

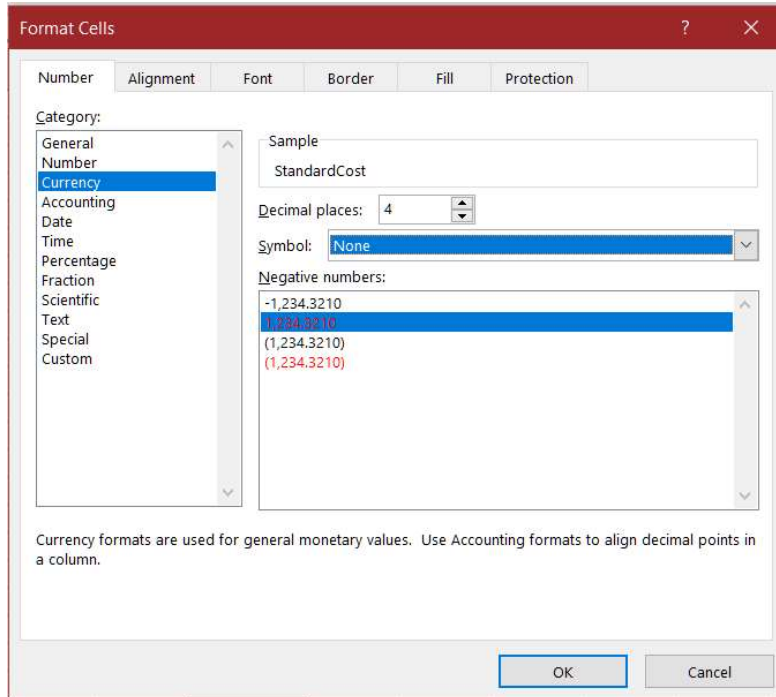
Data Warehousing and Business Intelligence

Cleaning Data

Part of the cleaning process, I have performed the following things on excel sheets.

Product Sheet:

- Formatted the StandardCost, ListPrice, DealerPrice columns in the Product sheet as currency with decimal places upto 4 and No \$ symbol.

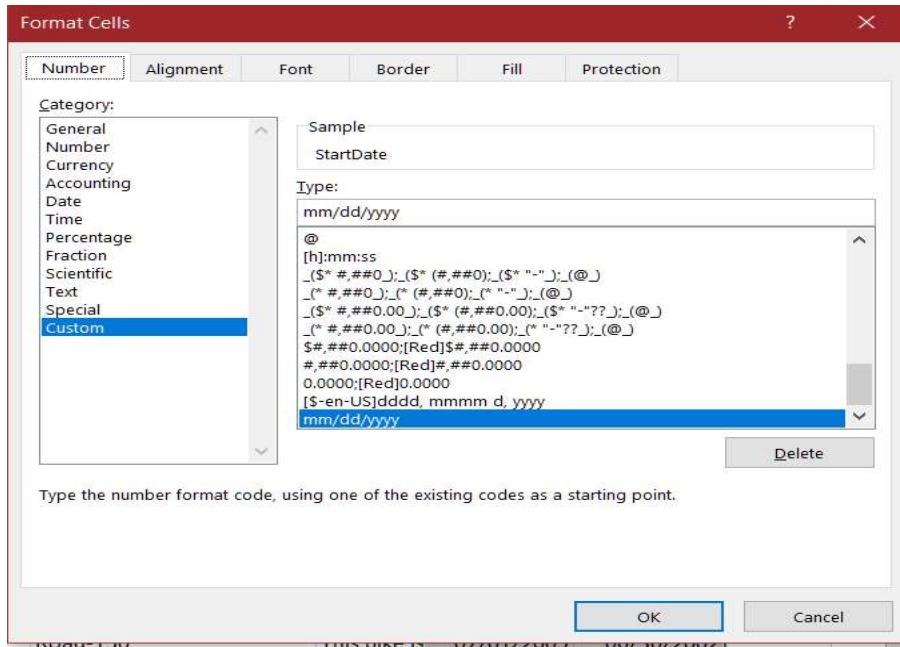


Formatted Start date and end date columns to mm/dd/yyyy format

StartDate	EndDate
7/1/2005 0:00	6/30/2002 0:00
7/1/2006 0:00	6/30/2003 0:00
7/1/2007 0:00	
7/1/2005 0:00	6/30/2002 0:00
7/1/2006 0:00	6/30/2003 0:00
7/1/2007 0:00	
7/1/2005 0:00	6/30/2002 0:00
7/1/2005 0:00	6/30/2002 0:00
7/1/2005 0:00	6/30/2002 0:00
7/1/2006 0:00	6/30/2003 0:00
7/1/2007 0:00	
7/1/2005 0:00	6/30/2002 0:00
7/1/2006 0:00	6/30/2003 0:00

-

Data Warehousing and Business Intelligence



StartDate	EndDate
07/01/2005	06/30/2002
07/01/2006	06/30/2003
07/01/2007	
07/01/2005	06/30/2002
07/01/2006	06/30/2003
07/01/2007	
07/01/2005	06/30/2002
07/01/2005	06/30/2002

Promotions Sheet:

- Formatted Start Date and End date to match the mm/dd/yyyy format.

Customer Sheet:

- Formatted BirthDate, DateFirstPurchase to match the mm/dd/yyyy format.
- Formatted YearlyIncome to match the currency format.
- Few of the Address Lines had incorrect symbol, which were changed accordingly.

Data Warehousing and Business Intelligence

Roßstr 9928	Roßstr 9928
Alderstr 7690	Alderstr 7690
6516 Beauver Lane	6516 Beauver Lane
88, avenue des Champs-Élysées	88, avenue des Champs-Élysées
Essener Straße 123	Essener Straße 123
4185 Kewwood Ct	4185 Kewwood Ct

Time Dimension Sheet:

- Formatted the FullDate to column to match mm/dd/yyyy format.

Transactions Sheet:

- Formatted OrderDate, DueDate and ShipDate columns to match the mm/dd/yyyy format.
- Formatted UnitPrice, ExtendedAmount, ProductStandardCost, TotalProductCost, SalesAmount, TaxAmount and Freight columns to match the currency format.
- OrderDateKey, DueDateKey and ShipDateKey were initially given in Date format, but their actual source i.e. date key in the time dimension table was an integer. For that reason, I have modified OrderDateKey, DueDateKey and ShipDateKey to text format (yyyymmdd) and then changed it to integer.

[illegible][illegible]

Data Warehousing and Business Intelligence

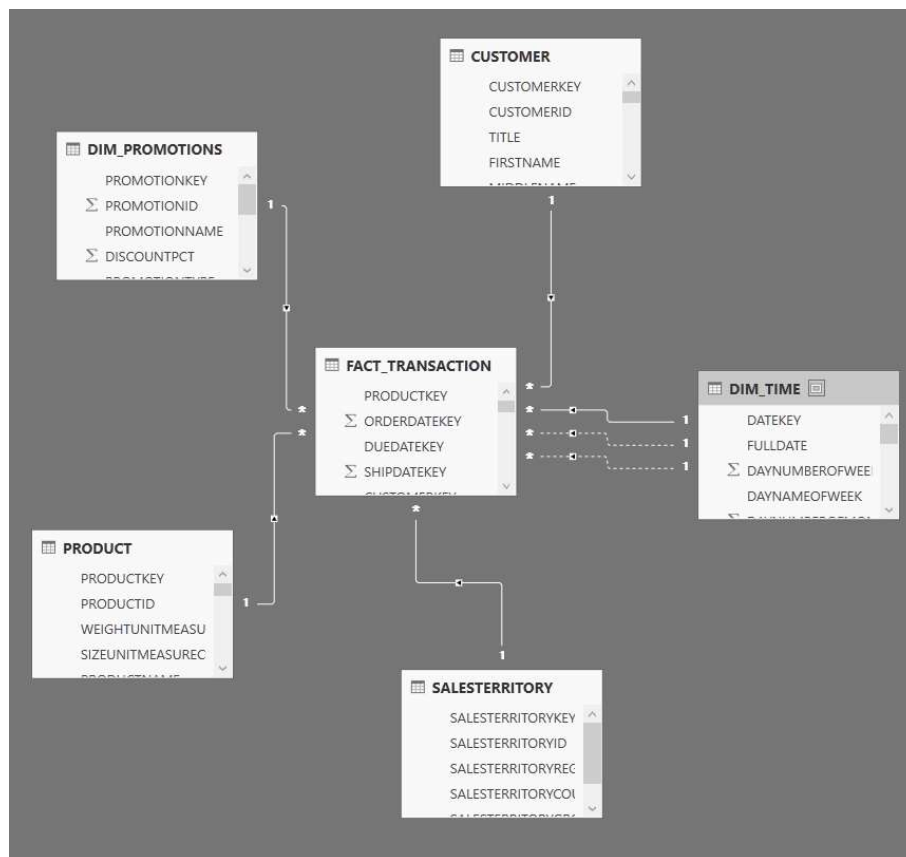
Data Warehouse:

- Part of the Data Warehouse implementation, I have created tables for each sheet in **Oracle** using oracle sql developer.
- Primary Keys were created based on the key values available in each sheet.
- For creation of Transaction table, foreign key references were made for CustomerKey, Productkey, PromotionKey, SalesTerritoryKey, DateKey associated to their respective columns in the table.

1. Based on the data available, I have implemented a data warehouse with Star schema. The data given had 5 excel sheets which can be considered as five dimensions that are Product, Customer, Sales Territory, Promotions and Time. As we have keys from all these tables referenced in the transaction table, we can consider the transaction table as a Fact table, which is linked to various Dimension tables.

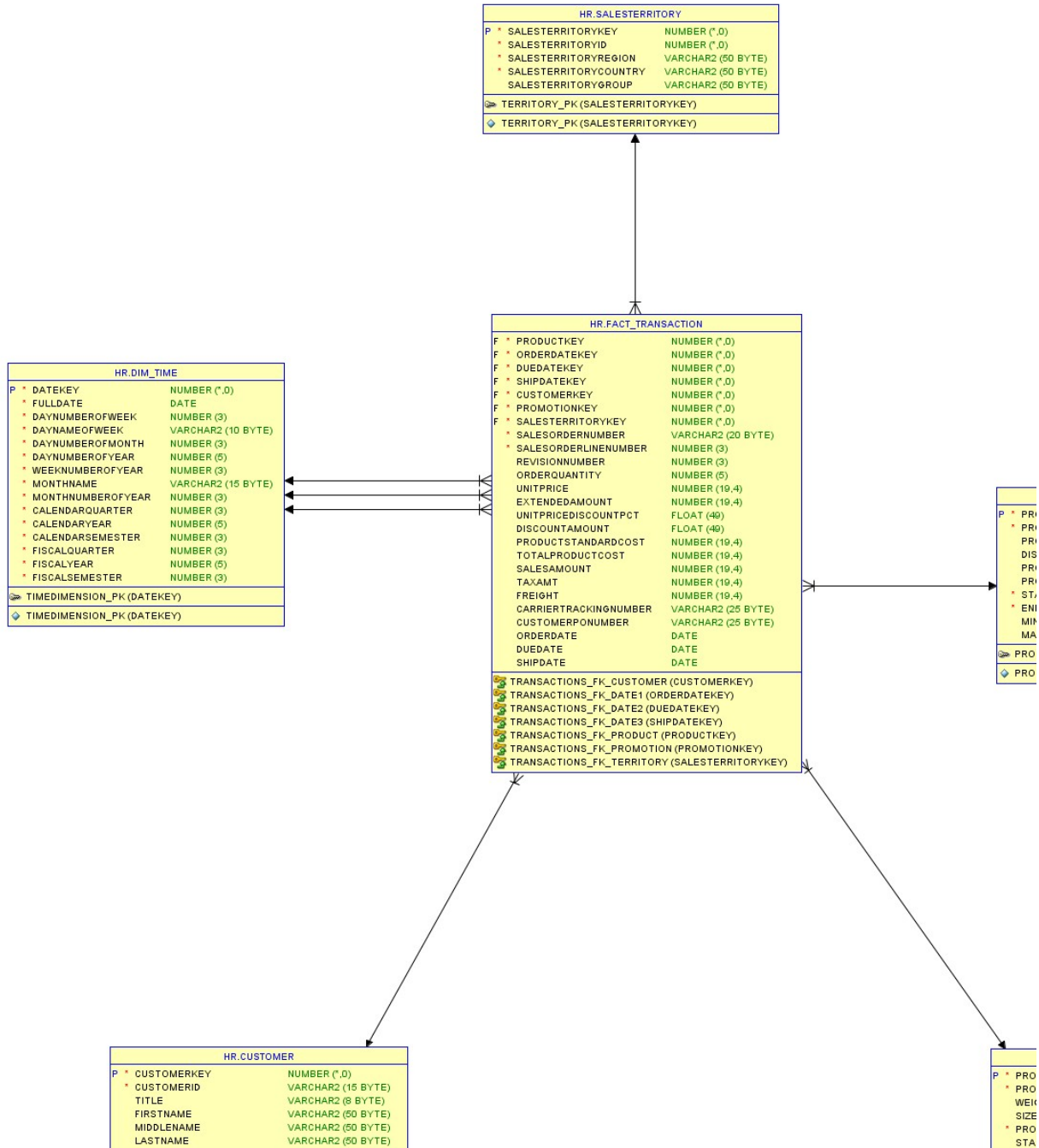
Using Star Schema enables us in analyzing the facts present in the transaction table across various dimensions. For that reason, I chose to implement star schema.

2. DW Diagram of my implementation:



Data Warehousing and Business Intelligence

Detailed DW Diagram:

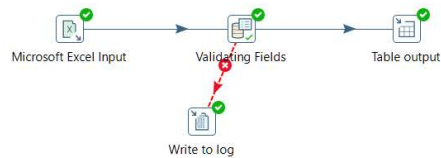


Note: Zoom in for details.

Data Warehousing and Business Intelligence

Loading Data

- I have used Pentaho to load my data from excels into the oracle tables that I have created.
- Have inputted each of the excel sheets individually and loaded data into tables separately.
- For each of the Dimension table loads, I have performed a validation transformation to make sure that No Null Data is inserted into the tables.
- Following screenshots show loads performed into oracle tables from excel sheets.



Execution Results										
<div>Execution History Logging Step Metrics Performance Graph Metrics Preview data</div>										
<div>● First rows ○ Last rows ○ Off</div>										
#	ProductKey	ProductID	WeightUnitMeasureCode	SizeUnitMeasureCode	ProductName	StandardCost	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint
1	293.0	FR-M94S-46	LB	CM	HL Mountain Frame - Silver, 46	623.8403	Y	Silver	500.0	375.0
2	294.0	FR-M94S-46	LB	CM	HL Mountain Frame - Silver, 46	660.9142	Y	Silver	500.0	375.0
3	295.0	FR-M94S-46	LB	CM	HL Mountain Frame - Silver, 46	747.2002	Y	Silver	500.0	375.0
4	296.0	FR-M94B-42	LB	CM	HL Mountain Frame - Black, 42	617.0281	Y	Black	500.0	375.0
5	297.0	FR-M94B-42	LB	CM	HL Mountain Frame - Black, 42	653.6971	Y	Black	500.0	375.0
6	298.0	FR-M94B-42	LB	CM	HL Mountain Frame - Black, 42	739.041	Y	Black	500.0	375.0
7	299.0	FR-M94B-44	LB	CM	HL Mountain Frame - Black, 44	699.0928	Y	Black	500.0	375.0
8	300.0	FR-M94B-48	LB	CM	HL Mountain Frame - Black, 48	699.0928	Y	Black	500.0	375.0
9	301.0	FR-M94B-46	LB	CM	HL Mountain Frame - Black, 46	617.0281	Y	Black	500.0	375.0

