**Documentation**

5/12/2021

# Table of Contents

[Data set selection 3](#_Toc71593832)

[overview and description of the data set 3](#_Toc71593833)

[ER Diagram 3](#_Toc71593834)

[Preparation of the data sources 4](#_Toc71593835)

[Source Details 4](#_Toc71593836)

[Source Table Details 5](#_Toc71593837)

[Solution Architecture 10](#_Toc71593838)

[Data Warehouse Design and Development 12](#_Toc71593839)

[ETL Development 14](#_Toc71593840)

[Data Stage ETL (ETL between Data Source layer and Staging layer) 14](#_Toc71593841)

[Data Warehouse ETL (ETL between Staging Layer and Data warehouse Layer ) 18](#_Toc71593842)

[Proofs in database level to validate the successful execution of both ETL’s 27](#_Toc71593843)

# Data set selection

Data set for the assignment was obtained from Kaggle.com. The link to the data set is mentioned below.

<https://www.kaggle.com/gdaley/hkracing>

## overview and description of the data set

This data set basically describes the performance of each horse participated to the horse races in Shatin and Happy Valley in Hong Kong. Performance of each horse is described regarding the parameters such as result of the race, position of the horse in different sections through the race, time taken by horse to pass different sections and the finish time of the race along with the details of the trainer of the horse at the time horse performs in the race and the details of the jockey. And also this dataset includes in depth details of each race that the horse participated such as race class, venue, race track configuration and condition, distance of the race, winning prize etc.

## ER Diagram

Chart, diagram

Description automatically generated

# Preparation of the data sources

* Source data set which is taken from the Kaggle consisted of mainly two csv files named runs and races.
* Details of the horse such as horse id, horse type and horse country were separated from races.csv file into a horses\_updated\_excel excel file
* Separated trainer ids from the races.csv and prepared the other details of trainer such as trainer name and trainer location
* Separated jockey ids from the races.csv and prepared the other details of the jockey such as jockey name and the country of the jockey.
* Races.csv file completely imported to the source table

## Source Details

The two main sources of data are **SQL Database** and one **Excel File**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source | Source Type | Object Name | Scheme | Object Type | Description |
| Horse\_Racing\_Source | SQL Database | jockey | dbo | Table | Includes details of the jockey such as jockey id , name and country |
| Horse\_Racing\_Source | SQL Database | trainer | dbo | Table | Include details of the trainer of the horse such as trainer id, name and the location |
| Horse\_Racing\_Source | SQL Database | race | dbo | Table | Includes details of each race the horses are participated such as race id, race no, venue , prize etc |
| Horse\_Racing\_Source | SQL Database | runs | dbo | Table | Includes details of performance of the horses in each race such as age of the horse at the race time along with the results of the race |
| Horses-updated-excel | Excel File |  |  |  | Includes details of each horse such as horse id, horse name , horse type and horse country |

