



**ORACLE<sup>®</sup>**

## **3NF Modelling**

Wim Villano

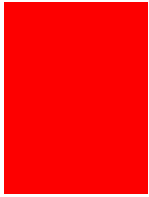
Sales Consultant BI, Oracle Nederland



# Agenda

- Oracle BI EE+ and 3NF
- 3NF Modelling Considerations
- Demonstration 3NF Modelling
- Summary
- Q&A



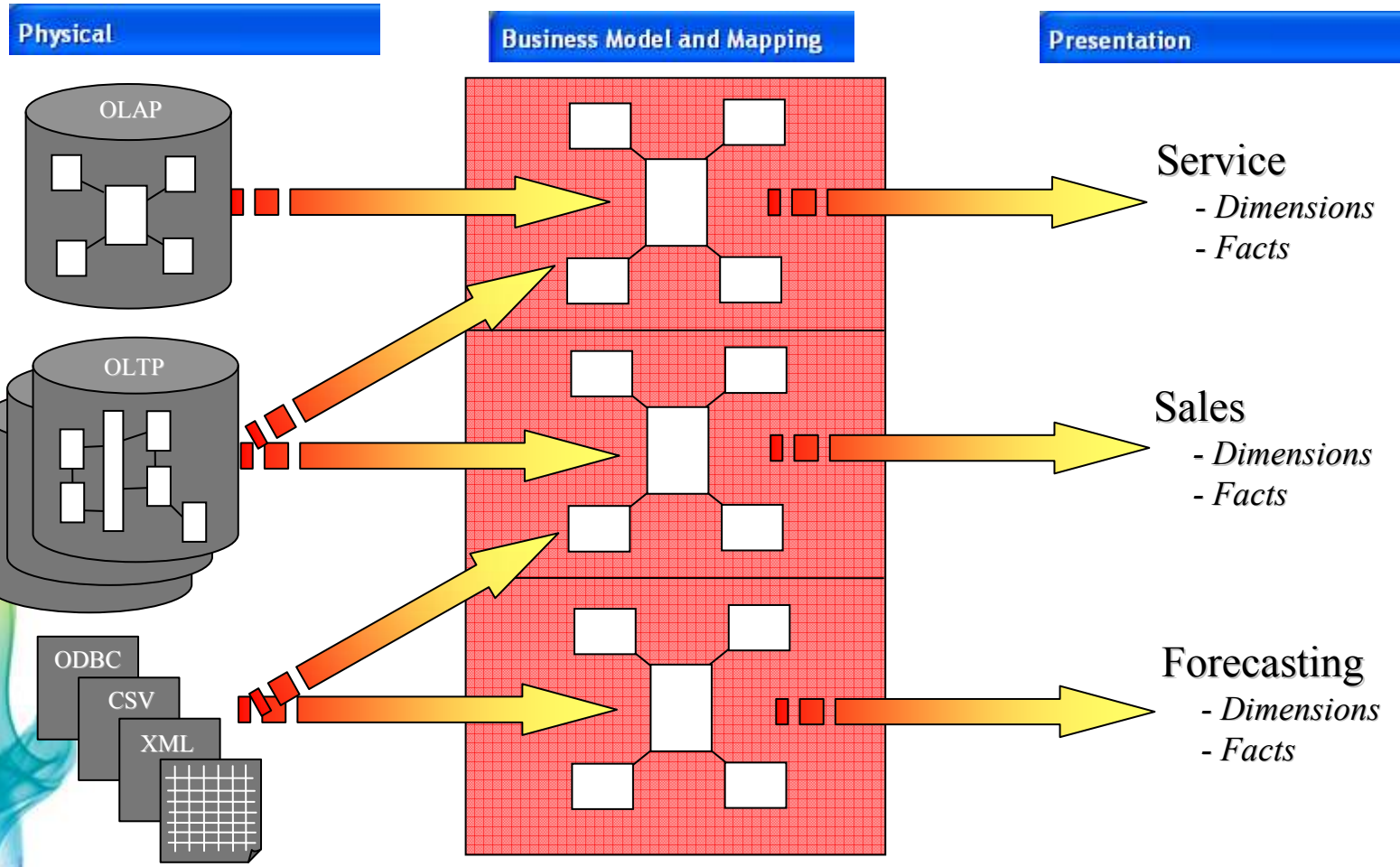
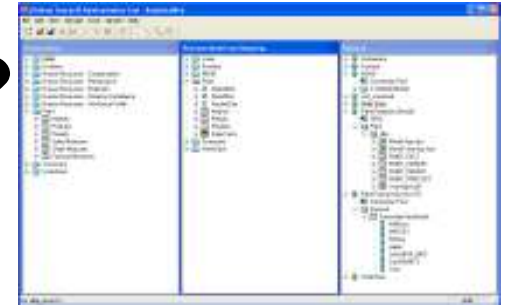


