1.2. Data cleaning process

Cleaning strategies we used

Before operational databases are ready to be transformed to a data warehouse, we need to identify dirty data and clean them — a crucial step in the activities of pre-data warehousing.

We based the week 4’s contents where we learnt five aspects of problems and run those queries we learnt one by one through every entity in the operational database. The reason why we do this — despite not being efficient — is that this is an effective approach where we covered all aspects of problems and ensure that all entitles are CLEANED.

Full SQL script for cleaning data

```
-- =============================  Problem 1: Duplicate
SELECT BOOKINGID, COUNT(*) as duplicate_records
FROM MonCity.BOOKING
GROUP BY BOOKINGID
HAVING COUNT (*) >1;

-- solution
DROP TABLE Clean_Booking CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_Booking as
SELECT DISTINCT *
FROM MonCity.BOOKING;


-- =============================  Problem 2: Null value problem
SELECT *
FROM moncity.ACCIDENTINFO
WHERE ACCIDENTID IS NULL;

-- solution
DROP TABLE Clean_Accidentinfo CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_Accidentinfo as
SELECT *
FROM MonCity.ACCIDENTINFO
WHERE ACCIDENTID IS NOT NULL;

-- =============================  Problem 3: Incorrect Values
SELECT *
FROM moncity.maintenance
WHERE maintenancecost < 0;

-- solution
DROP TABLE Clean_maintenance CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_maintenance as
```

```
SELECT *
FROM MonCity.MAINTENANCE
WHERE MAINTENANCECOST > 0;

-- =============================  Problem 4: Relationship Problem
SELECT ERRORcode
FROM ACCIDENTINFO
WHERE ERRORcode  NOT IN ( SELECT ERRORcode FROM MonCity.ERROR );

-- solution
DELETE
FROM CLEAN_ACCIDENTINFO
WHERE ERRORcode NOT IN ( SELECT ERRORcode FROM MonCity.ERROR );

-- =============================  Problem 5: Relationship Problem
SELECT *
FROM MONCITY.passenger
WHERE FACULTYID NOT IN
(SELECT FACULTYID
FROM moncity.faculty);

-- solution
DROP TABLE Clean_passenger CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_passenger as
SELECT DISTINCT *
FROM MonCity.passenger;

DELETE
FROM Clean_passenger
WHERE FACULTYID NOT IN
( SELECT FACULTYID
FROM moncity.faculty );
```

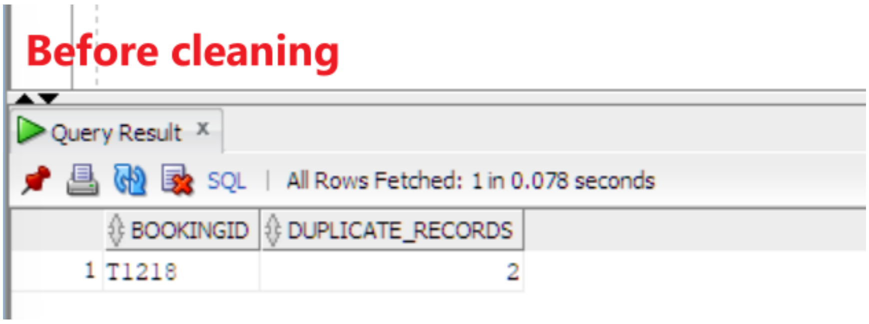
Demonstration of the difference between before and after cleaning

At the end, 5 types of problem are identified corresponding to the problems illustrated below.

- Problem 1: Duplicate
- Problem 2: Null value problem
- Problem 3: Incorrect Values
- Problem 4: Relationship Problem and Inconsistent Values
- Problem 5: Relationship Problem

Here we demonstrated 5 specific problems we found:

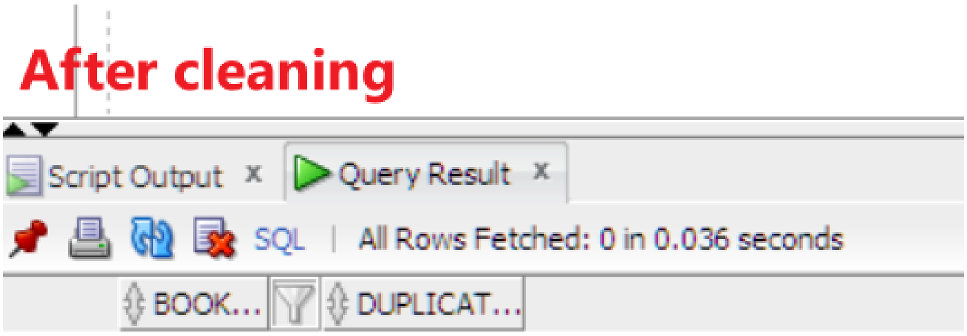
▼ Problem 1: Two duplicate entries found in `MonCity.BOOKING` where `bookingid = 'T1218'`



