



INFORMATION MANAGEMENT

MIS 381N

Project Report
on

FORMULA 1

Knowledge Management System & Trend Analysis

Submitted by

Aniket Patil	AAP3788
Apurva Audi	AA85254
Chyavan M Chandrashekar	CM65624
Kavya Angara	KA32577
Kyle Tobia	KJT887
Soumya Agarwal	SA55638



Contents

Contents	2
1. Project Design	3
1.1 Meet the contributors	3
1.2 Project Selection	3
2. Introduction	4
3. Data	4
4. Data Management	5
4.1 Data Lake	6
4.2 Data Warehouse	7
4.3 Data Transactions	10
4.3.1 Application-1	10
4.3.2 Application-2	10
4.3.3 Application-3	11
5. Analysis	12
6. Outcome	13
6.1 Insights	13
6.2 Roadblocks	14
6.3 Conclusion	14

1. Project Design

1.1 Meet the contributors

- **Aniket Patil** - Data Warehousing and preparing data for the Transaction Applications
- **Apurva Audi** - Analysis of data from Transaction Applications
- **Chyavan M Chandrashekar** - Reporting, Presentation, and some part of the analysis
- **Kavya Angara** - Data Preprocessing, exploring technologies to adopt for the project, and Data Lake setup
- **Kyle Tobia** - Presentation and understanding insights to provide recommendations
- **Soumya Agarwal** - Reporting, Presentation, and some part of the analysis

1.2 Project Selection

As a group, we all brought ideas and datasets to the table. After collating the ideas, we had a brainstorming session where we discussed what could be done with each dataset. We also included the learning and presentation aspect to pick a project that would let us explore and dive deeper into a data management system while still being riveting to the audience to whom we present our results. Some of the notable ideas that we discussed thoroughly were

- Credit Card Dataset - Perform defensive analysis to predict defaulters, and attempt to detect fraudulent transactions
- Uber Dataset - Perform offensive and supply-management-related analysis to find peak time and demand
- **Formula-1 dataset** - Perform analysis to help Audi enter the top-tier motorsport prepared to challenge the other big contenders like Mercedes, Ferrari, and Redbull.

We selected Formula One for our project because it provides a unique platform for us to explore the potential for technology and engineering innovation. The high speed, the intense competition, and the technical innovations in the cars and tracks make Formula One an ideal choice to highlight the advancements in technology and engineering that have been made over the years, and would provide us with an excellent understanding of the type of data and the importance of extracting insights from that data.

2. Introduction

Formula 1, or F1 as it is more commonly known, is a form of motorsport based on open-wheel race cars. It is considered to be the pinnacle of motorsport and is the most prestigious and expensive form of single-seat auto racing. F1 is contested by ten racing teams composed of two drivers each, with each team competing in events to win championships for the constructors and the drivers.

F1 is a highly technical and sophisticated sport and requires a great deal of skill and preparation from drivers, teams, and engineers. The cars used in F1 are designed and built to the highest standards, with teams investing heavily in the research and development of new technologies to gain a competitive edge. Teams also have access to advanced data analysis to help optimize their performance. This high precision, highly agile environment requires a lot of preparation and fine-tuning to perfect. Since Audi is planning to enter Formula-1 in 2026, our team planned to put ourselves in the driver's seat of the race entry analysis team to come up with insights that would help the company make the most optimal decisions to enter the competition.

3. Data

Formula 1 has been around since the early 1900s, although the first official F1 race took place in 1950. Our data consists of all the information about drivers, teams, circuits, lap-wise statistics, race and qualifying results from this official commencement from 1950, up to the latest race of the 2022 season, i.e. Abu Dhabi circuit. This data has been acquired from the public [Ergast Developer API](#) which provides the official data about the following information

- Seasons
- Race Schedule
- Race Results
- Qualifying Results
- Sprint Qualifying Results
- Standings
- Driver Information
- Constructor Information
- Circuit Information

- Finishing Status
- Lap Times
- Pit Stops

4. Data Management

In this project, we have used Oracle Cloud to set up the data management repositories and pipelines.

We retrieved the data in the form of CSVs from the aforementioned API and proceeded to upload into Oracle Cloud, which provides an option to assemble storage buckets. We, therefore, treat this Oracle Cloud Bucket as our data lake.

Further, Oracle Autonomous Database architecture provides a default workload of the type “Data Warehouse”. So, we did not explicitly construct a warehouse. Instead, we created a database with the default workload type and moved the CSV files from the data lake to tables in the warehouse.

For our applications, we created more tables in the same database (since communicating with multiple autonomous databases was out of the scope of our access) and treated these aggregated tables as our Transactional Application Databases.

We have directly accessed these Transactional Application Databases through Tableau to leverage its power to analyze and visualize the data.

