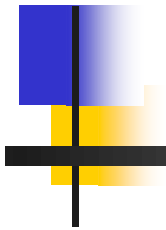
SQL Part II

IOE 373 Lecture 05

Review

- SELECT statement

  SELECT [Fields] FROM [Table(s)] WHERE [Conditions]
- ORDER BY
- MAX(), MIN(), AVG(), COUNT()
- GROUP BY
- Other online references for SQL (and Python):
  - https://www.w3schools.com/sql/default.asp
  - https://stackoverflow.com/



Topics

- INSERT
- UPDATE
- DELETE
- Queries on multiple tables

INSERT Statement

- INSERT is used to add rows into a table
- Syntax:

  INSERT INTO table (field1, field2, …) **VALUES** (value1, value2, …)
- Example:

  INSERT INTO Students (UMID, SSN, Name, Email) **VALUES** (37339830, 334332190, 'Joseph Woods', 'joew@umich.edu')

UMID*	SSN	Name	Email
37339822	344021945	Edward Jones	edjones@umich.edu
37339823	342122843	Steven Hanks	shanks@umich.edu
37339824	564231347	Edward Jones	edjones2@umich.edu
37339829	473293828	Edward Jones	edwardj@umich.edu
37339830	334332190	Joseph Woods	joew@umich.edu

New Row →

INSERT Statement

- You can omit some fields
- But not for field(s) that belong to primary key!

INSERT INTO Students (UMID, Name) **VALUES** (37339830, 'Joseph Woods')

- This is equivalent to

INSERT INTO Students (UMID, SSN, Name, Email) **VALUES** (37339830, NULL, 'Joseph Woods', NULL)

UMID*	SSN	Name	Email
37339822	344-02-1945	Edward Jones	edjones@umich.edu
37339823	342-12-2843	Steven Hanks	shanks@umich.edu
37339824	564-23-1347	Edward Jones	edjones2@umich.edu
37339829	473-29-3828	Edward Jones	edwardj@umich.edu
37339830		Joseph Woods	

New Row →

INSERT Statement

- You can omit the list of field names,
- But in this case, you have to include a value for every field in the order defined by the table
- For example

INSERT INTO Students
VALUES (37339830, NULL, 'Joseph Woods', NULL)

- Is equivalent to the query in previous slide

UPDATE Statement

- The UPDATE statement modifies a record in a table
- Syntax:

UPDATE table **SET** field1=value1, field2=value2, … **WHERE** conditions

- Warning: UPDATE queries can change lots of data very quickly, and the results may be irreversible!

UPDATE Example

- Update a record in Student table:

  UPDATE Students **SET**
  Name='Joseph Woodson', SSN='333-23-3444'
  WHERE UMID=37339830

- Result:

UMID*	SSN	Name	Email
37339822	344-02-1945	Edward Jones	edjones@umich.edu
37339823	342-12-2843	Steven Hanks	shanks@umich.edu
37339824	564-23-1347	Edward Jones	edjones2@umich.edu
37339829	473-29-3828	Edward Jones	edwardj@umich.edu
37339830	333-23-3444	Joseph Woodson	

- Without a **WHERE** clause, **UPDATE** commands will change ALL records in a table!

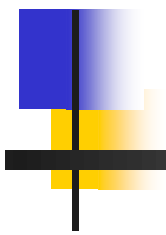
  UPDATE Students SET Name='Amy'

  will make EVERY student have the name 'Amy'

- Whereas

  UPDATE Students SET Name='Amy' WHERE UMID=44403200

  Will only change the name of one student



More Examples

- Add 5 points to everyone's grade

  UPDATE Grades **SET** Score = Score + 5
- Increase the grade by 4% for all students with grade less than 60
- **UPDATE** Grades **SET** Score = Score*1.04 **WHERE** Score < 60

DELETE Statement

- The DELETE statement removes a record from a table
- Syntax:

  DELETE * FROM table **WHERE** conditions
- Warning: DELETE queries can change lots of data very quickly, and the results may be irreversible!

