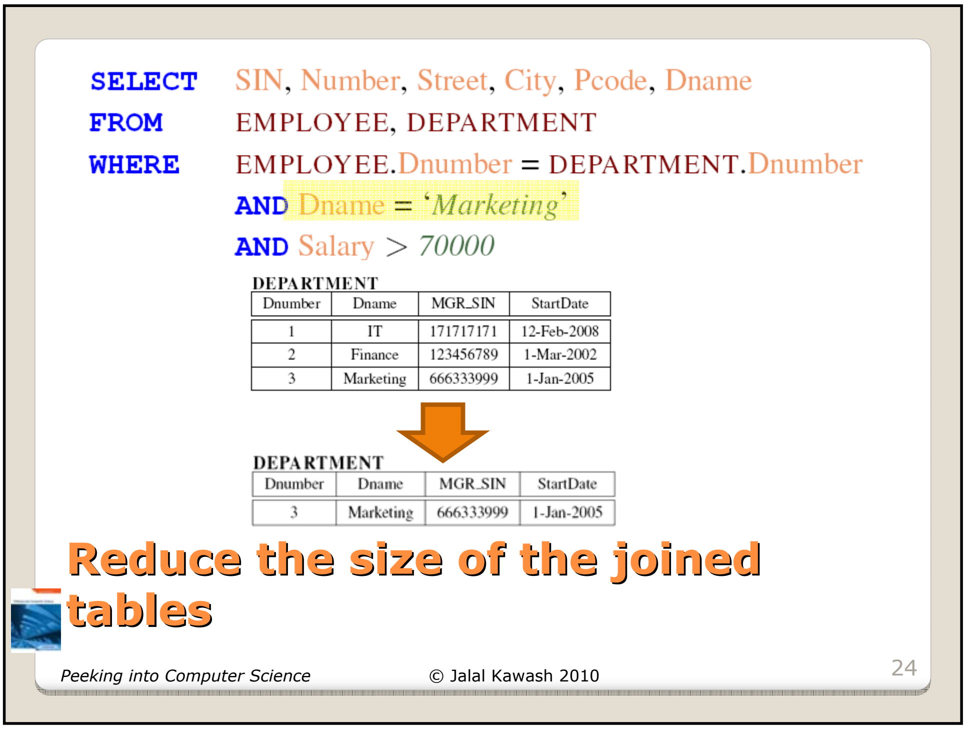
```
SELECT   Number, Street, City, Pcode
FROM     EMPLOYEE, DEPARTMENT
WHERE    EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
         AND Dname = 'IT'
```

Join Example



```
SELECT   SIN, Number, Street, City, Pcode, Dname
FROM     EMPLOYEE, DEPARTMENT
WHERE    EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
         AND Dname = 'Marketing'
         AND Salary > 70000
```

Determining the Result of a Join Query



```
SELECT   SIN, Number, Street, City, Pcode, Dname
FROM     EMPLOYEE, DEPARTMENT
WHERE    EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
         AND Dname = 'Marketing'
         AND Salary > 70000
```

