ASSIGNMENT 4

SUBMITTED BY-

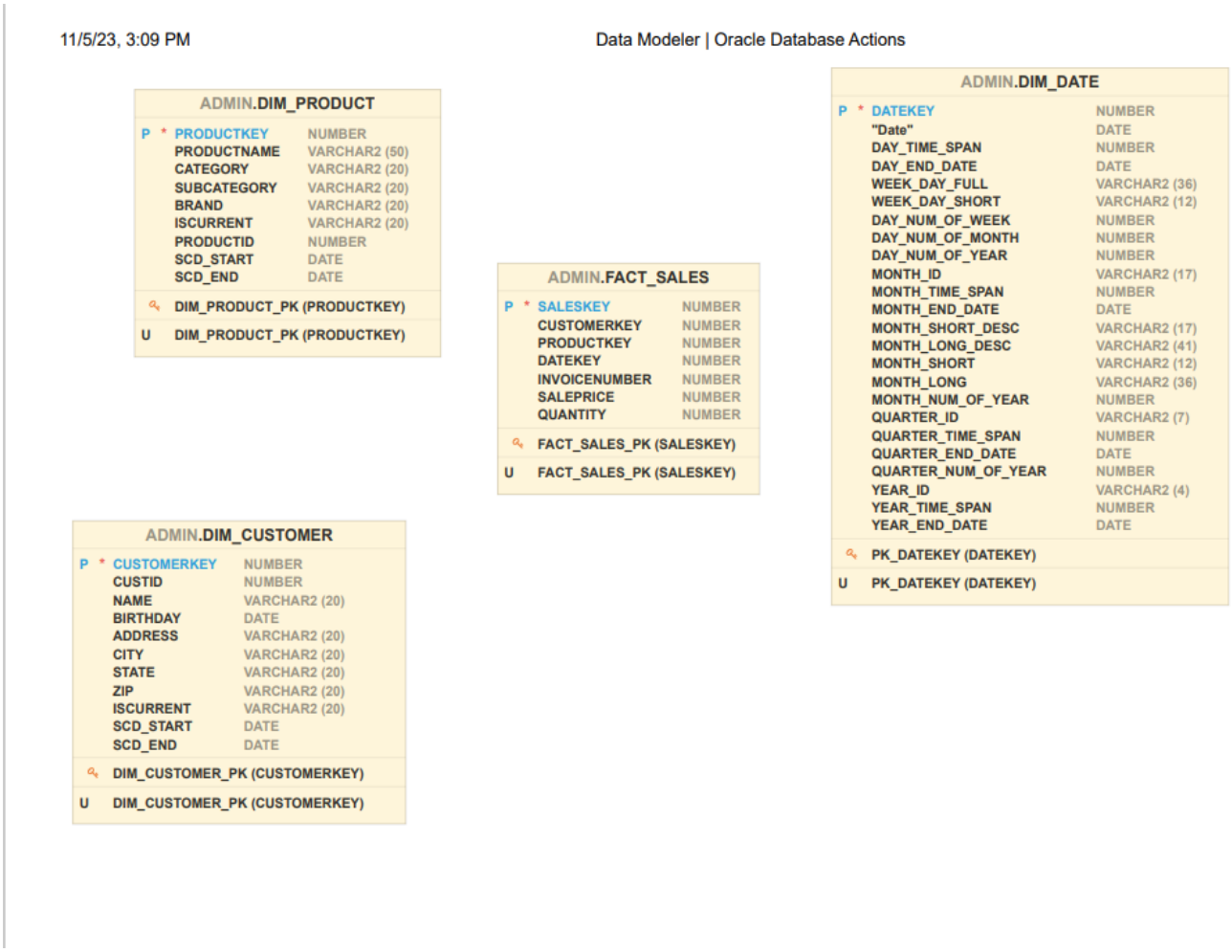
. Anjali (50526420) and .Kanupriya (50518452)

Executive Summary

After reviewing all proposals, our team has decided to merge multiple proposed database designs to form the final production database for the organization. So, we renamed the existing database tables and created new tables: Dim_Product, Dim_Customer, Dim_Date and Fact_Sales using the queries mentioned in the text files. After creating the tables, we generated the database diagram. We modified the date dimension according to the requirements of the client. We added a few records manually to the customer and product dimension tables. After creating the data warehouse design in Oracle, the next step is to create an ETL pipeline and populate data in our data warehouse. So, we installed Talend Open Studio which is freely available on the internet. To establish connection between oracle cloud and talend we need to address password- based and certificate-based authentication. For this, we create a Wallet folder and put certificate files needed by the JDBC driver to complete the connection. We place this folder in C drive so it's easy to locate. Then we download ojdbc8-full.tar file, unzip it and copy the following files in a separate folder on our pc - ojdbc8.jar, oraclepki.jar, osdt_core.jar, osdt_cert.jar. Then we open Talend and create a new job TestConnect to test the database connectivity. We create 3 classpath variables for oraclepki.jar, osdt_core.jar, osdt_cert.jar to help OJDBC driver to access our wallet files to establish a secure connection to our database. Then we select ojdbc8.jar file in the modules folder. Upon entering login name, password, connection string, and additional parameters in the database connection dialog box we finally establish a connection with the database. After we have established connectivity with the database, it's time to test the ability to read the database. For this, we open the TestConnect job, and drag and drop components on the design canvas of the job. We use library

components, OracleCloud input component and file output component for this job. We connect OracleCloud input to the file output component, and we run a select SQL command and ensure that the data can be read. After the test job runs successfully, we create different jobs to load our slowly changing dimension tables using the data files (in csv form) provided by the client. First, we retrieve the schema from Oracle database. Then we again use library components, the database, and the file-delimited loader for both the product and customer dimension tables. Finally, we create a job to load the fact table by establishing foreign key constraints using tMap component in Talend and run the job. The successful run of the job completes our ETL pipeline.