DELETE Example

- Delete a student with UMID 37339824

  DELETE * FROM Students **WHERE** UMID=37339824

- This will delete only one record
- But if you use

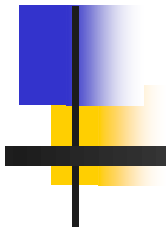
  DELETE * FROM Students **WHERE** Name='Edward Jones'

- This will delete all students named 'Edward Jones'

UMID*	SSN	Name	Email
37339822	344-02-1945	Edward Jones	edjones@umich.edu
37339823	342-12-2843	Steven Hanks	shanks@umich.edu
37339824	564-23-1347	Edward Jones	edjones2@umich.edu
37339829	473-29-3828	Edward Jones	edwardj@umich.edu
37339830	333-23-3444	Joseph Woodson	

DELETE Statement

- Use WHERE clause
  - This will remove all students records:
    DELETE * FROM Students
- Use primary key as condition if possible



Query On Multiple Tables

- Now we know how to select data from one table.
- But sometimes we need to get data from multiple tables
- Use **INNER JOIN** clause to “join” two or more tables

Example of Two Tables Query

- Suppose we want to list employees' Name, Email and Department Name with salary >=3000

Employees

PersonID	PersonName	Address	Email	Phone	Salary	DeptID
1	Steve	1234 Fuller	steve@notexist.com	734-333-9999	2000	2
2	John	234 Huron St.	john@notexist.com	734-233-8777	3000	1
3	Mary	2489 Stone Road	mary@notexist.com	734-876-8888	4000	1
4	Emily	1254 Green road	emily@notexist.com	734-233-9089	2500	2
5	Mike	333 Fifth	mike@notexist.com	734-344-0934	2900	1
6	James	255 Plymouth	james@notexist.com	734-3 4000		

Departments

DeptID	DeptName	DeptPhone	ManagerID
1	Finance	222-222-2233	2
2	Marketing	222-222-2333	1
3	IT	222-222-2345	6

Example

- Basically you need to match up records in two tables with same DeptID

Employees

PersonID	PersonName	Address	Email	Phone	Salary	DeptID
1	Steve	1234 Fuller	steve@notexist.com	734-333-9999	2000	2
2	John	234 Huron St.	john@notexist.com	734-233-8777	3000	1
3	Mary	2489 Stone Road	mary@notexist.com	734-876-8888	4000	1
4	Emily	1254 Green road	emily@notexist.com	734-233-9089		
5	Mike	333 Fifth	mike@notexist.com	734-344-0934		
6	James	255 Plymouth	james@notexist.com	734-333-4000		

Departments

DeptID	DeptName	DeptPhone	Manager ID
1	Finance	222-222-2233	2
2	Marketing	222-222-2333	1
3	IT	222-222-2345	6

- SELECT Employees.PersonName, Employees.Email, Departments.DepartmentName

FROM Employees, Departments

WHERE Employees.DeptID=Departments.DeptID

AND

Employees.Salary>=3000

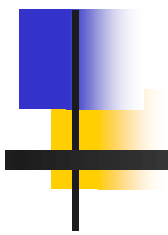
Example

- Syntax:

```
SELECT Employees.PersonName,
Employees.Email,
      Departments.DeptName
FROM Employees INNER JOIN Departments
ON Employees.DeptID = Departments.DeptID
WHERE Employees.Salary >= 3000
```

Query Result

PersonName	Email	DeptName
John	john@notexist.com	Finance
Mary	mary@notexist.com	Finance
James	james@notexist.com	HumanResources

