



Generating OAC Semantic Model Using LLMs

Konstantin Zhernevskiy/Data Intensity

- Expertise
 - OAC, OAS, OBIEE
 - FDI, FAW
 - BI Apps
 - ADW, Kimball, Data vault
 - ODI, CDC, Informatica, Python
 - Fusion, EBS as a sources
 - Cloud Migration & Optimal Data Architecture design
- Frameworks
 - Agile, Scrum
 - PMBOK
 - ITIL



Konstantin Zhernevskiy
Senior DWH Architect
PhD in Economics

Oracle Migration & Managed Services Specialists

FULL ORACLE STACK SUPPORT

3,500 Implementations

7,000 Upgrades

16,000 Managed environments

Our Mission

We manage **complex Oracle workloads** on any cloud, delivering high touch, quality service

Architected & migrated
80+ Oracle landscapes
to OCI in the past 18
months



Cloud Credentials

Oracle Partner

50+ Oracle Specializations
Cloud Services Provider (CSP)
Strategic MSP (SMSP)

Microsoft Partner

AWS Partner

Autonomous DB
Oracle Databases
E-Business Suite
PeopleSoft, JD Edwards
Analytics, Integration
Exadata
License Review and Management

**Largest independent Oracle
Managed Services Provider
globally for 20+ years**

RIGHT CLOUD. RIGHT LICENSING. RIGHT SERVICES.

ORACLE

Service
Partner

*Advanced Expertise in
Oracle E-Business Suite
Applications to Oracle Cloud
in North America*

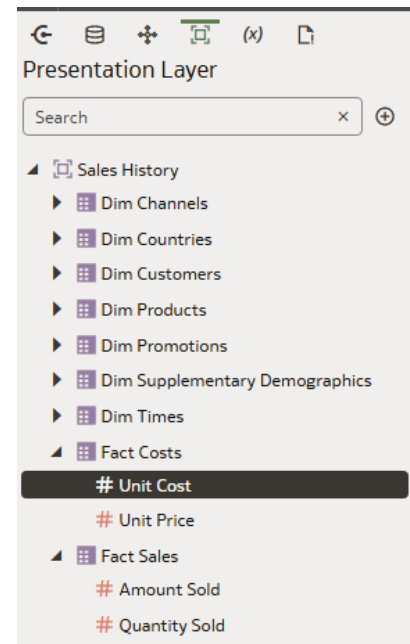
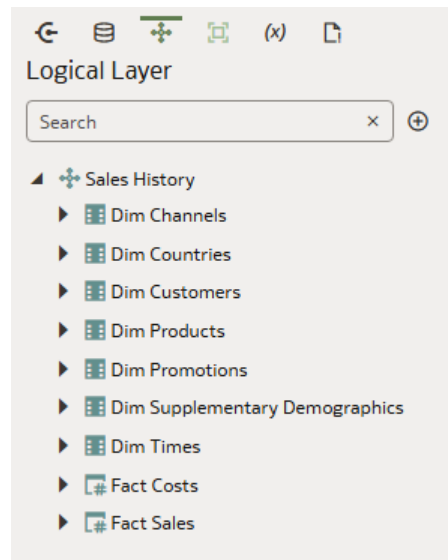
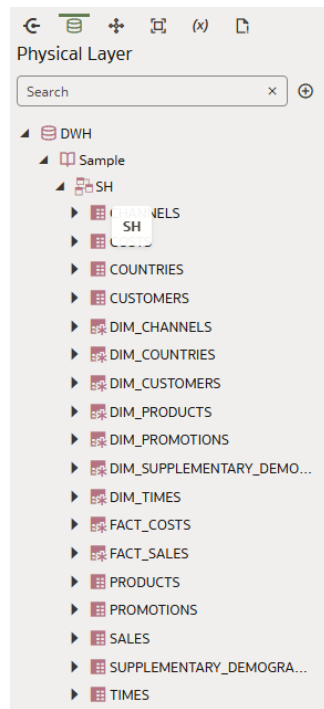
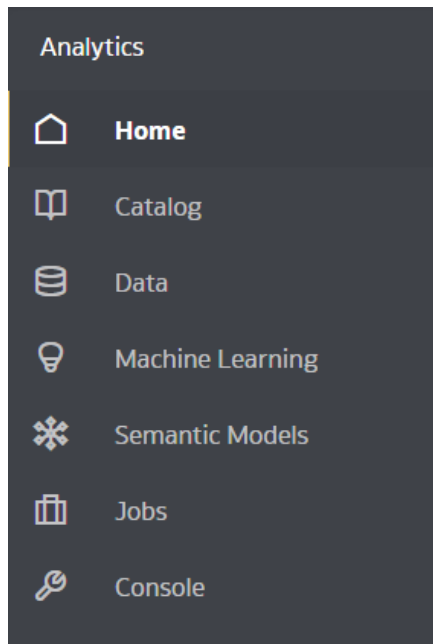


What is Semantic Model in OAC?

- **OAC** – is Oracle cloud-based platform that helps users analyze their data
- Semantic model is a **metadata model** that contains three layers: **physical, logical and presentation**. A semantic model is designed to present data for analysis according to the structure of the business.
- Well-designed semantic model meets the business requirements of the stakeholders **simplifying** of the underlying **data structure**.
- Semantic Model can be created and modified in GUI or using JSON-based markup language **SMML language**.

What is Semantic Model in OAC?

?



Semantic Model is a replacement for OBIEE Repository

The screenshot displays the Oracle BI Administration Tool interface for a Semantic Model named "Sales History". The tool is titled "Oracle BI Administration Tool - test4.rpd" and features a menu bar (File, Edit, View, Manage, Tools, Diagram, Window, Help) and a toolbar with various icons.

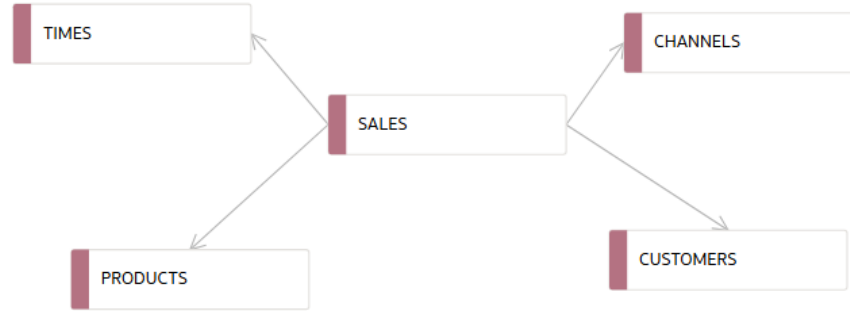
The interface is divided into three main panes:

- Presentation:** Shows the logical structure of the Semantic Model. It includes a hierarchy starting with "Sales History", which branches into "Dim Supplementary Demographics", "Dim Channels", "Dim Countries", and "Dim Customers". "Dim Customers" further branches into attributes like "Cust First Name", "Cust Last Name", "Cust Gender", "Cust Year Of Birth", "Cust Marital Status", "Cust Street Address", "Cust Postal Code", "Cust City", "Cust State Province", "Cust Main Phone Number", "Cust Income Level", "Cust Credit Limit", "Cust Email", "Cust Total", "Cust Eff From", "Cust Eff To", and "Cust Valid". Other dimensions include "Dim Products", "Dim Promotions", "Dim Times", "Fact Costs", and "Fact Sales".
- Business Model and Mapping:** Shows the mapping of the Semantic Model to the underlying Business Model. It mirrors the structure of the Presentation pane, showing how logical dimensions and facts are mapped to physical tables and columns. For example, "Fact Sales" is mapped to the "SALES" table, with columns like "PROD_ID", "CUST_ID", "TIME_ID", "CHANNEL_ID", "PROMO_ID", "Quantity Sold", and "Amount Sold".
- Physical:** Shows the physical database schema. It includes a "DWH" (Data Warehouse) container, which contains a "DAT" (Data Table) container. The "DAT" container includes a "Sample" table, which in turn contains a "SH" (Sales History) table. The "SH" table is mapped to various physical tables, including "CHANNELS", "COSTS", "COUNTRIES", "CUSTOMERS", "DIM_CHANNELS", "DIM_COUNTRIES", "DIM_CUSTOMERS", "DIM_PRODUCTS", "DIM_PROMOTIONS", "DIM_SUPPLEMENTARY_DEMOGRAPHICS", "DIM_TIMES", "FACT_COSTS", "FACT_SALES", "PRODUCTS", "PROMOTIONS", "SALES", "SUPPLEMENTARY_DEMOGRAPHICS", and "TIMES".

At the bottom of the window, the status bar indicates "Business Model : 'Sales History'" and a "NUM" field.

?

What are Star Schema, Fact, and Dimension?



- The **Star Schema** consists of one or more fact tables referencing any number of dimension tables.
- **Fact tables** record measurements or metrics for a specific event. Fact tables generally consist of numeric values, and foreign keys to dimensional data where descriptive information is kept.
- **Dimension** tables usually have a relatively small number of records compared to fact tables, but each record may have a very large number of attributes to describe the fact data.

How would you estimate the efforts for design ?



East Coast
Oracle Users
Conference



- Analyse 9 Tables and identify facts and dimensions
- Create Physical Tables and Aliases
- Create a Logical Model, define measures, and give business/human-readable names for columns
- Create Subject Area at the Presentation level

What is the challenge with OAC SM and why we need AI ?



- Creating SM is **time consuming (8-16 hours** per one star)
- It demands both experience and expertise
- LLMs can work with SQL, JSON, XML
- As a test case I used SH (Sales History) sample schema
- I decided use Chat GPT and DeepSeek for the research

Steps for the Semantic Model Design

1.Extract metadata for analysis (DDL, statistic, data samples)



2.Fact/Dimensions classification



3.Generating Json for Physical Tables



4.Generating Aliases



5.Generating Json for Logical tables



6.Generating Json for Presentation tables



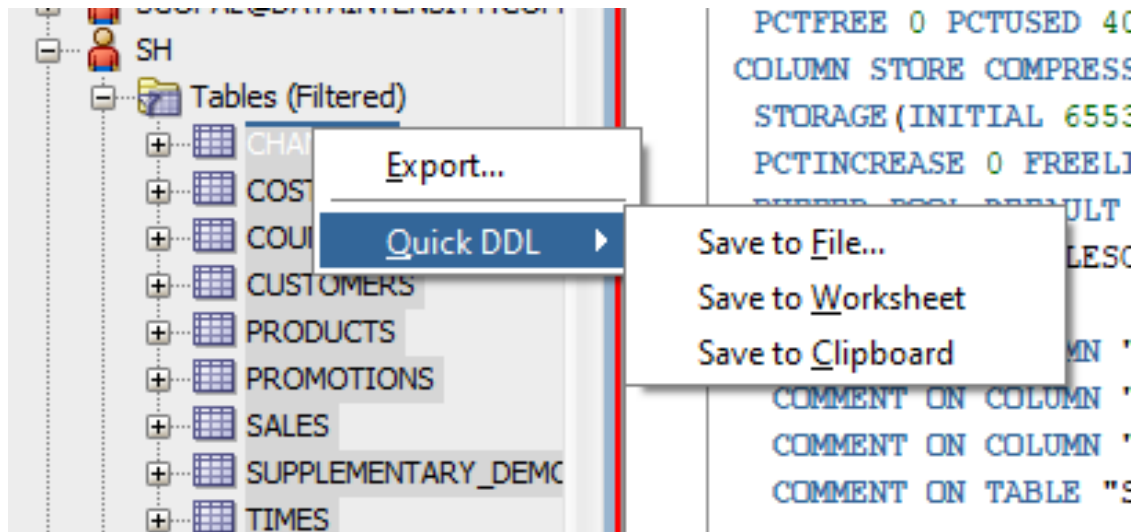
7.Zip and Upload Sematic Model into OAC

1. Extract metadata for analysis (DDL, statistic, data samples)

1 Extracting the data

- Extract DDL from the Database/DataWarehouse
- Extract Statistic for tables
- Extract Data Samples

1 Extracting DDL from the Database



1 Extracting DDL from the Database



```
1  -----
2  -- DDL for Table SALES
3  -----
4  CREATE TABLE "SH"."SALES"
5  (
6      "PROD_ID" NUMBER,
7      "CUST_ID" NUMBER,
8      "TIME_ID" DATE,
9      "CHANNEL_ID" NUMBER,
10     "PROMO_ID" NUMBER,
11     "QUANTITY_SOLD" NUMBER(10,2),
12     "AMOUNT_SOLD" NUMBER(10,2)
13 ) DEFAULT COLLATION "USING NLS_COMP" SEGMENT CREATION IMMEDIATE
14 PCTFREE 0 PCTUSED 40 INITRANS 1 MAXTRANS 255
15 COLUMN STORE COMPRESS FOR QUERY HIGH NO ROW LEVEL LOCKING LOGGING
16 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
17 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
18 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
19 TABLESPACE "SAMPLESCHEMA" ;
20
21 COMMENT ON COLUMN "SH"."SALES"."PROD_ID" IS 'FK to the products dimension table';
22 COMMENT ON COLUMN "SH"."SALES"."CUST_ID" IS 'FK to the customers dimension table';
23 COMMENT ON COLUMN "SH"."SALES"."TIME_ID" IS 'FK to the times dimension table';
24 COMMENT ON COLUMN "SH"."SALES"."CHANNEL_ID" IS 'FK to the channels dimension table';
25 COMMENT ON COLUMN "SH"."SALES"."PROMO_ID" IS 'promotion identifier, without
26 FK constraint (intentionally) to show outer join optimization';
27 COMMENT ON COLUMN "SH"."SALES"."QUANTITY_SOLD" IS 'product quantity sold with the transaction';
28 COMMENT ON COLUMN "SH"."SALES"."AMOUNT_SOLD" IS 'invoiced amount to the customer';
29 COMMENT ON TABLE "SH"."SALES" IS 'facts table, without a primary key; all rows
30 are uniquely identified by the combination of all foreign keys';
31 GRANT READ ON "SH"."SALES" TO PUBLIC;
32 GRANT READ ON "SH"."SALES" TO "ADMIN" WITH GRANT OPTION;\
33
34 -----
35 -- Ref Constraints for Table SALES
36 -----
37 ALTER TABLE "SH"."SALES" ADD CONSTRAINT "SALES_PROMO_FK" FOREIGN KEY ("PROMO_ID")
38 REFERENCES "SH"."PROMOTIONS" ("PROMO_ID") RELY DISABLE;
39 ALTER TABLE "SH"."SALES" ADD CONSTRAINT "SALES_PRODUCT_FK" FOREIGN KEY ("PROD_ID")
40 REFERENCES "SH"."PRODUCTS" ("PROD_ID") RELY DISABLE;
41 ALTER TABLE "SH"."SALES" ADD CONSTRAINT "SALES_CUSTOMER_FK" FOREIGN KEY ("CUST_ID")
42 REFERENCES "SH"."CUSTOMERS" ("CUST_ID") RELY DISABLE;
43 ALTER TABLE "SH"."SALES" ADD CONSTRAINT "SALES_CHANNEL_FK" FOREIGN KEY ("CHANNEL_ID")
44 REFERENCES "SH"."CHANNELS" ("CHANNEL_ID") RELY DISABLE;
45 ALTER TABLE "SH"."SALES" ADD CONSTRAINT "SALES_TIME_FK" FOREIGN KEY ("TIME_ID")
46 REFERENCES "SH"."TIMES" ("TIME_ID") RELY DISABLE;
```