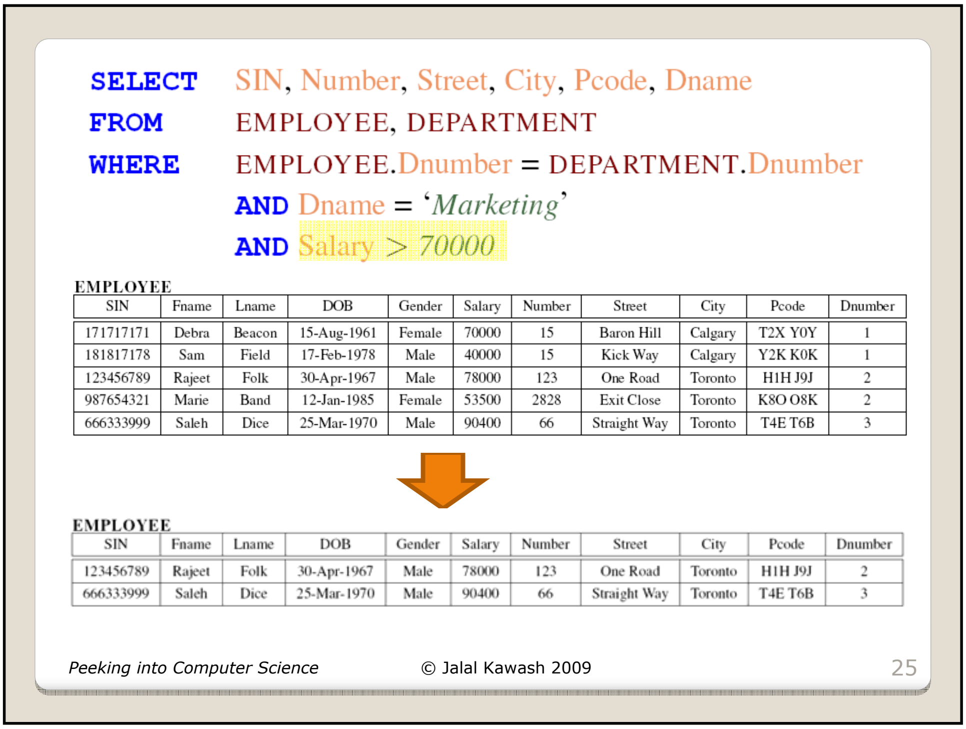
DEPARTMENT

Dnumber	Dname	MGR_SIN	StartDate
1	IT	171717171	12-Feb-2008
2	Finance	123456789	1-Mar-2002
3	Marketing	666333999	1-Jan-2005

DEPARTMENT

Dnumber	Dname	MGR_SIN	StartDate
3	Marketing	666333999	1-Jan-2005

Reduce the size of the joined tables



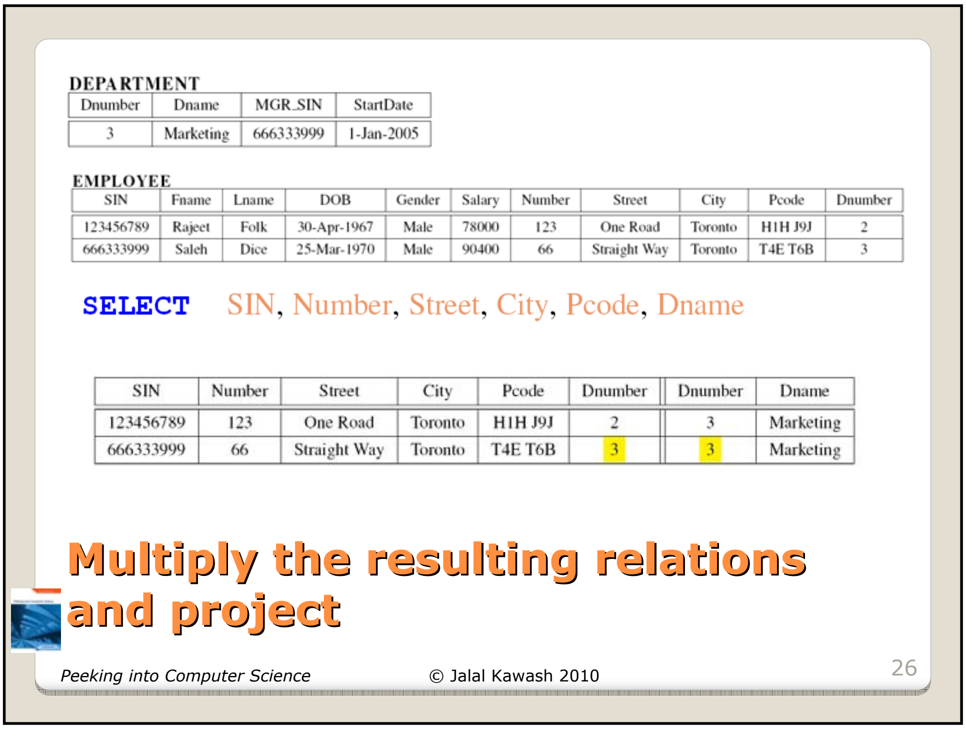
```
SELECT    SIN, Number, Street, City, Pcode, Dname
FROM      EMPLOYEE, DEPARTMENT
WHERE     EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
          AND Dname = 'Marketing'
          AND Salary > 70000
```