Product Table Load

Data Warehousing and Business Intelligence

Oracle SQL Developer - HR_SCHEMA

File Edit View Navigate Run Source Tools Window Help

NEW.sql HR_SCHEMA OLAP_Taketome.sql Relational_1 (Untitled_1)

Worksheet Query Builder

select * from product;

Script Output Query Result Explain Plan Query Result 1

SQL Fetched 50 rows in 0.008 seconds

	PRODUCTKEY	PRODUCTID	WEIGHT...	SIZEUNIT...	PRODUCT...	STANDARDCOST	FINISHED...	COLOR	SAFETYST...	REORDER...	LISTPRICE	PRODUCT...	SIZERANGE	WEIGHT	DAYSTOM...	PRODUCT...	DEALERP...
1	336 BK-R50B-62 LB	CM	Road-65...	413.1463 1	Black	100	75	699.0982 62.0	60-62 CM	20	4 R	419.4589 L					
2	337 BK-R50B-62 LB	CM	Road-65...	466.7066 1	Black	100	75	782.99 62.0	60-62 CM	20	4 R	469.794 L					
3	338 BK-R50B-44 LB	CM	Road-65...	413.1463 1	Black	100	75	699.0982 44.0	42-46 CM	18.77	4 R	419.4589 L					
4	339 BK-R50B-44 LB	CM	Road-65...	466.7066 1	Black	100	75	782.99 44.0	42-46 CM	18.77	4 R	469.794 L					
5	340 BK-R50B-48 LB	CM	Road-65...	413.1463 1	Black	100	75	699.0982 48.0	48-52 CM	19.1299...	4 R	419.4589 L					
6	341 BK-R50B-48 LB	CM	Road-65...	466.7066 1	Black	100	75	782.99 48.0	48-52 CM	19.1299...	4 R	469.794 L					
7	342 BK-R50B-52 LB	CM	Road-65...	413.1463 1	Black	100	75	699.0982 52.0	48-52 CM	19.42	4 R	419.4589 L					
8	343 BK-R50B-52 LB	CM	Road-65...	466.7066 1	Black	100	75	782.99 52.0	48-52 CM	19.42	4 R	469.794 L					
9	344 BK-M92S-38 LB	CM	Mountai...	1912.1544 1	Silver	100	75	3399.99 38.0	38-40 CM	20.35	4 M	2039.994 M					
10	345 BK-M92S-42 LB	CM	Mountai...	1912.1544 1	Silver	100	75	3399.99 42.0	42-46 CM	20.77	4 M	2039.994 M					
11	346 BK-M92S-44 LB	CM	Mountai...	1912.1544 1	Silver	100	75	3399.99 44.0	42-46 CM	21.1299...	4 M	2039.994 M					
12	347 BK-M92S-48 LB	CM	Mountai...	1912.1544 1	Silver	100	75	3399.99 48.0	48-52 CM	21.42	4 M	2039.994 M					
13	348 BK-M92S-38 LB	CM	Mountai...	1898.0944 1	Black	100	75	3374.99 38.0	38-40 CM	20.35	4 M	2024.994 M					
14	349 BK-M92S-42 LB	CM	Mountai...	1898.0944 1	Black	100	75	3374.99 42.0	42-46 CM	20.77	4 M	2024.994 M					
15	350 BK-M92S-44 LB	CM	Mountai...	1898.0944 1	Black	100	75	3374.99 44.0	42-46 CM	21.1299...	4 M	2024.994 M					
16	351 BK-M92S-48 LB	CM	Mountai...	1898.0944 1	Black	100	75	3374.99 48.0	48-52 CM	21.42	4 M	2024.994 M					
17	352 BK-M60S-38 LB	CM	Mountai...	1117.8559 1	Silver	100	75	2071.4196 38.0	38-40 CM	23.35	4 M	1242.8518 M					
18	353 BK-M60S-38 LB	CM	Mountai...	1265.6195 1	Silver	100	75	2319.99 38.0	38-40 CM	23.35	4 M	1391.994 M					
19	354 BK-M60S-42 LB	CM	Mountai...	1117.8559 1	Silver	100	75	2071.4196 42.0	42-46 CM	23.77	4 M	1242.8518 M					
20	355 BK-M60S-42 LB	CM	Mountai...	1265.6195 1	Silver	100	75	2319.99 42.0	42-46 CM	23.77	4 M	1391.994 M					
21	356 BK-M60S-46 LB	CM	Mountai...	1117.8559 1	Silver	100	75	2071.4196 46.0	42-46 CM	24.1299...	4 M	1242.8518 M					
22	357 BK-M60S-46 LB	CM	Mountai...	1265.6195 1	Silver	100	75	2319.99 46.0	42-46 CM	24.1299...	4 M	1391.994 M					

SQL History

Line 12 Column 23 Insert Modified Windows

1208 PM

Data Loaded into Oracle



Execution Results

Execution History Logging Step Metrics Performance Graph Metrics Preview data

First rows Last rows Off

#	SalesTerritoryKey	SalesTerritoryID	SalesTerritoryRegion	SalesTerritoryCountry	SalesTerritoryGroup
1	1.0	1.0	Northwest	United States	North America
2	2.0	2.0	Northeast	United States	North America
3	3.0	3.0	Central	United States	North America
4	4.0	4.0	Southwest	United States	North America
5	5.0	5.0	Southeast	United States	North America
6	6.0	6.0	Canada	Canada	North America
7	7.0	7.0	France	France	Europe
8	8.0	8.0	Germany	Germany	Europe
9	9.0	9.0	Australia	Australia	Pacific
10	10.0	10.0	United Kingdom	United Kingdom	Europe

Territory Table Load

Data Warehousing and Business Intelligence

The screenshot shows the Oracle SQL Developer interface. The query window contains the SQL statement: `select * from sales_territory;`. The query result is displayed in a table with 11 rows and 5 columns: SALES_TERRITORYKEY, SALES_TERRITORYID, SALES_TERRITORYREGION, SALES_TERRITORYCOUNTRY, and SALES_TERRITORYGROUP.

SALES_TERRITORYKEY	SALES_TERRITORYID	SALES_TERRITORYREGION	SALES_TERRITORYCOUNTRY	SALES_TERRITORYGROUP
1	1	Northwest	United States	North America
2	2	Northeast	United States	North America
3	3	Central	United States	North America
4	4	Southwest	United States	North America
5	5	Southeast	United States	North America
6	6	Canada	Canada	North America
7	7	France	France	Europe
8	8	Germany	Germany	Europe
9	9	Australia	Australia	Pacific
10	10	United Kingdom	United Kingdom	Europe
11	11	0 NA	NA	NA

Data Loaded into Territory Table



Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

☒ First rows ☐ Last rows ☐ Off

#	CustomerKey	CustomerID	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress
1	11000.0	AW00011000	<null>	Jon	V	Yang	N	1966/04/08 00:00:00.000	M	<null>	M	jon24@adventure-works.com
2	11001.0	AW00011001	<null>	Eugene	L	Huang	N	1965/05/14 00:00:00.000	S	<null>	M	eugene10@adventure-works.cc
3	11002.0	AW00011002	<null>	Ruben	<null>	Torres	N	1965/08/12 00:00:00.000	M	<null>	M	ruben35@adventure-works.con
4	11003.0	AW00011003	<null>	Christy	<null>	Zhu	N	1968/02/15 00:00:00.000	S	<null>	F	christy12@adventure-works.coi
5	11004.0	AW00011004	<null>	Elizabeth	<null>	Johnson	N	1968/08/08 00:00:00.000	S	<null>	F	elizabeth5@adventure-works.c
6	11005.0	AW00011005	<null>	Julio	<null>	Ruiz	N	1965/08/05 00:00:00.000	S	<null>	M	julio1@adventure-works.com
7	11006.0	AW00011006	<null>	Janet	G	Alvarez	N	1965/12/06 00:00:00.000	S	<null>	F	janet9@adventure-works.com
8	11007.0	AW00011007	<null>	Marco	<null>	Mehta	N	1964/05/09 00:00:00.000	M	<null>	M	marco14@adventure-works.coi