Oracle BI EE+ and 3NF



# Why is BI EE so good for 3NF?

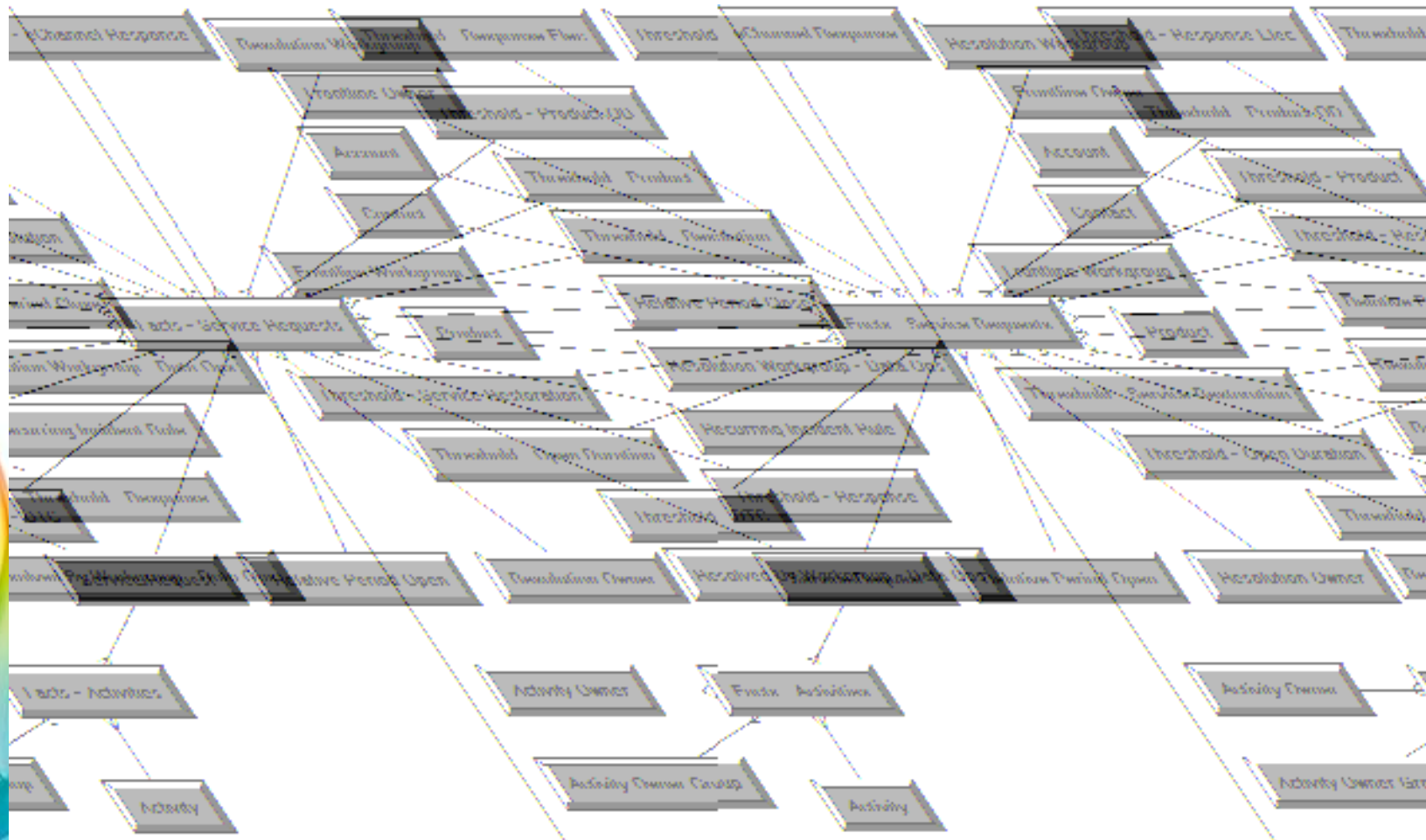
Logical Star Schemas and intuitive "Subject Areas" ....  
regardless of underlying complexity