EMPLOYEE

SIN	Fname	Lname	DOB	Gender	Salary	Number	Street	City	Pcode	Dnumber
171717171	Debra	Beacon	15-Aug-1961	Female	70000	15	Baron Hill	Calgary	T2X Y0Y	1
181817178	Sam	Field	17-Feb-1978	Male	40000	15	Kick Way	Calgary	Y2K K0K	1
123456789	Rajeet	Folk	30-Apr-1967	Male	78000	123	One Road	Toronto	H1H J9J	2
987654321	Marie	Band	12-Jan-1985	Female	53500	2828	Exit Close	Toronto	K8O O8K	2
666333999	Saleh	Dice	25-Mar-1970	Male	90400	66	Straight Way	Toronto	T4E T6B	3

EMPLOYEE

SIN	Fname	Lname	DOB	Gender	Salary	Number	Street	City	Pcode	Dnumber
123456789	Rajeet	Folk	30-Apr-1967	Male	78000	123	One Road	Toronto	H1H J9J	2
666333999	Saleh	Dice	25-Mar-1970	Male	90400	66	Straight Way	Toronto	T4E T6B	3



DEPARTMENT

Dnumber	Dname	MGR_SIN	StartDate
3	Marketing	666333999	1-Jan-2005

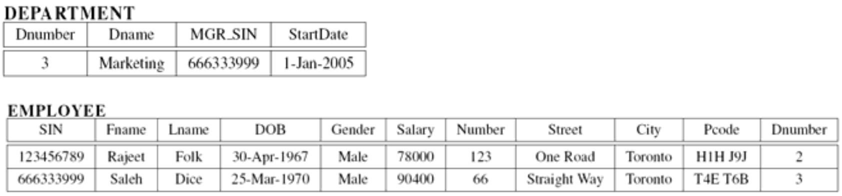
EMPLOYEE

SIN	Fname	Lname	DOB	Gender	Salary	Number	Street	City	Pcode	Dnumber
123456789	Rajeet	Folk	30-Apr-1967	Male	78000	123	One Road	Toronto	H1H J9J	2
666333999	Saleh	Dice	25-Mar-1970	Male	90400	66	Straight Way	Toronto	T4E T6B	3

```
SELECT    SIN, Number, Street, City, Pcode, Dname
```

SIN	Number	Street	City	Pcode	Dnumber	Dnumber	Dname
123456789	123	One Road	Toronto	H1H J9J	2	3	Marketing
666333999	66	Straight Way	Toronto	T4E T6B	3	3	Marketing

Multiply the resulting relations and project



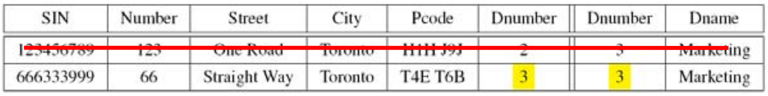
DEPARTMENT

Dnumber	Dname	MGR_SIN	StartDate
3	Marketing	666333999	1-Jan-2005

EMPLOYEE

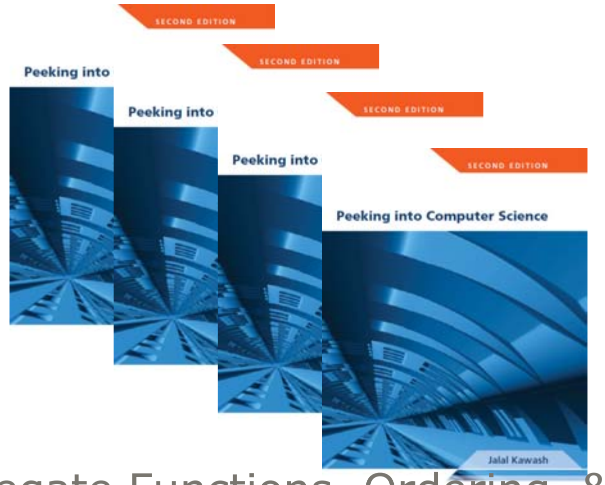
SIN	Fname	Lname	DOB	Gender	Salary	Number	Street	City	Pcode	Dnumber
123456789	Rajeet	Folk	30-Apr-1967	Male	78000	123	One Road	Toronto	H1H J9J	2
666333999	Saleh	Dice	25-Mar-1970	Male	90400	66	Straight Way	Toronto	T4E T6B	3