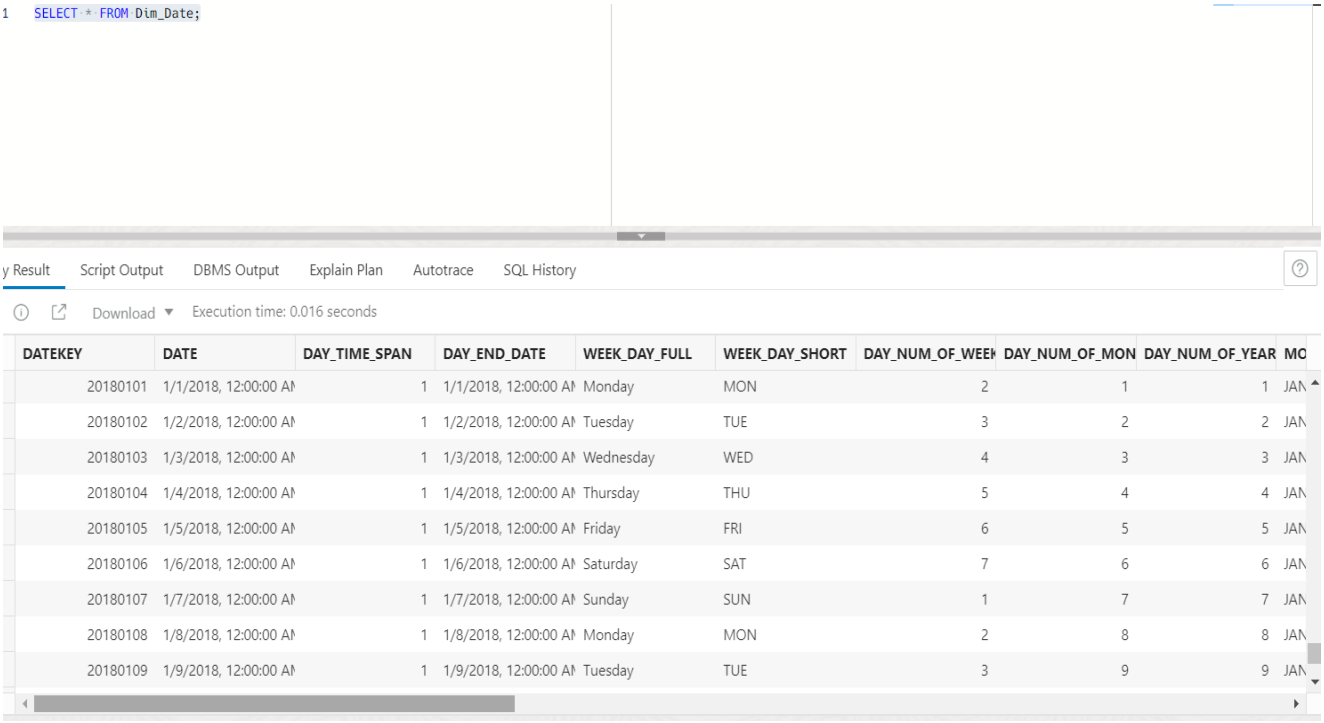
Database Diagram :



Answer the following questions in your report: Based on the diagram generated, what is this database missing that you'd expect to see? Why might it be missing this component? Use materials we've discussed in class and research (citing sources) to write no more than 2-4 sentences in response.

The diagram is missing the relationships between the tables. It is missing as we did not establish foreign key relationships between the fact table and other dimension tables in Oracle Cloud. We didn't do so as enforcing constraints in the database slows down the load process.