# Why is BI EE so good for 3NF?


And this is what it looks once converted into “Logical” Star Schemas

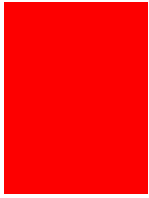




# BI EE and 3NF

**BI EE provides many benefits with 3NF / OLTP data-sources**

- Ability to map any physical data-model into Logical Star Schemas
  - End-user is transparent to the complexity on the physical layer
  - Simplified calculations
  - Simplified security
  - Using BI EE “Fragmentation”, multiple OLTPs can appear as one
  - Excellent performance (response times & throughput)
  - Many features to suit “Operational” reporting:
    - Traffic light colours
    - In-built drill-down ability
    - Conditional formatting
    - Multiple reports on a single dashboard
    - Guided Navigation
    - BI Delivers can alert via a number of channels
    - For urgent requirements, a report can run physical SQL direct against source db
- 



## 3NF Modelling Considerations

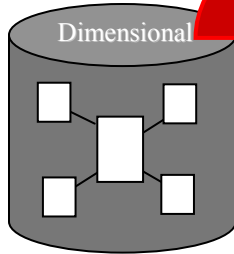


# Difference 3NF and Dimensional

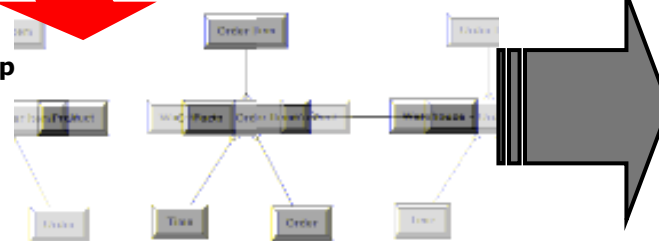
Physical

Business Model and Mapping

Dimensional Model

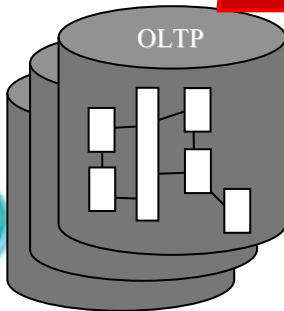


Drag/Drop

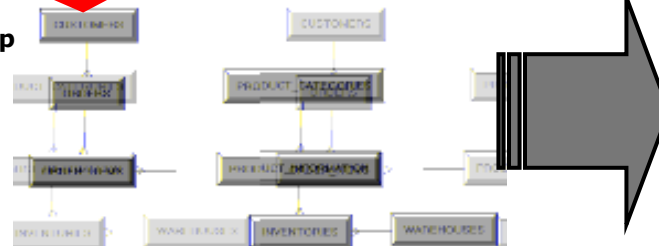


Region	Brand	Dollars	Units
CENTRAL REGION	Enterprise	\$36,557,766	11,457,019
	Magcolor	\$8,330,004	2,655,654
	McCluskey	\$73,429,694	20,366,826
CENTRAL REGION Total		\$118,317,464	34,479,500
EASTERN REGION	Enterprise	\$57,203,750	23,041,024
	Magcolor	\$17,127,516	4,643,990
	McCluskey	\$152,331,562	44,239,374
EASTERN REGION Total		\$226,662,828	71,924,388
SOUTHERN REGION	Enterprise	\$48,224,036	17,646,814
	Magcolor	\$31,225,642	7,856,282
	McCluskey	\$52,499,552	14,877,918
SOUTHERN REGION Total		\$131,949,230	49,380,814
WESTERN REGION	Enterprise	\$51,439,136	15,182,939
	Magcolor	\$43,191,822	9,513,999
	McCluskey	\$104,129,638	28,044,350
WESTERN REGION Total		\$198,760,606	52,741,288

3NF



Drag/Drop



View Display Error

Odbc driver returned an error (SQLEXPDIRECTW).