WHERE EMPLOYEE.Dnumber = DEPARTMENT.Dnumber



SIN	Number	Street	City	Pcode	Dnumber	Dnumber	Dname
~~123456789~~	~~123~~	~~One Road~~	~~Toronto~~	~~H1H J9J~~	~~2~~	~~3~~	~~Marketing~~
666333999	66	Straight Way	Toronto	T4E T6B	3	3	Marketing

Apply the join condition



Aggregate Functions, Ordering, & Grouping

Objectives

By the end of this section, you will be able to:

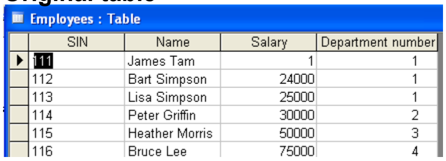
1. Use aggregate functions
2. Group the calculations of aggregate functions
3. Formulate SQL queries with HAVING and ORDER BY clauses

JT’s Extra: Basic Parts Of An SQL Query

- **SELECT**: Specifies the fields/columns shown in the query results e.g., SIN field.
- **FROM**: Lists the tables from which the data is to be selected e.g., look in the Employees table.
- **WHERE**: Provides the conditions to determine if rows/records are shown by the query.
- **ORDER BY**: Specifies the order in which rows are to be returned by the query.

JT’s Extra: ‘Order By’ (Data and Query)

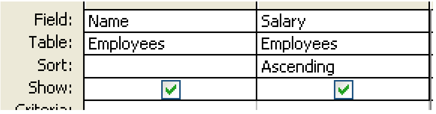
Original table



Employees : Table

	SIN	Name	Salary	Department number
▶	111	James Tam	1	1
	112	Bart Simpson	24000	1
	113	Lisa Simpson	25000	1
	114	Peter Griffin	30000	2
	115	Heather Morris	50000	3
	116	Bruce Lee	75000	4

QBE

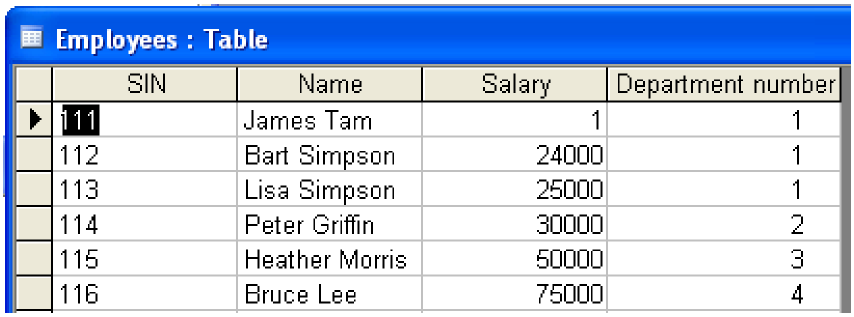


Field:	Name	Salary
Table:	Employees	Employees
Sort:		Ascending
Show:	☑	☑
Criteria:		

SQL

SELECT Employees.Name,
Employees.Salary
FROM Employees
ORDER BY Employees.Salary;

JT’s Extra: Query Results



Employees : Table

	SIN	Name	Salary	Department number
▶	111	James Tam	1	1
	112	Bart Simpson	24000	1
	113	Lisa Simpson	25000	1
	114	Peter Griffin	30000	2
	115	Heather Morris	50000	3
	116	Bruce Lee	75000	4

Order By Clause

- Orders the result of a query

```
SELECT      *
FROM        EMPLOYEE
WHERE       City != 'Calgary'
ORDER BY    Lname
```

Descending Order

```
SELECT      *
FROM        EMPLOYEE
WHERE       City != 'Calgary'
ORDER BY    Lname DESC
```

```
SELECT      *
FROM        EMPLOYEE
WHERE       City != 'Calgary'
ORDER BY    Lname, Fname
```

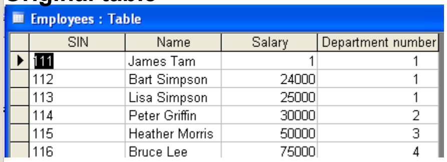
Order By Clause

- Sum
- Average
- Minimum
- Maximum
- Count

[1]: JT's extra, Aggregate functions allow you to perform calculations on multiple rows of data, but will only return a single value in the response.

