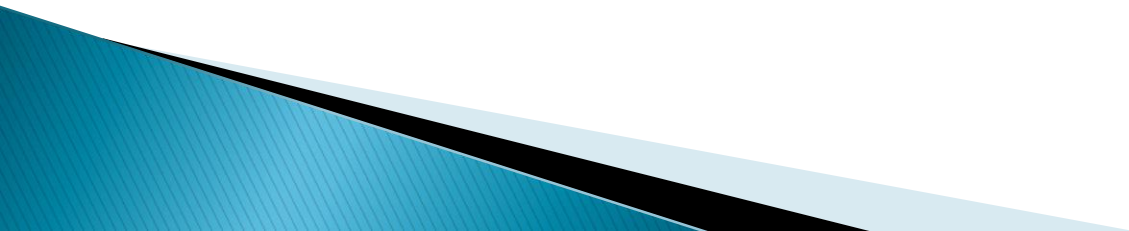# UNION, INTERSECT, EXCEPT, Chapter 6

# SQL - Syntax

SELECT [DISTINCT | ALL]

              {* | colExpression [AS newName]] [,...]}

FROM TableName [alias][,...]

[WHERE condition]

[GROUP BY columnList] [HAVING condition]

[ORDER BY columnList];

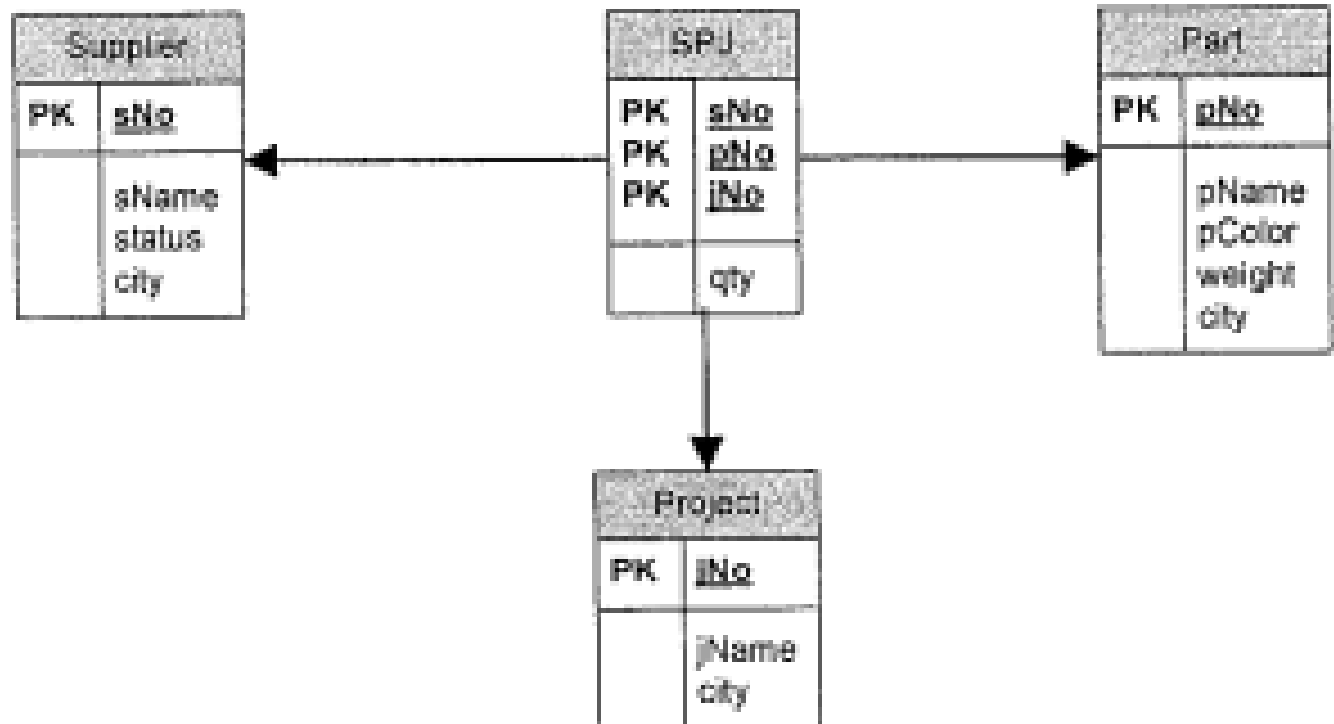UNION, INTERSECT and EXCEPT can be used to connect two queries as above when the queries are "union-compatible"

# Union-Compatible

Two tables, or the result of two queries, are union-compatible if they have the same structure.