[Error Details](#)



Error Codes: OPR4ONWY:U9IM8TAC:OI2DL65P

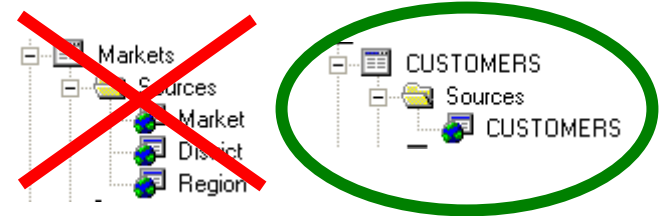
State: HY000. Code: 10058. [nQODBC] [SQL\_STATE: HY000] [nQSError: 10058] A general error has occurred. [nQSError: 15018] Incorrectly defined logical table source (for fact table INVENTORIES) does not contain mapping for [CUSTOMERS.CUST\_LAST\_NAME]. (HY000)

SQL Issued: SELECT CUSTOMERS.CUST\_LAST\_NAME saw\_0, INVENTORIES.QUANTITY\_ON\_HAND saw\_1 FROM OE ORDER BY saw\_0, saw\_1

[Refresh](#)

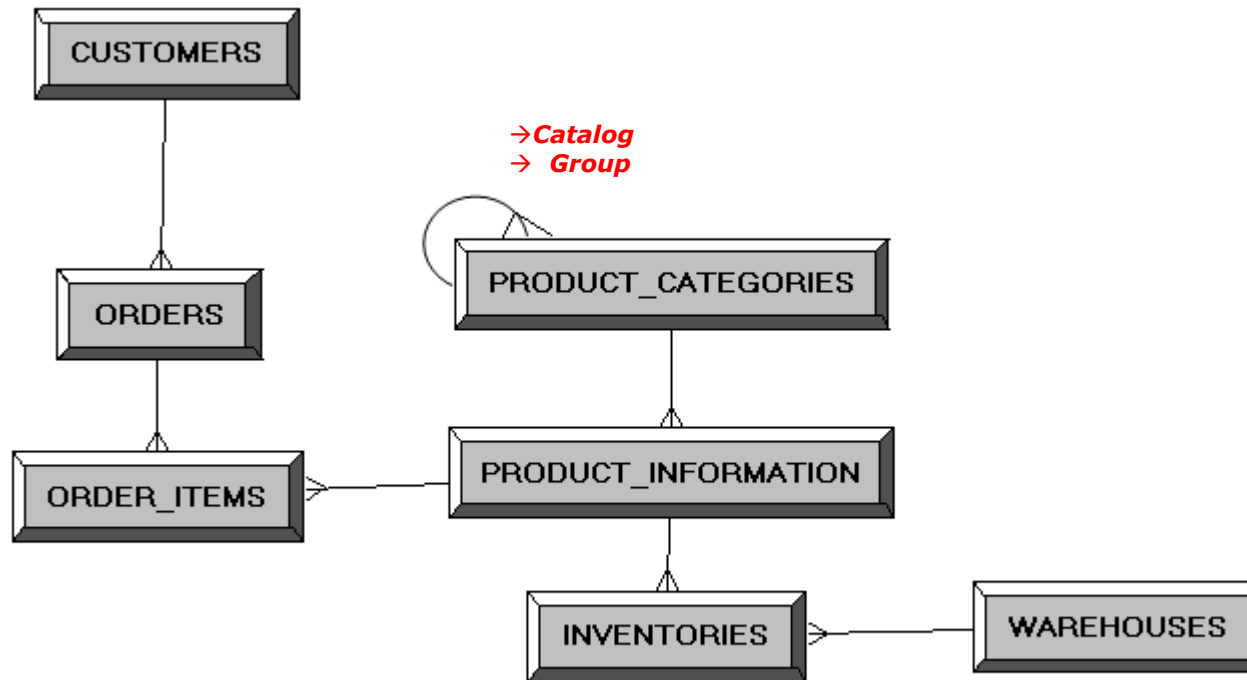
# Considerations

- Think in Dimensions and Facts (dependent on user requirements)
- Remove multiple & circular joins in Physical Layer -> Alias 
- Remove self-joins in Physical Layer -> Alias 
- Basic Modelling rule: 1 LTS per dimension!
- Deal with Time (Dimension)



# Time Dimension with 3NF

“OE” schema supplied on Oracle db’s





# Time Dimensions with 3NF

1. Don't have a Time dimension
2. Use an existing table to obtain Time dimension data (via Opaque View)
3. Import a Time dimension physical table and join using the calendar date columns






# Time Dimensions with 3NF


## Option 1

- Don't have a Time dimension and force users to perform "BETWEEN" queries

e.g. **SELECT "# Orders"**  
**WHERE "Order Date" BETWEEN '01-JAN-2006'**  
**AND '01-JAN-2007'**



Order Date  
Between  and

- 
- Indexes on ORDER\_DATE will still be usable
  - However, no real possibility of drilling-down Year → Quarter → Month → Day
  - Not the best in terms of user-friendliness
  - No "Canonical Time"

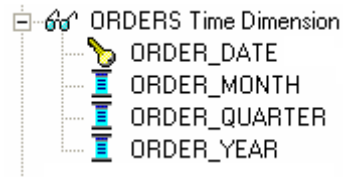
# Time Dimensions with 3NF

## Option 2 (a bit messy)

- Use an existing table to obtain Time dimension data, “Function Based Indexes” will be required for all levels of the Time hierarchy

