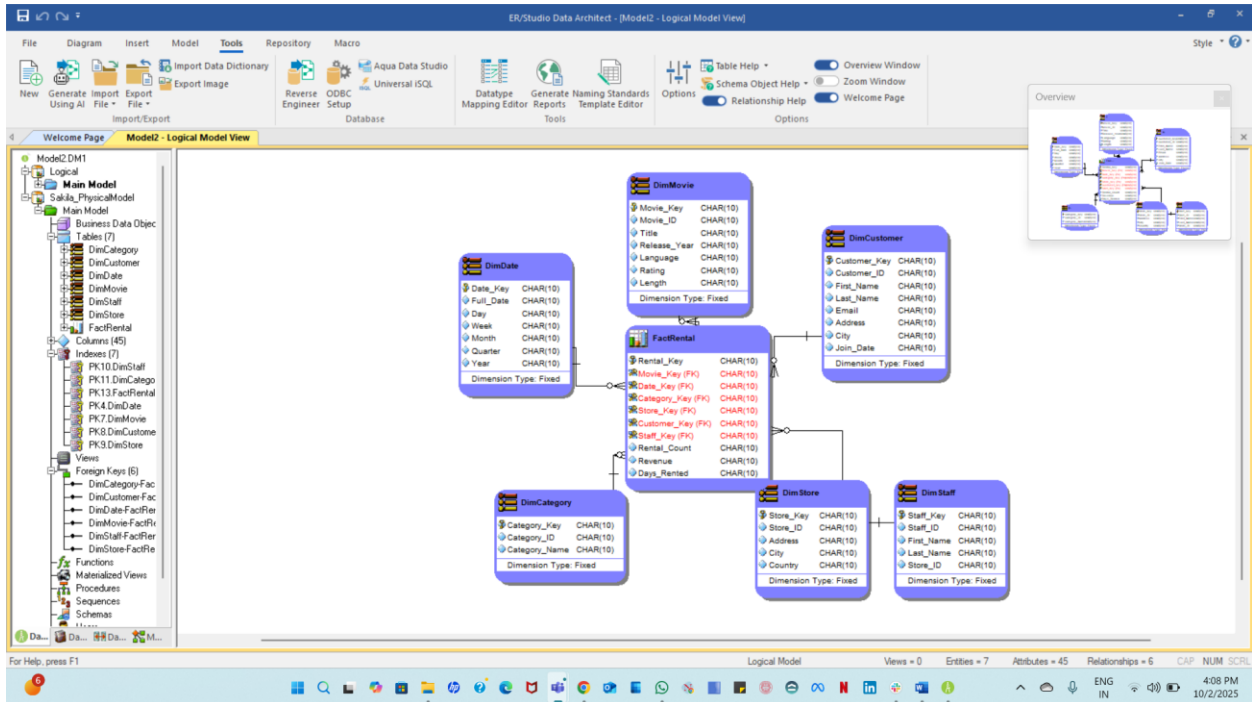


Designing Advanced Data Architectures for Business Intelligence

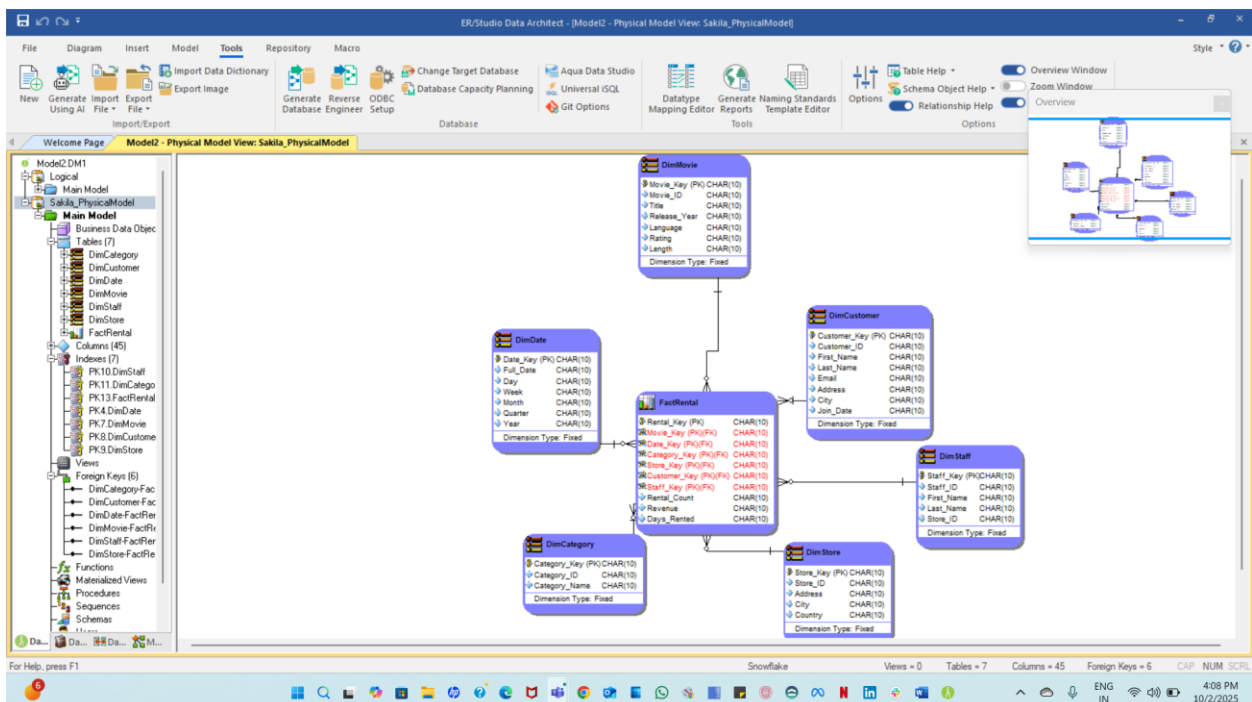
Dimension Modeling

Name – Krisha Lakhani
NUID - 002334794

Logical Model



Physical Model

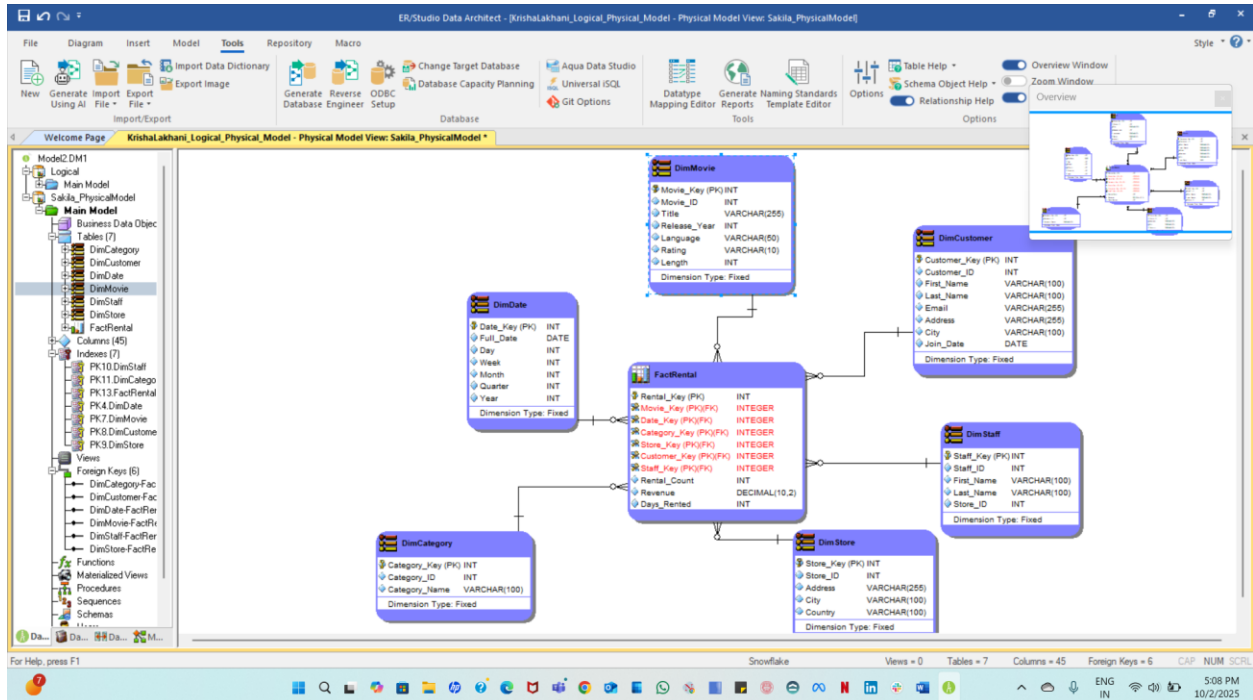


Designing Advanced Data Architectures for Business Intelligence

Dimension Modeling

Name – Krisha Lakhani
NUID - 002334794

Changing Datatypes



The constructed **Star Schema** efficiently meets the analytical and reporting requirements of the organization based on the **Sakila Movie Rental** business operations. It provides insights like monitoring **customer onboarding trends**, comparing **revenue at the store** level across various time frames, pinpointing **high-performing customers and staff**, and examining the **frequency of movie rentals**.

All these business needs can be fulfilled by executing **aggregations and joins** on the central **Fact_Rental** table along with the relevant **dimension tables** (Date, Customer, Movie, Staff, Store, and Category). This dimensional approach guarantees enhanced query performance, streamlined analysis, and flexibility for upcoming analytical requirements.