01 Extracting Statistic for Tables

OWNER	TABLE_NAME	NUM_ROWS	AVG_ROW_LEN
SH	CHANNELS	5	41
SH	PROMOTIONS	503	98
SH	CUSTOMERS	55500	189
SH	SALES	918843	29
SH	COUNTRIES	23	64
SH	SUPPLEMENTARY_DEMOGRAPHICS	4500	153
SH	PRODUCTS	72	173
SH	TIMES	1826	198
SH	COSTS	82112	26

01 Extract Data Samples

```
select 'select * from '||OWNER||'.'||TABLE_NAME||' WHERE ROWNUM<11;'  
from all_tab_statistics  
where owner='SH';
```

```
select  
'select '||LISTAGG(COLUMN_NAME,',') WITHIN GROUP (ORDER BY COLUMN_ID)||' from  
'||OWNER||'.'||TABLE_NAME||' WHERE ROWNUM<11;'  
from all_tab_columns  
where owner='SH'  
group by owner,table_name
```


01 Extract Data Samples



East Coast
Oracle Users
Conference



- `select PROD_ID,TIME_ID,PROMO_ID,CHANNEL_ID,UNIT_COST,UNIT_PRICE from SH.COSTS WHERE ROWNUM<11;`
- `select PROD_ID,CUST_ID,TIME_ID,CHANNEL_ID,PROMO_ID,QUANTITY_SOLD,AMOUNT_SOLD from SH.SALES WHERE ROWNUM<11;`
- `select`
`TIME_ID,DAY_NAME,DAY_NUMBER_IN_WEEK,DAY_NUMBER_IN_MONTH,CALENDAR_WEEK_NUMBER,FISCAL_WEEK_NUMBER,WEEK_ENDING_DAY,WEEK_ENDING_DAY_ID,CALENDAR_M`
`ONTH_NUMBER,FISCAL_MONTH_NUMBER,CALENDAR_MONTH_DESC,CALENDAR_MONTH_ID,FISCAL_MONTH_DESC,FISCAL_MONTH_ID,DAYS_IN_CAL_MONTH,DAYS_IN_FIS_MONTH,END`
`OF_CAL_MONTH,END_OF_FIS_MONTH,CALENDAR_MONTH_NAME,FISCAL_MONTH_NAME,CALENDAR_QUARTER_DESC,CALENDAR_QUARTER_ID,FISCAL_QUARTER_DESC,FISCAL_QUARTER`
`_ID,DAYS_IN_CAL_QUARTER,DAYS_IN_FIS_QUARTER,END_OF_CAL_QUARTER,END_OF_FIS_QUARTER,CALENDAR_QUARTER_NUMBER,FISCAL_QUARTER_NUMBER,CALENDAR_YEAR,CALEN`
`DAR_YEAR_ID,FISCAL_YEAR,FISCAL_YEAR_ID,DAYS_IN_CAL_YEAR,DAYS_IN_FIS_YEAR,END_OF_CAL_YEAR,END_OF_FIS_YEAR from SH.TIMES WHERE ROWNUM<11;`
- `select CHANNEL_ID,CHANNEL_DESC,CHANNEL_CLASS,CHANNEL_CLASS_ID,CHANNEL_TOTAL,CHANNEL_TOTAL_ID from SH.CHANNELS WHERE ROWNUM<11;`
- `select`
`PROD_ID,PROD_NAME,PROD_DESC,PROD_SUBCATEGORY,PROD_SUBCATEGORY_ID,PROD_SUBCATEGORY_DESC,PROD_CATEGORY,PROD_CATEGORY_ID,PROD_CATEGORY_DESC,PROD_WEI`
`GHT_CLASS,PROD_UNIT_OF_MEASURE,PROD_PACK_SIZE,SUPPLIER_ID,PROD_STATUS,PROD_LIST_PRICE,PROD_MIN_PRICE,PROD_TOTAL,PROD_TOTAL_ID,PROD_SRC_ID,PROD_EFF_FROM,`
`PROD_EFF_TO,PROD_VALID from SH.PRODUCTS WHERE ROWNUM<11;`
- `select`
`COUNTRY_ID,COUNTRY_ISO_CODE,COUNTRY_NAME,COUNTRY_SUBREGION,COUNTRY_SUBREGION_ID,COUNTRY_REGION,COUNTRY_REGION_ID,COUNTRY_TOTAL,COUNTRY_TOTAL_ID,CO`
`UNTRY_NAME_HIST from SH.COUNTRIES WHERE ROWNUM<11;`
- `select`
`CUST_ID,CUST_FIRST_NAME,CUST_LAST_NAME,CUST_GENDER,CUST_YEAR_OF_BIRTH,CUST_MARITAL_STATUS,CUST_STREET_ADDRESS,CUST_POSTAL_CODE,CUST_CITY,CUST_CITY_ID,CUS`
`T_STATE_PROVINCE,CUST_STATE_PROVINCE_ID,COUNTRY_ID,CUST_MAIN_PHONE_NUMBER,CUST_INCOME_LEVEL,CUST_CREDIT_LIMIT,CUST_EMAIL,CUST_TOTAL,CUST_TOTAL_ID,CUST_S`
`RC_ID,CUST_EFF_FROM,CUST_EFF_TO,CUST_VALID from SH.CUSTOMERS WHERE ROWNUM<11;`
- `select`
`PROMO_ID,PROMO_NAME,PROMO_SUBCATEGORY,PROMO_SUBCATEGORY_ID,PROMO_CATEGORY,PROMO_CATEGORY_ID,PROMO_COST,PROMO_BEGIN_DATE,PROMO_END_DATE,PRO`
`MO_TOTAL,PROMO_TOTAL_ID from SH.PROMOTIONS WHERE ROWNUM<11;`
- `select`
`CUST_ID,EDUCATION,OCCUPATION,HOUSEHOLD_SIZE,YRS_RESIDENCE,AFFINITY_CARD,BULK_PACK_DISKETTES,FLAT_PANEL_MONITOR,HOME_THEATER_PACKAGE,BOOKKEEPING_APPLICAT`
`ION,PRINTER_SUPPLIES,Y_BOX_GAMES,OS_DOC_SET_KANJI,COMMENTS from SH.SUPPLEMENTARY_DEMOGRAPHICS WHERE ROWNUM<11;`

01 Extract Data Samples

Example of records:

PROD_ID	TIME_ID	PROMO_ID	CHANNEL_ID	UNIT_COST	UNIT_PRICE
118	31-DEC-00	999	4	7	7
120	06-JUL-00	999	2	8	8
120	23-JUL-00	999	2	8	8
120	01-AUG-00	999	2	8	8
120	09-AUG-00	999	2	8	8
120	14-AUG-00	999	2	8	8
120	31-AUG-00	999	2	8	8
120	04-OCT-00	999	2	8	8
120	02-NOV-00	999	2	8	8
120	03-NOV-00	999	2	8	8

10 rows selected.

PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_SOLD
13	524	20-JAN-98	2	999	1	1205.99
13	2128	05-APR-98	2	999	1	1250.25
13	3212	05-APR-98	2	999	1	1250.25
13	3375	05-APR-98	2	999	1	1250.25
13	5204	05-APR-98	2	999	1	1250.25
13	7082	05-APR-98	2	999	1	1250.25
13	9316	05-APR-98	2	999	1	1250.25
13	12782	05-APR-98	2	999	1	1250.25
13	13869	05-APR-98	2	999	1	1250.25
13	1949	05-JUL-98	2	999	1	1210.21

10 rows selected.

2.Fact/Dimensions classification

2.Fact/Dimension classification: Prompt (DeepSeek)

?

- I'll give you the list of the tables and you need to identify the type of the table, show it to me and **remember** it.
- It will be needed on the further steps.
- There are **two types** of tables: fact or dimension.
- Please use the following **rules** to identify the type of the table:
 - 1.If the number of rows in the table is significantly more rows than others it is most likely a fact.
 - 2.If a table is referenced by other tables, then it is most likely a dimension.
 - 3.If a table references other tables, but is not referenced by others, then most likely this is a fact.
 - 4.If table have columns like SUM, AMOUNT, Quantity , QTY, TIME, VALUE, COUNT then most likely this is a fact.
- Information about number of records:
 - ...
 - DDL:
 - ...

2. Fact/Dimension classification: Response (DeepSeek)

Based on the provided information, I will classify each table as either a fact or a dimension using the rules you provided. Here's the classification:

Fact Tables

1. SALES

- Reason:
 - It has the largest number of rows (918,843).
 - It references other tables (e.g., **PRODUCTS**, **CUSTOMERS**, **TIMES**, **CHANNELS**, **PROMOTIONS**).
 - It contains columns like **QUANTITY_SOLD** and **AMOUNT_SOLD**, which are typical of fact tables.

2. COSTS

- Reason:
 - It has a significant number of rows (82,112).
 - It references other tables (e.g., **PRODUCTS**, **TIMES**, **CHANNELS**, **PROMOTIONS**).
 - It contains columns like **UNIT_COST** and **UNIT_PRICE**, which are typical of fact tables.

2. Fact/Dimension classification: Response (DeepSeek)

Dimension Tables

- 1.CHANNELS
 - Reason:
 - It has a small number of rows (5).
 - It is referenced by other tables (e.g., **SALES**, **COSTS**).
- 2.PROMOTIONS
 - Reason:
 - It has a moderate number of rows (503).
 - It is referenced by other tables (e.g., **SALES**, **COSTS**).
- 3.CUSTOMERS
 - Reason:
 - It has a moderate number of rows (55,500).
 - It is referenced by other tables (e.g., **SALES**).
- 4.COUNTRIES
 - Reason:
 - It has a small number of rows (23).
 - It is referenced by other tables (e.g., **CUSTOMERS**).

2. Fact/Dimension classification: Response (DeepSeek)

- 5. SUPPLEMENTARY_DEMOGRAPHICS
 - Reason:
 - It has a moderate number of rows (4,500).
 - It is referenced by other tables (e.g., **CUSTOMERS**).
- 6. PRODUCTS
 - Reason:
 - It has a small number of rows (72).
 - It is referenced by other tables (e.g., **SALES**, **COSTS**).
- 7. TIMES
 - Reason:
 - It has a moderate number of rows (1,826).
 - It is referenced by other tables (e.g., **SALES**, **COSTS**).

2. Fact/Dimension classification: Response (DeepSeek)



East Coast
Oracle Users
Conference



Summary of Table Types:

Table Name	Type
SALES	Fact
COSTS	Fact
CHANNELS	Dimension
PROMOTIONS	Dimension
CUSTOMERS	Dimension
COUNTRIES	Dimension
SUPPLEMENTARY_DEMOGRAPHICS	Dimension
PRODUCTS	Dimension
TIMES	Dimension

Let me know if you need further clarification or assistance with the next steps!

3. Generating Json for Physical Tables

3.1. Generating Json for Physical Tables - Prompt



East Coast
Oracle Users
Conference

Eastern States
OATUG
ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP

← → ↺ 🌐 chat.deepseek.com/a/chat/s/c37981f7-38d6-4cb0-8cfb-16a617b71a07

deepseek

New chat

Today

New chat

New chat

New chat

New chat

New chat

New chat

New chat

New chat

Yesterday

New chat

Below is the SQL input script that creates a table on the physical layer in Oracle Analytic Cloud repository. Please give me a JSON that represents a table created by this script. Do not include storage,tablespace,logging and grants, but add joins for foreign keys and do it only if FOREIGN KEY CONSTRAINT exists:

-- DDL for Table SALES

```
CREATE TABLE "SH"."SALES"
(  "PROD_ID" NUMBER,
   "CUST_ID" NUMBER,
   "TIME_ID" DATE,
   "CHANNEL_ID" NUMBER,
   "PROMO_ID" NUMBER,
   "QUANTITY_SOLD" NUMBER(10,2),
   "AMOUNT_SOLD" NUMBER(10,2)
) DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE
```

3.1. Generating Json for Physical Tables - Response



East Coast
Oracle Users
Conference

Eastern States
OATUG
ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP

← → ↺

chat.deepseek.com/a/chat/s/c37981f7-38d6-4cb0-8cfb-16a617b71a07

deepseek

New chat

Today

New chat

New chat

New chat

New chat

New chat

New chat

New chat

New chat

Yesterday

New chat

Here is the JSON representation of the SALES table based on the provided SQL script, including the joins for foreign keys where applicable:

json

Copy

```
{
  "physicalTable": {
    "name": "SALES",
    "sourceType": "TABLE",
    "physicalColumns": [
      {
        "name": "PROD_ID",
        "dataType": "NUMERIC",
        "length": 0,
        "nullable": false
      },
      {
        "name": "CUST_ID",
        "dataType": "NUMERIC",
        "length": 0,
        "nullable": false
      },
      {
        "name": "TIME_ID"
      }
    ]
  }
}
```

▼

4. Generating Aliases


4. Generate aliases json for Physical tables: Response



I need you help to create names for JSON files representing aliases for each table, indicating whether it's a fact or dimension.

There must be only one alias for each table.

Use the INNER type of joins. Do not use LEFT joins unless explicitly stated.

For each table, use the information from the SQL files I gave you to generate all the ames of all the aliases that could be derived from table foreign keys.

One again, for each dimension and fact table, there must be only one alias file.

The format of the filenames is ("FACT" / "DIM") + "_" + TableName + ".json".

Give the answer as Python list. There should be no other text in your response.

The example:

```
["DIM_CUSTOMERS.json", "DIM_CUSTOMERS_SHIP_TO.json"]
```

4. Generate aliases json for Physical tables: Response

Finished reasoning >

python



Copy

```
["DIM_CHANNELS.json", "FACT_COSTS.json", "DIM_COUNTRIES.json", "DIM_CUSTOMERS.json", "DIM_PROI
```



4. Generate aliases json for Physical tables: Response



East Coast
Oracle Users
Conference



In the following prompts, I'll send you the JSON filenames representing physical tables. For each, you'll have to return me the alias JSON for the corresponding table that will reference the initial physical table JSON and have **joins** with the other tables. As you join, **join aliases not tables**. Use INNER joins. Do not use LEFT joins unless explicitly stated. Confirm that you understood the assignment. Your responses for the following prompts should not contain anything but a JSON.