e.g. Define your Time dimension by creating an “Opaque View” against the ORDERS table in BI EE:

```
create index FI_ORDERS_1 on ORDERS TRUNC(order_date);
create index FI_ORDERS_2 on ORDERS (TO_CHAR(order_date,'YYYY / MM'));
create index FI_ORDERS_3 on ORDERS (TO_CHAR(order_date,'YYYY / Q'));
create index FI_ORDERS_4 on ORDERS (TO_CHAR(order_date,'YYYY'));
```



```
SELECT trunc(order_date)      order_date,
       to_char(order_date,'YYYY / MM') order_month,
       to_char(order_date,'YYYY / Q')  order_quarter,
       to_char(order_date,'YYYY')     order_year
from orders
```

- Year → Quarter → Month → Day drill-down now possible
- Indexes used for all levels of Time hierarchy
- Dashboard filters may take a while to build as a full index scan is required to obtain all Year, Month, Quarter values

# Time Dimensions with 3NF

## Option 3 (preferred)

- Import a Time dimension physical table and join using the calendar date columns:

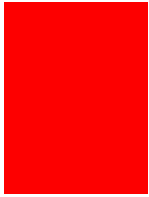
ORDER_ID	ORDER_DATE	STATUS	CUST_ID
2354	15/07/2000	0	104
2355	26/01/1998	8	104
2356	26/01/2000	5	105
2357	09/01/1998	5	108

Date	Month	Year	Year Month	Year Quarter
26/01/2000	January	2000	2000 / 01	2000 Q 1

- Year → Quarter → Month → Day drill-downs and “Canonical Time” are achievable
- If the 3NF date column is not truncated then create a “Function Based Index” and join to the date column using the TRUNC() function:

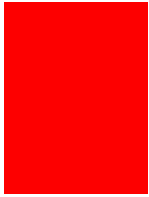
```
create index FI_ORDERS_1 on ORDERS TRUNC(order_date);
```

```
SELECT .....  
FROM TIME, ORDERS  
WHERE TIME.CALENDAR_DATE = TRUNC(ORDERS.ORDER_DATE)
```