Customer table Load

Data Warehousing and Business Intelligence

Oracle SQL Developer: HR_SCHEMA

File Edit View Navigate Run Source Team Tools Window Help

Welcome PageNEW.sqlDIM_TIMEHR_SCHEMAOLAP_TakeHome.sqlRelational_1 (Untitled_1)

WorksheetQuery Builder

```
select * from Customers
```

Script OutputQuery ResultExplain PlanQuery Result 1

SQLFetched 50 rows in 0.04 seconds

	CUSTOMERKEY	CUSTOMERID	TITLE	FIRSTNAME	MIDDLENAME	LASTNAME	NAMESTYLE	BIRTHDATE	MARITALSTATUS	CUSTOMER_SUFFIX	GENDER	EMAILADDRESS	YEARLYINCOME	TOTALCHILDREN	NUMBERCHILDRENTATHOME
1	11082AW00011082	(null)	Angela	L	Butler	0	04-AUG-66	S	(null)	F	angelal17@adventure-works.com	130000	0	14	
2	11083AW00011083	(null)	Alyssa	F	Cox	0	15-MAR-66	M	(null)	F	alysa37@adventure-works.com	130000	0	14	
3	11084AW00011084	(null)	Lucas	(null)	Phillips	0	12-SEP-57	S	(null)	M	lucas7@adventure-works.com	80000	2	01	
4	11085AW00011085	(null)	Emily	(null)	Johnson	0	19-JUL-57	S	(null)	F	emily1@adventure-works.com	60000	2	01	
5	11086AW00011086	(null)	Ryan	(null)	Brown	0	23-DEC-57	M	(null)	M	ryan43@adventure-works.com	70000	2	11	
6	11087AW00011087	(null)	Tamara	L	Liang	0	03-OCT-57	M	(null)	F	tamara6@adventure-works.com	70000	3	21	
7	11088AW00011088	(null)	Hunter	(null)	Davis	0	25-NOV-57	M	(null)	M	hunter64@adventure-works.com	80000	2	11	
8	11089AW00011089	(null)	Abigail	M	Price	0	05-FEB-57	S	(null)	F	abigail125@adventure-works.com	80000	2	11	
9	11090AW00011090	(null)	Trevor	(null)	Bryant	0	17-DEC-57	S	(null)	M	trevor19@adventure-works.com	90000	2	01	
10	11091AW00011091	(null)	Dalton	(null)	Perez	0	04-APR-57	M	(null)	M	dalton37@adventure-works.com	90000	2	01	
11	11092AW00011092	(null)	Cheryl	A	Diaz	0	06-MAY-67	M	(null)	F	cheryl48@adventure-works.com	90000	2	21	
12	11093AW00011093	(null)	Aimee	A	He	0	10-SEP-67	M	(null)	F	aimel138@adventure-works.com	100000	0	01	
13	11094AW00011094	(null)	Cedric	W	Ma	0	01-APR-62	S	(null)	M	cedric138@adventure-works.com	70000	1	01	
14	11095AW00011095	(null)	Chad	(null)	Kumar	0	01-SEP-62	S	(null)	M	chad9@adventure-works.com	70000	1	01	
15	11096AW00011096	(null)	Andre	(null)	Amend	0	10-AUG-62	M	(null)	M	andre618@adventure-works.com	60000	1	01	
16	11097AW00011097	(null)	Edwin	R	Nara	0	27-OCT-61	M	(null)	M	edwin398@adventure-works.com	60000	1	01	
17	11098AW00011098	(null)	Mallory	S	Rubio	0	01-MAY-61	S	(null)	F	mallory7@adventure-works.com	60000	1	01	
18	11099AW00011099	(null)	Adam	(null)	Ross	0	08-MAR-61	M	(null)	M	adam38@adventure-works.com	60000	1	01	
19	11100AW00011100	(null)	Latasha	L	Havazzo	0	15-SEP-60	S	(null)	F	latashal0@adventure-works.com	60000	1	01	
20	11101AW00011101	(null)	Abby	L	Sai	0	08-MAY-65	S	(null)	F	abby4@adventure-works.com	70000	0	01	
21	11102AW00011102	(null)	Julia	L	Welson	0	21-APR-65	S	(null)	F	julia78@adventure-works.com	80000	5	51	
22	11103AW00011103	(null)	Cassie	(null)	Chande	0	17-OCT-64	S	(null)	F	cassiel38@adventure-works.com	70000	0	01	

SQL History

Line 12 Column 23InsertModifiedWindows

Data Loaded into Customers table



The screenshot shows the 'Execution Results' tab in SQL Server Enterprise Manager. It displays a table with 9 rows of promotion data. The columns include PromotionKey, PromotionID, PromotionName, DiscountPct, PromotionType, PromotionCategory, StartDate, EndDate, MinQty, and MaxQty. The data is fetched from the 'Promotions' table in the 'AdventureWorks' database.

#	PromotionKey	PromotionID	PromotionName	DiscountPct	PromotionType	PromotionCategory	StartDate	EndDate	MinQty	MaxQty
1	1.0	1.0	No Discount	0.0	No Discount	No Discount	2005/06/01 00:00:00.000	2008/12/31 00:00:00.000	0.0	<null>
2	2.0	2.0	Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2005/07/01 00:00:00.000	2008/06/30 00:00:00.000	11.0	14.0
3	3.0	3.0	Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2005/07/01 00:00:00.000	2008/06/30 00:00:00.000	15.0	24.0
4	4.0	4.0	Volume Discount 25 to 40	0.1	Volume Discount	Reseller	2005/07/01 00:00:00.000	2008/06/30 00:00:00.000	25.0	40.0
5	5.0	5.0	Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2005/07/01 00:00:00.000	2008/06/30 00:00:00.000	41.0	60.0
6	6.0	6.0	Volume Discount over 60	0.2	Volume Discount	Reseller	2005/07/01 00:00:00.000	2008/06/30 00:00:00.000	61.0	<null>
7	7.0	7.0	Mountain-100 Clearance Sale	0.35	Discontinued Product	Reseller	2006/05/15 00:00:00.000	2006/06/30 00:00:00.000	0.0	<null>
8	8.0	8.0	Sport Helmet Discount-2002	0.1	Seasonal Discount	Reseller	2006/07/01 00:00:00.000	2006/07/31 00:00:00.000	0.0	<null>
9	9.0	9.0	Road-650 Overstock	0.3	Excess Inventory	Reseller	2006/07/01 00:00:00.000	2006/08/31 00:00:00.000	0.0	<null>

Promotions Table Load

Data Warehousing and Business Intelligence

The screenshot shows the Oracle SQL Developer interface with a query executed against the HR_SCHEMA. The query is `select * from dim_promotions`. The result set contains 16 rows of promotion data.

PROMOTIONKEY	PROMOTIONID	PROMOTIONNAME	DISCOUNTPCT	PROMOTIONTYPE	PROMOTIONCATEGORY	STARTDATE	ENDDATE	MINQTY	MAXQTY
1	1	1 No Discount	0	No Discount	No Discount	01-JUN-05	31-DEC-08	0	(null)
2	2	2 Volume Discount 11 to 14	0.02	Volume Discount	Reseller	01-JUL-05	30-JUN-08	11	14
3	3	3 Volume Discount 15 to 24	0.05	Volume Discount	Reseller	01-JUL-05	30-JUN-08	15	24
4	4	4 Volume Discount 25 to 40	0.1	Volume Discount	Reseller	01-JUL-05	30-JUN-08	25	40
5	5	5 Volume Discount 41 to 60	0.149999999999999	Volume Discount	Reseller	01-JUL-05	30-JUN-08	41	60
6	6	6 Volume Discount over 60	0.2	Volume Discount	Reseller	01-JUL-05	30-JUN-08	61	(null)
7	7	7 Mountain-100 Clearance Sale	0.349999999999999	Discontinued Product	Reseller	15-MAY-06	30-JUN-06	0	(null)
8	8	8 Sport Helmet Discount-2002	0.1	Seasonal Discount	Reseller	01-JUL-06	31-JUL-06	0	(null)
9	9	9 Road-650 Overstock	0.299999999999999	Excess Inventory	Reseller	01-JUL-06	31-AUG-06	0	(null)
10	10	10 Mountain Tire Sale	0.5	Excess Inventory	Customer	15-JUN-07	30-AUG-07	0	(null)
11	11	11 Sport Helmet Discount-2003	0.149999999999999	Seasonal Discount	Reseller	01-JUL-07	31-JUL-07	0	(null)
12	12	12 LL Road Frame Sale	0.349999999999999	Excess Inventory	Reseller	01-JUL-07	15-AUG-07	0	(null)
13	13	13 Touring-3000 Promotion	0.149999999999999	New Product	Reseller	01-JUL-07	30-SEP-07	0	(null)
14	14	14 Touring-1000 Promotion	0.2	New Product	Reseller	01-JUL-07	30-SEP-07	0	(null)
15	15	15 Half-Price Pedal Sale	0.5	Seasonal Discount	Customer	15-AUG-07	15-SEP-07	0	(null)
16	16	16 Mountain-500 Silver Clearance Sale	0.4	Discontinued Product	Reseller	01-MAY-08	30-JUN-08	0	(null)