## Source Table Details

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Source | Source Type | Table Name | Column Name | Data Type | Link Table | Link Column | Decription |
| Horse\_Racing\_Source | SQL Database | jockey | jockey\_id | int |  |  | Unique jockey id |
| jockey\_name | nvarchar |  |  | Jockey name |
| country | nvarchar |  |  | Country |
| Horse\_Racing\_Source | SQL Database | trainer | trainer\_id | int |  |  | Unique trainer id |
| trainer\_name | nvarchar |  |  | Trainer name |
| trainer\_location | nvarchar |  |  | Trainer location |
| Horse\_Racing\_Source | SQL Database | race | race\_id | int |  |  | Unique race id |
| date | datetime |  |  | Date of the race |
| venue | nvarchar |  |  | Venue of the race |
| race\_no | int |  |  | Race number of the race in the day’s meeting |
| config | nvarchar |  |  | Race track configuration |
| surface | nvarchar |  |  | Type of race track surface |
| distance | int |  |  | Distance of the race in meters |
| going | nvarchar |  |  | Track condition |
| horse\_ratings | nvarchar |  |  | The range of horse ratings that may participate in this race |
| prize | money |  |  | The winning prize |
| race class | int |  |  | Class of the race |
| sec time1 | float |  |  | Time taken by leader of the race to reach the end of the 1st sectional point |
| sec time2 | float |  |  | Time taken by the leader of the race to reach the end of the 2nd sectional point |
| sec time3 | float |  |  | Time taken by the leader of the race to reach the end of the 3rd sectional point |
| sec time4 | float |  |  | Time taken by the leader of the race to reach the end of the 4th sectional point |
| sec time5 | float |  |  | Time taken by the leader of the race to reach the end of the 5th sectional point |
| sec time6 | float |  |  | Time taken by the leader of the race to reach the end of the 6th sectional point |
| sec time7 | float |  |  | Time taken by the leader of the race to reach the end of the 7th sectional point |
| time\_leader\_1 | float |  |  | time taken by the leader of the race in the 1st section only (sec) |
| time\_leader\_2 | float |  |  | time taken by the leader of the race in the 2nd section only (sec) |
| time\_leader\_3 | float |  |  | time taken by the leader of the race in the 3rd section only (sec) |
| time\_leader\_4 | float |  |  | time taken by the leader of the race in the 4th section only, if any (sec) |
| time\_leader\_5 | float |  |  | time taken by the leader of the race in the 5th section only, if any (sec) |
| time\_leader\_6 | float |  |  | time taken by the leader of the race in the 6th section only, if any (sec) |
| time\_leader\_7 | float |  |  | time taken by the leader of the race in the 7th section only, if any (sec) |
| place\_combination1 | int |  |  | placing horse no (1st) |
| place\_combination2 | int |  |  | placing horse no (2nd) |
| place\_combination3 | int |  |  | placing horse no (3rd) |
| place\_combination4 | int |  |  | placing horse no (4th) |
| place\_dividend1 | float |  |  | placing dividend paid (for placecombination1) |
| place\_dividend2 | float |  |  | placing dividend paid (for placecombination2) |
| place\_dividend3 | float |  |  | placing dividend paid (for placecombination3) |
| place\_dividend4 | float |  |  | placing dividend paid (for placecombination4) |
| win\_combination1 | int |  |  | winning horse no |
| win\_dividend1 | float |  |  | winning dividend paid (for wincombination1) |
| win\_ combination2 | int |  |  | joint winning horse no, if any |
| Win\_dividend2 | float |  |  | winning dividend paid (for wincombination2, if any) |
| Horse\_Racing\_Source | SQL Database | runs | race\_id | int | runs | race\_id | race id |
| horse\_no | int |  |  | The number assigned to this horse, in the race |
| horse\_id | int | horse | horse\_id | unique identifier of this horse |
| result | int |  |  | finishing position of this horse in the race |
| won | int |  |  | whether horse won (1) or otherwise (0) |
| lengths\_behind | float |  |  | finishing position, as the number of horse lengths behind the winner |
| horse\_age | int |  |  | current age of this horse at the time of the race |
| horse\_rating | int |  |  | rating number assigned by to this horse at the time of the race |
| horse\_gear | nvarchar |  |  | gear carried by the horse in the race |
| **declared\_weight** | float |  |  | declared weight of the horse and jockey, in lbs |
| **actual\_weight** | float |  |  | actual weight carried by the horse, in lbs |
| draw | int |  |  | post position number of the horse in this race |
| **position\_sec1** | int |  |  | position of this horse (ranking) in section 1 of the race |
| **position\_sec2** | int |  |  | position of this horse (ranking) in section 2 of the race |
| **position\_sec3** | int |  |  | position of this horse (ranking) in section 3 of the race |
| **position\_sec4** | int |  |  | position of this horse (ranking) in section 4 of the race, if any |
| **position\_sec5** | int |  |  | position of this horse (ranking) in section 5 of the race, if any |
| **position\_sec6** | int |  |  | position of this horse (ranking) in section 6 of the race, if any |
| **behind\_sec1** | float |  |  | position of this horse (lengths behind leader) in section 1 of the race |
| **behind\_sec2** | float |  |  | position of this horse (lengths behind leader) in section 2 of the race |
| **behind\_sec3** | float |  |  | position of this horse (lengths behind leader) in section 3 of the race |
| **behind\_sec4** | float |  |  | position of this horse (lengths behind leader) in section 4 of the race, if any |
| **behind\_sec5** | float |  |  | position of this horse (lengths behind leader) in section 5 of the race, if any |
| **behind\_sec6** | float |  |  | position of this horse (lengths behind leader) in section 6 of the race, if any |
| **time1** | float |  |  | time taken by the horse to pass through the 1st section of the race (sec) |
| **time2** | float |  |  | time taken by the horse to pass through the 2nd section of the race (sec) |
| time3 | float |  |  | time taken by the horse to pass through the 3rd section of the race (sec) |
| **time4** | float |  |  | time taken by the horse to pass through the 4th section of the race, if any (sec) |
| **time5** | float |  |  | time taken by the horse to pass through the 5th section of the race, if any (sec) |
| **time6** | float |  |  | time taken by the horse to pass through the 6th section of the race, if any (sec) |
| **win\_odds** | float |  |  | win odds for this horse at start of race |
| **place\_odds** | float |  |  | place (finishing in 1st, 2nd or 3rd position) odds for this horse at start of race |
| **trainer\_id** | int | triainer | trainer\_id | unique identifier of the horse's trainer at the time of the race |
| **jockey\_id** | int | jockey | jockey\_id | unique identifier of the jockey riding the horse in this race |
| horses-updated-excel | Excel file | horse | **horse\_id** | int |  |  | Unique identifier of the horse |
| **horse\_name** | nvarchar |  |  | name of the horse |
| **horse\_type** | nvarchar |  |  | Type of the horse |
| **horse\_country** | nvarchar |  |  | Country of the horse |