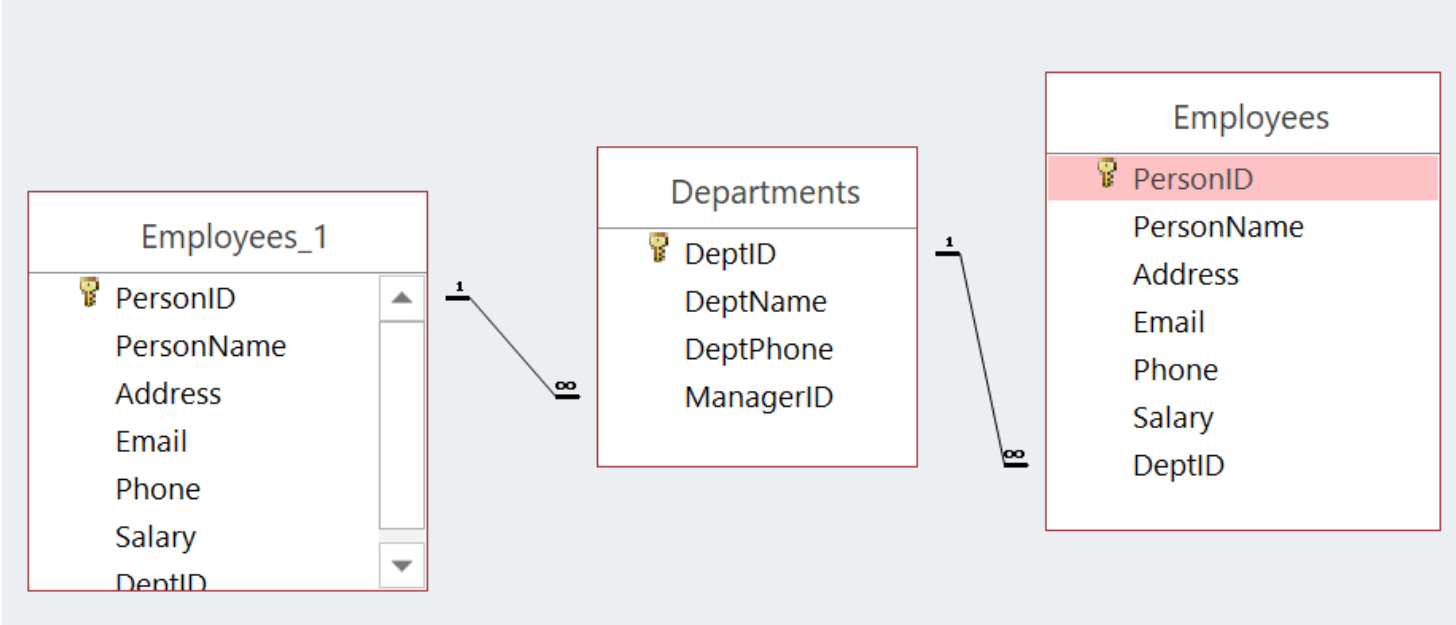


Example

- Since an INNER JOIN query returns fields from both tables, you must specify which table each field comes from.
- You do this by prefixing each field name with the name of the table it comes from, followed by a period:

  Employees.DeptID, Departments.DeptID

What if I want the name of the manager also?

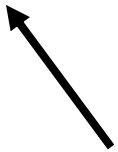


What if I want the name of the manager also?

SELECT Employees.PersonName, Employees.Email, Departments.DeptName,

What if I want the name of the manager also?

SELECT Employees.PersonName, Employees.Email, Departments.DeptName, **(Select Employees.PersonName from Employees Where Employees.PersonID=Departments.ManagerID) AS ManagerName**

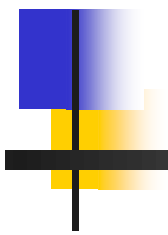


I need to figure out the name of the manager based on the managerID

This expression in parenthesis is also known as a **subquery**

What if I want the name of the manager also?

```
SELECT Employees.PersonName,
Employees.Email, Departments.DeptName,
(Select Employees.PersonName from Employees
Where
Employees.PersonID=Departments.ManagerID)
AS ManagerName
FROM Employees INNER JOIN Departments ON
Employees.DeptID = Departments.DeptID
WHERE Salary >=3000;
```



Example

- Using **INNER JOIN** is preferable (more efficient) to using **WHERE** to join two tables:

```
SELECT Employees.PersonName, Employees.Email,
    Departments.DeptName
FROM Employees, Departments
WHERE Employees.DeptID = Departments.DeptID
  AND Employees.Salary >= 3000
```

Although this query will give the same result, it is much slower in most database systems

Example

- You can use INNER JOIN together with other clauses:

```
SELECT Departments.DeptName,
AVG(Employees.Salary) AS AverageSalary
FROM Employees INNER JOIN Departments
ON Employees.DeptID = Departments.DeptID
GROUP BY Departments.DeptName
```

DeptName	AverageSalary
Finance	3300
HumanResources	4500
Marketing	2250

Other Ways of Joining Tables

- Sometimes INNER JOIN is not enough
- What if there are "mismatches" between two tables?