Example. If I give you the "DIM_CUSTOMERS" table name, you should generate the following JSON alias:

```
{
  "physicalTable": {
    "name": "DIM_CUSTOMERS",
    "sourceTable": "physicalTable:DWH.Sample.SH.CUSTOMERS",
    "overrideSourceCacheSetting": false,
    "joins": [
      {
        "rightTable": "physicalTable:DWH.Sample.SH.DIM_COUNTRIES",
        "useJoinExpression": false,
        "joinConditions": [
          {
            "leftColumn": "physicalColumn:DWH.Sample.SH.DIM_CUSTOMERS.COUNTRY_ID",
            "rightColumn": "physicalColumn:DWH.Sample.SH.DIM_COUNTRIES.COUNTRY_ID"
          }
        ],
        "joinType": "LEFT",
        "cardinality": "MANY_TO_ONE"
      }
    ]
  }
}
```

4. Generate aliases json for Physical tables: Response

  ChatGPT o3-mini ▾

```
]
}
}
```

Finished reasoning

json

 Copy

```
{
  "physicalTable": {
    "name": "FACT_SALES",
    "sourceTable": "physicalTable:DWH.Sample.SH.SALES",
    "overrideSourceCacheSetting": false,
    "joins": [
      {
        "rightTable": "physicalTable:DWH.Sample.SH.DIM_PRODUCTS",
        "useJoinExpression": false,
        "joinConditions": [
          {
            "leftColumn": "physicalColumn:DWH.Sample.SH.FACT_SALES.PROD_ID",
            "rightColumn":
```

Ask anything

5. Generating Json for Logical tables

5. Chat GPT: Generate json for logical tables: Prompt

In the following prompts, for each alias, I'll need you to **generate logical jsons** using physical jsons that you have done before for all physical aliases that example, done before.

Please use the examples below.

Use INNER joins for tables. Do not use LEFT joins unless explicitly stated.

Please **capitalize Table names** and column names and replace underscores with space.

Make all the columns names (except IDs) in Init Cap, ensuring the first letter of every word is capitalized while other letters are not.

There are **three types of columns** in the tables: **"measures"**, **"attributes"** and **"keys"**.

"measures" represent quantitative data that can be **aggregated** like sales figures or total region, and it is usually numeric or decimal type.

"attributes" are descriptive characteristics like product category or example, and it is usually varchar or varchar2 or char type.

"keys" are identifiers used to link different data tables together and it is usually numeric, integer or GUID type and has suffix ID.

For Dimension Tables please **add hierarchy Type**.

For Fact tables please add SUM aggregation for measures. You can identify measures using definition above.

Please **pull the information "description"** for each column from the comment in the DDL SQL for this column.

For example, for column PROD_MIN_PRICE add description "product minimum price" because of "COMMENT ON COLUMN 'SH'. 'PRODUCTS'. 'PROD_MIN_PRICE' IS 'product minimum price';"

Example for Fact Table "COSTS":

5. Chat GPT: Generate json for logical tables: Response



East Coast
Oracle Users
Conference

Eastern States
OATUG
ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP

Fact Dimension Classification x DeepSeek x +

chatgpt.com/c/67de3db1-07c0-8000-9c5c-a31b8de91c2e

ChatGPT o3-mini

```
}  
}  
}
```

Finished reasoning >

json Copy

```
{  
  "logicalTable": {  
    "name": "Fact Sales",  
    "type": "FACT",  
    "logicalColumns": [  
      {  
        "name": "Prod Id",  
        "description": "FK to the products dimension table",  
        "dataType": "NUMERIC",  
        "writeable": false,  
        "logicalColumnSource": {  
          "derivedFrom": "PHYSICAL_COLUMNS",  
          "physicalMappings": [  

```

Ask anything

+ Deep research

6. Generating Json for Presentation tables

6. Chat GPT:Generate json for presentation tables:Prompt



East Coast
Oracle Users
Conference



In the following prompts, for each logical table, please generate presentation json.

Please use the examples below.

Make all the columns names **in Init Cap**, ensuring the first letter of every word is capitalized while other letters are not.

Recall, that there are three types of columns in the tables: "measures", "attributes" and "keys".

"measures" represent quantitative data that can be aggregated like sales figures or total count and it is usually numeric or decimal type.

"attributes" are descriptive characteristics like product category or region and it is usually varchar or varchar2 or char type.

"keys" are identifiers used to link different data tables together and it is usually numeric, integer or GUID type and has suffix ID.

In other words, copy everything from the logical tables but remove all the ids.

Please add into presentation json only attributes and measures and exclude keys

Example:

```
{
  "presentationTable" : {
    "name" : "Fact Sales",
    "presentationColumns" : [ {
      "name" : "Quantity Sold",
      "sourceLogicalColumn" : "logicalColumn:Sales History.Fact Sales.Quantity Sold"
    }, {
      "name" : "Amount Sold",
      "description" : "invoiced amount to the customer",
      "sourceLogicalColumn" : "logicalColumn:Sales History.Fact Sales.Amount Sold"
    } ]
  }
}
```

6. Chat GPT: Generate json for presentation tables:Response

← → ↺ chatgpt.com/c/67de3db1-07c0-8000-9c5c-a31b8de91c2e

📄 ✎ ChatGPT o3-mini ▾

```
    ]  
  }  
}
```

Finished reasoning >

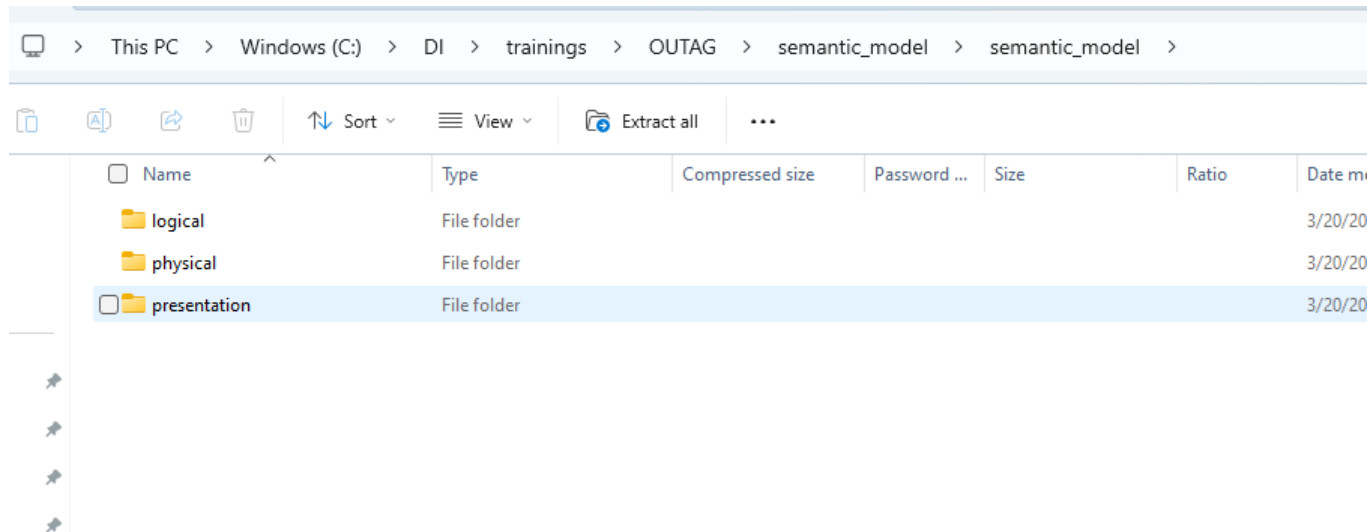
json

📄 Copy

```
{  
  "presentationTable": {  
    "name": "Fact Sales",  
    "presentationColumns": [  
      {  
        "name": "Quantity Sold",  
        "description": "product quantity sold with the transaction",  
        "sourceLogicalColumn": "logicalColumn:Sales History.Fact Sales.Quantity Sold"  
      },  
      {  
        "name":
```

7.Zip and Upload Sematic Model into OAC

7 Upload data into OAC



7 Upload data into OAC

Import From File

Choose a file containing content to replace or add to the semantic model's existing content.