# Solution Architecture

Diagram

Description automatically generated

**Layers**

1. **Data Sources Layer**

This layer represents the Horse\_Racing\_Source database and the excel file which contains the details of the horses

1. **Staging Layer**

This layer represents the Horse\_Racing\_Stage database which contains the staging tables created after the execution of Data stage ETL process

1. **Data Warehouse Layer**

Data Warehouse layer contains the Dimension tables and Fact table which is designed according to the star schema

**ETL Processes**

1. **Data Stage ETL (ETL process between Data sources layer and Staging layer)**

This ETL process mainly implemented to load data in the Data Sources Layer (SQL Databases and Excel file) to Staging layer. In this ETL Process

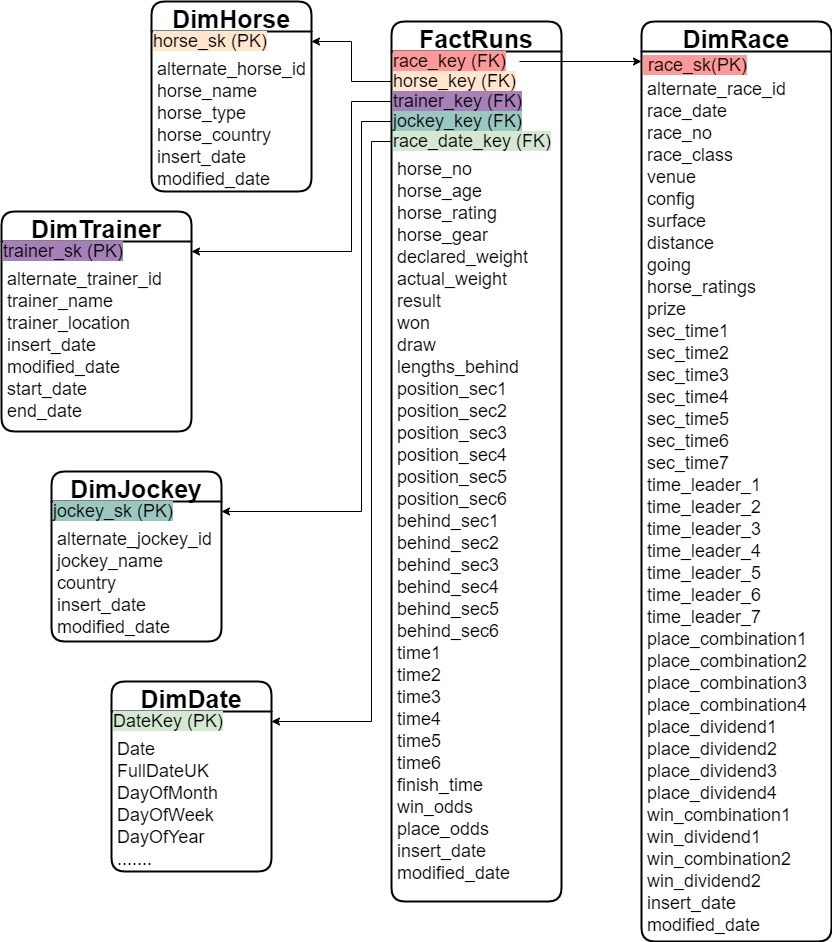
After executing this ETL Process below tables are created in the Horse\_Racing\_Stage Data base and data will be loaded to those tables.