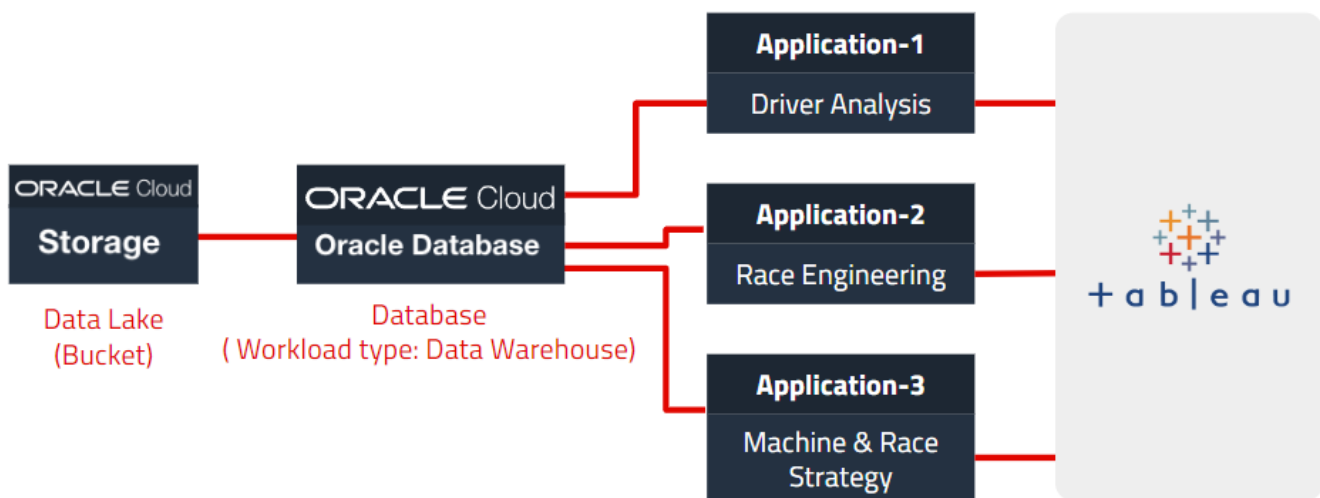
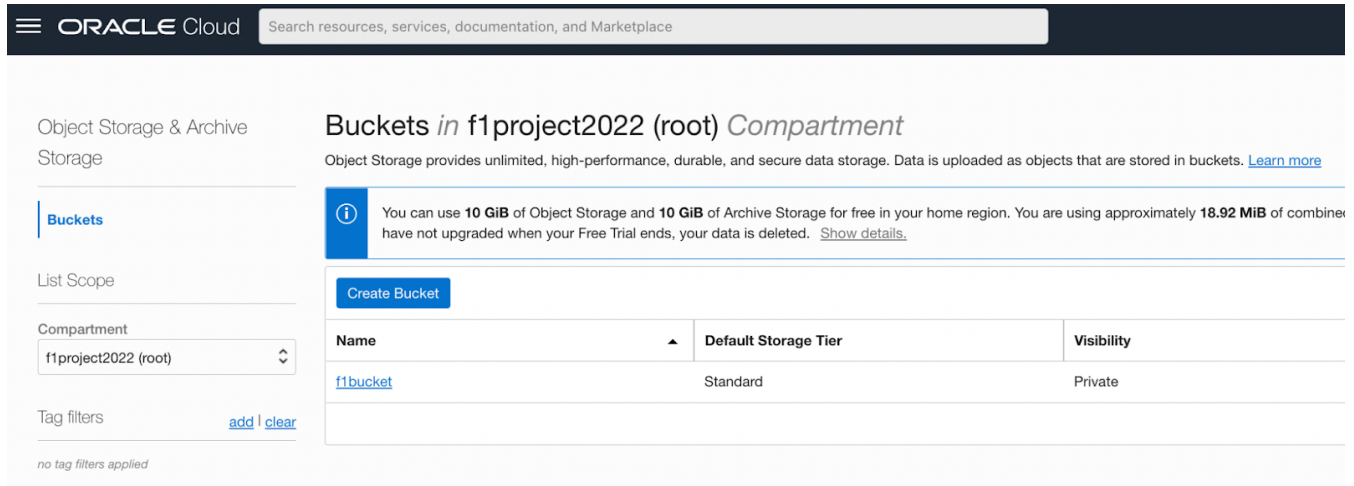


Figure-1: Setup of the Data Management System

4.1 Data Lake

Data is loaded into Oracle cloud by creating a bucket in cloud storage where we can just load our files. It is a straightforward process and does not require any code.



