

Comandi di Oracle per la gestione del DW

```
CREATE DIMENSION <nome>
{LEVEL <nome_l> IS <nome tabella>.<attr>}
HIERARCHY <nome gerarchia> (
    {<nomelivello> CHILD OF}
    ...)
{ATTRIBUTE <nome livello> DETERMINES <nome<tabella>.<attr>}
```

```
CREATE MATERIALIZED VIEW nome
BUILD [IMMEDIATE|DEFERRED]
REFRESH [FAST|COMPLETE|FORCE] [ON COMMIT|ON DEMAND|START WITH <date> NEXT <date expr>]
[ENABLE QUERY REWRITE]
AS <subquery >
```

```
DROP MATERIALIZED VIEW nome
```

```
ALTER MATERIALIZED VIEW ...
```

```
SELECT ...FROM... WHERE... GROUP BY CUBE (elenco colonne)
```

```
SELECT .... FROM... WHERE... GROUP BY ROLLUP (elenco colonne)
```

```
SELECT A1,...,An
FROM
    (SELECT B1,...,Bm, RANK() OVER(ORDER BY Ai ASC ORDER BY Aj DESC) AS rank
FROM ...
WHERE ...
GROUP BY (A1,...,An)
WHERE rank <= N;
```

Comandi di XSL

```
<xsl:template
  match = pattern
  name = qname
  priority = number>
<!-- Content: (xsl:param*, template) -->
</xsl:template>

<xsl:apply-templates select = node-set-expression >
<!-- Content:(xsl:sort | xsl:with-param)* -->
</xsl:apply-templates>

<xsl:value-of select= node-set-expression />

<xsl:for-each
  select = node-set-expression>
<!-- Content: (xsl:sort*, template) -->
</xsl:for-each>

<xsl:choose>
<xsl:when test= node-set-expression>
... some code ...</xsl:when>
<xsl:otherwise>... some code .... </xsl:otherwise>
</xsl:choose>

<xsl:if test= node-set-expression> some output ... </xsl:if>

<xsl:sort select= node-set-expression />

node-set-expression e' un'espressione Xpath
```

Tabella in HTML

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage.
<b>This text is bold</b>

<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>

</body>
</html>
```

Comandi di XML schema

```
<xsd:element name="name" type="type" minOccurs="int" maxOccurs="int"/>
```

basicType: xsd:integer, xsd:string, xsd:date

```
<xsd:element name="name" minOccurs="int" maxOccurs="int">
  <xsd:complexType>
    <xsd:[sequence|choise]>
    ....
  </xsd:[sequence|choise]>
  </xsd:complexType>
</xsd:element>
```

DEFAULT: minOccurs 1, maxOccurs 1

```
<xsd:simpleType name= "name">
  <xsd:restriction base= "xsd:source">
    [ <xsd:facet value= "value"/>| <xsd:enumerate value= "value"/>]
  </xsd:restriction>
</xsd:simpleType>
```

Element	Explanation
attribute	Defines an attribute
choice	Allows only one of the elements contained in the <choice> declaration to be present within the containing element
complexType	Defines a complex type element
element	Defines an element
extension	Extends an existing simpleType or complexType element
restriction	Defines restrictions on a simpleType, simpleContent, or a complexContent
schema	Defines the root element of a schema
sequence	Specifies that the child elements must appear in a sequence.
simpleContent	Contains extensions or restrictions on a text-only complex type or on a simple type as content and contains no elements
simpleType	Defines a simple type and specifies the constraints and information about the values of attributes or text-only elements

XSD Restrictions/Facets for Datatypes

Constraint	Description
enumeration	Defines a list of acceptable values
length	Specifies the exact number of characters or list items allowed.
[max min]Exclusive	Specifies the [upper lower] bounds for numeric values
[max min]Inclusive	Specifies the [upper lower] bounds for numeric values
[max min]Length	Specifies the [maximum minimum] number of characters or list items allowed.