

# XQueries Presentation

**Abdelaziz Abushark 20332134**

**Ali Al Ani 19314029**

**Saif Ali 203332469**

**Noel Austin 20333159**

**Adam Beatty 20332203**

**Anastasiya Bogoslovskaya 20332234**

<b>Query status table</b>	<b>2</b>
Bank:	4
DTD document:	4
XML document:	5
xQueries:	6
1st xQuery:	6
2nd xQuery:	6
Just_Eat:	7
DTD document:	7
XML document:	7
xQueries:	8
1st xQuery:	8
2nd xQuery:	8
Finder:	8
DTD document:	8
XML document:	9
xQueries:	9
1st xQuery:	9
Customer:	10
DTD Document:	10
XML document:	11
xQueries:	12
1st xQuery:	12
2nd xQuery:	12
Order:	13

DTD Document:	13
XML Document:	13
xQueries:	14
1st xQuery:	14
2nd xQuery:	14
Payment:	15
DTD Document:	15
XML Document:	15
xQueries:	16
1st xQuery:	16
2nd xQuery:	16
3rd xQuery	16
Settings:	17
DTD Document:	17
XML Document:	17
xQueries:	18
1st xQuery:	18
2nd xQuery:	18

## Query status table

Query	Related Use Case	For	Let	Interlinked	Built in function	User Defined function	Working Yes/No
1 Bank details	Bank	X					Yes
2 Bank Payment details	Payment	X					Yes
3 Just Eat information	Just Eat	X		X			Yes

4 Just Eat information	Just Eat	X		X			Yes
5 Finder for restaurant and food	Search	X		X			Yes
6 Settings to check if phone is valid	Settings		X		X		Yes
7 Settings to check if email is valid	Settings		X		X		Yes
8 Customer phone number	Customer	X				X	Yes

9 Customers total bill amount for orders	Customers	X				X	Yes
10 Payment types	Payment	X	X		X		Yes
11 Payment sum	Payment	X	X		X		Yes

--	--	--	--	--	--	--	--

12 order type	Order		X				Yes
13 order number	Order		X		X		Yes

14 payment max	Payment	X	X				Yes
----------------------	---------	---	---	--	--	--	-----

---

Bank:

DTD document:

```

<?xml version="1.0" ?>
<!DOCTYPE doc [
<!ELEMENT doc (payment* , bank*)>

<!ELEMENT payment ( bank*,amount? , type+)>
<!ELEMENT bank ( number*, name?, type+)>
<!ATTLIST doc type CDATA #REQUIRED >
<!ATTLIST doc xml:lang CDATA #IMPLIED >
<!ATTLIST bank name CDATA #IMPLIED >

<!ELEMENT amount (#PCDATA) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT type (#PCDATA) >

```

<!ELEMENT number (#PCDATA) >

] >

---

## XML document:

```
<doc type="bank" xml:lang="en">
  <payment>
    <bank name="Allied Irish Banks">
      <number>20331234</number>
      <name>Folabi</name>
      <type>Student account</type>
    </bank>
    <bank name="Bank Of Ireland">
      <number>20554553</number>
      <name>Mousa</name>
      <type>Business account</type>
    </bank>
    <bank name="Permenant TSB">
      <number>20784553</number>
      <name>Anton</name>
      <type>Personal account</type>
    </bank>
  </payment>
  <payment>
    <amount>1000</amount>
    <type>Apple pay</type>
  </payment>
  <payment>
    <amount>12.95</amount>
    <type>Cash</type>
  </payment>
  <payment>
    <amount>43.65</amount>
    <type>card</type>
  </payment>
</doc>
```

---

## xQueries:

### 1st xQuery:

```
for $x in doc("bankDone.xml")/doc/payment/bank
```

```
return
```

```
<bank_details>
```

```
<bank>
```

```
{ string($x/name )}
```

```
</bank>
```

```
<number>
```

```
{ string($x/number) }
```

```
</number>
```

```
<type>
```