Object Storage & Archive Storage

Buckets in f1project2022 (root) Compartment

Object Storage provides unlimited, high-performance, durable, and secure data storage. Data is uploaded as objects that are stored in buckets. [Learn more](#)

You can use **10 GiB** of Object Storage and **10 GiB** of Archive Storage for free in your home region. You are using approximately **18.92 MiB** of combined storage. [Show details](#)

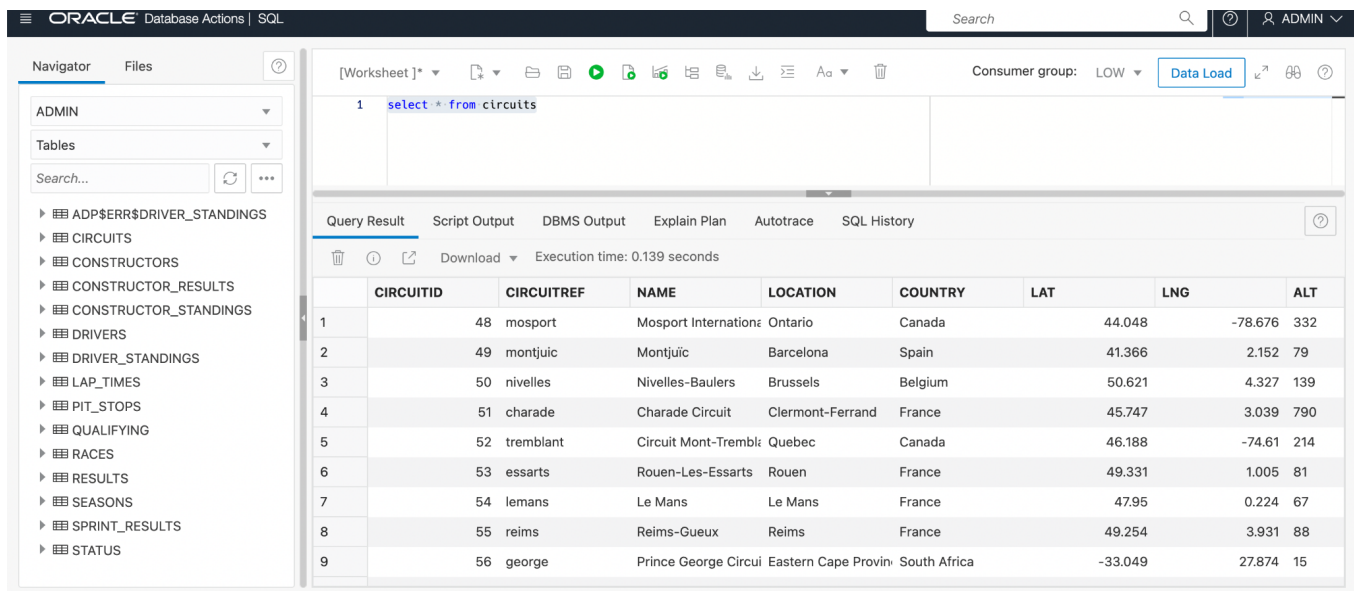
[Create Bucket](#)

Name	Default Storage Tier	Visibility
f1bucket	Standard	Private

Tag filters: [add](#) | [clear](#)

no tag filters applied

Figure-2: Bucket in Oracle Storage



Oracle Database Actions | SQL

Search

Consumer group: LOW [Data Load](#)

1 `select * from circuits`

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

Download Execution time: 0.139 seconds

	CIRCUITID	CIRCUITREF	NAME	LOCATION	COUNTRY	LAT	LNG	ALT
1	48	mosport	Mosport International	Ontario	Canada	44.048	-78.676	332
2	49	montjuic	Montjuïc	Barcelona	Spain	41.366	2.152	79
3	50	nivelles	Nivelles-Baulers	Brussels	Belgium	50.621	4.327	139
4	51	charade	Charade Circuit	Clermont-Ferrand	France	45.747	3.039	790
5	52	tremblant	Circuit Mont-Tremblant	Quebec	Canada	46.188	-74.61	214
6	53	essarts	Rouen-Les-Essarts	Rouen	France	49.331	1.005	81
7	54	lemans	Le Mans	Le Mans	France	47.95	0.224	67
8	55	reims	Reims-Gueux	Reims	France	49.254	3.931	88
9	56	george	Prince George Circuit	Eastern Cape Province	South Africa	-33.049	27.874	15

Figure-3: Files in the Bucket and table view

4.2 Data Warehouse

Oracle Cloud gives the convenience of auto generated DDL when we upload the data. However, in order to create a schema, we altered and executed a tailored DDL query. The following is the auto generated Oracle DDL code against the code we executed for the tables we used in our data model:

Table Creation	Constraint Addition
<pre>CREATE TABLE "ADMIN"."RESULTS" ("RESULTID" NUMBER, "RACEID" NUMBER, "DRIVERID" NUMBER, "CONSTRUCTORID" NUMBER, "NUMBER_RW" NUMBER, "GRID" NUMBER, "POSITION" NUMBER, "POSITIONTEXT" VARCHAR2(64) COLLATE "USING_NLS_COMP", "POSITIONORDER" NUMBER, "POINTS" NUMBER, "LAPS" NUMBER, "TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "MILLISECONDS" NUMBER, "FASTESTLAP" NUMBER, "RANK" NUMBER, "FASTESTLAPTIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FASTESTLAPSPEED" NUMBER, "STATUSID" NUMBER) DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE RESULTS ADD CONSTRAINT FK_resultId PRIMARY KEY (resultId); ALTER TABLE RESULTS ADD CONSTRAINT FK_raceId01 FOREIGN KEY (raceId) REFERENCES RACES (raceId); ALTER TABLE RESULTS ADD CONSTRAINT FK_driverId01 FOREIGN KEY (driverId) REFERENCES DRIVERS (driverId); ALTER TABLE RESULTS ADD CONSTRAINT FK_constructors02 FOREIGN KEY (constructorId) REFERENCES CONSTRUCTORS (constructorId); ALTER TABLE RESULTS ADD CONSTRAINT FK_statusId03 FOREIGN KEY (statusId) REFERENCES STATUS (statusId);</pre>
<pre>CREATE TABLE "ADMIN"."QUALIFYING" ("QUALIFYID" NUMBER, "RACEID" NUMBER, "DRIVERID" NUMBER, "CONSTRUCTORID" NUMBER, "NUMBER_RW" NUMBER, "POSITION" NUMBER, "Q1" VARCHAR2(64) COLLATE "USING_NLS_COMP", "Q2" VARCHAR2(64) COLLATE "USING_NLS_COMP", "Q3" VARCHAR2(64) COLLATE "USING_NLS_COMP") DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE QUALIFYING ADD CONSTRAINT FK_qualifyId PRIMARY KEY (qualifyId); ALTER TABLE QUALIFYING ADD CONSTRAINT FK_raceId04 FOREIGN KEY (raceId) REFERENCES RACES (raceId); ALTER TABLE QUALIFYING ADD CONSTRAINT FK_driverId05 FOREIGN KEY (driverId) REFERENCES DRIVERS (driverId); ALTER TABLE QUALIFYING ADD CONSTRAINT FK_constructorId06 FOREIGN KEY (constructorId) REFERENCES CONSTRUCTORS (constructorId);</pre>

Table Creation	Constraint Addition
<pre>CREATE TABLE "ADMIN"."DRIVER_STANDINGS" ("DRIVERSTANDINGSID" NUMBER, "RACEID" NUMBER, "DRIVERID" NUMBER, "POINTS" NUMBER, "POSITION" NUMBER, "POSITIONTEXT" NUMBER, "WINS" NUMBER) DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE DRIVER_STANDINGS ADD CONSTRAINT fk_driverStandingsId PRIMARY KEY (driverStandingsId); ALTER TABLE DRIVER_STANDINGS ADD CONSTRAINT FK_raceId09 FOREIGN KEY (raceId) REFERENCES RACES (raceId); ALTER TABLE DRIVER_STANDINGS ADD CONSTRAINT FK_driverId10 FOREIGN KEY (driverId) REFERENCES DRIVERS (driverId);</pre>
<pre>CREATE TABLE "ADMIN"."RACES" ("RACEID" NUMBER, "YEAR" NUMBER, "ROUND" NUMBER, "CIRCUITID" NUMBER, "NAME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "DATE_RW" DATE, "TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "URL" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FP1_DATE" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FP1_TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FP2_DATE" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FP2_TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FP3_DATE" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FP3_TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "QUALI_DATE" VARCHAR2(64) COLLATE "USING_NLS_COMP", "QUALI_TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "SPRINT_DATE" VARCHAR2(64) COLLATE "USING_NLS_COMP", "SPRINT_TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP") DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE RACES ADD CONSTRAINT FK_raceId PRIMARY KEY (raceId); ALTER TABLE RACES ADD CONSTRAINT FK_circuitId11 FOREIGN KEY (circuitId) REFERENCES CIRCUITS (circuitId);</pre>
<pre>CREATE TABLE "ADMIN"."PIT_STOPS" ("RACEID" NUMBER, "DRIVERID" NUMBER, "STOP" NUMBER, "LAP" NUMBER, "TIME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "DURATION" VARCHAR2(64) COLLATE "USING_NLS_COMP", "MILLISECONDS" NUMBER) DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE PIT_STOPS ADD CONSTRAINT FK_raceId1 FOREIGN KEY (raceId) REFERENCES RACES (raceId); ALTER TABLE PIT_STOPS ADD CONSTRAINT FK_driverId12 FOREIGN KEY (driverId) REFERENCES DRIVERS (driverId);</pre>

