Xpath11

C1	C2	C3	C4	C5	C6	C7
X						

Get all teamId of teams with less than or equals 3 employees.
//Team[@employees <= "3"]/@teamid

Xpath12

C1	C2	C3	C4	C5	C6	C7
		X				

Get all teamId of teams which are based in country IT.
//ISO_Code[.="IT"]/ancestor::Team/@teamid

Xpath13

C1	C2	C3	C4	C5	C6	C7
				X		

Get the sum of seats the first team in the XML data has produced.
sum(/Teams/Team[1]/Units/Unit/seatconfig/seat/amount)

Xpath14

C1	C2	C3	C4	C5	C6	C7
X						X

Get the team with the smallest teamId
/Teams/Team/@teamid[not(. > /Teams/Team/@teamid)]

Xpath21

C1	C2	C3	C4	C5	C6	C7
	X	X	X		X	

Union of registration of plane with Testdate of the specificied protocolId.
//Protocol[@protocolid="AKYHEFGNCP"]/ancestor::Protocols/ancestor::Plane/@registration |
//Protocol[@protocolid="AKYHEFGNCP"]/descendant::Testdate

Xpath22

C1	C2	C3	C4	C5	C6	C7
X				X		

Number of planes customer with Id 119 has purchased.
count(/Customers/Customer[@customerid = "119"]/Planes/Plane)

Xpath23

C1	C2	C3	C4	C5	C6	C7
X					X	

Union of customer with Id 119's first and lastname
//Customer[@customerid = "119"]/Firstname | //Customer[@customerid = "119"]/Lastname

Xpath24

C1	C2	C3	C4	C5	C6	C7
		X		X		

Number of times the looping test has been mentioned in all protocols.
count(//test[.="looping"]/ancestor::results/ancestor::Protocol/@protocolid)

Xpath31

C1	C2	C3	C4	C5	C6	C7
X	X					

Get street names in zips that are less than 10000.
//Address[@zip<10000]/descendant::Street/text()

Xpath32

C1	C2	C3	C4	C5	C6	C7
X		X				

Get all customers in zips that are less than 10000.
//Address[@zip<10000]/ancestor::Customer

Xpath33

C1	C2	C3	C4	C5	C6	C7
				X		

Get the total sum of Capacity of toilets in which the Flowrate exceeds 4.0.
sum(//Flowrate[.>4.0]/preceding-sibling::Capacity)

Xpath34

C1	C2	C3	C4	C5	C6	C7
X				X		

Get the amount of planes in which the number of their registration is less than 4000 (Reg format XXX0000).
count(//Plane[number(substring(./@registration,4))<4000])

Xpath41

C1	C2	C3	C4	C5	C6	C7
	X					

Get all bar unitid of first plane.
//Plane[1]/Bars/descendant::Bar/@unitid

Xpath42

C1	C2	C3	C4	C5	C6	C7
	X	X	X		X	

Union of all planes that are either white or have a bar that has less than 6 minifridges.
//Color[.="white"]//ancestor::Plane/@registration | //Plane[descendant::Minifridges/Amount[.<6]]/@registration

Xpath43

C1	C2	C3	C4	C5	C6	C7
	X			X		

Get all registrations of planes with more than 3 bars.
//Plane[count(descendant::Bar)>3]/@registration

Xpath44

C1	C2	C3	C4	C5	C6	C7
						X

Cost of the most expensive drink.
//Bar/Beverages/Beverage/Cost[not(. < //Cost)]

Xpath	C1	C2	C3	C4	C5	C6	C7
11	X						
12			X				
13					X		
14							X
21		X	X	X		X	
22	X				X		
23	X					X	
24			X		X		
31	X	X					
32	X		X				
33					X		
34	X				X		
41		X					
42		X	X	X		X	
43		X			X		
44							X
SUM	6	5	5	2	6	3	2