Aggregate Functions[1]

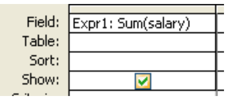
Original table



Employees : Table

SIN	Name	Salary	Department number
111	James Tam	1	1
112	Bart Simpson	24000	1
113	Lisa Simpson	25000	1
114	Peter Griffin	30000	2
115	Heather Morris	50000	3
116	Bruce Lee	75000	4

QBE

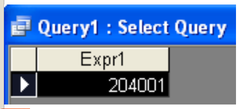


Field: Expr1: Sum(salary)
Table:
Sort:
Show: ☑

SQL

SELECT Sum(salary) AS Expr1
FROM Employees;

Query result

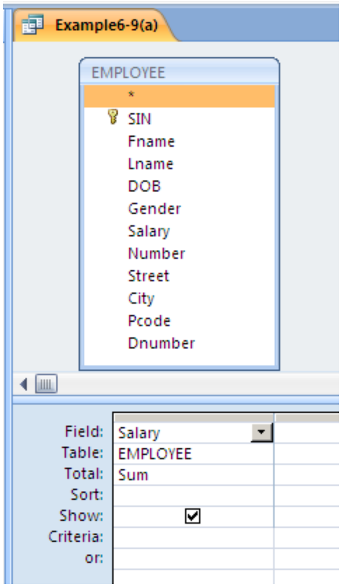


Query1 : Select Query

Expr1
204001

JT's Extra: SUM

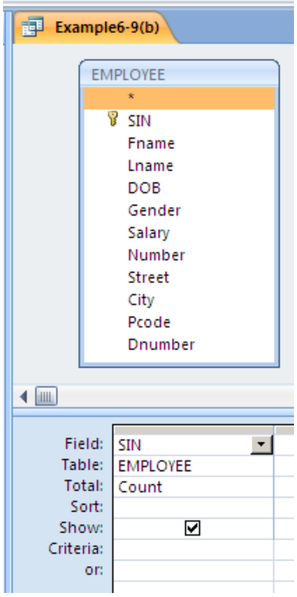
```
SELECT   SUM(Salary)
FROM     EMPLOYEE
```



Field: Salary
Table: EMPLOYEE
Total: Sum
Sort:
Show: ☑
Criteria:
or:

Sum

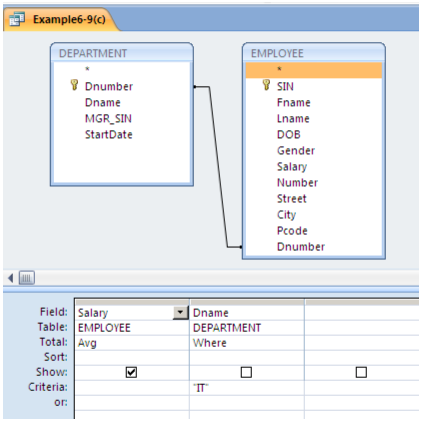
```
SELECT   COUNT(SIN)
FROM     EMPLOYEE
```



Field:	SIN
Table:	EMPLOYEE
Total:	Count
Sort:	
Show:	☑
Criteria:	
or:	

COUNT

```
SELECT   AVG(Salary)
FROM     EMPLOYEE, DEPARTMENT
WHERE    EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
         AND Dname = ‘IT’
```



Field:	Salary	Dname	
Table:	EMPLOYEE	DEPARTMENT	
Total:	Avg	Where	
Sort:			
Show:	☑	□	□
Criteria:		'IT'	
or:			

AVG

- Retrieve the min salary of female employees who work for the Finance department
- Use function Min()

EMPLOYEE

SIN	Fname	Lname	DOB	Gender	Salary	Number	Street	City	Pcode	Dnumber
171717171	Debra	Beacon	15-Aug-1961	Female	70000	15	Baron Hill	Calgary	T2X Y0Y	1
181817178	Sam	Field	17-Feb-1978	Male	40000	15	Kick Way	Calgary	Y2K K0K	1
123456789	Rajeet	Folk	30-Apr-1967	Male	78000	123	One Road	Toronto	H1H J9J	2
987654321	Marie	Band	12-Jan-1985	Female	53500	2828	Exit Close	Toronto	K8O O8K	2
666333999	Saleh	Dice	25-Mar-1970	Male	90400	66	Straight Way	Toronto	T4E T6B	3