Table Creation	Constraint Addition
<pre>CREATE TABLE "ADMIN"."CONSTRUCTORS" ("CONSTRUCTORID" NUMBER, "CONSTRUCTORREF" VARCHAR2(64) COLLATE "USING_NLS_COMP", "NAME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "NATIONALITY" VARCHAR2(64) COLLATE "USING_NLS_COMP", "URL" VARCHAR2(256) COLLATE "USING_NLS_COMP") DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE CONSTRUCTORS ADD CONSTRAINT FK_constructorId PRIMARY KEY(constructorId);</pre>
<pre>CREATE TABLE "ADMIN"."DRIVERS" ("DRIVERID" NUMBER, "DRIVERREF" VARCHAR2(64) COLLATE "USING_NLS_COMP", "NUMBER_RW" VARCHAR2(64) COLLATE "USING_NLS_COMP", "CODE" VARCHAR2(64) COLLATE "USING_NLS_COMP", "FORENAME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "SURNAME" VARCHAR2(64) COLLATE "USING_NLS_COMP", "DOB" DATE, "NATIONALITY" VARCHAR2(64) COLLATE "USING_NLS_COMP", "URL" VARCHAR2(256) COLLATE "USING_NLS_COMP") DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 10 MAXTRANS 255 COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "DATA"</pre>	<pre>ALTER TABLE DRIVERS ADD CONSTRAINT FK_driverId PRIMARY KEY(driverId);</pre>

Table-1: DDL code for the Data Warehouse setup

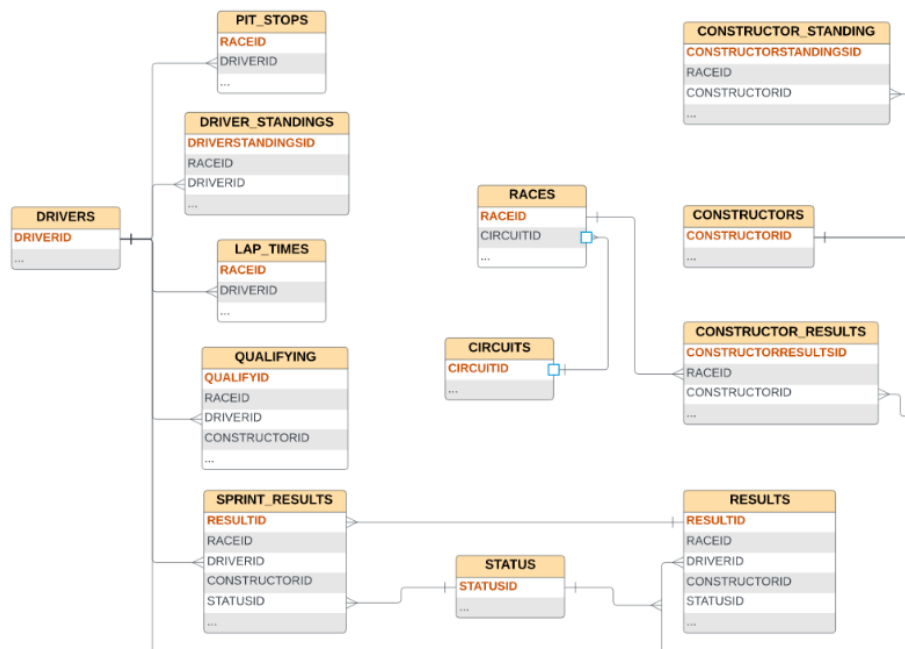


Figure-3: Data Model for the Data Warehouse

4.3 Data Transactions

4.3.1 Application-1

Driver Perspective: Who should we sign?

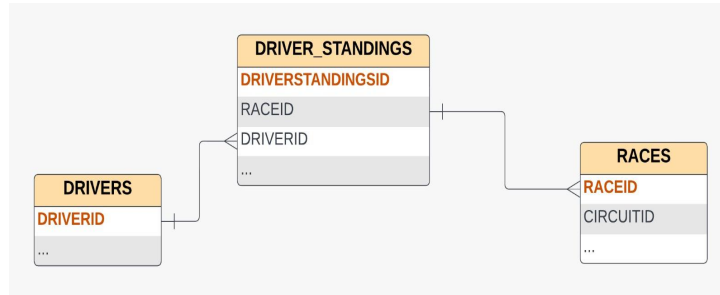


Figure-4: Data Model for the First Transaction Application - Driver Analysis

```

14 create table transactiona_app_driver_analysis as
15 select * from DRIVER_STANDINGS d1
16 left join DRIVERS d2 on d1.DRIVERID=d2.DRIVERID
17 left join Races r1 on r1.RACEID=d1.RACEID;
  
