

Designing Indexed Views



Gail Shaw

TECHNICAL LEAD

@SQLintheWild <http://sqlinthewild.co.za>



Agenda



Introducing Indexed Views

Limitations and restrictions

Where are they useful?



Introducing Indexed Views

Normally, views are just saved
SELECT statements

It is possible to create a
clustered index on a view

This materialises the view,
persists its results as if it were
a table

Contents of the view are
automatically kept up to date
as underlying tables change





Indexed views have a massive number of limitations



Limitations of Indexed Views

```
CREATE VIEW ClientsLatestShipment  
WITH SCHEMABINDING  
AS
```

```
SELECT c.*,  
       (SELECT OfficialName  
        FROM dbo.StarSystems ss  
        WHERE ss.StarSystemID = c.HeadquarterSystemID) AS HQ  
FROM dbo.Clients c LEFT OUTER JOIN (  
  SELECT s.ClientID, s.ReferenceNumber, MAX(sd.Mass) AS HighestMass,  
         AVG(sd.Volume) AS AvgVolume,  
         ROW_NUMBER() OVER (PARTITION BY s.ClientID ORDER BY s.CreationDate  
DESC) AS RN  
  FROM dbo.Shipments s INNER JOIN dbo.ShipmentDetails sd ON sd.ShipmentID =  
s.ShipmentID  
  GROUP BY s.ClientID, s.CreationDate, s.ReferenceNumber) sub ON c.ClientID =  
sub.ClientID  
WHERE sub.RN = 1
```



SET Options Required

SET ANSI_NULLS ON

SET ANSI_PADDING ON

SET ANSI_WARNINGS ON

SET ARITHABORT ON

SET CONCAT_NULL_YIELDS_NULL ON

SET NUMERIC_ROUNDABORT OFF

SET QUOTED_IDENTIFIER ON



Where Are Indexed Views Useful?

Aggregates

Simple joins with filters



Demo



Creating an indexed view and using it in queries



Downsides of Indexed Views

Additional write overhead

Additional space requirement

Require specific SET options



Summary



What indexed views are

Limitations and restrictions

Where they are useful