- Same number of columns
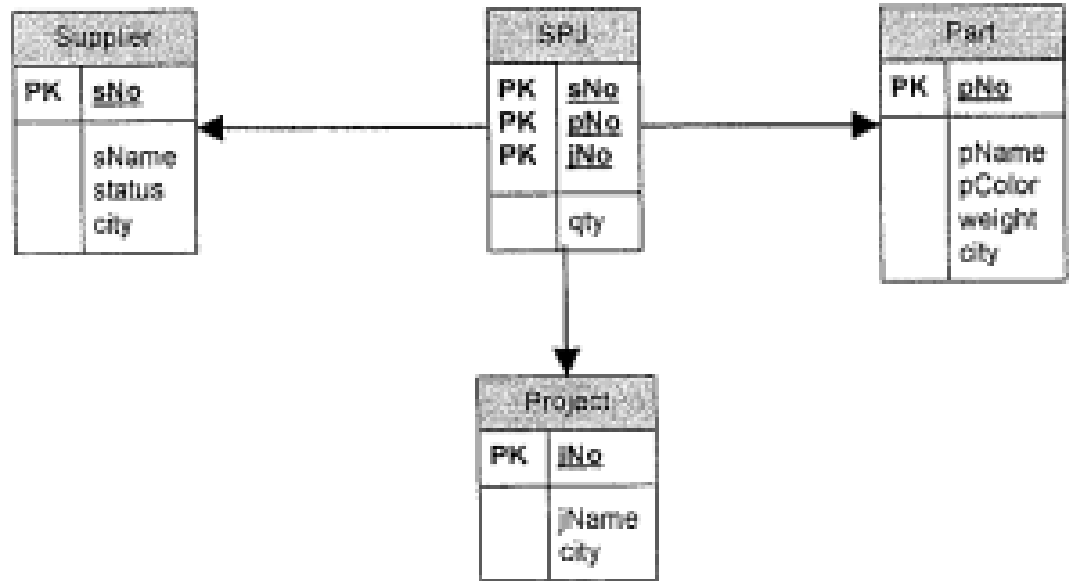- Corresponding columns have the same data types

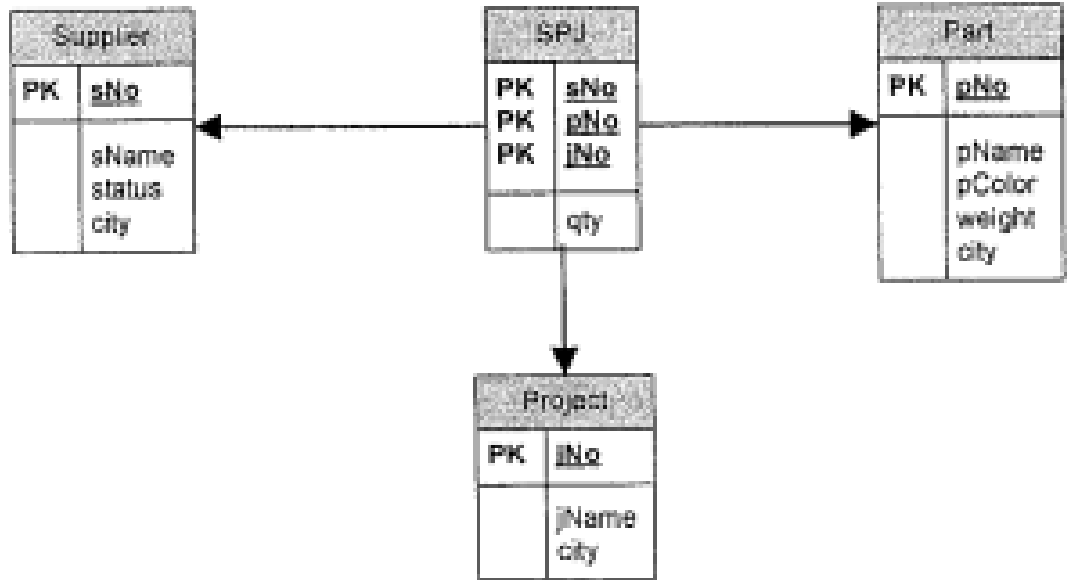# SPJ (Suppliers / Parts / Projects) Database