```

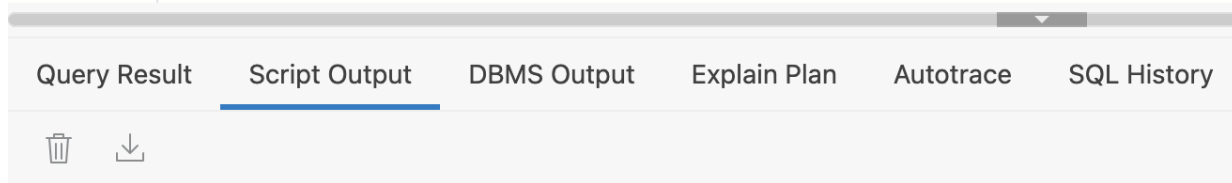


Table TRANSACTIONA_APP_DRIVER_ANALYSIS created.

Elapsed: 00:00:00.659

Figure-5: Creating Data Model for Application-1 in Oracle Cloud

4.3.2 Application-2

Race Engineering Perspective: Should we focus on pit-stops?

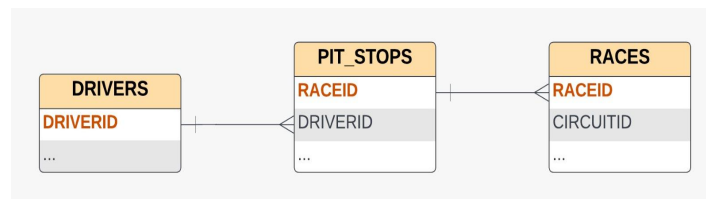


Figure-6: Data Model for the Second Transaction Application - Race Engineering Analysis

```

14 create table transactiona_app_race_engg as
15 select *
16 from PIT_STOPS p left join RACES r on p.RACEID = r.RACEID as tmp
17 left join drivers d on tmp.DRIVERID = d.DRIVERID

```

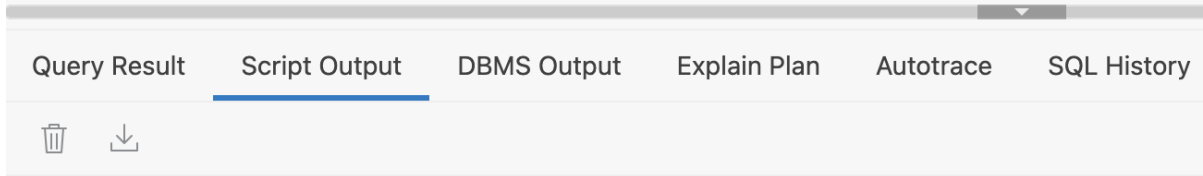


Table TRANSACTIONA_APP_RACE_ENGG created.

Elapsed: 00:00:00.196

Figure-7: Creating Data Model for Application-2 in Oracle Cloud

4.3.3 Application-3

Race Strategy Perspective: Success in the Grand Prix based on Qualifying

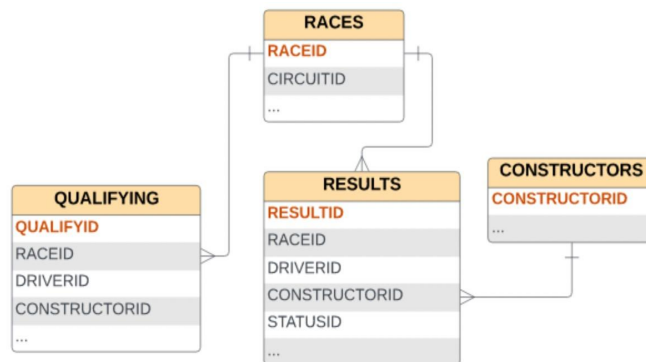


Figure-8: Data Model for the Third Transaction Application - Race Strategy Analysis

```

14 create table transactiona_app_race_strategy as
15 select *
16 from CONSTRUCTORS c inner join RESULTS res on c.CONSTRUCTORID = res.CONSTRUCTORID as tmp1
17 inner join races r on tmp1.RACEID = r.RACEID and tmp1.CONSTRUCTORID = r.CONSTRUCTORID as tmp2
18 inner join QUALIFYING q on q.RACEID = tmp2.RACEID and q.CONSTRUCTORID = tmp2.CONSTRUCTORID
19 where r.year > 2016;

```

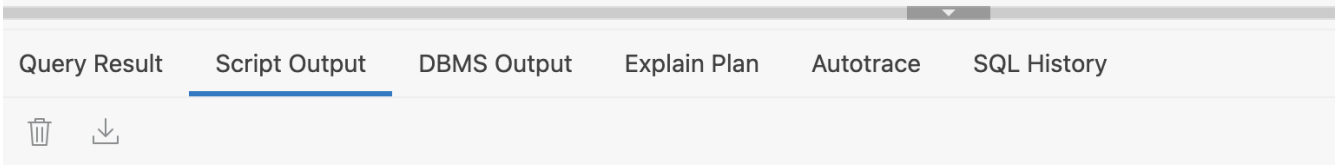


Table TRANSACTIONA_APP_RACE_STRATEGY created.