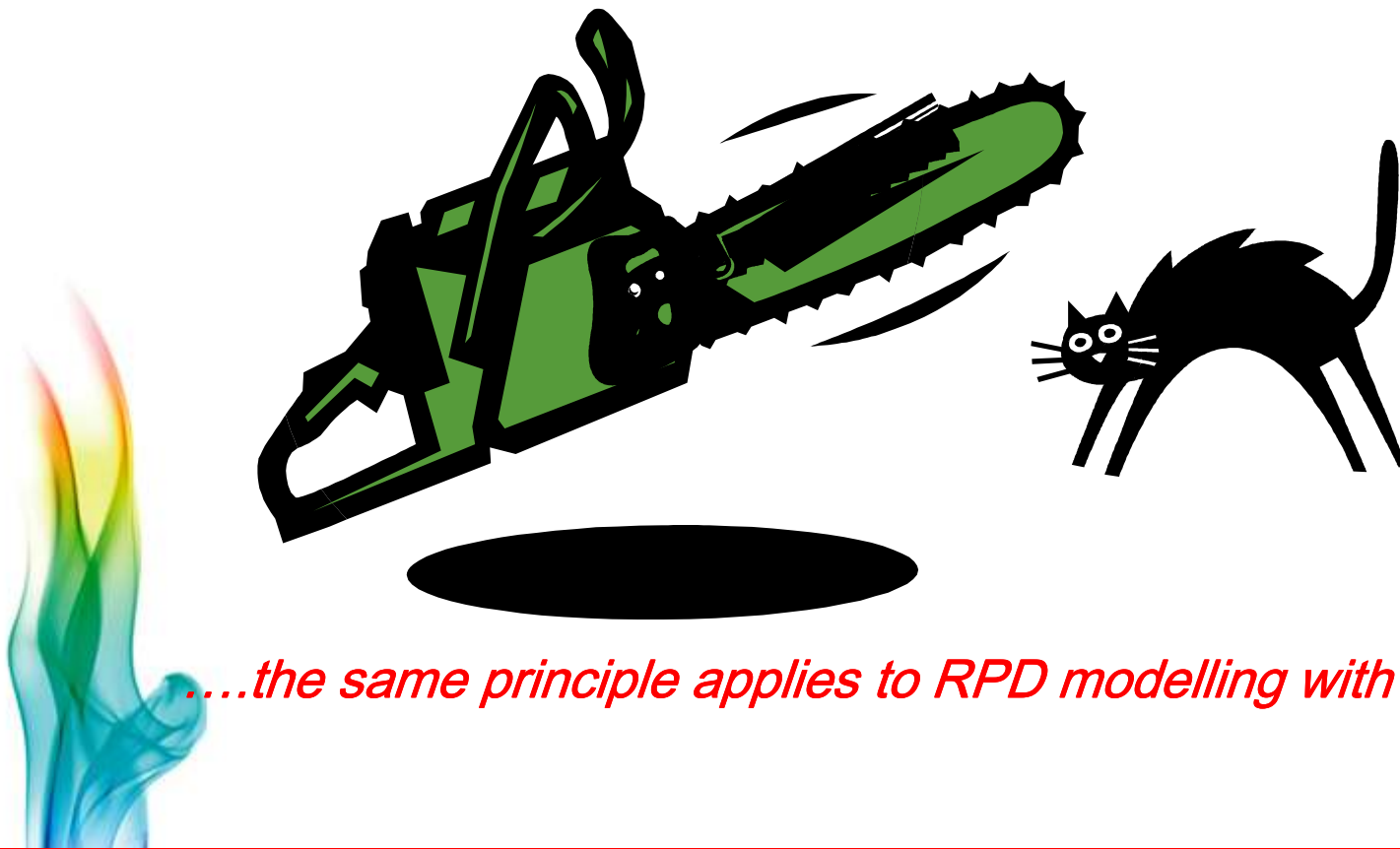


# Demonstration 3NF Modelling