Suppliers-Parts-Project
Database

# Example

Write a query to list all cities in which part manufacturers and suppliers are located

**Supplier**

| PK | sNo |
|----|-----|
|    | sName |
|    | status |
|    | city |

**SPJ**

| PK | sNo |
|----|-----|
| PK | pNo |
| PK | jNo |
|    | qty |

**Part**

| PK | pNo |
|----|-----|
|    | pName |
|    | pColor |
|    | weight |
|    | city |

**Project**

| PK | jNo |
|----|-----|
|    | jName |
|    | city |

# Example



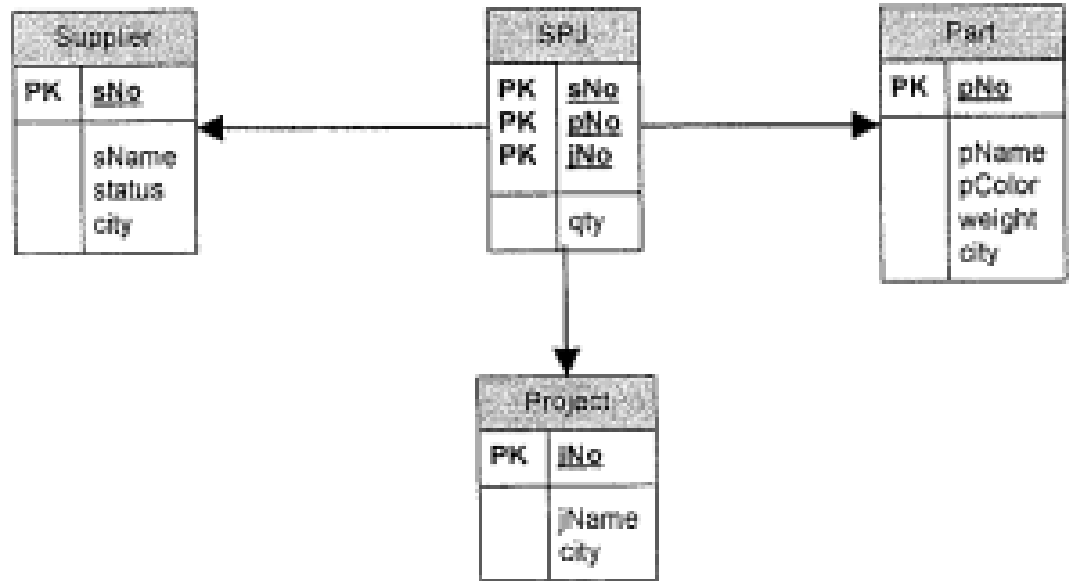Write a query to list all cities in which part suppliers are located:
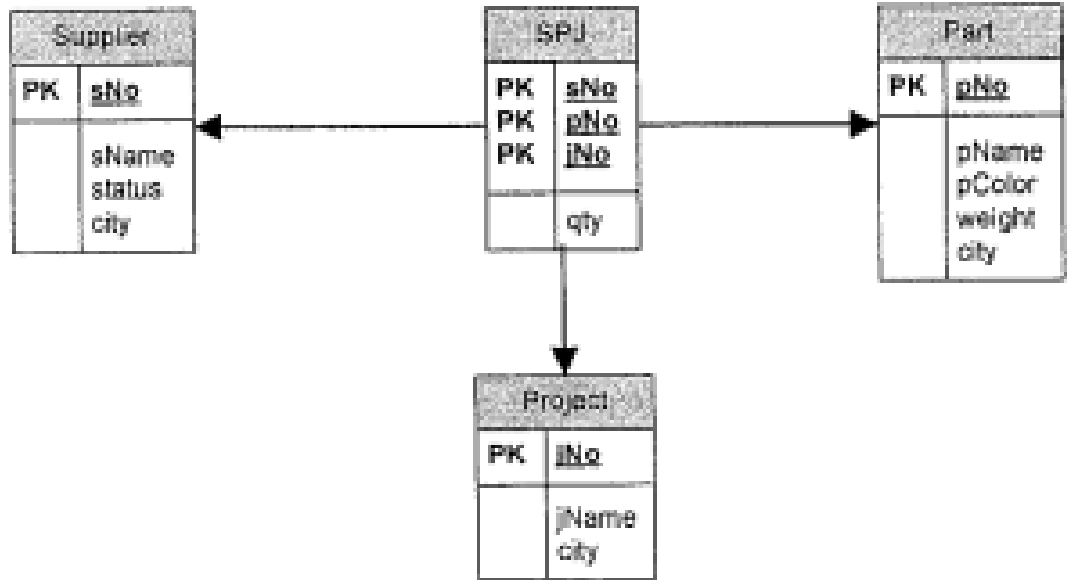
```
SELECT city
FROM P

INTERSECT

SELECT city
FROM S
```

# Example

Write a query to list all cities in which part manufacturers are located, but no supplier is located.