* StgHorse
* StgJockey
* StgRace
* StgRuns
* StgTrainer

1. **ETL between Staging layer and Data warehouse layer**

This ETL process is implemented to load data in the Staging Layer (Horse\_Racing\_StageDB) to Data warehouse. In this ETL process, data is validated, cleaned, transformed and aggregated. And also necessary transformations such as lookups, derived columns, merge, union, sort and replacement of null values were performed.

# Data Warehouse Design and Development



Data warehouse is designed according to the Star Schema with one fact table and five dimension tables

|  |  |  |
| --- | --- | --- |
|  | Dimension / Fact | Description |
| FactRuns | Fact Table | This Fact table is designed to store details of performance of each horse participated to races along with the details of race date key, trainer\_ key (SK of trainer at the time of race), jockey\_key (SK of the jockey at the time of the race) |
| DimHorse | Dimension Table | This dimension table stores the details of the each horse |
| DimTrainer | Dimension Table | Stores the details of trainers |
| DimJockey | Dimension Table | Stores the details of jockies |
| DimRace | Dimension Table | Stores the details of each race |
| DimDate | Dimension Table | Stores the details of dates |

DimTrainer is considered as a type 2 slowly changing dimension.

**Grain** of the fact table is one record for performance of each horse per each race horse participated.

**Hierarchies considered**

DimHorse

* horse\_type -> horse
* horse\_country -> horse

DimRace

* venue -> race
* surface -> race
* horse\_ratings -> race
* race\_class -> race

# ETL Development

## Data Stage ETL (ETL between Data Source layer and Staging layer)

This ETL is mainly implemented to extract and load data in Horse\_Racing\_Source database into Horse\_Racing\_Stage database