Select * from Dim_Date Query Screenshot



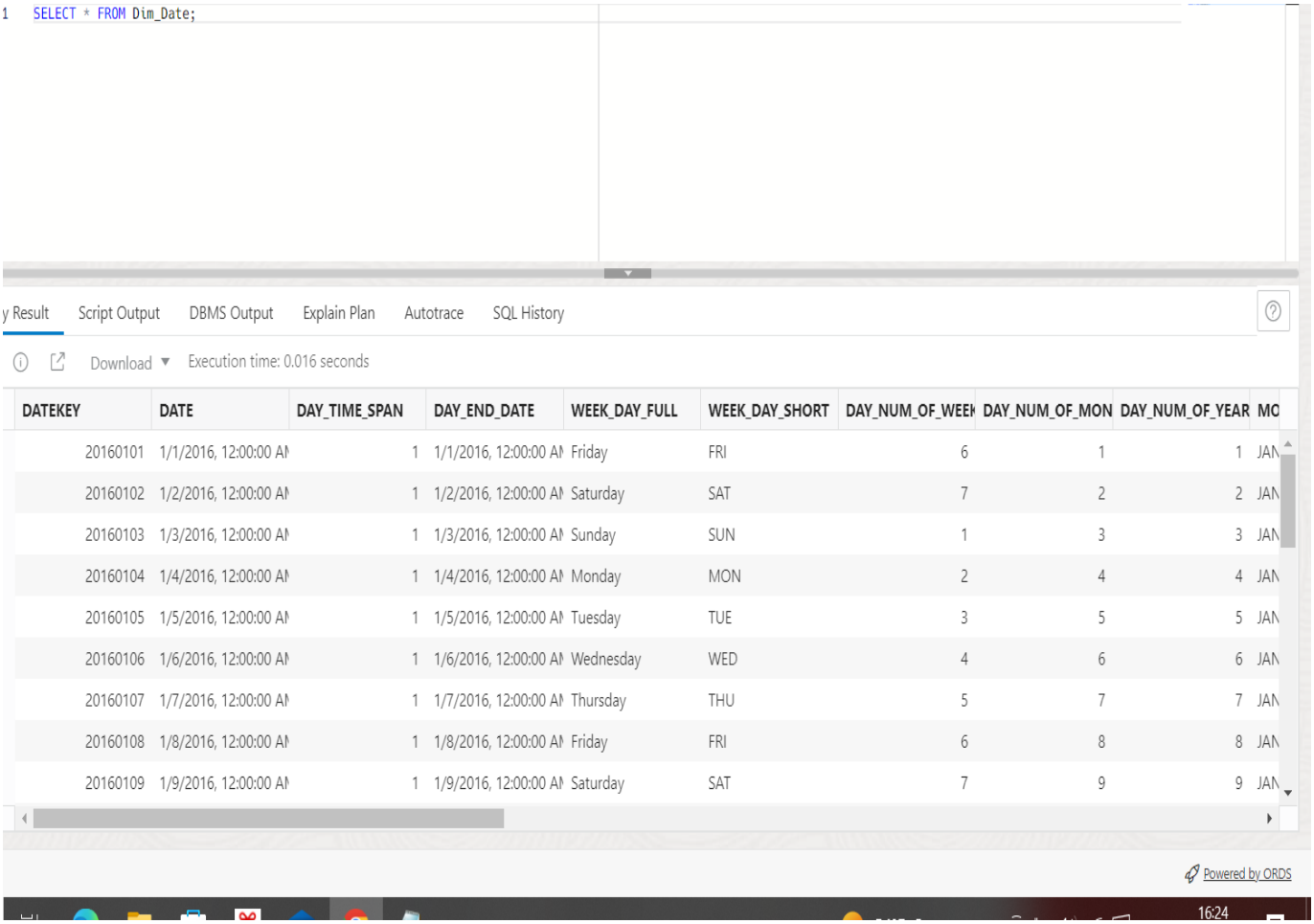
```
1  SELECT * FROM Dim_Date;
```

y Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▾ Execution time: 0.016 seconds

DATEKEY	DATE	DAY_TIME_SPAN	DAY_END_DATE	WEEK_DAY_FULL	WEEK_DAY_SHORT	DAY_NUM_OF_WEEK	DAY_NUM_OF_MON	DAY_NUM_OF_YEAR	MO
20180101	1/1/2018, 12:00:00 AM	1	1/1/2018, 12:00:00 AM	Monday	MON	2	1	1	JAN
20180102	1/2/2018, 12:00:00 AM	1	1/2/2018, 12:00:00 AM	Tuesday	TUE	3	2	2	JAN
20180103	1/3/2018, 12:00:00 AM	1	1/3/2018, 12:00:00 AM	Wednesday	WED	4	3	3	JAN
20180104	1/4/2018, 12:00:00 AM	1	1/4/2018, 12:00:00 AM	Thursday	THU	5	4	4	JAN
20180105	1/5/2018, 12:00:00 AM	1	1/5/2018, 12:00:00 AM	Friday	FRI	6	5	5	JAN
20180106	1/6/2018, 12:00:00 AM	1	1/6/2018, 12:00:00 AM	Saturday	SAT	7	6	6	JAN
20180107	1/7/2018, 12:00:00 AM	1	1/7/2018, 12:00:00 AM	Sunday	SUN	1	7	7	JAN
20180108	1/8/2018, 12:00:00 AM	1	1/8/2018, 12:00:00 AM	Monday	MON	2	8	8	JAN
20180109	1/9/2018, 12:00:00 AM	1	1/9/2018, 12:00:00 AM	Tuesday	TUE	3	9	9	JAN

Updated Select * from Dim_Date Query Screenshot



```
1  SELECT * FROM Dim_Date;
```

y Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▼ Execution time: 0.016 seconds

DATEKEY	DATE	DAY_TIME_SPAN	DAY_END_DATE	WEEK_DAY_FULL	WEEK_DAY_SHORT	DAY_NUM_OF_WEEK	DAY_NUM_OF_MON	DAY_NUM_OF_YEAR	MO
20160101	1/1/2016, 12:00:00 AM	1	1/1/2016, 12:00:00 AM	Friday	FRI	6	1	1	JAN
20160102	1/2/2016, 12:00:00 AM	1	1/2/2016, 12:00:00 AM	Saturday	SAT	7	2	2	JAN
20160103	1/3/2016, 12:00:00 AM	1	1/3/2016, 12:00:00 AM	Sunday	SUN	1	3	3	JAN
20160104	1/4/2016, 12:00:00 AM	1	1/4/2016, 12:00:00 AM	Monday	MON	2	4	4	JAN
20160105	1/5/2016, 12:00:00 AM	1	1/5/2016, 12:00:00 AM	Tuesday	TUE	3	5	5	JAN
20160106	1/6/2016, 12:00:00 AM	1	1/6/2016, 12:00:00 AM	Wednesday	WED	4	6	6	JAN
20160107	1/7/2016, 12:00:00 AM	1	1/7/2016, 12:00:00 AM	Thursday	THU	5	7	7	JAN
20160108	1/8/2016, 12:00:00 AM	1	1/8/2016, 12:00:00 AM	Friday	FRI	6	8	8	JAN
20160109	1/9/2016, 12:00:00 AM	1	1/9/2016, 12:00:00 AM	Saturday	SAT	7	9	9	JAN