# Example



Write a query to list all cities in which part manufacturers are located, but no supplier is located.

SELECT city
FROM P

EXCEPT
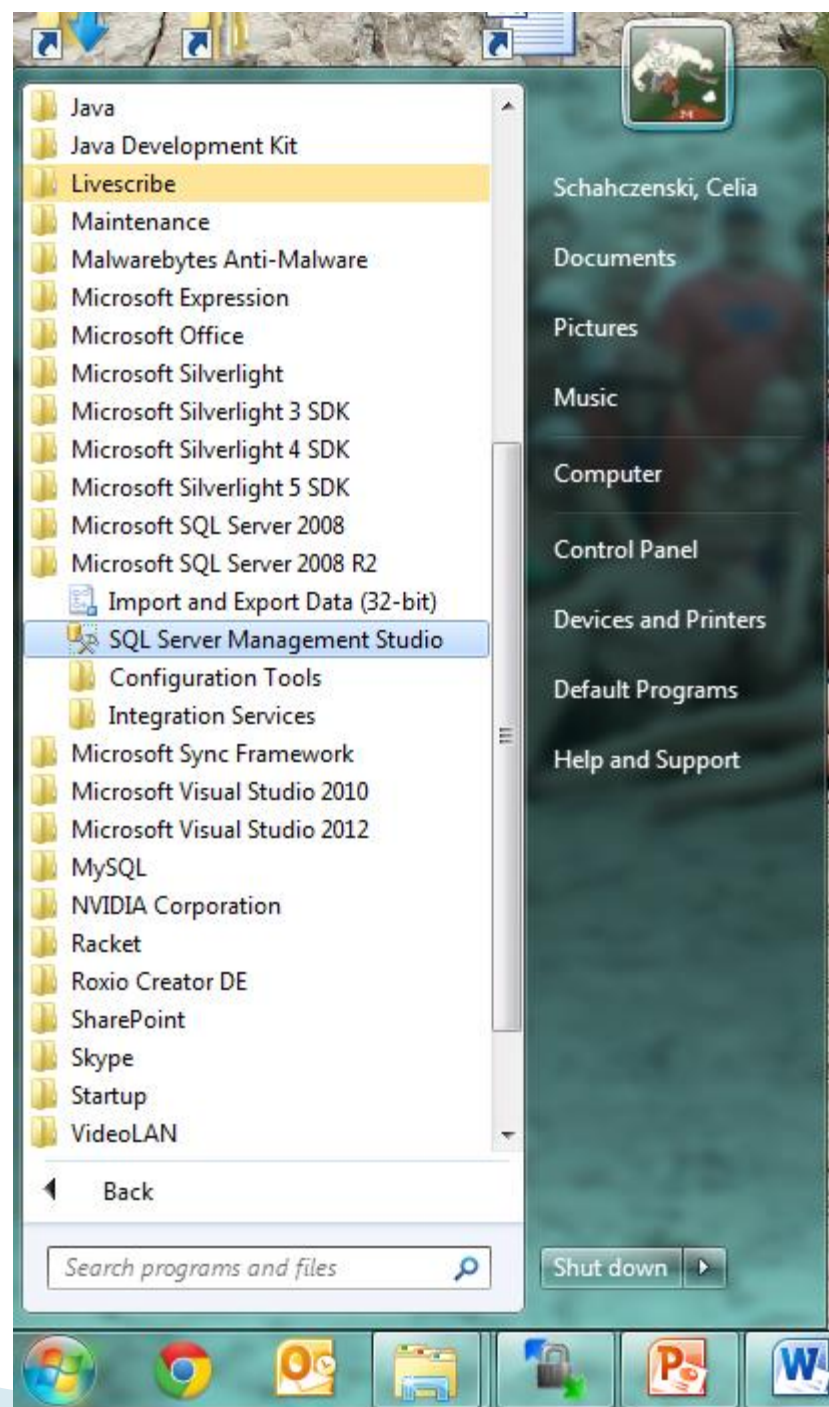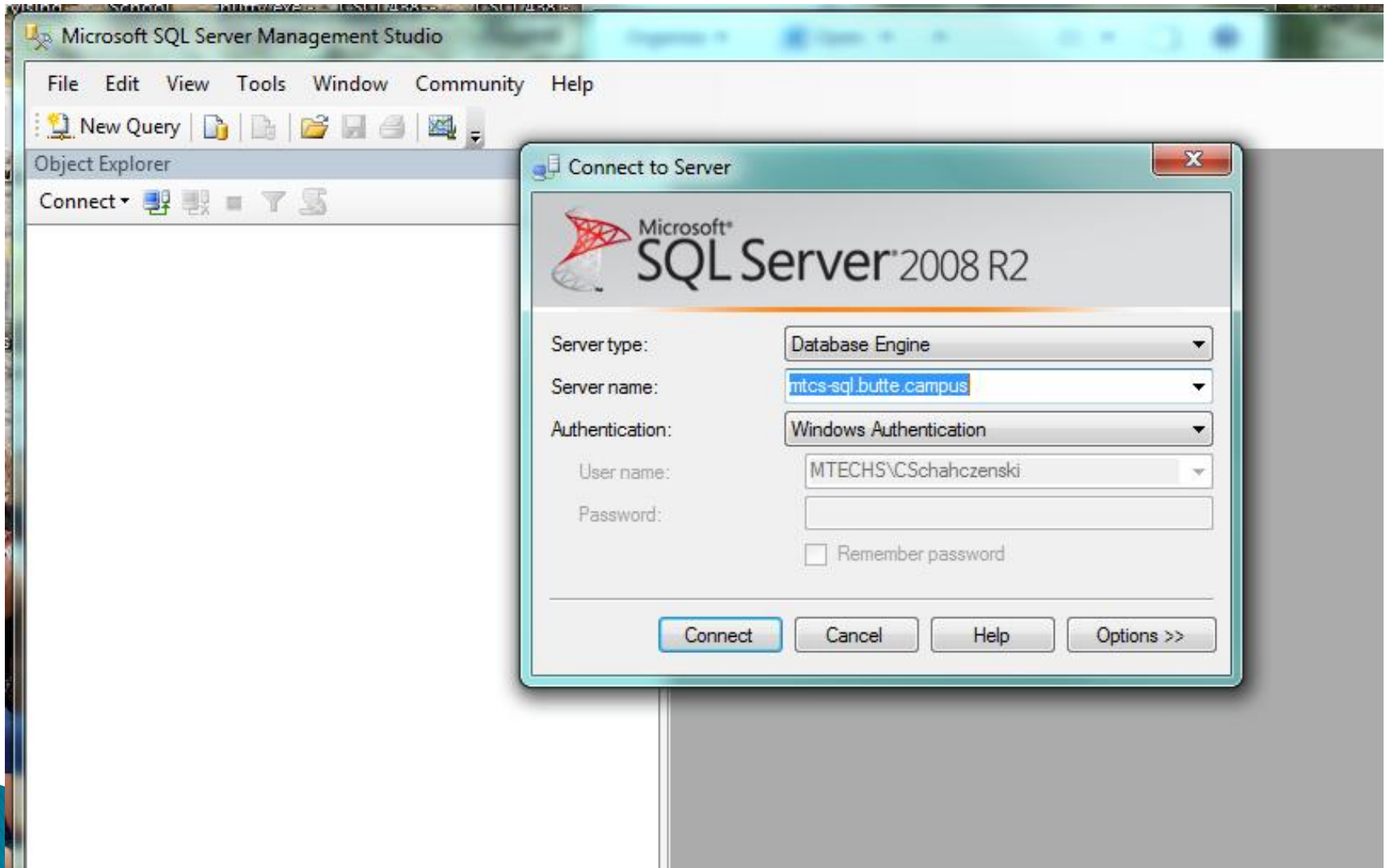
SELECT city
FROM S

# SQL Server (Microsoft)

- Account has been set up
- Windows authentication is used

# Connecting to SQL Server

# Connecting to SQL Server

# Connecting to HotelReservations