DEPARTMENT

Dnumber	Dname	MGR_SIN	StartDate
1	IT	171717171	12-Feb-2008
2	Finance	123456789	1-Mar-2002
3	Marketing	666333999	1-Jan-2005

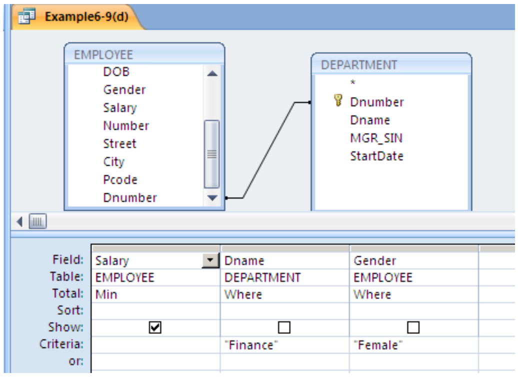
PROJ_EMP

SIN	Pnumber	Hours
171717171	1	15
171717171	2	20
171717171	4	5
181817178	1	30
181817178	2	10
123456789	3	40
666333999	4	40

PROJECT

Pnumber	Pname	Location	Dnumber
1	Web Shopping	Calgary	1
2	Network Upgrade	Calgary	1
3	New Benefits	Toronto	2
4	Product XT345	Toronto	3

```
SELECT  MIN(Salary)
FROM    EMPLOYEE, DEPARTMENT
WHERE   EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
        AND Dname = ‘Finance’
        AND Gender = ‘Female’
```



Field:	Salary	Dname	Gender
Table:	EMPLOYEE	DEPARTMENT	EMPLOYEE
Total:	Min	Where	Where
Sort:			
Show:	☑	☐	☐
Criteria:		"Finance"	"Female"
or:			

MIN

- Retrieve the max salary of employees who work on projects located in Toronto
- Use function Max()

EMPLOYEE

SIN	Fname	Lname	DOB	Gender	Salary	Number	Street	City	Pcode	Dnumber
171717171	Debra	Beacon	15-Aug-1961	Female	70000	15	Baron Hill	Calgary	T2X Y0Y	1
181817178	Sam	Field	17-Feb-1978	Male	40000	15	Kick Way	Calgary	Y2K K0K	1
123456789	Rajeet	Folk	30-Apr-1967	Male	78000	123	One Road	Toronto	H1H J9J	2
987654321	Marie	Band	12-Jan-1985	Female	53500	2828	Exit Close	Toronto	K8O O8K	2
666333999	Saleh	Dice	25-Mar-1970	Male	90400	66	Straight Way	Toronto	T4E T6B	3

DEPARTMENT

Dnumber	Dname	MGR_SIN	StartDate
1	IT	171717171	12-Feb-2008
2	Finance	123456789	1-Mar-2002
3	Marketing	666333999	1-Jan-2005

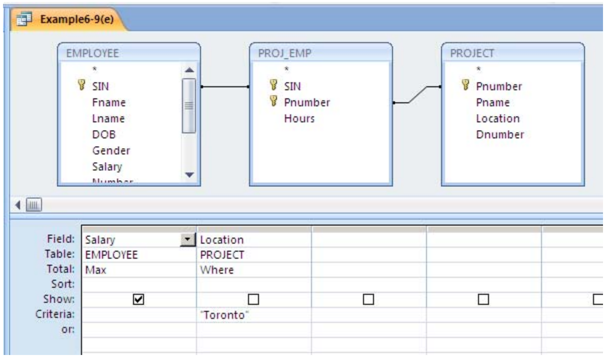
PROJ_EMP

SIN	Pnumber	Hours
171717171	1	15
171717171	2	20
171717171	4	5
181817178	1	30
181817178	2	10
123456789	3	40
666333999	4	40

PROJECT

Pnumber	Pname	Location	Dnumber
1	Web Shopping	Calgary	1
2	Network Upgrade	Calgary	1
3	New Benefits	Toronto	2
4	Product XT345	Toronto	3

```
SELECT   MAX (Salary)
FROM     EMPLOYEE, EMP_PROJ, PROJECT
WHERE    EMPLOYEE.SIN = EMP_PROJ.SIN
         AND PROJECT.Pnumber = EMP_PROJ.Pnumber
         AND Location = 'Toronto'
```