Powered by ORDS

16:24

Updated Dim_Date Script Screenshot

Modified_Dim_Date Query - Notepad

File Edit Format View Help

```
SELECT
TO_NUMBER(TRIM(leading '0' FROM TO_CHAR(CurrDate,'yyyymmdd'))) as DATEKEY,
CurrDate AS "Date",
1 AS Day_Time_Span,
CurrDate AS Day_End_Date,
TO_CHAR(CurrDate,'Day') AS Week_Day_Full,
TO_CHAR(CurrDate,'DY') AS Week_Day_Short,
TO_NUMBER(TRIM(leading '0' FROM TO_CHAR(CurrDate,'D'))) AS Day_Num_of_Week,
TO_NUMBER(TRIM(leading '0' FROM TO_CHAR(CurrDate,'DD'))) AS Day_Num_of_Month,
TO_NUMBER(TRIM(leading '0' FROM TO_CHAR(CurrDate,'DDD'))) AS Day_Num_of_Year,
UPPER(TO_CHAR(CurrDate,'Mon') || '-' || TO_CHAR(CurrDate,'YYYY')) AS Month_ID,
-- 31 AS Month_Time_Span,
MAX(TO_NUMBER(TO_CHAR(CurrDate, 'DD'))) OVER (PARTITION BY TO_CHAR(CurrDate,'Mon')) AS Month_Time_Span,
--to_date('31-JAN-2010','DD-MON-YYYY') AS Month_End_Date,
MAX(CurrDate) OVER (PARTITION BY TO_CHAR(CurrDate,'Mon')) as Month_End_Date,
TO_CHAR(CurrDate,'Mon') || ' ' || TO_CHAR(CurrDate,'YYYY') AS Month_Short_Desc,
RTRIM(TO_CHAR(CurrDate,'Month')) || ' ' || TO_CHAR(CurrDate,'YYYY') AS Month_Long_Desc,
TO_CHAR(CurrDate,'Mon') AS Month_Short,
TO_CHAR(CurrDate,'Month') AS Month_Long,
TO_NUMBER(TRIM(leading '0'FROM TO_CHAR(CurrDate,'MM'))) AS Month_Num_of_Year,
'Q' || UPPER(TO_CHAR(CurrDate,'Q') || '-' || TO_CHAR(CurrDate,'YYYY')) AS Quarter_ID,
-- 31 AS Quarter_Time_Span,
COUNT(*) OVER (PARTITION BY TO_CHAR(CurrDate,'Q')) AS Quarter_Time_Span,
-- to_date('31-JAN-2010','DD-MON-YYYY') AS Quarter_End_Date,
MAX(CurrDate) OVER (PARTITION BY TO_CHAR(CurrDate,'Q')) AS Quarter_End_Date,
TO_NUMBER(TO_CHAR(CurrDate,'Q')) AS Quarter_Num_of_Year,
TO_CHAR(CurrDate,'YYYY') AS Year_ID,
--31 AS Year_Time_Span,
COUNT(*) OVER (PARTITION BY TO_CHAR(CurrDate,'YYYY')) AS Year_Time_Span,
-- to_date('31-JAN-2010','DD-MON-YYYY') AS Year_End_Date
MAX(CurrDate) OVER (PARTITION BY TO_CHAR(CurrDate,'YYYY')) Year_End_Date
FROM
(SELECT level n,
-- Calendar starts at the day after this date.
TO_DATE('31/12/2015','DD/MM/YYYY') + NUMTODSINTERVAL(level,'day') CurrDate
FROM dual
-- Change for the number of days to be added to the table.
CONNECT BY level <= 4018)
ORDER BY CurrDate
;

ALTER TABLE DIM_DATE
ADD CONSTRAINT pk_datekey PRIMARY KEY (DATEKEY);
```