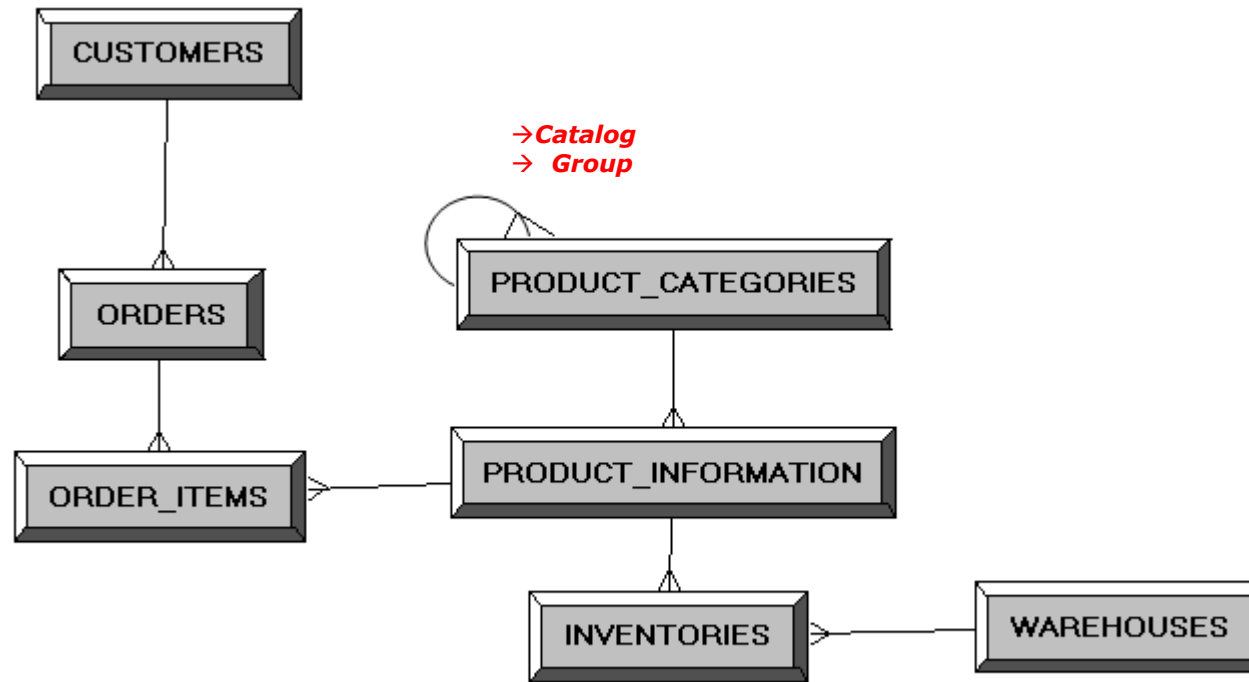
*"There is more than one way to skin a cat!"*