- ☒ Replace all existing content
- ☐ Add content and replace matching objects

Import File

Select

Cancel

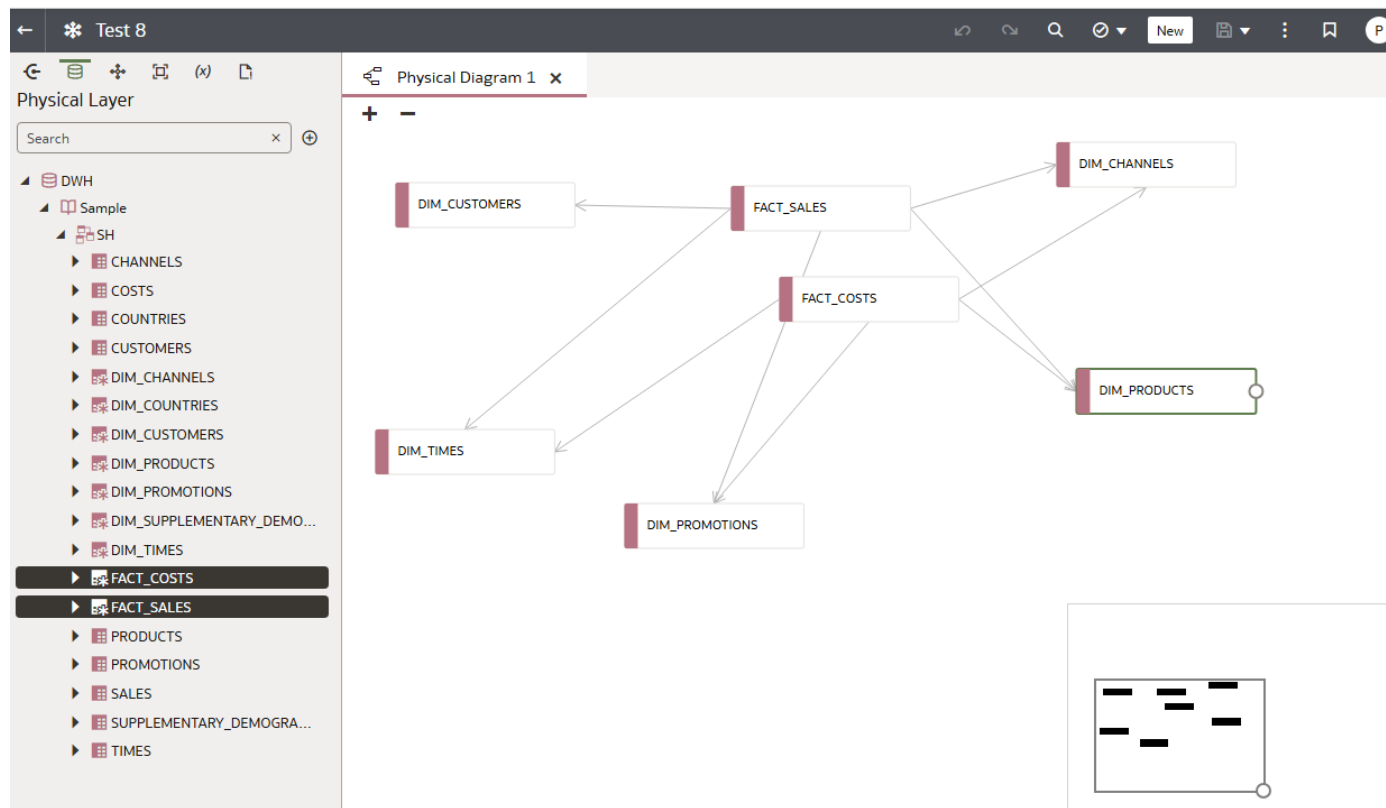
Import

Import successful

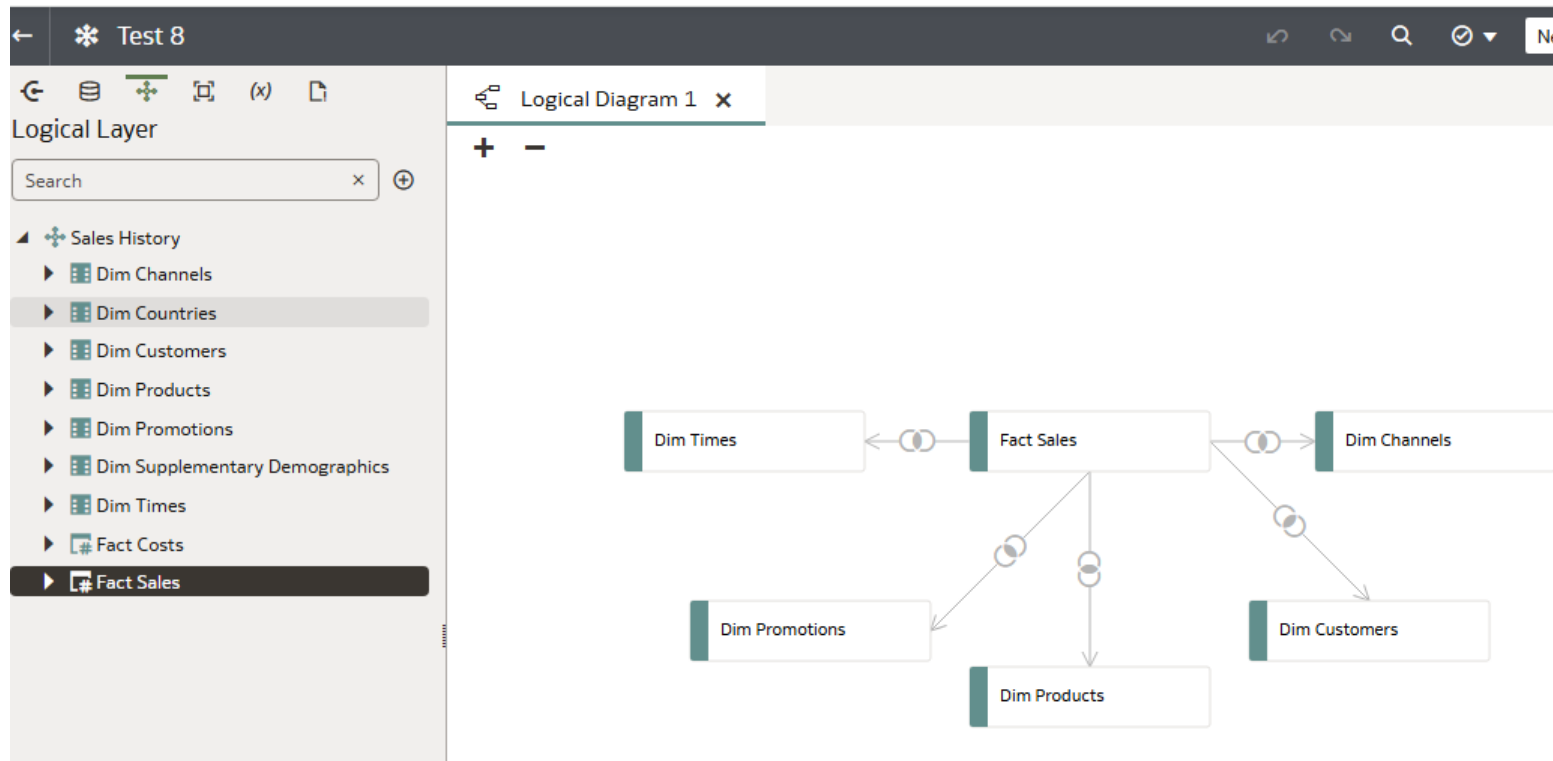
The semantic model was created, but its connections weren't imported. Before continuing, create the required system data connection and assign it to connection pool(s).