Select * from Dim_Product Query Screenshot

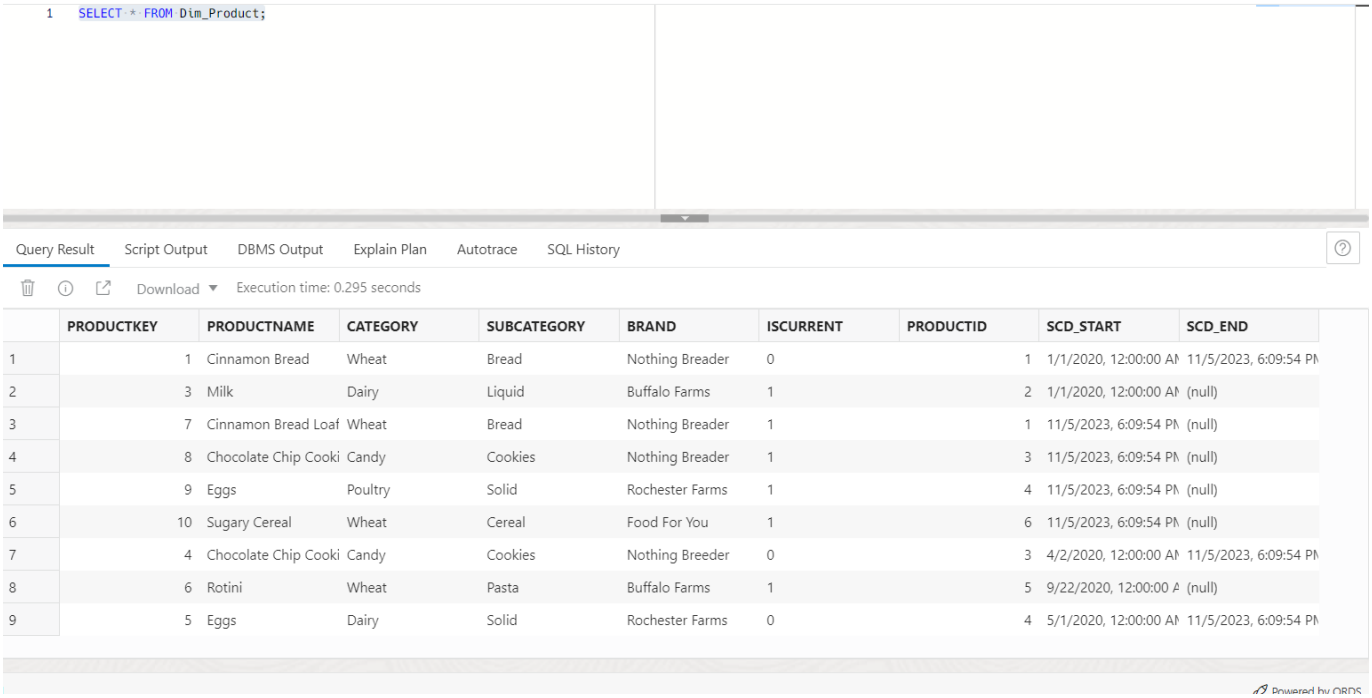
```
28  SELECT * FROM DIM_PRODUCT;
```

Query Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▼ Execution time: 0.004 seconds

	PRODUCTKEY	PRODUCTNAME	CATEGORY	SUBCATEGORY	BRAND	ISCURRENT	PRODUCTID	SCD_START	SCD_END
1	1	Cinnamon Bread	Wheat	Bread	Nothing Breader	1	1	1/1/2020, 12:00:00 AM	(null)
2	3	Milk	Dairy	Liquid	Buffalo Farms	1	2	1/1/2020, 12:00:00 AM	(null)
3	4	Chocolate Chip Cooki	Candy	Cookies	Nothing Breeder	1	3	4/2/2020, 12:00:00 AM	(null)
4	6	Rotini	Wheat	Pasta	Buffalo Farms	1	5	9/22/2020, 12:00:00 A	(null)
5	5	Eggs	Dairy	Solid	Rochester Farms	1	4	5/1/2020, 12:00:00 AM	(null)

Updated SELECT * FROM Dim_Product Query Screenshot



```
1  SELECT * FROM Dim_Product;
```

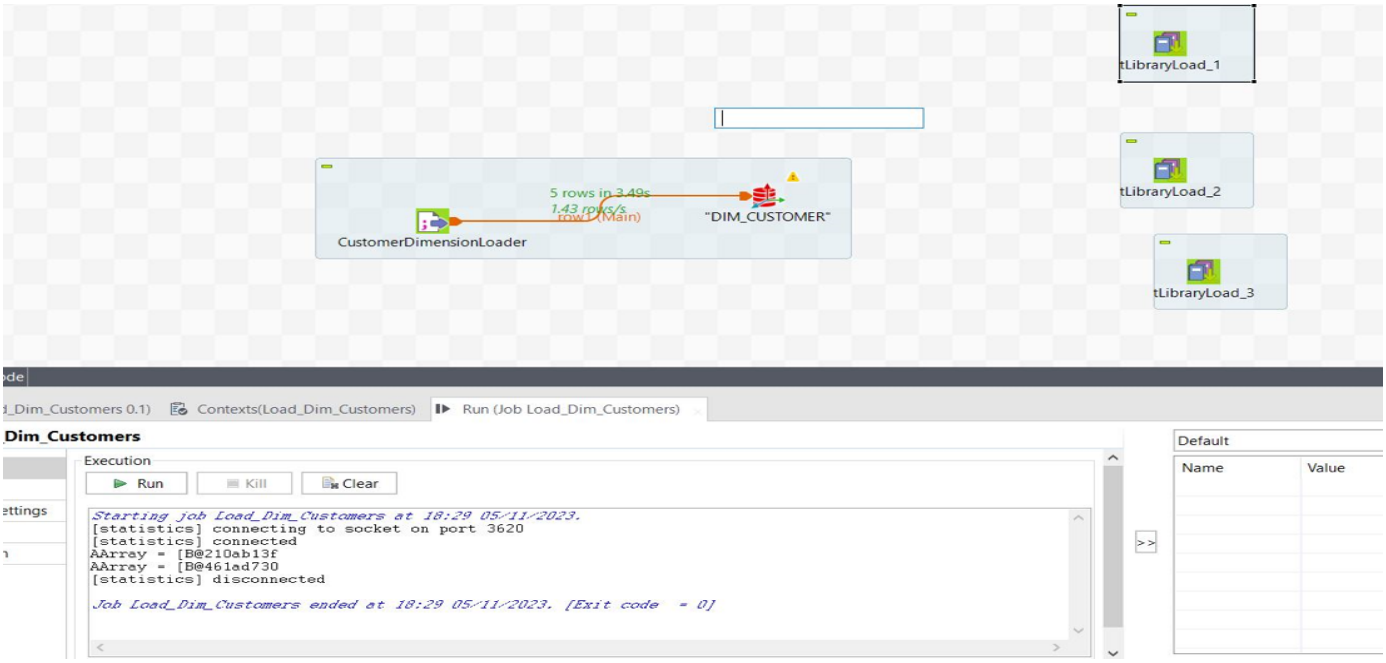
Query Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▾ Execution time: 0.295 seconds