*....the same principle applies to RPD modelling with BI EE!*



# “OE” schema supplied on Oracle db’s





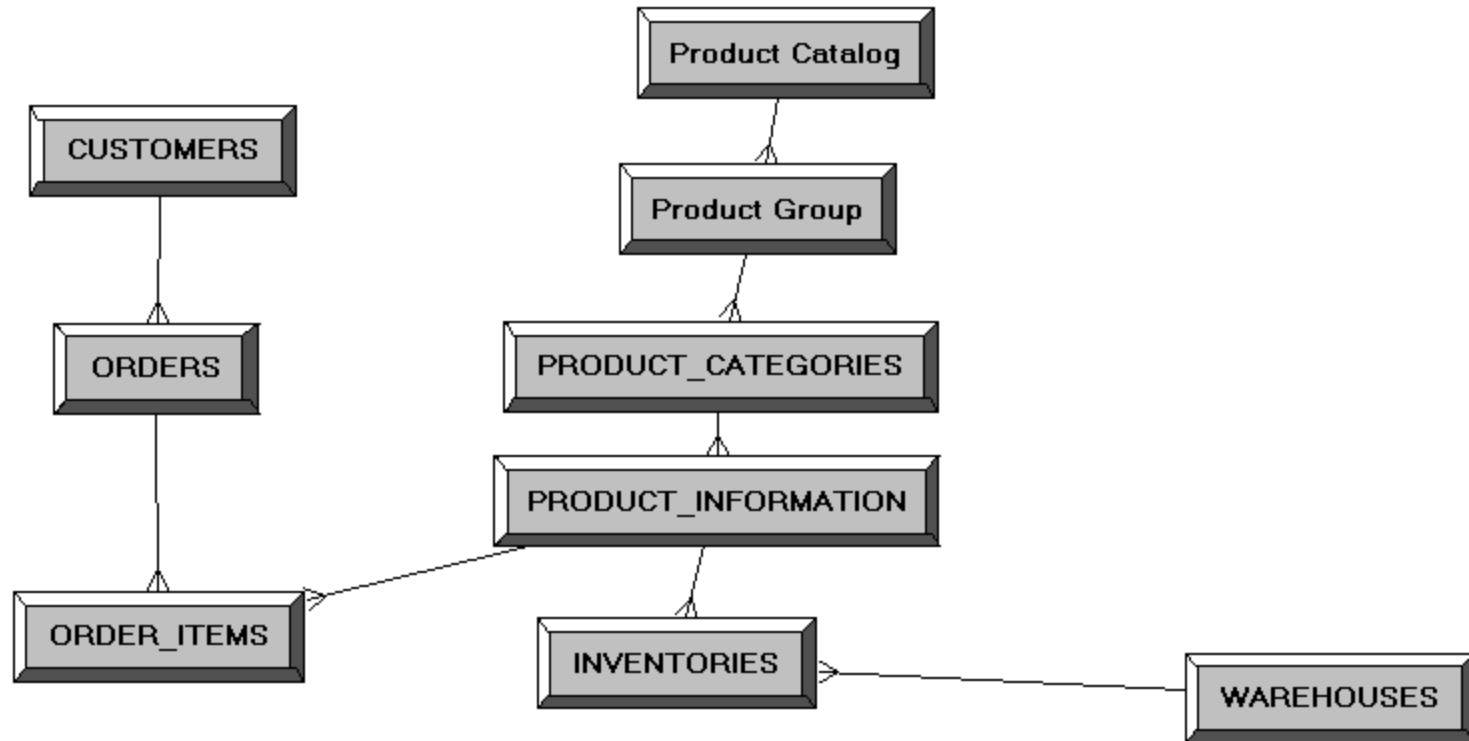


# How to Start



- One logical star at a time
- Model the Fact logical table first
- Model Time Dimension
- Model Other Dimensions (keep them related to the fact: 1 LTS)
- Create Aliases to
  - Solve self joins
  - Remove Multiple Joins
  - Remove Circulair Joins

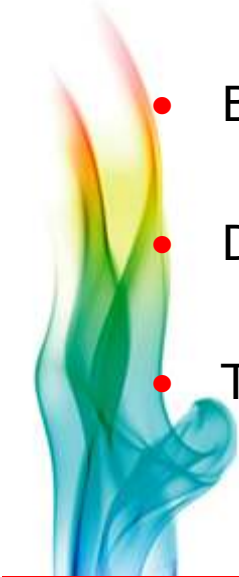
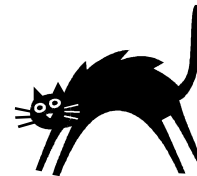
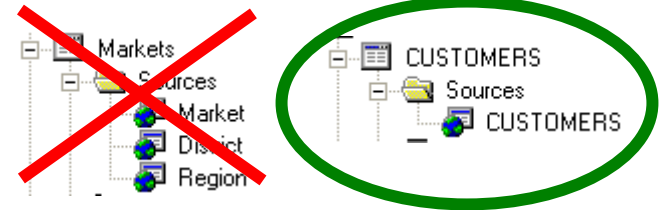


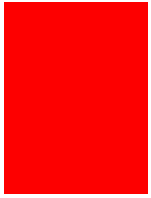
# Product Dimension



# Summary

- Think in Dimensions and Facts (dependent on user requirements)
- Remove multiple & circular joins in Physical Layer -> Alias 
- Remove self-joins in Physical Layer -> Alias 
- Basic Modelling rule: 1 LTS per dimension!
- Deal with Time (Dimension)
- There are more ways to skin the cat!





Q & A

