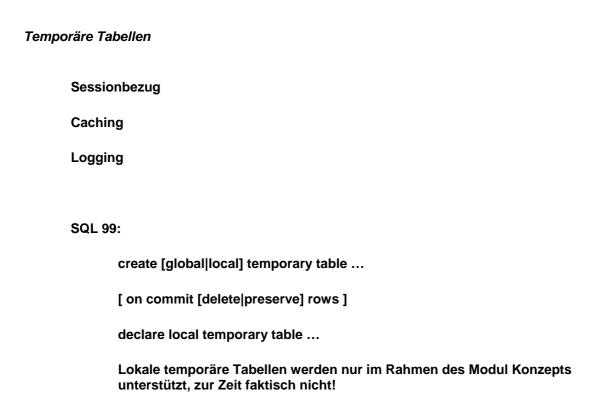
# 8. Datendefinition in SQL (2)



Verschiedene proprietäre Ansätze

Views	
	Ziel:
	Vereinfachung lesender Zugriffe
	und der Verwaltung der Zugriffsrechte
	View Definition:
	Create view Viewname [ (Spaltenliste) ] as
	Select-Anweisung
	Die View Definition wird zur Laufzeit umgesetzt
	I.A. werden die Abfrage auf die View und die Viewdefinition zusammengefasst (Merge), eine Materialisierung der View stellt häufig ein Performance Problem dar.

Es hängt vom verwendeten DBS ab, welche Selects im Rahmen einer Viewdefinition verwendet werden dürfen!

Materialized Views (Automatic Summary Tables)

**Einsatz im Data Warehouse Bereich** 

### View Updates

View Updates sind nur eingeschränkt möglich, wobei Unterschiede von DBS zu DBS zu beachten sind

### **Update Problematik:**

Es kann sein, daß durch eine Änderung von Daten über eine View die Daten über die View nicht mehr sichtbar sind.

### **Check Option**

sorgt dafür, daß dieses Phänomen nicht auftritt

#### Ziel:

Vereinfachung lesender und schreibender Zugriffe und der Verwaltung der Zugriffsrechte

Realisierung in verschiedenen 3GLs (C/C++, Java)

Realisierung in SQL:

SQL Anweisungen (i.a. DML)

Problematik von Select - Anweisungen

**Prozedurale Anweisungen** 

**Exception Handling** 

#### Vorteile:

Zentrale Datenbankzugriffsschicht

Weniger Client/Server Kommunikation

Wiederverwendung des Query Execution Plan

#### Nachteile:

Mangelnde Portabilität

(Implementierung + Schnittstelle + Mächtigkeit)

Probleme mit dynamischem SQL

Performance Probleme bei "Missbrauch"

**Zusammenfassung zu Moduln (Packages)** 

# Trigger

## Spezifikation durch:

Auslösende Aktion

Tabelle

Zeitpunkt

**Begleitende Aktion** 

Portablitätsaspekte

Instead of - Trigger

Select Trigger – Auditing

# Spezielle Konstrukte:

Verteilte DB (Links)

Replikation (Snapshots, ggf. mit Refresh)