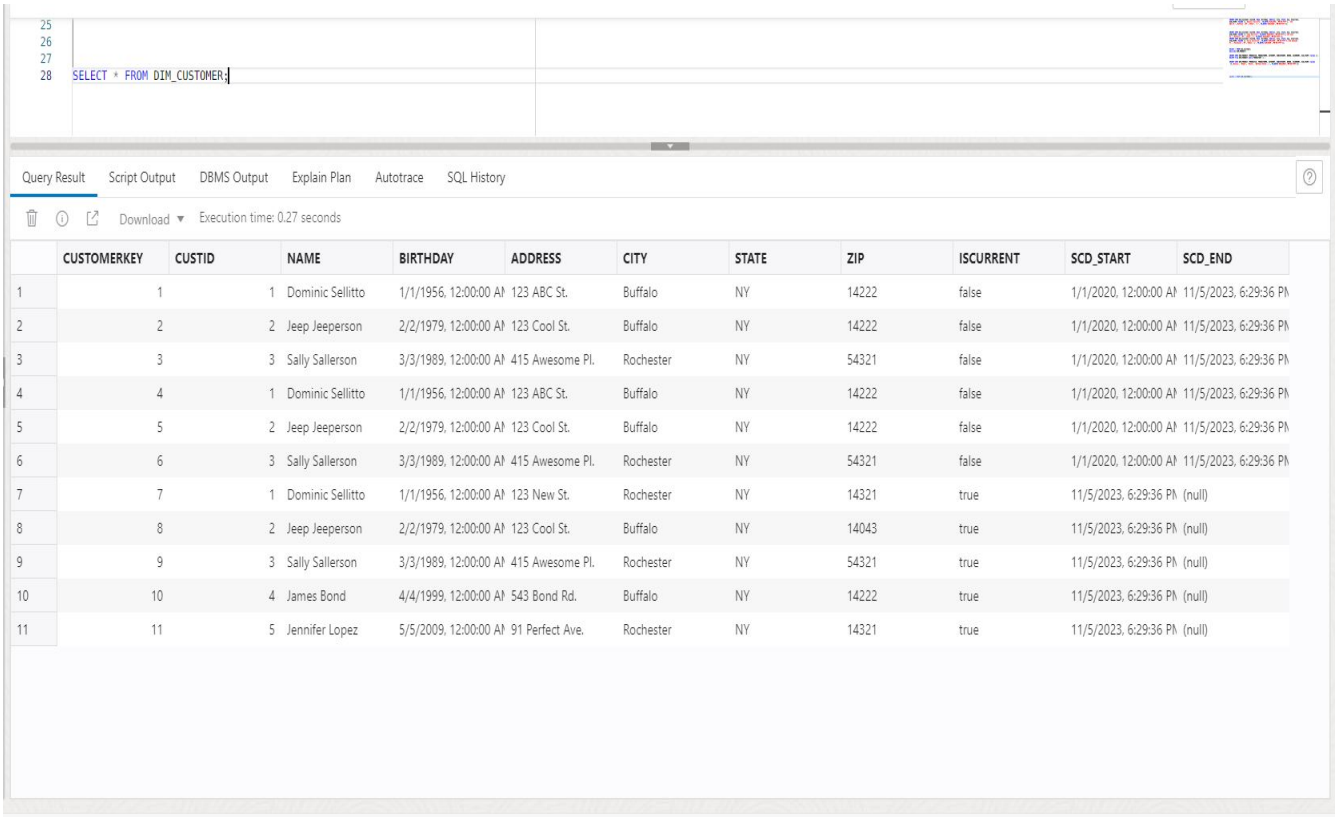
	PRODUCTKEY	PRODUCTNAME	CATEGORY	SUBCATEGORY	BRAND	ISCURRENT	PRODUCTID	SCD_START	SCD_END
1	1	Cinnamon Bread	Wheat	Bread	Nothing Breader	0	1	1/1/2020, 12:00:00 AM	11/5/2023, 6:09:54 PM
2	3	Milk	Dairy	Liquid	Buffalo Farms	1	2	1/1/2020, 12:00:00 AM	(null)
3	7	Cinnamon Bread Loaf	Wheat	Bread	Nothing Breader	1	1	11/5/2023, 6:09:54 PM	(null)
4	8	Chocolate Chip Cooki	Candy	Cookies	Nothing Breader	1	3	11/5/2023, 6:09:54 PM	(null)
5	9	Eggs	Poultry	Solid	Rochester Farms	1	4	11/5/2023, 6:09:54 PM	(null)
6	10	Sugary Cereal	Wheat	Cereal	Food For You	1	6	11/5/2023, 6:09:54 PM	(null)
7	4	Chocolate Chip Cooki	Candy	Cookies	Nothing Breeder	0	3	4/2/2020, 12:00:00 AM	11/5/2023, 6:09:54 PM
8	6	Rotini	Wheat	Pasta	Buffalo Farms	1	5	9/22/2020, 12:00:00 A	(null)
9	5	Eggs	Dairy	Solid	Rochester Farms	0	4	5/1/2020, 12:00:00 AM	11/5/2023, 6:09:54 PM