Field:	Salary	Location			
Table:	EMPLOYEE	PROJECT			
Total:	Max	Where			
Sort:					
Show:	☑	□	□	□	□
Criteria:		"Toronto"			
or:					

MAX

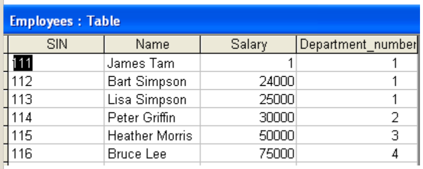
Grouping Calculations

- How to find the sum of salary ***per*** *department*

- SQL has the GROUP BY clause
- Groups calculations of an aggregate function

JT’s Extra: Group By (Small Example)

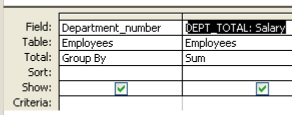
Employees table



Employees : Table

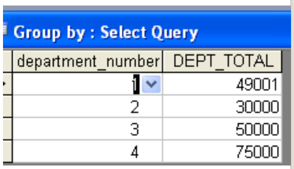
SIN	Name	Salary	Department_number
111	James Tam	1	1
112	Bart Simpson	24000	1
113	Lisa Simpson	25000	1
114	Peter Griffin	30000	2
115	Heather Morris	50000	3
116	Bruce Lee	75000	4

QBE



Field:	Department_number	DEPT_TOTAL: Salary
Table:	Employees	Employees
Total:	Group By	Sum
Sort:		
Show:	☑	☑
Criteria:		

Query result



Group by : Select Query

department_number	DEPT_TOTAL
1	49001
2	30000
3	50000
4	75000

SQL

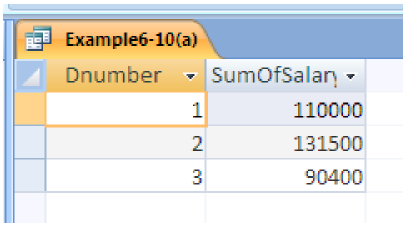
SELECT department_number, sum(salary) AS DEPT_TOTAL

FROM Employees

GROUP BY department_number;

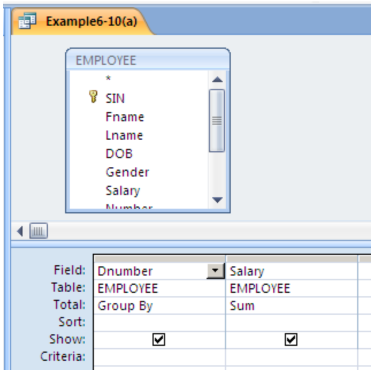
GROUP BY

```
SELECT      Dnumber, SUM(Salary)
FROM        EMPLOYEE
GROUP BY    Dnumber
```



Example6-10(a)

Dnumber	SumOfSalary
1	110000
2	131500
3	90400



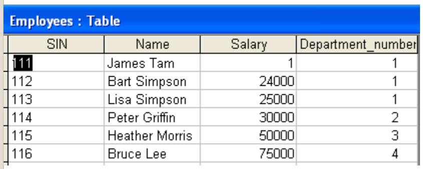
Example6-10(a)

Field:	Dnumber	Salary
Table:	EMPLOYEE	EMPLOYEE
Total:	Group By	Sum
Sort:		
Show:	☑	☑
Criteria:		

HAVING Clause

- To show only some of the groups
- WHERE filters tuples
- HAVING filters groups

Employees table



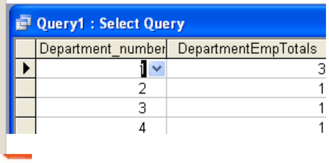
Employees : Table

SIN	Name	Salary	Department_number
111	James Tam	1	1
112	Bart Simpson	24000	1
113	Lisa Simpson	25000	1
114	Peter Griffin	30000	2
115	Heather Morris	50000	3
116	Bruce Lee	75000	4

SQL

```
SELECT Department_number, COUNT(*)  as DepartmentEmpTotals
FROM Employees
GROUP BY Department_number;
```