OK

7 Upload data into OAC



7 Upload data into OAC



8. Automation work with Chat GPT using REST API

8. Automation using Chat GPT Open API – Main Steps

- Automation using Python
- Chat GPT helped to write the Python script
- 7 steps – 7 function in Python
- All Prompts are in txt files
- All DDL are in sql files
- 1.581 ML Input tokens + 95k Output tokens
- Cost of one run on o3-mini is \$1.36
- 15 minutes instead of ~1 hour manual work

8. Chat GPT API - Solved challenges

- OpenAI o3-mini shown better results than GPT-4o mini
- It used STRING type instead of VARCHAR. I had to modify the prompt
- It used Left join for joins and OAC did not load such joins. I added explicit instructions to use INNER
- It did not understand the Capitalization instruction and I had to add the description of Init Cap explicitly
- After all those correction I was able to get uploadable semantic model

8. Automation using Chat GPT Open API – Main Steps



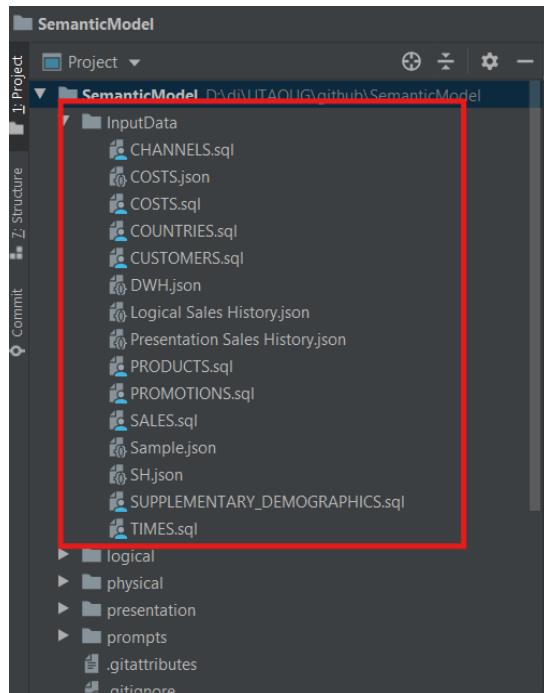
East Coast
Oracle Users
Conference

Eastern States
OATUG
ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP

The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'SemanticModel' located at 'D:\di\UTAOUg\github\SemanticModel'. The project contains several folders: 'InputData', 'logical', 'physical', 'presentation', and 'prompts'. It also includes files like '.gitattributes', '.gitignore', 'cache_context.txt', 'config.py', 'generate.py', 'Input01_DDL.sql', 'Input2_Statistic.txt', 'Input03_random_10_records_from_each_table.txt', and 'print_directories.py'. The main editor window shows the 'generate.py' file with the following Python code:

```
361
362 if __name__ == '__main__':
363     context = []
364     load_cached_context()
365     classify_tables_into_facts_and_dimensions()
366     cache_context()
367     create_directory_structure_of_three_layers()
368     cache_context()
369     process_sql_files_and_generate_jsons_for_physical_layer()
370     cache_context()
371     generate_json_aliases_for_physical_tables()
372     cache_context()
373     process_alias_files_and_generate_jsons_for_logical_layer()
374     cache_context()
375     process_logical_files_and_generate_jsons_for_presentation_layer()
376     cache_context()
377     zip_three_layers()
378
```

8. Automation using Chat GPT Open API – Export Data



8.Automation using Chat GPT Open API – Prompt Fact/Dim



The screenshot shows a code editor with a project structure on the left and a generated SQL script in the main editor. The project structure includes folders for 'Project', 'SemanticModel', 'InputData', 'logical', 'physical', 'presentation', and 'prompts'. The 'prompts' folder contains several files, including 'prompt_step3.txt' which is currently selected. The main editor displays the content of 'prompt_step3.txt', which is a prompt for a Chat GPT API to generate a list of tables and their types (fact or dimension) based on specific rules. The prompt also includes a table of information about the number of records for various tables.

```
1 I'll give you the list of the tables and you need to identify the type of the table, show it to me and
2 It will be needed on the further steps.
3 There are two types of tables: fact or dimension.
4 Please use the following rules to identify the type of the table:
5 1.If the number of rows in the table is significantly more rows than others it is most likely a fact.
6 2.If a table is referenced by other tables, then it is most likely a dimension.
7 3.If a table references other tables, but is not referenced by others, then most likely this is a fact
8 4.If table have numeric columns like SUM, AMOUNT, Quantity, QTY, TIME, VALUE, COUNT, COST, PRICE then
9 5.If table have char columns like NAME, CODE, DESCRIPTION then most likely this is a dimension.
10 Information about number of records:
11 OWNER, TABLE_NAME, NUM_ROWS, AVG_ROW_LEN
12 SH CHANNELS 5 41
13 SH PROMOTIONS 503 98
14 SH CUSTOMERS 55500 189
15 SH SALES 918843 29
16 SH COUNTRIES 23 64
17 SH SUPPLEMENTARY_DEMOGRAPHICS 4500 153
18 SH PRODUCTS 72 173
19 SH TIMES 1826 198
20 SH COSTS 82112 26
21
22 Oracle SQL scripts:
```

10. Comparing Chat GPT and Deep Seek

Chat GPT

- Free GUI version is not enough, but \$20 subscription works decently and it is better than free DeepSeek
- Chat GPT has internal storage, but it has limitations and not stable
- You have multiple model
- GPT o3-mini better then GPT4o
- API pay as you go and works smoothly
- Good price/quality ratio



DeepSeek

- Free GUI – works decently and better than free Chat GPT
- Only one model
- Does not have internal storage
- API pay as you go
- API not so smooth as Chat GPT
- Concerns about data protection



Summary



East Coast
Oracle Users
Conference



- It was proved that **SM can be generated using GUI Chat GPT and DeepSeek**
- Generating SM using GUI AI takes **1 h instead of 8-16 hours** and can be performed even by junior developer
- It was proved that SM can be generated using Chat GPT API
- Generating SM using API takes **15 minutes** comparing with **8-16 hours** manual work.
- SM may be not ideal, but it **covers 80% of the manual work** and can be improved after generation

Q&A



Thank you!

kozhernevskiy@dataintensity.com
www.dataintensity.com

Appendix

Appendix 1. DeepSeek does not have internal storage



Hi, I'm DeepSeek.