Powered by ORDS

Dim_Customer Talend Job Screenshot



SELECT * FROM Dim_Customer Query Screenshot



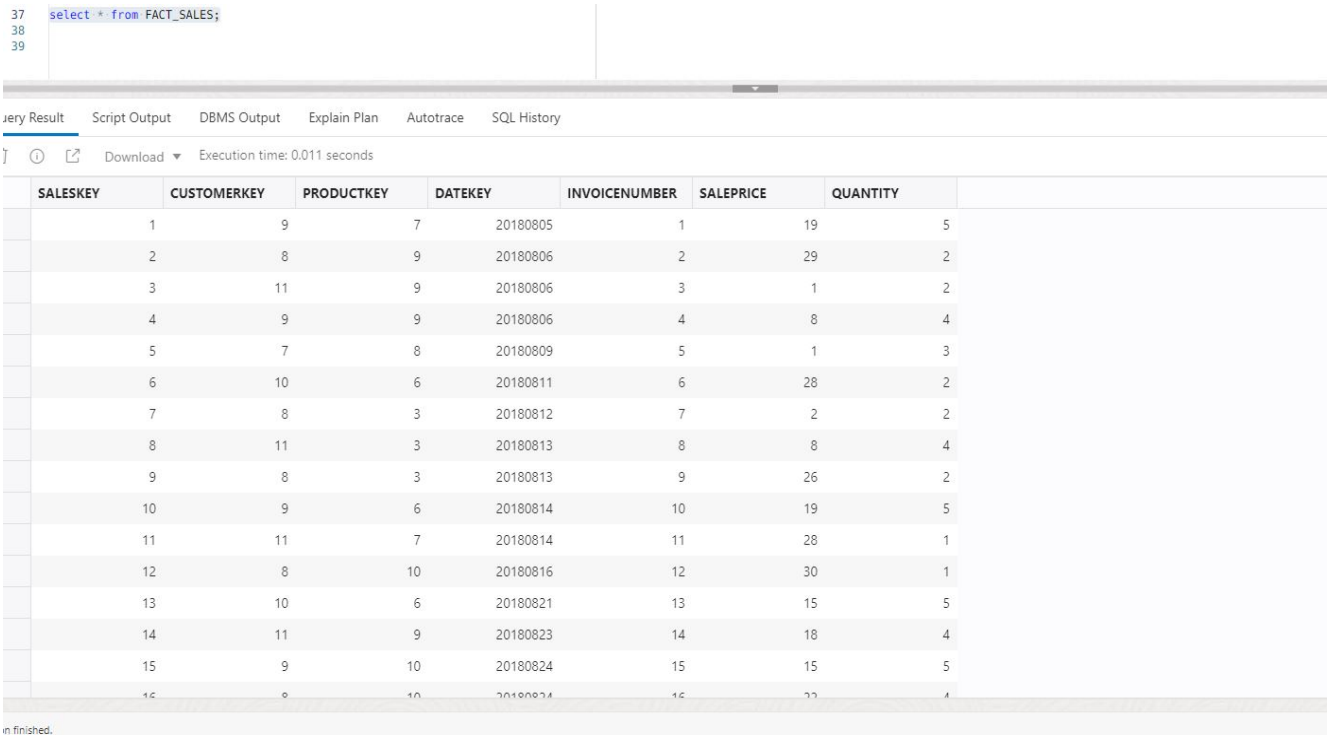
```
25
26
27
28  SELECT * FROM DIM_CUSTOMER;
```

Query Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▼ Execution time: 0.27 seconds

	CUSTOMERKEY	CUSTID	NAME	BIRTHDAY	ADDRESS	CITY	STATE	ZIP	ISCURRENT	SCD_START	SCD_END
1	1	1	Dominic Sellitto	1/1/1956, 12:00:00 AM	123 ABC St.	Buffalo	NY	14222	false	1/1/2020, 12:00:00 AM	11/5/2023, 6:29:36 PM
2	2	2	Jeep Jeeperson	2/2/1979, 12:00:00 AM	123 Cool St.	Buffalo	NY	14222	false	1/1/2020, 12:00:00 AM	11/5/2023, 6:29:36 PM
3	3	3	Sally Sallerson	3/3/1989, 12:00:00 AM	415 Awesome Pl.	Rochester	NY	54321	false	1/1/2020, 12:00:00 AM	11/5/2023, 6:29:36 PM
4	4	1	Dominic Sellitto	1/1/1956, 12:00:00 AM	123 ABC St.	Buffalo	NY	14222	false	1/1/2020, 12:00:00 AM	11/5/2023, 6:29:36 PM
5	5	2	Jeep Jeeperson	2/2/1979, 12:00:00 AM	123 Cool St.	Buffalo	NY	14222	false	1/1/2020, 12:00:00 AM	11/5/2023, 6:29:36 PM
6	6	3	Sally Sallerson	3/3/1989, 12:00:00 AM	415 Awesome Pl.	Rochester	NY	54321	false	1/1/2020, 12:00:00 AM	11/5/2023, 6:29:36 PM
7	7	1	Dominic Sellitto	1/1/1956, 12:00:00 AM	123 New St.	Rochester	NY	14321	true	11/5/2023, 6:29:36 PM	(null)
8	8	2	Jeep Jeeperson	2/2/1979, 12:00:00 AM	123 Cool St.	Buffalo	NY	14043	true	11/5/2023, 6:29:36 PM	(null)
9	9	3	Sally Sallerson	3/3/1989, 12:00:00 AM	415 Awesome Pl.	Rochester	NY	54321	true	11/5/2023, 6:29:36 PM	(null)
10	10	4	James Bond	4/4/1999, 12:00:00 AM	543 Bond Rd.	Buffalo	NY	14222	true	11/5/2023, 6:29:36 PM	(null)
11	11	5	Jennifer Lopez	5/5/2009, 12:00:00 AM	91 Perfect Ave.	Rochester	NY	14321	true	11/5/2023, 6:29:36 PM	(null)