Query result

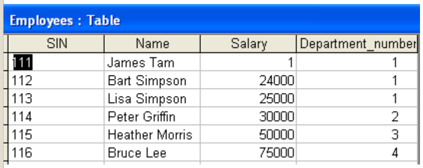


Query1 : Select Query

Department_number	DepartmentEmpTotals
1	3
2	1
3	1
4	1

JT’s Extra: Group By (All Departments)

Employees table



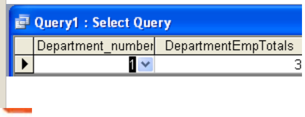
Employees : Table

SIN	Name	Salary	Department_number
111	James Tam	1	1
112	Bart Simpson	24000	1
113	Lisa Simpson	25000	1
114	Peter Griffin	30000	2
115	Heather Morris	50000	3
116	Bruce Lee	75000	4

SQL

```
SELECT Department_number, COUNT(*)  as DepartmentEmpTotals
FROM Employees
GROUP BY Department_number
HAVING COUNT(*)>1
```

Query result

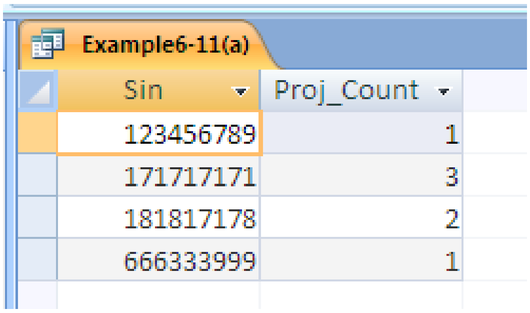


Query1 : Select Query

Department_number	DepartmentEmpTotals
1	3

JT’s Extra: Having Clause (Bigger Dept.’s)

```
SELECT      SIN, COUNT(*)
FROM        EMP_PROJ
GROUP BY    SIN
```

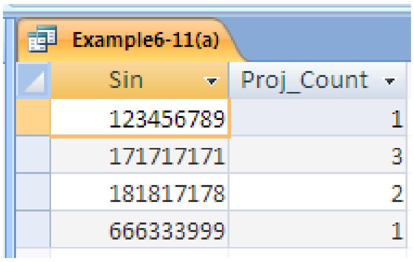


Example6-11(a)

Sin	Proj_Count
123456789	1
171717171	3
181817178	2
666333999	1

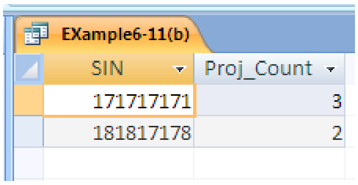
HAVING Clause

```
SELECT      SIN, COUNT(*)
FROM        EMP_PROJ
GROUP BY    SIN
HAVING      COUNT(*) > 1
```



Example6-11(a)

Sin	Proj_Count
123456789	1
171717171	3
181817178	2
666333999	1



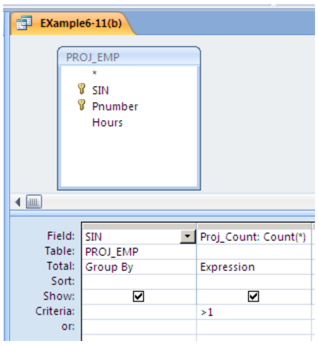
EXample6-11(b)

SIN	Proj_Count
171717171	3
181817178	2

HAVING Clause

HAVING Clause

```
SELECT      SIN, COUNT(*)
FROM        EMP_PROJ
GROUP BY    SIN
HAVING      COUNT(*) > 1
```



EXample6-11(b)

PROJ_EMP
- *
- SIN
- Pnumber
- Hours

Field:	SIN	Proj_Count: Count(*)
Table:	PROJ_EMP	
Total:	Group By	Expression
Sort:		
Show:	☑	☑
Criteria:		>1
or:		

EXISTS – Optional Reading

```
SELECT   *
FROM     DEPARTMENT
WHERE    NOT EXISTS (
                      SELECT   *
                      FROM     EMPLOYEE
                      WHERE    EMPLOYEE.Dnumber = DEPARTMENT.Dnumber
                               AND Gender != 'Female'
                      )
```