Ordinals

Number	OrdinalValue
1	First
2	Second
3	Third
4	Fourth

SpelledValues

Number	SpelledValue
1	One
2	Two
5	Five
6	Six

Joining Tables

- INNER JOIN can only join records with corresponding field:

SELECT * **FROM** Ordinals **INNER JOIN** SpelledValues
ON Ordinals.Number = SpelledValues.Number
ORDER BY Ordinals.Number

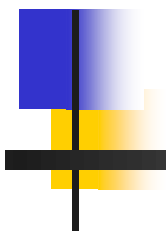
Ordinals

Number	OrdinalValue
1	First
2	Second
3	Third
4	Fourth

SpelledValues

Number	SpelledValue
1	One
2	Two
5	Five
6	Six

Number	OrdinalValue	SpelledValue
1	First	One
2	Second	Two



Other Ways of Joining Tables

- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

LEFT JOIN

- **LEFT JOIN** select all records from the left table even if there are no matching records in the right table

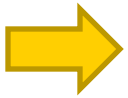
SELECT * **FROM** Ordinals **LEFT JOIN** SpelledValues
ON Ordinals.Number = SpelledValues.Number
ORDER BY Ordinals.Number

Ordinals

Number	OrdinalValue
1	First
2	Second
3	Third
4	Fourth

SpelledValues

Number	SpelledValue
1	One
2	Two
5	Five
6	Six



Number	Ordinal Value	Spelled Value
1	First	One
2	Second	Two
3	Third	NULL
4	Fourth	NULL

RIGHT JOIN

- **RIGHT JOIN** select all records from the right table even if there are no matching records in the left table

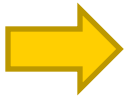
SELECT * **FROM** Ordinals **RIGHT JOIN** SpelledValues
ON Ordinals.Number = SpelledValues.Number
ORDER BY Ordinals.Number

Ordinals

Number	OrdinalValue
1	First
2	Second
3	Third
4	Fourth

SpelledValues

Number	SpelledValue
1	One
2	Two
5	Five
6	Six



Number	Ordinal Value	Spelled Value
1	First	One
2	Second	Two
5	NULL	Five
6	NULL	Six

FULL JOIN (not in access)

- **FULL JOIN** select all records from both tables even if there are no matching records in the other table

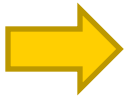
SELECT * **FROM** Ordinals **FULL JOIN** SpelledValues
ON Ordinals.Number = SpelledValues.Number
ORDER BY Ordinals.NumberID

Ordinals

Number ID	OrdinalValue
1	First
2	Second
3	Third
4	Fourth

SpelledValues

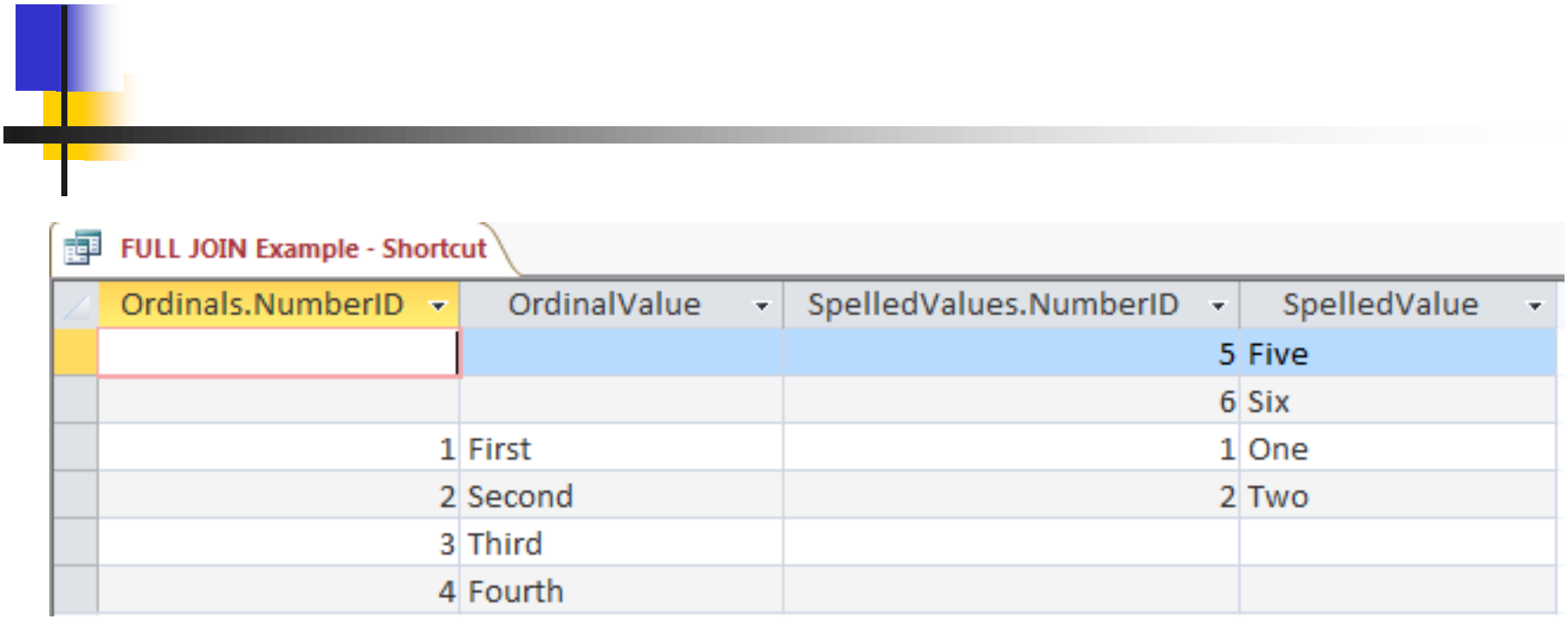
Number ID	SpelledValue
1	One
2	Two
5	Five
6	Six