Add Headings (Format > Paragraph styles) and they will be displayed in your table of contents.

```
{ string($x/type) }
```

```
</type>
```

```
</bank_details>
```

### 2nd xQuery:

```
for $x in doc("bankDone.xml")/doc/payment
```

```
return
```

```
<payment_details>
```

```
<amount>
```

```
{ string($x/amount) }
```

```
</amount>
```

```
<type>
```

```
{ string($x/type) }
```

```
</type>
```

```
</payment_details>
```

---

## Just\_Eat:

DTD document:

```
<?xml version="1.0" ?>

<!DOCTYPE Just_Eat [
<!ELEMENT Just_Eat (Search*)>
<!ELEMENT Search (name+, email, phone, dateofbirth)>
<!ATTLIST Search restaurant_search CDATA #REQUIRED>
<!ATTLIST Search food_search CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
]>
```

---

XML document:

```
<Just_Eat>
  <Search restaurant_search = 'Mad Egg' food_search = 'Beef Burger'>
    <name> Abdel Azuzzy </name>
    <email> elrayah@hotmail.com </email>
    <phone> "089334252 </phone>
    <dateofbirth> 9/01/2000 </dateofbirth>
  </Search>
  <Search restaurant_search = 'Burger King' food_search = 'Triple Cheesy Whopper'>
    <name> Ben Dover </name>
    <email> bdove@gmail.com </email>
    <phone> 0872637478 </phone>
    <dateofbirth> 27/09/1998 </dateofbirth>
  </Search>
  <Search restaurant_search = 'Dominos Pizza' food_search = 'Mighty Meaty'>
    <name> John Kennedy </name>
    <email> jken@gmail.com </email>
    <phone> 0856974751 </phone>
```

```
        <dateofbirth> 06/03/1991 </dateofbirth>
    </Search>
</Just_Eat>
```

---

## xQueries:

### 1st xQuery:

```
for $b in /Just_Eat/Search
where $b/name = "John Kennedy" and $b/phone = 0856974751
return
    <Search>
        <dob>{string ($b/dateofbirth)}</dob>
        <email>{string ($b/email)}</email>
    </Search>
```

### 2nd xQuery:

```
for $b in /Just_Eat/Search
where $b/name = "Ben Dover" and $b/phone = 0872637478
return
    <Search>
        <dob>{string ($b/dateofbirth)}</dob>
        <phone>{string ($b/phone)}</phone>
    </Search>
```

---

## Finder:

### DTD document:

```
<?xml version="1.0" ?>
<!DOCTYPE Finder[
```



```
<!ELEMENT Finder (Search* )>
<!ELEMENT Search (orderId* , orderNumber*)>
<!ATTLIST Search restaurant_search CDATA #REQUIRED>
<!ATTLIST Search food_search CDATA #REQUIRED>
<!ELEMENT orderId (#PCDATA)>
<!ELEMENT orderNumber (#PCDATA)>
```

```
] >
```

---

## XML document:

```
<Finder>

<Search restaurant_search = 'Mad Egg' food_search = 'Beef Burger'>
<orderId> 2020213 </orderId>
<orderNumber> 20 </orderNumber>
</Search>
<Search restaurant_search = 'Burger King' food_search = 'Triple Cheesy Whopper'>
<orderId> 2020213 </orderId>
<orderNumber> 30 </orderNumber>
</Search>
<Search restaurant_search = 'Dominos Pizza' food_search = 'Mighty Meaty'>
<orderId> 2020213 </orderId>
<orderNumber> 40 </orderNumber>
</Search>
</Finder>
```

---

## xQueries:

### 1st xQuery:

```
for $b in /Finder/Search
where $b/orderId = 2020213 and $b/orderNumber = 20
return
  <Search>
  <OI>{string ($b/orderId)}</OI>
  <OD>{string ($b/orderNumber)}</OD>
  </Search>
```