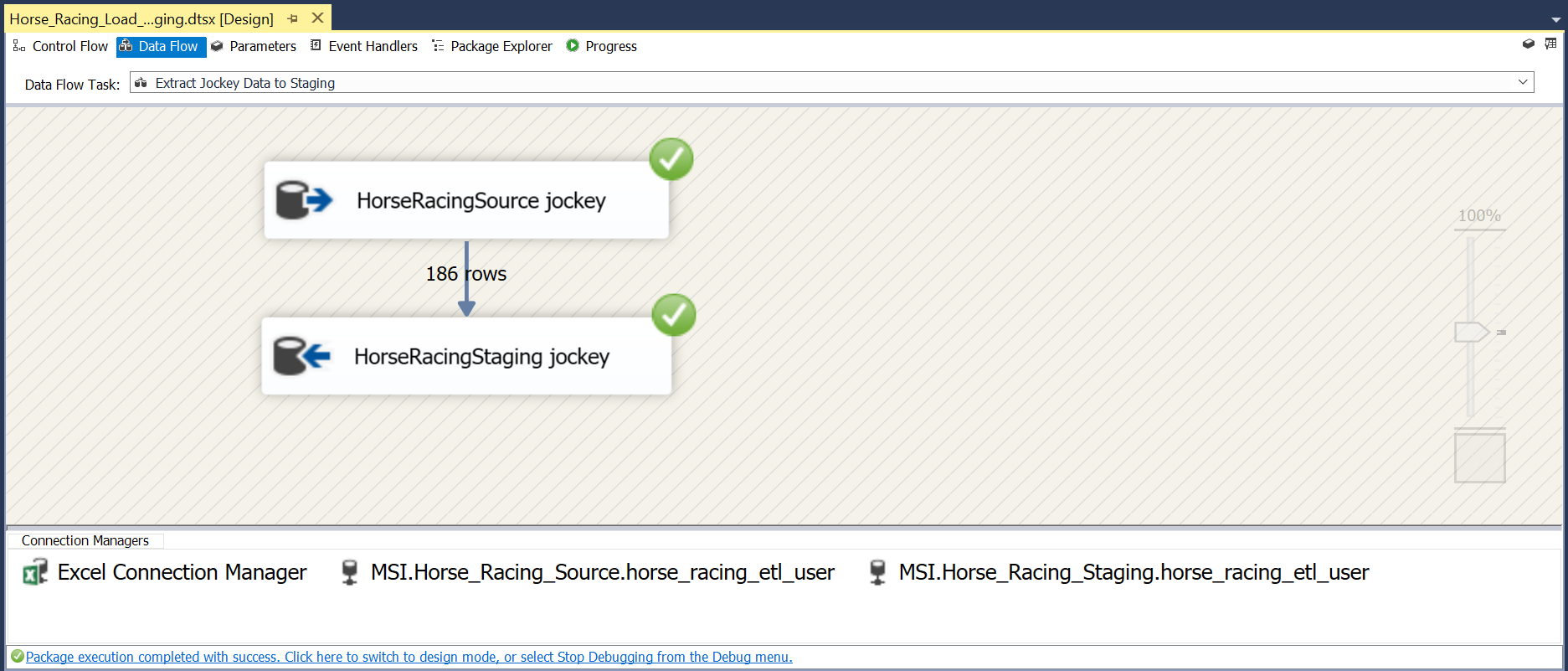
**Control flow of the ETL**

**Below diagram describes the orchestration of the ETL process**

**Graphical user interface, text, application

Description automatically generated**

**Data flow of Extract Jockey Data to Staging**



**Data flow of Extract Trainer Data to Staging**

**A picture containing graphical user interface

Description automatically generated**

**Data flow extract Horse Data to Staging**

**A picture containing graphical user interface

Description automatically generated**

**Data flow of Extract Race Data to Staging**

A picture containing graphical user interface

Description automatically generated

**Data Flow Extract Runs data to Staging**

A picture containing diagram

Description automatically generated

**Execution Results of the ETL**

Graphical user interface, text, application, email

Description automatically generated

## Data Warehouse ETL (ETL between Staging Layer and Data warehouse Layer )

**Control flow of the ETL**

**Below diagram describes the orchestration of the ETL process**

**Graphical user interface, application

Description automatically generated**

**Data Flow of Transform and Load Jockey Data**

Graphical user interface

Description automatically generated

Below Stored Procedure is used to handle the insertion of new records and modified records

Graphical user interface, text, application, email

Description automatically generated

**Data Flow of Transform and Load Horses Data**

Timeline

Description automatically generated

Below Stored Procedure is used to handle the insertion of new records and modified records

Graphical user interface, text, application

Description automatically generated

**Data Flow of Transform and Load Race Data**

* Data type of the surface (type of race track surface: 1 = dirt and 0 = turf) column in StgRace was integer type.
* It is converted to accept Unicode String data type.
* Then replaced 1’s to dirt and 0’s to turf

Graphical user interface, text, email

Description automatically generated

Below stored procedure is used to handle the insertion of new records and modified records

CREATE PROCEDURE dbo.UpdateDimRace

@raceID int,

@race\_date date,

@venue nvarchar(50),

@race\_no int,

@config nvarchar(50),

@surface nvarchar(50),

@distance int,

@going nvarchar(50),

@horse\_ratings nvarchar(50),

@prize money,

@race\_class int,

@sec\_time1 float,

@sec\_time2 float,

@sec\_time3 float,

@sec\_time4 float,

@sec\_time5 float,

@sec\_time6 float,

@sec\_time7 float,

@time\_leader\_1 float,

@time\_leader\_2 float,

@time\_leader\_3 float,

@time\_leader\_4 float,

@time\_leader\_5 float,

@time\_leader\_6 float,

@time\_leader\_7 float,

@place\_combination1 int,

@place\_combination2 int,

@place\_combination3 int,

@place\_combination4 int,

@place\_dividend1 float,

@place\_dividend2 float,

@place\_dividend3 float,

@place\_dividend4 float,

@win\_combination1 int,

@win\_dividend1 float,

@win\_combination2 int,

@win\_dividend2 float

AS

BEGIN

if not exists(select race\_sk

from dbo.DimRace

where alternate\_race\_id = @raceID)

BEGIN

insert into dbo.DimRace

(alternate\_race\_id,race\_date,venue,race\_no,config,surface,distance,going,horse\_ratings,

prize,race\_class,sec\_time1,sec\_time2,sec\_time3,sec\_time4,sec\_time5,sec\_time6,sec\_time7,

time\_leader\_1,time\_leader\_2,time\_leader\_3,time\_leader\_4,time\_leader\_5,time\_leader\_6,time\_leader\_7,place\_combination1,place\_combination2,

place\_combination3,place\_combination4,place\_dividend1,place\_dividend2,place\_dividend3,place\_dividend4,

win\_combination1,win\_dividend1,win\_combination2,win\_dividend2,insert\_date,modified\_date)

VALUES

(@raceID,@race\_date,@venue,@race\_no,@config,@surface,@distance,@going,@horse\_ratings,@prize,@race\_class,

@sec\_time1, @sec\_time2,@sec\_time3,@sec\_time4,@sec\_time5,@sec\_time6,@sec\_time7,