Number ID	Ordinal Value	Spelled Value
1	First	One
2	Second	Two
3	Third	NULL
4	Fourth	NULL
5	NULL	Five
6	NULL	Six

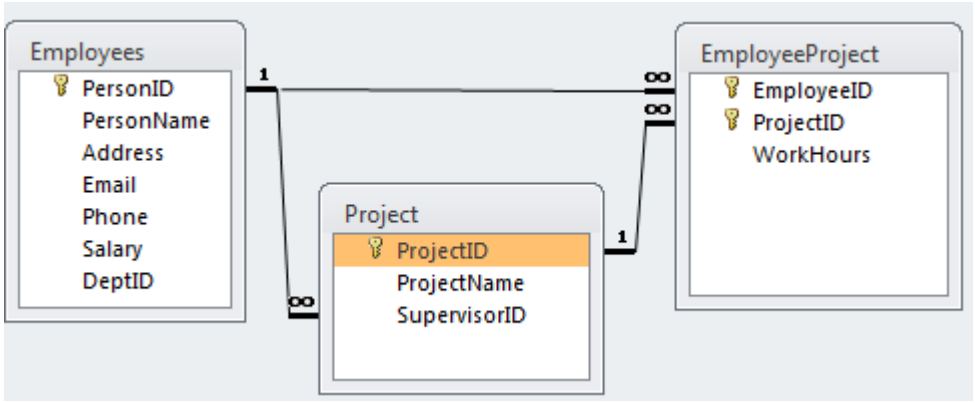
Equivalent expression (for Access)

```
SELECT Ordinals.*, SpelledValues.*
FROM Ordinals LEFT JOIN SpelledValues ON
Ordinals.NumberID = SpelledValues.NumberID
UNION
SELECT Ordinals.*, SpelledValues.*
FROM SpelledValues LEFT JOIN Ordinals ON
SpelledValues.NumberID = Ordinals.NumberID
ORDER BY Ordinals.NumberID;
```


FULL JOIN Example - Shortcut
Ordinals.NumberID
OrdinalValue
SpelledValues.NumberID
SpelledValue
5 Five
6 Six
1 First
1 One
2 Second
2 Two
3 Third
4 Fourth