Designing Advanced Data Architectures for Business Intelligence Dimension Modeling

Name – Krisha Lakhani
NUID - 002334794

- **Customer Onboarding:** Derived from Dim_Customer joined with Dim_Date (using customer creation date) to track new customers by day, week, month, or year.
- **Most Rented Movie:** Fact_Rental joined with Dim_Movie and Dim_Date to find the most frequently rented movie per day.
- **Store Revenue (Daily):** Fact_Rental (or Fact_Payment) joined with Dim_Store and Dim_Date to calculate daily revenue per store.
- **Revenue Comparison (Monthly):** Same fact joined with Dim_Store and Dim_Date to compare monthly revenue trends.
- **Monthly Rentals vs Last Month:** Fact_Rental joined with Dim_Store and Dim_Date to compare month-over-month rental counts.
- **Top Customers:** Fact_Rental joined with Dim_Customer and Dim_Store to identify top 5 customers by rental frequency or total spending.
- **Top Employees:** Fact_Rental joined with Dim_Staff and Dim_Store to identify top 3 staff members assisting customers.

The **grain** of the fact table defines the level of detail captured for each record.

In this model, each record in the **Fact_Rental** table represents one rental transaction at a specific store, for a specific customer, and on a specific date.

This means the grain is:

“One record per movie rental transaction (per customer, per store, per date).”

This level of granularity ensures that all business requirements — such as comparing rentals over time, identifying popular movies, and calculating revenues — can be answered using summarized aggregations.

Designing Advanced Data Architectures for Business Intelligence

Dimension Modeling

Name – Krisha Lakhani
NUID - 002334794

In Account Admin

The screenshot shows the Snowflake SQL Editor interface. The query editor contains the following SQL code:

```
1 --Creating a warehouse, Database, schema, and role
2
3 create or replace warehouse MEDIA_WH warehouse_size='XSMALL';
4 create or replace database MEDIA_DB;
5 create or replace schema MEDIA_DB.SAKILA;
6 create or replace role MEDIA_ROLE;
7
8 --First, take ownership of the DB to which you want to manage so you can grant permissions
9 grant ownership on database MEDIA_DB to role ACCOUNTADMIN;
10
11 --Grant access permissions to the role on the created warehouse, database, and schema
12
13 GRANT USAGE ON WAREHOUSE MEDIA_WH TO ROLE MEDIA_ROLE;
14 --Database
15 GRANT USAGE ON DATABASE MEDIA_DB to role MEDIA_ROLE;
16 --Schema
```