@time\_leader\_1,@time\_leader\_2,@time\_leader\_3,@time\_leader\_4,@time\_leader\_5,@time\_leader\_6,@time\_leader\_7,@place\_combination1,@place\_combination2,

@place\_combination3,@place\_combination4,@place\_dividend1,@place\_dividend2,@place\_dividend3,@place\_dividend4,

@win\_combination1,@win\_dividend1,@win\_combination2,@win\_dividend2,GETDATE(),GETDATE())

END;

if exists(select race\_sk

from dbo.DimRace

where alternate\_race\_id = @raceID)

BEGIN

update dbo.DimRace

set race\_date=@race\_date,

venue=@venue,

race\_no= @race\_no,

config = @config,

surface = @surface,

distance = @distance,

going = @going,

horse\_ratings = @horse\_ratings,

prize = @prize,

race\_class = @race\_class,

sec\_time1 = @sec\_time1,

sec\_time2 = @sec\_time2,

sec\_time3 = @sec\_time3,

sec\_time4 = @sec\_time4,

sec\_time5 = @sec\_time5,

sec\_time6 = @sec\_time6,

sec\_time7 = @sec\_time7,

time\_leader\_1 = @time\_leader\_1,

time\_leader\_2 = @time\_leader\_2,

time\_leader\_3 = @time\_leader\_3,

time\_leader\_4 = @time\_leader\_4,

time\_leader\_5 = @time\_leader\_5,

time\_leader\_6 = @time\_leader\_6,

time\_leader\_7 = @time\_leader\_7,

place\_combination1 = @place\_combination1,

place\_combination2 = @place\_combination2,

place\_combination3 = @place\_combination3,

place\_combination4 = @place\_combination4,

place\_dividend1 = @place\_dividend1,

place\_dividend2 = @place\_dividend2,

place\_dividend3 = @place\_dividend3,

place\_dividend4 = @place\_dividend4,

win\_combination1 = @win\_combination1,

win\_dividend1 = @win\_dividend1,

win\_combination2 = @win\_combination2,

win\_dividend2 = @win\_dividend2,

modified\_date = GETDATE()

where alternate\_race\_id = @raceID

END;

END;

**Data flow of Transform and Load Trainer Data (Slow Changing Dimension)**

* Trainer Dimension is implemented as a type 2 slowly changing dimension
* trainer\_location is considered as the historical attribute
* trainer\_name is considered as the changing attribute

A picture containing diagram

Description automatically generated

Text

Description automatically generated with low confidence

Diagram

Description automatically generated

A picture containing text, stationary, envelope, businesscard

Description automatically generated

**Data flow of Transform and Load Runs Fact Table**

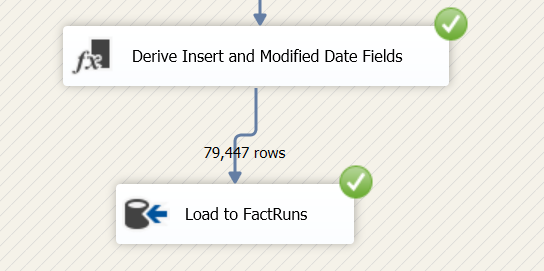
Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated



* When extracting from runs staging, sort was used to sort the data by race\_id and passed through all the columns
* When extracting from race staging, sort was used to sort the date by race\_id and passed through only the date column
* Then two sources were joined using merge join (inner join is used) by considering race\_id as the join key
* “Race sk Lookup” is used to insert respective surrogate key belonged to race\_id into race\_key column
* “Race Date Lookup” is used to insert respective race\_date\_key
* “Horse Lookup” is used to insert respective surrogate key belonged to horse\_id into horse\_key column
* “Trainer Lookup” is used to insert respective surrogate key belonged to trainer\_id into trainer\_key column
* “Jockey Lookup” is used to insert respective surrogate key belonged to jockey\_id into jockey\_key column
* Replaced null values in time4, time5, time6 columns to 0
* By using derived column insert date and modified date inserted
* Finally loaded data into fact table

**Execution Results of the ETL**

Graphical user interface, text, application, email

Description automatically generated

# Proofs in database level to validate the successful execution of both ETL’s

**Jockey**

Graphical user interface, application

Description automatically generated

**Horse**

Graphical user interface, application

Description automatically generated

**Race**

Graphical user interface, application

Description automatically generated

**Trainer**

Graphical user interface, text, application

Description automatically generated

**Fact Runs**

Graphical user interface, application

Description automatically generated