Data loaded into Promotions



The screenshot shows the 'Execution Results' window in SQL Developer. The query executed is `select * from time_table_load`. The result set contains 8 rows of time-related data.

#	DateKey	FullDate	DayNumberOfWeek	DayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	WeekNumberOfYear	MonthName	MonthNumberOfYear	CalendarQuarter	CalendarYear
1	20050101.0	2005/01/01	7.0	Saturday	1.0	1.0	1.0	January	1.0	1.0	2005
2	20050102.0	2005/01/02	1.0	Sunday	2.0	2.0	2.0	January	1.0	1.0	2005
3	20050103.0	2005/01/03	2.0	Monday	3.0	3.0	2.0	January	1.0	1.0	2005
4	20050104.0	2005/01/04	3.0	Tuesday	4.0	4.0	2.0	January	1.0	1.0	2005
5	20050105.0	2005/01/05	4.0	Wednesday	5.0	5.0	2.0	January	1.0	1.0	2005
6	20050106.0	2005/01/06	5.0	Thursday	6.0	6.0	2.0	January	1.0	1.0	2005
7	20050107.0	2005/01/07	6.0	Friday	7.0	7.0	2.0	January	1.0	1.0	2005
8	20050108.0	2005/01/08	7.0	Saturday	8.0	8.0	2.0	January	1.0	1.0	2005

Time Table Load

Data Warehousing and Business Intelligence

Oracle SQL Developer: HR_SCHEMA

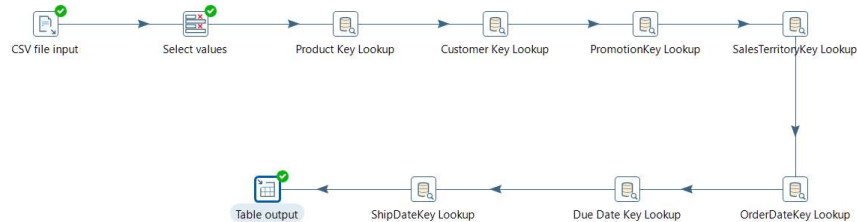
Script Output x Query Result x Explain Plan x Query Result 1 x

SQL Fetched 50 rows in 0.003 seconds

DATEKEY	FULLDATE	DAYNUMBEROFWEEK	DAYNAMEOFWEEK	DAYNUMBEROFMONTH	DAYNUMBEROFYEAR	WEEKNUMBEROFYEAR	MONTH-NAME	MONTHNUMBEROFYEAR	CALENDARQUARTER	CALENDARYEAR	CALENDARSEMIESTER	FISCALQUARTER	FISCAL
1	20050101 01-JAN-05	7	Saturday	1	1	1	1 January	1	1	2005	1	3	
2	20050102 02-JAN-05	1	Sunday	2	2	2	2 January	1	1	2005	1	3	
3	20050103 03-JAN-05	2	Monday	3	3	3	3 January	1	1	2005	1	3	
4	20050104 04-JAN-05	3	Tuesday	4	4	4	4 January	1	1	2005	1	3	
5	20050105 05-JAN-05	4	Wednesday	5	5	5	5 January	1	1	2005	1	3	
6	20050106 06-JAN-05	5	Thursday	6	6	6	6 January	1	1	2005	1	3	
7	20050107 07-JAN-05	6	Friday	7	7	7	7 January	1	1	2005	1	3	
8	20050108 08-JAN-05	7	Saturday	8	8	8	8 January	1	1	2005	1	3	
9	20050109 09-JAN-05	1	Sunday	9	9	9	9 January	1	1	2005	1	3	
10	20050110 10-JAN-05	2	Monday	10	10	10	10 January	1	1	2005	1	3	
11	20050111 11-JAN-05	3	Tuesday	11	11	11	11 January	1	1	2005	1	3	
12	20050112 12-JAN-05	4	Wednesday	12	12	12	12 January	1	1	2005	1	3	
13	20050113 13-JAN-05	5	Thursday	13	13	13	13 January	1	1	2005	1	3	
14	20050114 14-JAN-05	6	Friday	14	14	14	14 January	1	1	2005	1	3	
15	20050115 15-JAN-05	7	Saturday	15	15	15	15 January	1	1	2005	1	3	
16	20050116 16-JAN-05	1	Sunday	16	16	16	16 January	1	1	2005	1	3	
17	20050117 17-JAN-05	2	Monday	17	17	17	17 January	1	1	2005	1	3	
18	20050118 18-JAN-05	3	Tuesday	18	18	18	18 January	1	1	2005	1	3	
19	20050119 19-JAN-05	4	Wednesday	19	19	19	19 January	1	1	2005	1	3	
20	20050120 20-JAN-05	5	Thursday	20	20	20	20 January	1	1	2005	1	3	
21	20050121 21-JAN-05	6	Friday	21	21	21	21 January	1	1	2005	1	3	
22	20050122 22-JAN-05	7	Saturday	22	22	22	22 January	1	1	2005	1	3	

Data Loaded into Time dimension table.

- After all the dimension tables were loaded, I have loaded Transactions table from excel into oracle.
- Using Excel input in pentaho for transactions table was causing problem, for that reason had to save the sheet in csv format and then used in pentaho with csv input.
- Various lookup transformations were applied for all the referencing foreign keys in order to pass the foreign key value only if it is available in the dimension tables.



Execution Results

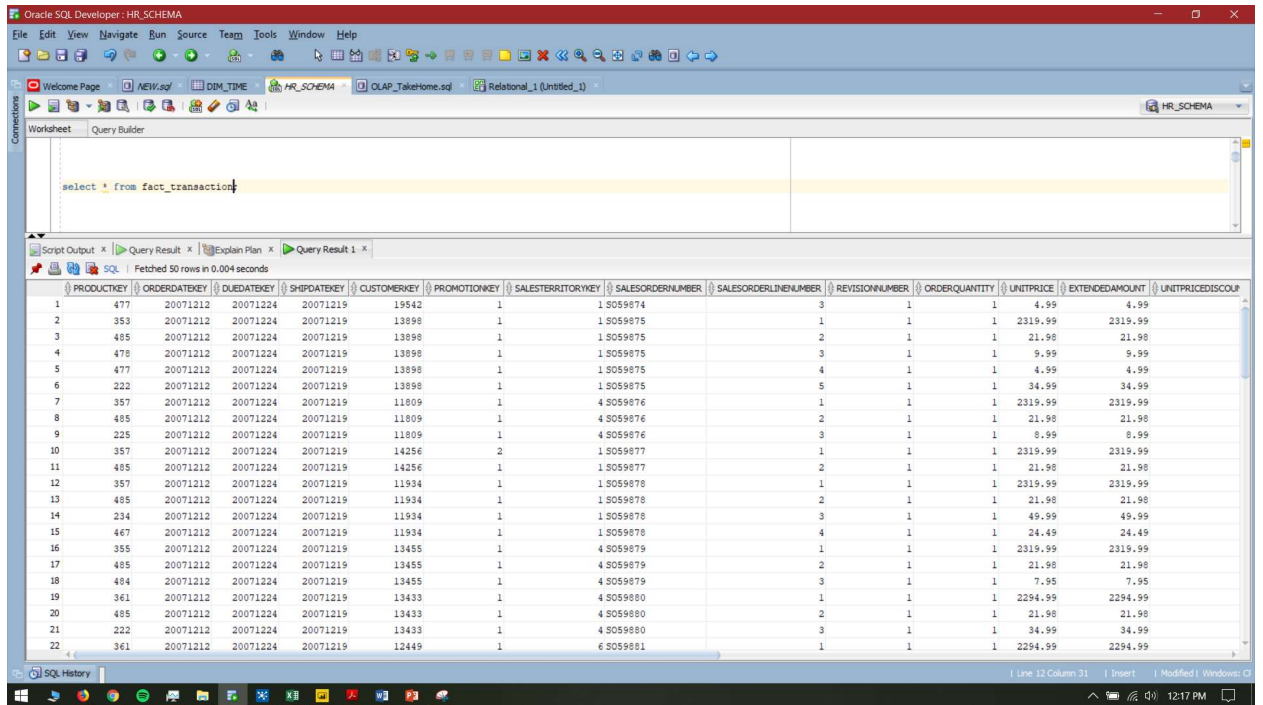
Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