The Results tab shows the following table:

created_on	name	is_default	is_current	origin	owner	comment	options	retention_time
2025-10-02 14:16:25.164 -0700	MEDIA_DB	N	Y		ACCOUNTADMIN			1
2025-09-29 17:50:31.013 -0700	SNOWFLAKE	N	N	SNOWFLAKE.ACCOUN				0
2025-09-29 17:50:34.556 -0700	SNOWFLAKE.LE	N	N		ACCOUNTADMIN	Created by Snowflake		1
2025-09-29 17:50:33.706 -0700	SNOWFLAKE.SA	N	N	SFSALESHARED.SFC	ACCOUNTADMIN	Preloaded TPC H Data		0
2025-10-02 13:52:35.600 -0700	USERSLAKHANI	N	N					1

The screenshot shows the Snowflake SQL Editor interface. The query editor contains the following SQL code:

```
5 create or replace schema MEDIA_DB.SAKILA;
6 create or replace role MEDIA_ROLE;
7
8 --First, take ownership of the DB to which you want to manage so you can grant permissions
9 grant ownership on database MEDIA_DB to role ACCOUNTADMIN;
10
11 --Grant access permissions to the role on the created warehouse, database, and schema
12
13 GRANT USAGE ON WAREHOUSE MEDIA_WH TO ROLE MEDIA_ROLE;
14 --Database
15 GRANT USAGE ON DATABASE MEDIA_DB to role MEDIA_ROLE;
16 --Schema
17 grant ownership on schema MEDIA_DB.SAKILA to role MEDIA_ROLE;
18 grant usage on schema MEDIA_DB.SAKILA to role MEDIA_ROLE;
19
20
```

The Results tab shows the following table:

created_on	name	is_def	is_current	origin	owner	comment	options	retention_time
2025-10-02 14:16:25.164 -0700	MEDIA_DB	N	Y		ACCOUNTADMIN			1
2025-09-29 17:50:31.013 -0700	SNOWFLAKE	N	N	SNOWFLAKE.ACCOUN				0
2025-09-29 17:50:34.556 -0700	SNOWFLAKE.LE	N	N		ACCOUNTADMIN	Created by Snowflake		1
2025-09-29 17:50:33.706 -0700	SNOWFLAKE.SA	N	N	SFSALESHARED.SFC	ACCOUNTADMIN	Preloaded TPC H Data		0
2025-10-02 13:52:35.600 -0700	USERSLAKHANI	N	N					1

Designing Advanced Data Architectures for Business Intelligence Dimension Modeling

Name – Krisha Lakhani
NUID - 002334794

In Media Role

The screenshot shows a SQL IDE interface with a workspace named 'My Workspace'. The 'Database Explorer' on the left lists databases: MEDIA_DB, SNOWFLAKE, SNOWFLAKE_LEARNING_DB, SNOWFLAKE_SAMPLE_DATA, and USER\$MEDIA_USER. The main editor shows a query in 'Untitled.sql' with the following SQL code:

```
--quick check
SELECT CURRENT_USER(), CURRENT_ROLE(), CURRENT_WAREHOUSE(), CURRENT_DATABASE(), CURRENT_SCHEMA();
```

The 'Results (just now)' pane shows the output of the query:

CURRENT_USER()	CURRENT_ROLE()	CURRENT_WAREHOUSE()	CURRENT_DATABASE()	CURRENT_SCHEMA()
MEDIA_USER	MEDIA_ROLE	MEDIA_WH	null	null

The 'Query History' pane shows the executed query with a duration of 22ms.