How can I help you today?

Extract only text from images and files.



CHANNELS.sql
SQL 2.39KB



COSTS.sql
SQL 2.06KB



COUNTRIES.sql
SQL 2.82KB



CUSTOMERS.sql
SQL 5.14KB



PRODUCTS.sql
SQL 4.75KB



PROMOTIONS.sql
SQL 3.4KB



SALES.sql
SQL 3.1KB



SUPPLEMENTARY_...
SQL 1.69KB



TIMES.sql
SQL 8.21KB

load files into your internal storage into folder "InputData"



DeepThink (R1)



Search



Appendix 1. DeepSeek does not have internal storage



East Coast
Oracle Users
Conference



Loading SQL Files into InputData Folder

I do **not** have persistent internal storage, databases, or filesystems. Here's what you need to know:

1. No Persistent Storage

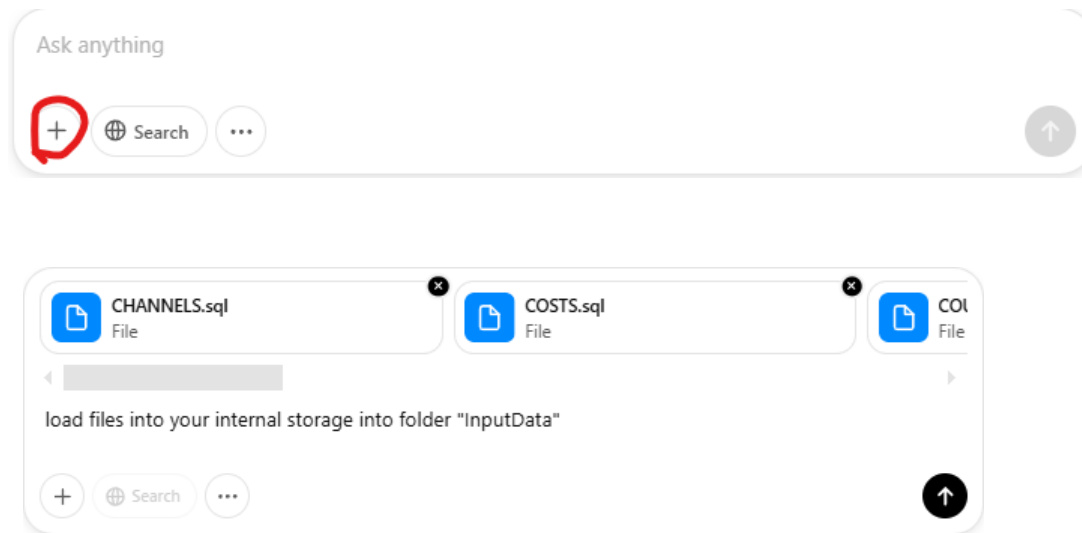
- I operate in a **stateless** environment.
- I **cannot retain files, databases, or data** between interactions.
- Any code or instructions I provide must be executed manually by you in your own environment.

2. Handling Files Like `InputData`

If you want to use the SQL scripts you shared:

- **You** must save them to your local machine, cloud storage, or database server.
- **You** need to run the SQL commands (e.g., in Oracle, MySQL, etc.) to create tables.

Appendix 2.Chat GPT: Internal Storage



Appendix 2.Chat GPT: Internal Storage

All uploaded SQL files have been successfully moved to the "InputData" folder in the internal storage.

The stored files are:

- CHANNELS.sql
- COSTS.sql
- COUNTRIES.sql
- CUSTOMERS.sql
- PRODUCTS.sql
- PROMOTIONS.sql
- SALES.sql
- SUPPLEMENTARY_DEMOGRAPHICS.sql
- TIMES.sql

Let me know if you need any further modifications! 🚀 [~]



Appendix 2.Chat GPT: Internal Storage

Please show all structures and files in your internal storage in folder "SemanticModel/InputData/DDL" in the tree view

Here is the structure of the folder `SemanticModel/InputData/DDL` in the tree view:

```
pgsql

SemanticModel/
  InputData/
    DDL/
      CHANNELS.sql
      COSTS.sql
      COUNTRIES.sql
      CUSTOMERS.sql
      PRODUCTS.sql
      PROMOTIONS.sql
      SALES.sql
      SUPPLEMENTARY_DEMOGRAPHICS.sql
      TIMES.sql
```

Let me know if you need to view, edit, or analyze any specific file!



Appendix 2.Chat GPT: Internal Storage



East Coast
Oracle Users
Conference

Eastern States
OATUG
ORACLE APPLICATIONS & TECHNOLOGY USERS GROUP

The code execution state was reset, which means all stored files were lost. Please re-upload the necessary files so I can proceed with your request. [~]



Please show me the content of the file
"SemanticModel/InputData/DDL/SALES.sql" in your internal
storage

I currently do not have direct access to the contents of the file
"SemanticModel/InputData/DDL/SALES.sql". However, if you upload the file, I can read its contents and
assist you with any modifications or analysis you need. Let me know how you'd like to proceed!

⚠ You may only upload 10 files at a time. ✕

Appendix 2.Chat GPT: Internal Storage

Please analyze all DDL scripts in .sql files in your internal storage folder `"/SemanticModel/InputData/DDL"` and table statistics below.

All those files represent the list of tables, and you need to identify the type of each table, show it to me, and remember it.

It will be needed on the further steps.

There are two types of tables: fact and dimension.

Please use the following rules to identify the type of the table:

1. If the number of rows in the table is significantly more rows than others it is most likely a fact.
2. If a table is referenced by other tables, then it is most likely a dimension.
3. if a table references other tables, but is not referenced by others, then most likely this is a fact.
4. If the table has columns like SUM, AMOUNT, Quantity, QTY, TIME, VALUE, and COUNT then most likely this is a fact.

Information about the number of records:

Appendix 2.Chat GPT: Internal Storage

📄 Memory updated

Based on the analysis of the DDL scripts and the provided table statistics, here is the classification of tables into **fact** and **dimension** categories:

Fact Tables:

1. **SALES** (918,843 rows) - High number of rows, contains **AMOUNT_SOLD**, **QUANTITY_SOLD**, references multiple dimension tables.
2. **COSTS** (82,112 rows) - Contains **UNIT_COST**, **UNIT_PRICE**, references dimension tables, and follows fact table characteristics.

Dimension Tables:

1. **CHANNELS** (5 rows) - Small dataset, referenced by fact tables.
2. **PROMOTIONS** (503 rows) - Referenced by fact tables, small dataset.
3. **CUSTOMERS** (55,500 rows) - Referenced by **SALES**, fits the dimension table characteristics.
4. **COUNTRIES** (23 rows) - Small dataset, most likely used as a lookup/reference.
5. **SUPPLEMENTARY_DEMOGRAPHICS** (4,500 rows) - Fits the dimension table pattern, possibly used for extended customer analysis.
6. **PRODUCTS** (72 rows) - Referenced by fact tables, small dataset.
7. **TIMES** (1,826 rows) - Contains **TIME_ID**, used for time-based analysis, commonly a dimension.

This classification is now stored for further steps. Let me know how you'd like to proceed!