UnitPrice	ExtendedAmount	UnitPriceDiscountPct	DiscountAmount	ProductStandardCost	TotalProductCost	SalesAmount	TaxAmt	Freight	CarrierTrackingNumber	CustomerPONumber
3.99	3.99	0	0	1.4923	1.4923	3.99	0.3192	0.0998	<null>	<null>
34.99	34.99	0	0	13.0863	13.0863	34.99	2.7992	0.8748	<null>	<null>
1700.99	1700.99	0	0	1082.51	1082.51	1700.99	136.0792	42.5248	<null>	<null>
34.99	34.99	0	0	13.0863	13.0863	34.99	2.7992	0.8748	<null>	<null>
53.99	53.99	0	0	41.5723	41.5723	53.99	4.3192	1.3498	<null>	<null>
8.99	8.99	0	0	6.9223	6.9223	8.99	0.7192	0.2248	<null>	<null>
539.99	539.99	0	0	343.6496	343.6496	539.99	43.1992	13.4998	<null>	<null>
21.49	21.49	0	0	8.0373	8.0373	21.49	1.7192	0.5373	<null>	<null>
539.99	539.99	0	0	343.6496	343.6496	539.99	43.1992	13.4998	<null>	<null>

Transaction data load

Data Warehousing and Business Intelligence



The screenshot displays the Oracle SQL Developer interface. The 'Query Result' tab shows the results of a query executed on the 'fact_transaction' table. The query is 'select * from fact_transaction'. The results are displayed in a table with 12 columns: PRODUCTKEY, ORDERDATEKEY, DUEDATEKEY, SHIPDATEKEY, CUSTOMERKEY, PROMOTIONKEY, SALESTERRITORYKEY, SALESORDERNUMBER, SALESORDERLINENUMBER, REVISIONNUMBER, ORDERQUANTITY, UNITPRICE, EXTENDEDAMOUNT, and UNITPRICEDISCOUNT. The table contains 22 rows of data, representing transactions from 2007. The status bar at the bottom indicates 'Line 12 Column 21' and 'Insert' mode.

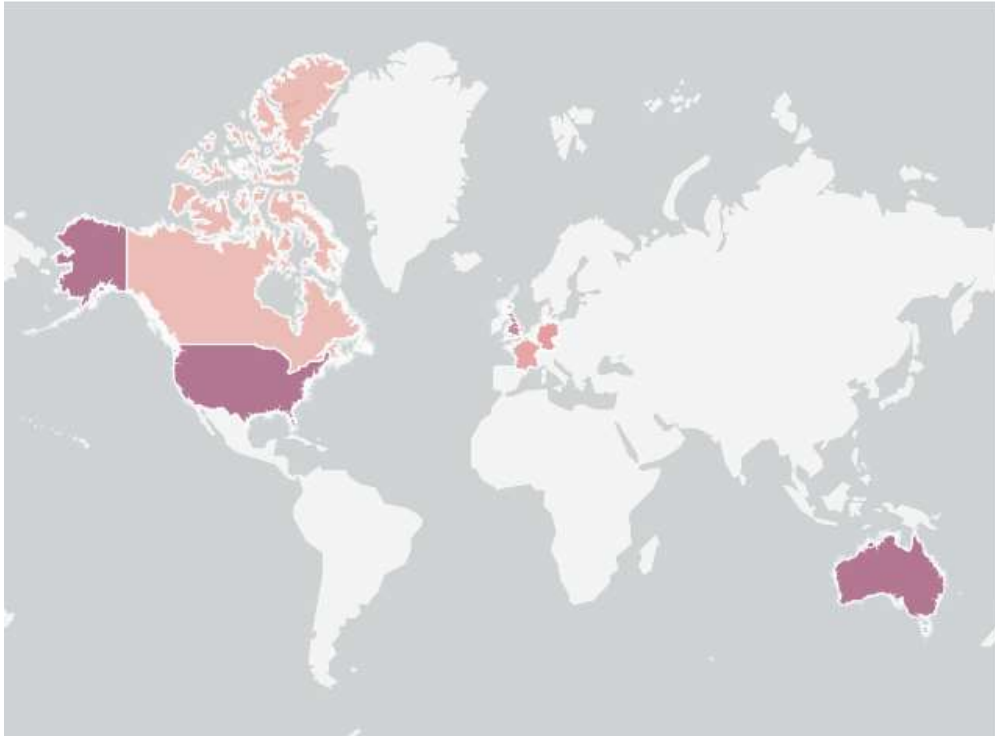
	PRODUCTKEY	ORDERDATEKEY	DUEDATEKEY	SHIPDATEKEY	CUSTOMERKEY	PROMOTIONKEY	SALESTERRITORYKEY	SALESORDERNUMBER	SALESORDERLINENUMBER	REVISIONNUMBER	ORDERQUANTITY	UNITPRICE	EXTENDEDAMOUNT	UNITPRICEDISCOUNT
1	477	20071212	20071224	20071219	19542	1	1	1	3	1	1	4.99	4.99	
2	353	20071212	20071224	20071219	13898	1	1	1	1	1	1	2319.99	2319.99	
3	485	20071212	20071224	20071219	13898	1	1	1	2	1	1	21.98	21.98	
4	478	20071212	20071224	20071219	13898	1	1	1	3	1	1	9.99	9.99	
5	477	20071212	20071224	20071219	13898	1	1	1	4	1	1	4.99	4.99	
6	222	20071212	20071224	20071219	13898	1	1	1	5	1	1	34.99	34.99	
7	357	20071212	20071224	20071219	11809	1	1	1	1	1	1	2319.99	2319.99	
8	485	20071212	20071224	20071219	11809	1	1	1	2	1	1	21.98	21.98	
9	225	20071212	20071224	20071219	11809	1	1	1	3	1	1	8.99	8.99	
10	357	20071212	20071224	20071219	14256	2	1	1	1	1	1	2319.99	2319.99	
11	485	20071212	20071224	20071219	14256	1	1	1	2	1	1	21.98	21.98	
12	357	20071212	20071224	20071219	11934	1	1	1	1	1	1	2319.99	2319.99	
13	485	20071212	20071224	20071219	11934	1	1	1	2	1	1	21.98	21.98	
14	234	20071212	20071224	20071219	11934	1	1	1	3	1	1	49.99	49.99	
15	467	20071212	20071224	20071219	11934	1	1	1	4	1	1	24.49	24.49	
16	355	20071212	20071224	20071219	13455	1	1	1	1	1	1	2319.99	2319.99	
17	485	20071212	20071224	20071219	13455	1	1	1	2	1	1	21.98	21.98	
18	484	20071212	20071224	20071219	13455	1	1	1	3	1	1	7.95	7.95	
19	361	20071212	20071224	20071219	13433	1	1	1	1	1	1	2294.99	2294.99	
20	485	20071212	20071224	20071219	13433	1	1	1	2	1	1	21.98	21.98	
21	222	20071212	20071224	20071219	13433	1	1	1	3	1	1	34.99	34.99	
22	361	20071212	20071224	20071219	12449	1	1	1	1	1	1	2294.99	2294.99	

Data loaded into Transaction table.