SELECT * FROM FACT_SALES Query Screenshot



```
37  select * from FACT_SALES;
38
39
```

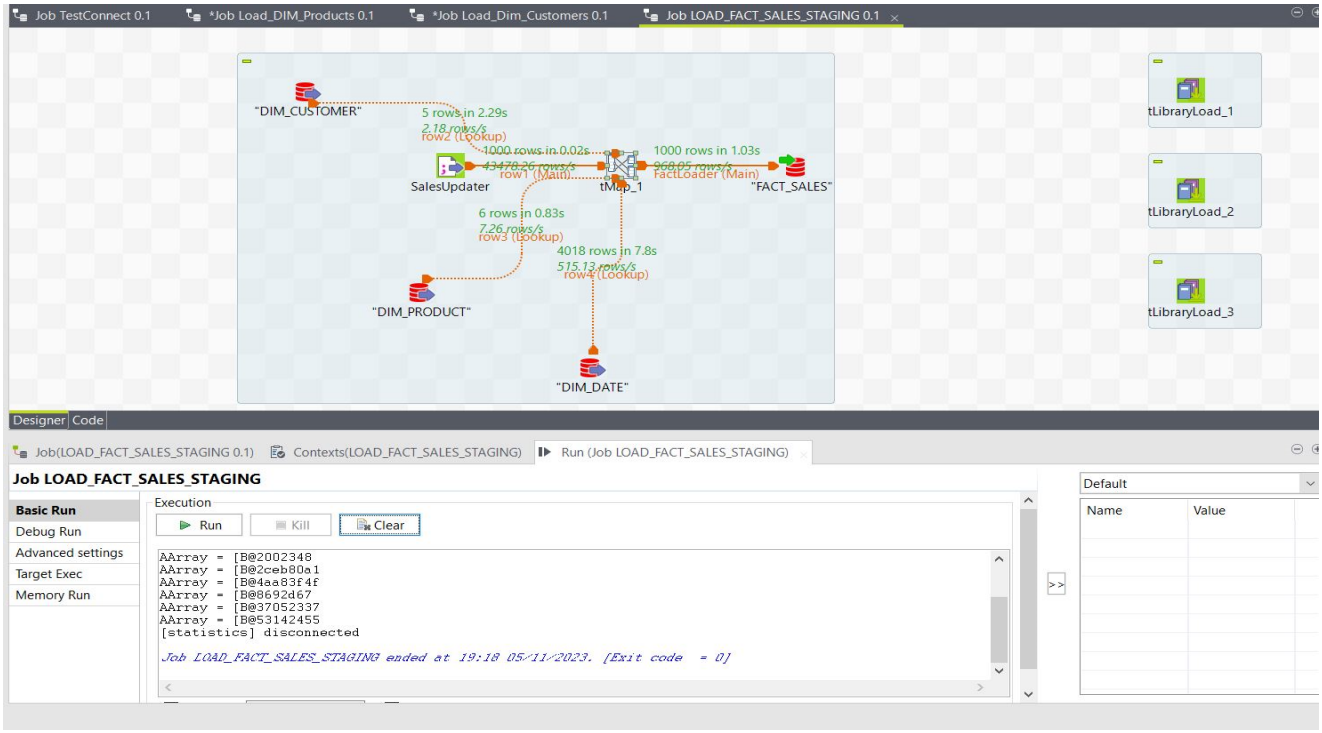
uery Result | Script Output | DBMS Output | Explain Plan | Autotrace | SQL History

Download ▾ Execution time: 0.011 seconds

SALESKEY	CUSTOMERKEY	PRODUCTKEY	DATEKEY	INVOICENUMBER	SALEPRICE	QUANTITY
1	9	7	20180805	1	19	5
2	8	9	20180806	2	29	2
3	11	9	20180806	3	1	2
4	9	9	20180806	4	8	4
5	7	8	20180809	5	1	3
6	10	6	20180811	6	28	2
7	8	3	20180812	7	2	2
8	11	3	20180813	8	8	4
9	8	3	20180813	9	26	2
10	9	6	20180814	10	19	5
11	11	7	20180814	11	28	1
12	8	10	20180816	12	30	1
13	10	6	20180821	13	15	5
14	11	9	20180823	14	18	4
15	9	10	20180824	15	15	5
16	8	10	20180824	16	22	4

n finished.

Completed Job Screenshot



Screenshot of tMap Screen