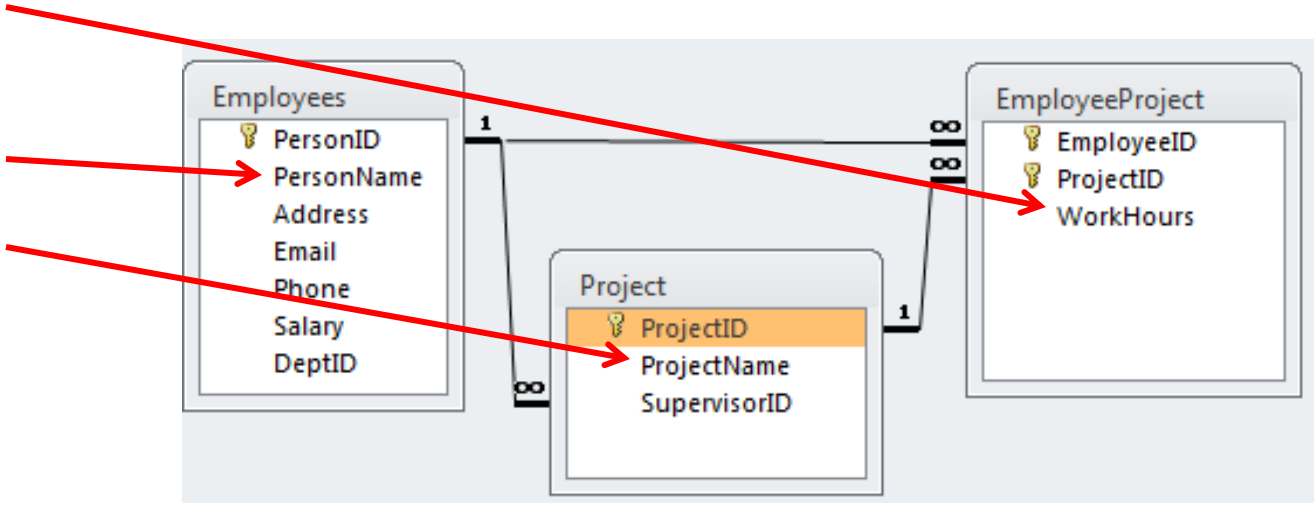
More Complex INNER JOIN

- If we want to select data from *Employees* and *EmployeeProject*, we may need to join 3 tables together



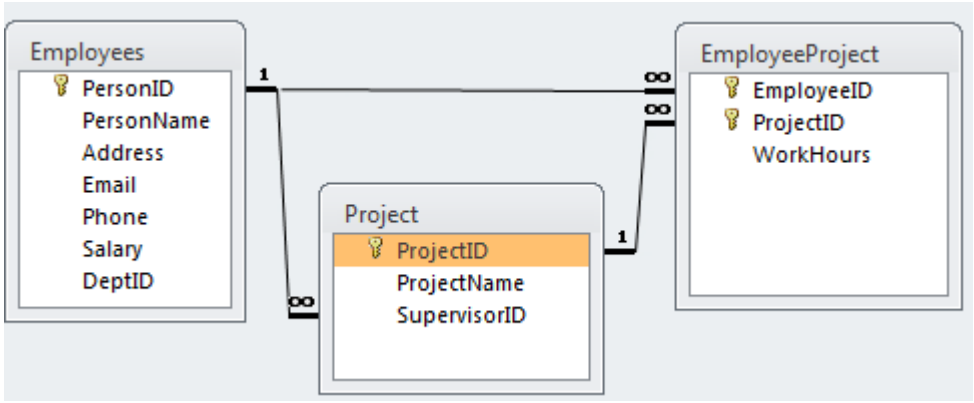
Complex INNER JOIN Example

- If we want to see which employee is working on which project, and how many hours they contribute to the project



Complex INNER JOIN Example

SELECT Employees.PersonName, Project.ProjectName, EmployeeProject.WorkHours
FROM (Employees **INNER JOIN** EmployeeProject
ON
Employees.PersonID=EmployeeProject.EmployeeID)
INNER JOIN Project
ON EmployeeProject.ProjectID=Project.ProjectID



- We first join the *Employees* and *EmployeeProject* table:

  (Employees **INNER JOIN** EmployeeProject
ON
Employees.PersonID=EmployeeProject.EmployeeID)

Then join the result with *Project* table:

  (…) **INNER JOIN** Project
ON EmployeeProject.ProjectID=Project.ProjectID

- Result:

PersonName	ProjectName	WorkHours
Steve	A	5
John	B	4
Mary	C	10
Emily	B	5
Mike	B	3
James	C	2
James	A	8
John	C	7
Mike	A	3