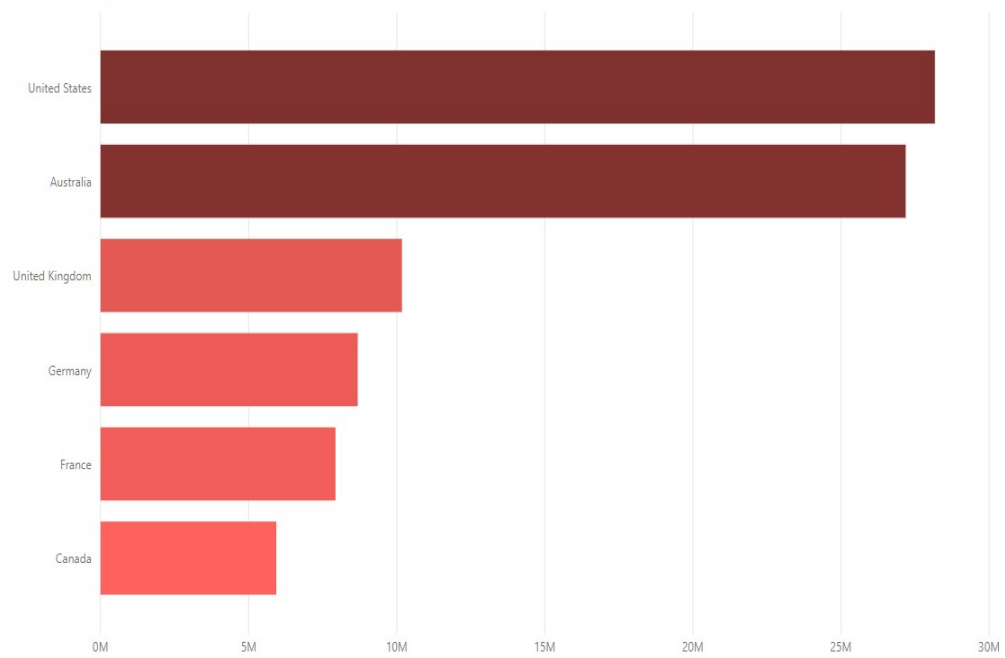
Data Warehousing and Business Intelligence

BI Dashboard

- The Data from oracle tables is imported into PowerBi to perform visualizations.

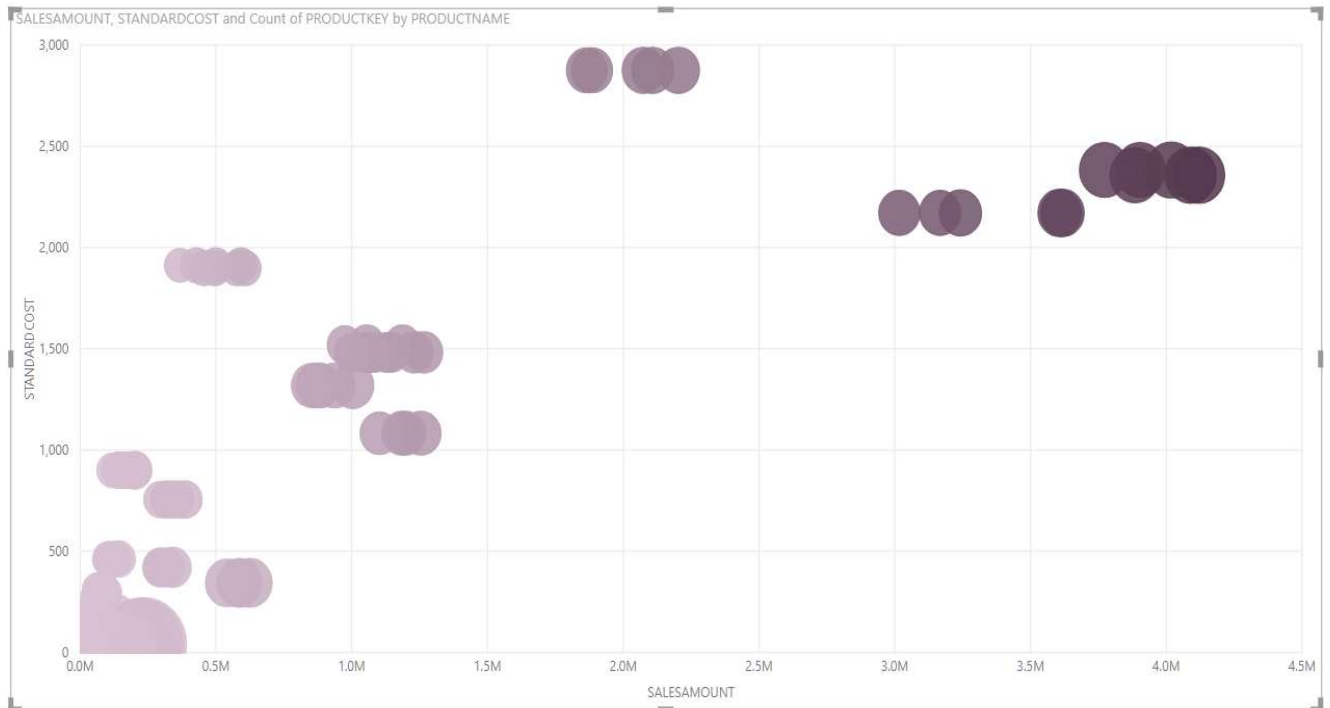


SALESAMOUNT by SALESTERRITORYCOUNTRY



Data Warehousing and Business Intelligence

- The above visualizations show, how the revenue is generated from various countries.
- From the observation, we can see that Australia and United States are two places with high sales amount generation, which gives us the insight that Production should be more focused in United States and Australia to increase profits, with lesser focus on France and Canada.



- The above Visualization of Products, their sales, their Standard cost and sales amount gives us an insight that there are few products with high standard cost (2,000) but they are not really generating great sales amount. So its better to reduce the production of those products and focus more on products generating high sales amount or products with low production cost.