Elapsed: 00:00:00.242

Figure-9: Creating Data Model for Application-3 in Oracle Cloud

5. Analysis

Since connecting Tableau directly to Oracle cloud is an easy and efficient way of analyzing data visually, we performed most of our analysis on Tableau. Here are some snippets from our exploration.

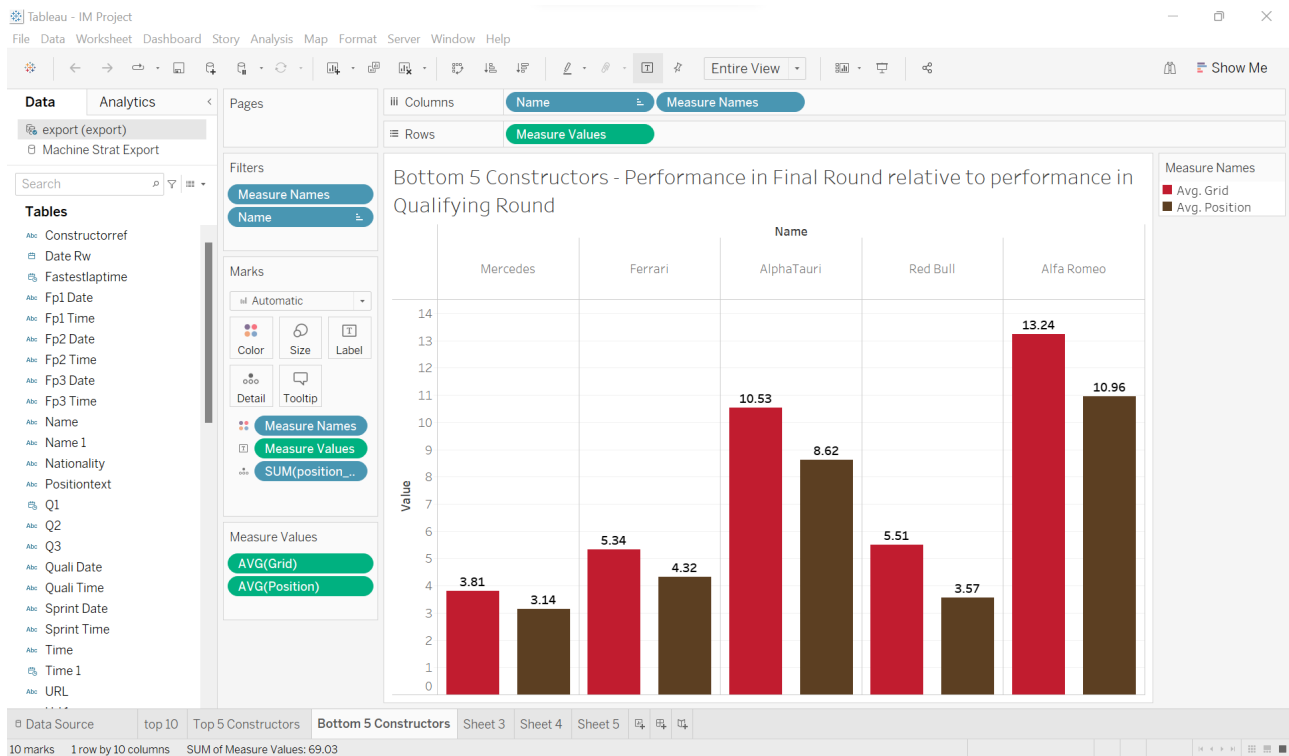


Figure- 10: High Ranking Qualifying vs Lower Ranking Grand Prix Performances

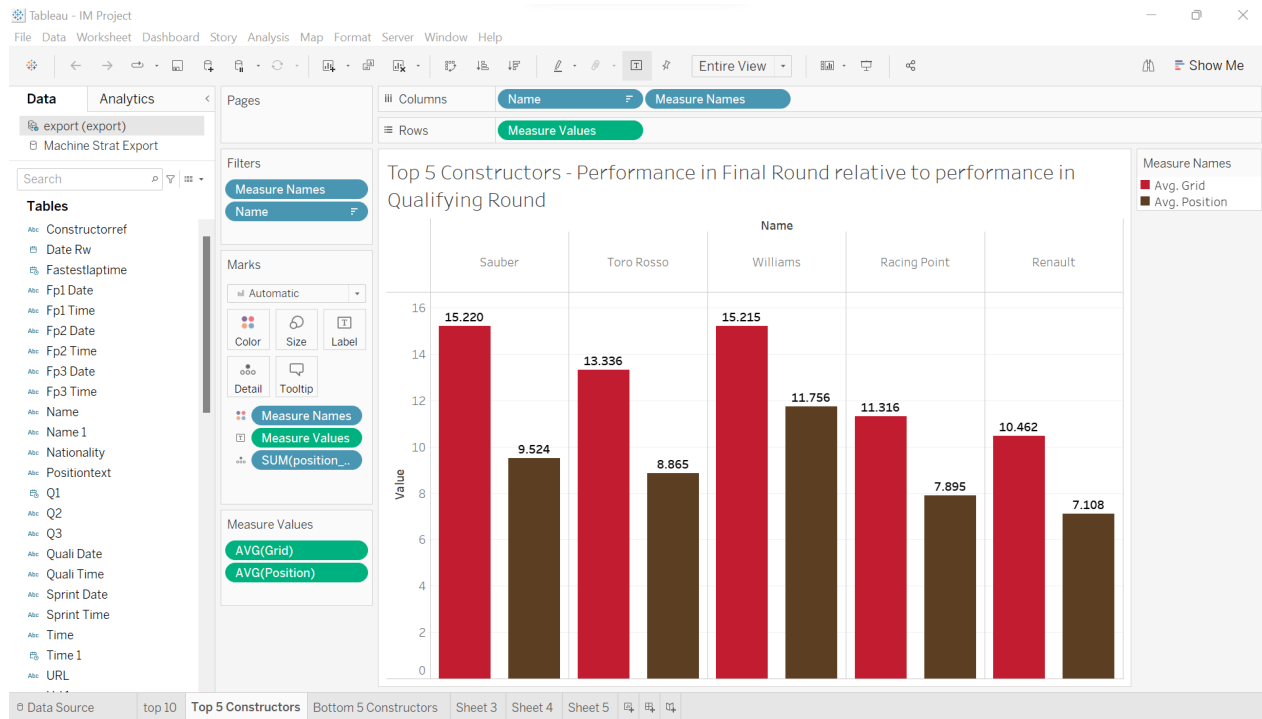


Figure- 11: Low Ranking Qualifying vs Higher Ranking Grand Prix Performances

6. Outcome

6.1 Insights

Some of the insights from our analysis from the *Driver Perspective* are as follows

- Using the last five years on an equal basis, we would buy Lewis Hamilton as our driver
- However, if we view the most recent year's performance as an indicator of future performance, we would instead choose to buy Max Verstappen
- If we were to do this analysis again, we would try to see how consistent a change is in the leaderboard year after year

For the *Race Engineering Perspective*, we concentrated on the comparison between the Pit-stop Timing and Final result, and here are a few aha moments from our analysis

- The relationship between pit-stop time and average position cannot easily be detected
 - In certain scenarios, a small decrease in pit-stop time may result in a change in results
 - However, there are cases in which someone has a very quick pit-stop, but they remain in last place, so that very fast pit-stop has little-to-no impact on the driver's final position
- If we were to do this analysis in the future, it may be best to find patterns between pit-stop time and

final position of only the drivers in the top-10

We also performed an analysis to examine the *Race Strategy Perspective*. For this, we analyzed the importance of Qualifying Round on the results of the race. The insights we got from this analysis is as follows

- We found that among the top teams, qualifying times are an important indicator of their final race positions
- However, for the mid-table constructors, qualifying is not a strong indicator of final race position

6.2 Roadblocks

- Our data was huge, so we were not able to use the ORACLE SQL Server provided in the class
- We were unable to use Google Cloud platform since it does not support referential integrity and is only used for LDM (Logical Data Modeling).
- Since we are using Oracle Autonomous Database, we were not able to communicate across databases. Hence we had to create our aggregated and raw tables in the same databases.
- We had to perform a lot of data preprocessing to extract meaningful insights out of it
 - When a driver doesn't finish a race, the position is noted as null
 - Date format was not supportive

6.3 Conclusion

We simulated how a business planning for the market entry would use publicly available data to make decisions. We learned how the knowledge base is set up, and how it can be molded in various forms for the purpose of analysis. We also discovered what problems a business could face when setting up these systems, from an IT perspective, as well as the Analyst perspective who has to deal with publicly available data which was not designed to be for the purpose of the analysis being performed.