---

## Customer:

### DTD Document:

```
<?xml version="1.0"?>
<!DOCTYPE Doc[
<!ELEMENT Doc ( Customer* , Payment* , Order+ , Search+ , Bank*)>
<!ELEMENT Customer (first_name , last_name , email , phone_number , address)>
<!ELEMENT Payment ( amount+ , type* )>
<!ELEMENT Order (name+ , order_type* , number*)>
<!ELEMENT Search ( resturant_search+ , food_search+)>
<!ELEMENT Bank ( bank_Name* , account_Number* , account_Name* , account_Type*)>
<!--ATTLIST Customer id CDATA #REQUIRED-->
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT order_type (#PCDATA)>
<!ELEMENT number (#PCDATA)>
<!ELEMENT resturant_search (#PCDATA)>
<!ELEMENT food_search (#PCDATA)>
<!ELEMENT bank_Name (#PCDATA)>
<!ELEMENT account_Number (#PCDATA)>
<!ELEMENT account_Name (#PCDATA)>
<!ELEMENT account_Type (#PCDATA)>

]>
```

---

## XML document:

<Doc>

<Customer id = '22323'>  
<first\_name>Philip </first\_name>  
<last\_name> Tairov </last\_name>  
<email> tariovp@gmail.com </email>  
<phone\_number> 089433435 </phone\_number>  
<address> 1 spagethi mafia </address>  
</Customer>

<Payment>  
<amount> 100 </amount>  
<type> Apple Pay </type>  
</Payment>

<Payment>  
<amount> 12.95</amount>  
<type> Cash </type>  
</Payment>

<Payment>  
<amount> 43.65 </amount>  
<type> card </type>  
</Payment>

<Order>  
<name> Chicken Nuggets </name>  
<order\_type> Takeaway</order\_type>  
<number> 203992123 </number>  
</Order>

<Search>  
<resturant\_search> Mad eggs </resturant\_search>  
<food\_search> Beef Burger </food\_search>  
</Search>

<Bank>  
<bank\_Name> Allied Irish Banks </bank\_Name>

```
<account_Number> 20331234
</account_Number>
<account_Name> Folabi
</account_Name>
<account_Type> Student account </account_Type>
</Bank>

</Doc>
```

---

## xQueries:

### 1st xQuery:

```
declare function local:all_customers ()
{
  for $x in
doc("Customer/customer.xml")/Doc/Customer/phone_number
  return
  <phone>
  {($x)}
  </phone>
};
<all>
{local:all_customers() }
</all>
```

### 2nd xQuery:

```
declare function local:find_id ($paymentT as xs:string)
{
  for $j in
doc("Customer/customer.xml")/Doc/Payment
  return
  <bill>
  { string($j/amount )}
  </bill>
};
local:find_id("All three orders")
```

---

## Order:

### DTD Document:

```
<?xml version="1.0"?>
<!DOCTYPE Orders [
<!ELEMENT Orders (Order*)>
<!ELEMENT Order ( Order_name* , Order_type+ , Order_number* , Order_Payment*)>
<!ATTLIST Order_Payment idref CDATA #REQUIRED>
<!ELEMENT Order_name (#PCDATA)>
<!ELEMENT Order_type (#PCDATA)>
<!ELEMENT Order_number (#PCDATA)>
<!ELEMENT Order_Payment (#PCDATA)>

]>
```

---

### XML Document:

```
<Orders>
  <Order>
    <Order_name>Chicken Nuggets</Order_name>
    <Order_type>Takeaway</Order_type>
    <Order_number>203992123</Order_number>
    <Order_Payment idref="201">
    </Order_Payment>
  </Order>
  <Order>
    <Order_name>Chicken Burger</Order_name>
    <Order_type>Delivery</Order_type>
    <Order_number>203992123</Order_number>
  <Order_Payment idref="207">
    </Order_Payment>

</Order>
```

</Orders>

---

## xQueries:

### 1st xQuery:

```
for $x in doc("order.xml")/Orders/Order
return
  <Order_type>

    {string($x/Order_type)}

</Order_type>
```