The screenshot shows the same SQL IDE interface. The 'Database Explorer' on the left lists databases: STAGES, STREAMLITS, TABLES, TABLE_CONSTRAINTS, TABLE_PRIVILEGES, and TABLE_STORAGE_METRICS. The main editor shows a series of SQL queries in 'Untitled.sql':

```
--quick check
SELECT CURRENT_USER(), CURRENT_ROLE(), CURRENT_WAREHOUSE(), CURRENT_DATABASE(), CURRENT_SCHEMA();

USE ROLE MEDIA_ROLE;
USE WAREHOUSE MEDIA_WH;
USE DATABASE MEDIA_DB;
USE SCHEMA SAKILA;
```

The 'Results (just now)' pane shows the output of the first query:

status
Statement executed successfully.

The 'Query History' pane shows the executed queries with durations: 99ms, 97ms, 93ms, and 69ms.

Designing Advanced Data Architectures for Business Intelligence

Dimension Modeling

Name – Krisha Lakhani
NUID - 002334794

The screenshot shows the Snowflake SQL Editor interface. The main window displays the SQL code for creating the FactRental table. The code includes comments and a CREATE TABLE statement with various columns and a primary key constraint.

```
113 )
114 ;
115
116
117
118 --
119 -- TABLE: FactRental
120 --
121
122 CREATE TABLE FactRental(
123     Rental_Key INT NOT NULL,
124     Movie_Key INTEGER NOT NULL,
125     Date_Key INTEGER NOT NULL,
126     Category_Key INTEGER NOT NULL,
127     Store_Key INTEGER NOT NULL,
128     Customer_Key INTEGER NOT NULL,
129     Staff_Key INTEGER NOT NULL,
130     Rental_Count INT,
131     Revenue DECIMAL(10, 2),
132     Days_Rented INT,
133     CONSTRAINT PK13 PRIMARY KEY (Movie_Key, Date_Key, Category_Key, Store_Key, Customer_Key, Staff_Key, Rental_Key)
134 )
135 ;
136
137
138
139 --
```

The Database Explorer on the left shows the schema structure, including the SAKILA database and its tables.

The screenshot shows the Snowflake SQL Editor interface with the execution of ALTER TABLE statements. The main window displays the SQL code for adding foreign key constraints to the FactRental table.

```
163 ALTER TABLE FactRental ADD CONSTRAINT RefDimStaff6
164 FOREIGN KEY (Staff_Key)
165 REFERENCES DimStaff(Staff_Key)
166 ;
167
168 ALTER TABLE FactRental ADD CONSTRAINT RefDimMovie1
169 FOREIGN KEY (Movie_Key)
170 REFERENCES DimMovie(Movie_Key)
171 ;
172
```

The Results (just now) section shows a table with 7 rows and 10 columns. The table is named FactRental and contains data for various dimensions.

Table	Chart	created_on	name	database_name	schema_name	kind	comment	cluster_by	# rows	# i
1		2025-10-02 14:42:37.030 -0700	DIMCATEGORY	MEDIA_DB	SAKILA	TABLE			0	
2		2025-10-02 14:42:37.436 -0700	DIMCUSTOMER	MEDIA_DB	SAKILA	TABLE			0	
3		2025-10-02 14:42:37.811 -0700	DIMDATE	MEDIA_DB	SAKILA	TABLE			0	
4		2025-10-02 14:42:38.085 -0700	DIMMOVIE	MEDIA_DB	SAKILA	TABLE			0	
5		2025-10-02 14:42:38.827 -0700	DIMSTAFF	MEDIA_DB	SAKILA	TABLE			0	
6		2025-10-02 14:42:39.083 -0700	DIMSTORE	MEDIA_DB	SAKILA	TABLE			0	
7		2025-10-02 14:42:39.478 -0700	FACTRENTAL	MEDIA_DB	SAKILA	TABLE			0	