Data Warehousing and Business Intelligence

SALESAMOUNT by PRODUCTNAME



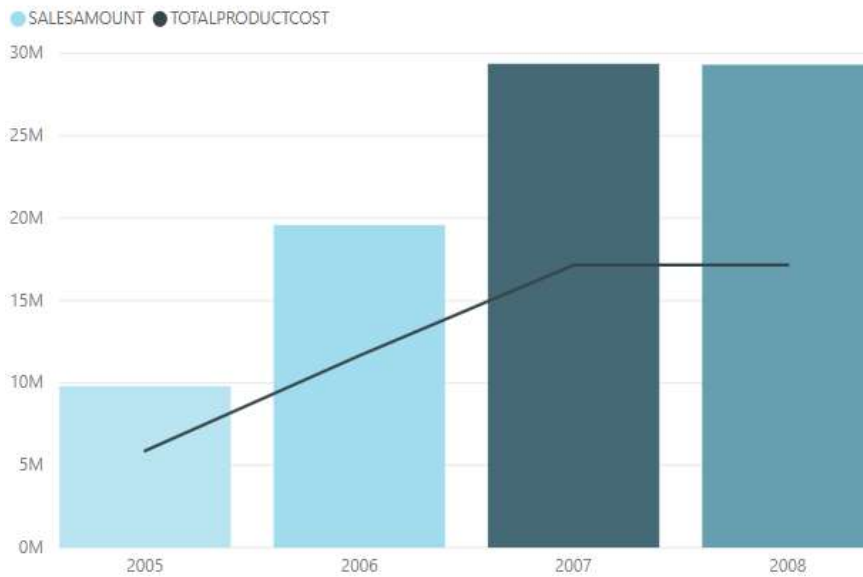
SALESAMOUNT by PRODUCTNAME



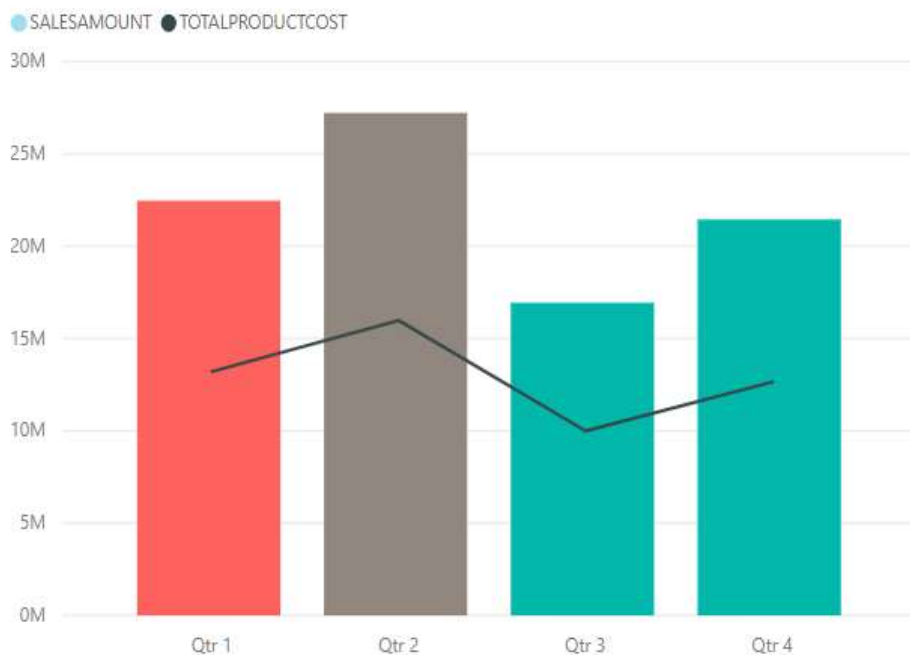
- The above visualization shows the bikes with highest sales (Green) amount generation and those with the lowest (Red) sales amount generation.
- This gives us an insight that, probably customers are preferring the bikes in green zone more than the ones in red zone. So its better to focus the production on Bikes with high sales amount generation and minimize the focus on bikes in red zone.

Data Warehousing and Business Intelligence

SALESAMOUNT and TOTALPRODUCTCOST by Year

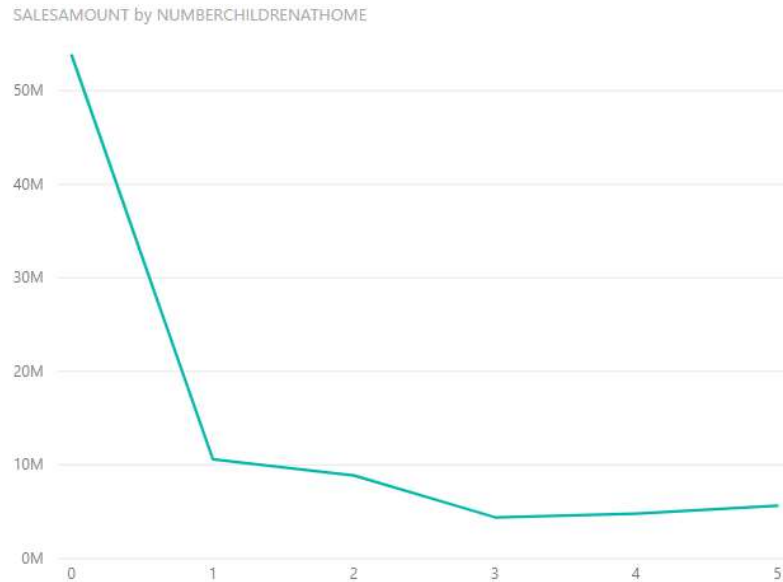


SALESAMOUNT and TOTALPRODUCTCOST by Quarter

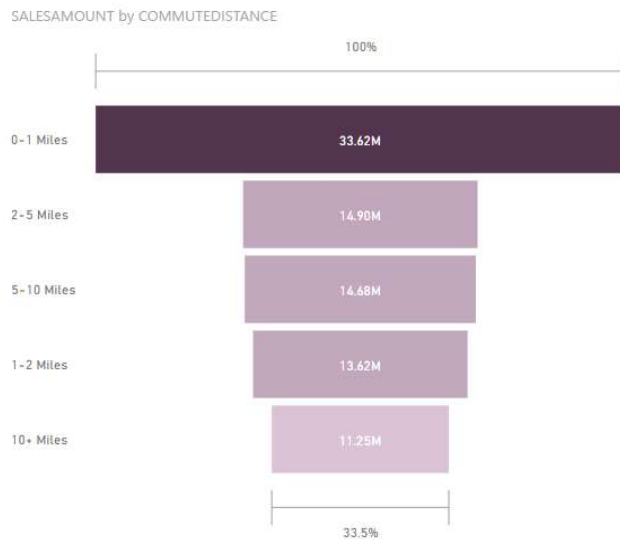


- The above visualizations show that the sales amount has not changed from 2007 and 2008; The Company should be implementing better marketing strategies in order to increase the sales revenue year by year.
- In the second bar graph, we can observe that there is more sales amount generation in Quarter 2, so it would be wise for the business to provide promotional offers to further increase the sales in Quarter 2, or business can provide promotional offers in Quarter 3 and 4 to increase sales in Q3 and Q4.

Data Warehousing and Business Intelligence



- The above graph between the number of children at home and the sales amount generated gives us the information that customers with no children are responsible for the majority of sales revenue, which might help the marketing team in focusing people with no children as potential customers.

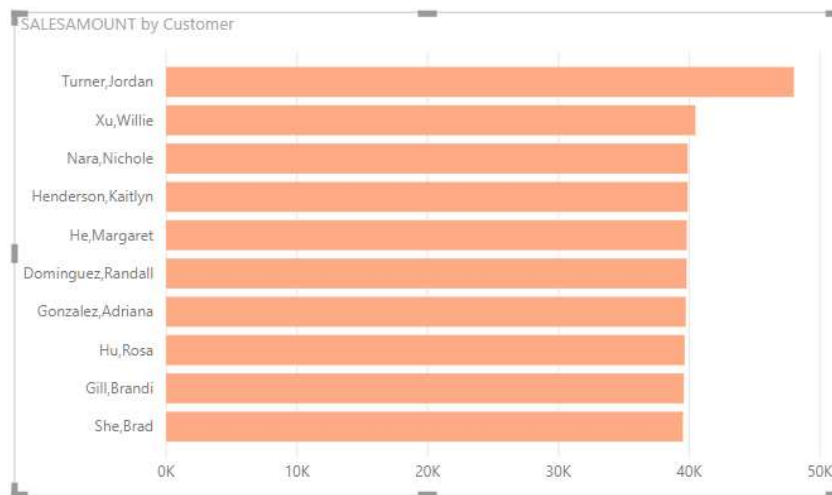


- The above visualization gives us information of our customer base with respect to their commute distances.
- Through this, we can observe that people who are having lesser commute distances are interested in our products than people having higher commute distances, which is somewhat obvious. However, we can also see that people with commute distances more 10+ Miles are also interested in our products, which alerts us not to overlook the people having higher commute distances, as they can also be our potential customers.

Data Warehousing and Business Intelligence



- The above visualization shows how owning a house is related to the generated sales amount.
- Through the above visualization, we can infer that people owning houses are responsible for more than 60% of sales amount. Therefore, we should be giving some good promotion offers to customers who live in rented houses to increase their sales.
- A strategy like “we would help you in moving your bike when you move to different place” might really encourage them in buying the products.



- The above visualization gives us the top 10 customers with respect to sales amount they have generated.
- This helps us in giving some gifts or extra discounts to these customers, which would work as a great marketing strategy.