Long term “investment” in access paths

Definition of index structures in SQL:

• Index on relations (primary/secondary):
  – CREATE INDEX IndexOnRel ON Rel(A)
• View index:
  – CREATE VIEW V1 AS SELECT ... FROM ... WHERE ...
  – CREATE INDEX IV1 ON V1 ...

Result of V1 is stored in database (Materialisation!)
Result of V1 might be used to answer queries

Definition of such access paths is the task of physical DB design!

Example from MS SQL Server Doc

These queries will use the above indexed view:

SELECT SUM(UnitPrice*Quantity*(1.00-Discount)) AS Rev,
   OrderDate, ProductID
FROM dbo.[Order Details] od, dbo.Orders o
WHERE od.OrderID=o.OrderID AND ProductID in (2, 4, 25,...)
   AND OrderDate >= '05/01/1998'
GROUP BY OrderDate, ProductID
ORDER BY Rev DESC

SELECT OrderDate, SUM(UnitPrice*Quantity*(1.00-Discount))
   AS Rev
FROM dbo.[Order Details] od, dbo.Orders o
WHERE od.OrderID=o.OrderID AND DATEPART(mm,OrderDate)= 3
   AND DATEPART(yy,OrderDate) = 1998
GROUP BY OrderDate
ORDER BY OrderDate

Example from MS SQL Server Doc

Create view:
CREATE VIEW V1
WITH SCHEMABINDING
AS
SELECT SUM(UnitPrice*Quantity*(1.00-Discount)) AS Revenue, OrderDate, ProductID, COUNT_BIG(*) AS COUNT
FROM dbo.[Order Details] od, dbo.Orders o
WHERE od.OrderID=o.OrderID
GROUP BY OrderDate, ProductID

Create index on the view:
CREATE UNIQUE CLUSTERED INDEX IV1 ON V1 (OrderDate, ProductID)

Query Execution Plan

Without indexed view!
Query Execution Plan

With indexed view!

SELECT SUM(UnitPrice*Quantity*(1.00-Discount)) AS Rev, OrderDate, ProductID
FROM dbo.[Order Details] od, dbo.[Order] o
WHERE od.OrderID = o.OrderID AND ProductID in (2, 4, 25, 13, 7, 89, 22, 34)
GROUP BY OrderDate, ProductID
ORDER BY Rev DESC

Query Execution Plan

Query might be slightly different from view definition (here: AVG instead of SUM, as COUNT is also contained in view)

Query Execution Plan

Backjoins are not considered