### 2nd xQuery:

```
for $x in doc("order.xml")/Orders/Order
return
  <Order_number>

    {string($x/Order_number)}

</Order_number>
```

---

## Payment:

### DTD Document:

```
<?xml version = "1.0" encoding = "UTF-8"?>

<!DOCTYPE Payments[ <!ELEMENT Payments(Payment)>
<!ELEMENT Payment (payment_amount, payment_type)>
<!ELEMENT payment_amount (#PCDATA)>
<!ELEMENT payment_type (#PCDATA)>
<!ATTLIST Payment id CDATA #REQUIRED>
]>
```

---

### XML Document:

```
<Payments>

<Payment id = "201">
<payment_amount> 100 </payment_amount>
<payment_type> Apple Pay </payment_type>
</Payment>

<Payment id = "101">
<payment_amount> 12.95</payment_amount>
<payment_type> Cash </payment_type>
</Payment>

<Payment id = "301">
<payment_amount> 43.65 </payment_amount>
<payment_type> card </payment_type>
</Payment>

</Payments>
```

---

### xQueries:

### 1st xQuery:

```
let $j := doc("payment.xml")/Payments/Payment/payment_type
```

```
return
```

```
<list_of_payment_types>  
  {distinct-values($j)}  
</list_of_payment_types>
```

### 2nd xQuery:

```
for $j in doc("payment.xml")/Payments/Payment/payment_amount  
order by $j/payment_amount descending
```

```
return
```

```
<payment_sum>  
  {$j}  
</payment_sum>
```

### 3rd xQuery

```
declare function local:payment_values()
```

```
{
```

```
for $payment-value in doc("payment.xml")/Payments/Payment/payment_amount
```

```
return
```

```
<payment_value>  
  {$payment_value}  
</payment_value>  
};
```

```
<payment>
```

```
{local:payment_values()}  
</payment>
```

---



## Settings:

### DTD Document:

```
<?xml version="1.0"?>
<!DOCTYPE doc[
<!ELEMENT doc ( settings, Just_eat )>
<!ELEMENT settings ( firstname*, lastname*, emailaddress?, password?, phonenumber*,
address)>
<!ELEMENT Just_eat ( settings.User_Name, settings.User_Email, settings.User_Phone)>
<!ATTLIST doc number CDATA #REQUIRED>
<!ELEMENT settings.User_Name (#PCDATA)>
<!ELEMENT settings.User_Email (#PCDATA)>
<!ELEMENT settings.User_Phone (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT emailaddress (#PCDATA)>
<!ELEMENT password (#PCDATA)>
<!ELEMENT phonenumber (#PCDATA)>
<!ELEMENT address (#PCDATA)>

]>
```

---

### XML Document:

```
<doc number = "2">

<settings>
  <firstname> Matt </firstname>
    <lastname> Murdock </lastname>
    <emailaddress> mattie@tcd.ie </emailaddress>
    <password> ***** </password>
    <phonenumber>0874532242</phonenumber>
    <address> 69 White Manor </address>
  </settings>
  <Just_eat>
    <settings.User_Name> Abdel</settings.User_Name>
    <settings.User_Email> abdel232@gmail.com </settings.User_Email>
    <settings.User_Phone> 0834355532</settings.User_Phone>
  </Just_eat>
```

</doc>

---

## xQueries:

### 1st xQuery:

```
let $x := doc("settings.xml") /doc/settings/phonenumbers
return
<phone_number_check>
{$x}
{if(fn:string-length($x) = 10)
then "this phone number is valid"
else "phone number is not valid"}
</phone_number_check>
```

### 2nd xQuery:

```
let $x := doc("settings.xml") /doc/settings/emailaddresses
return
<email_address_check>
{$x}
{if(fn:contains($x, "@") )
then "email address is valid"
else "email address is not valid"}
</email_address_check>
```

---

(: Thank you :)