All Rows Fetched: 1 in 0.078 seconds

	BOOKINGID	DUPLICATE_RECORDS
1	T1218	2

```
SELECT BOOKINGID, COUNT(*) as
duplicate_records
FROM MonCity.BOOKING
GROUP BY BOOKINGID
HAVING COUNT (*) >1;
```



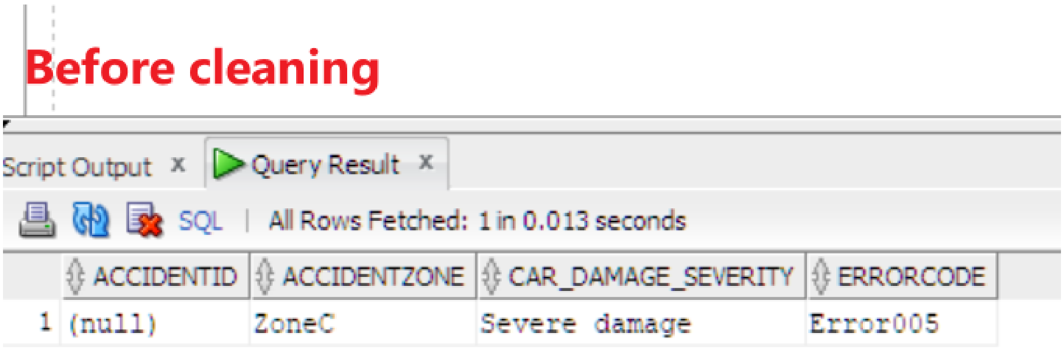
All Rows Fetched: 0 in 0.036 seconds

BOOK...	DUPLICAT...

```
DROP TABLE Clean_Booking CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_Booking as
SELECT DISTINCT *
FROM MonCity.BOOKING;

SELECT BOOKINGID,
COUNT(*) as duplicate_records
FROM Clean_Booking
GROUP BY BOOKINGID
HAVING COUNT (*) >1;
```

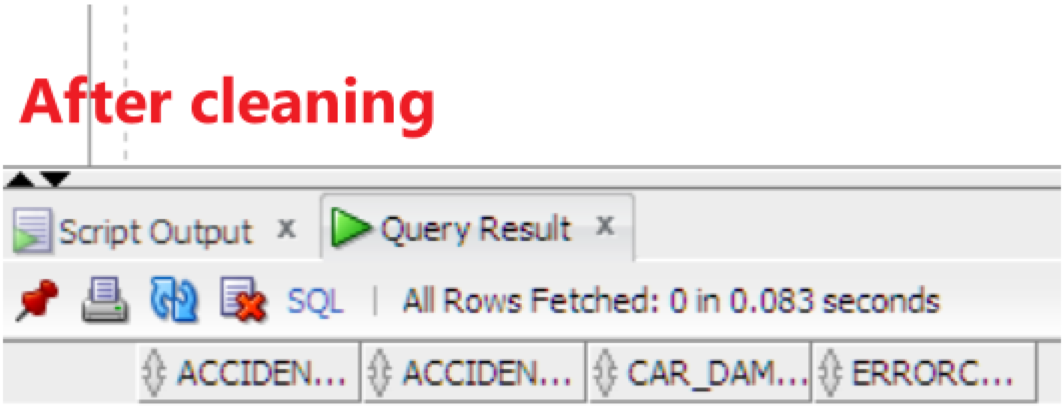
▼ Problem 2: One entry found with NULL value



All Rows Fetched: 1 in 0.013 seconds

	ACCIDENTID	ACCIDENTZONE	CAR_DAMAGE_SEVERITY	ERRORCODE
1	(null)	ZoneC	Severe damage	Error005

```
SELECT *
FROM moncity.ACCIDENTINFO
WHERE ACCIDENTID IS NULL;
```



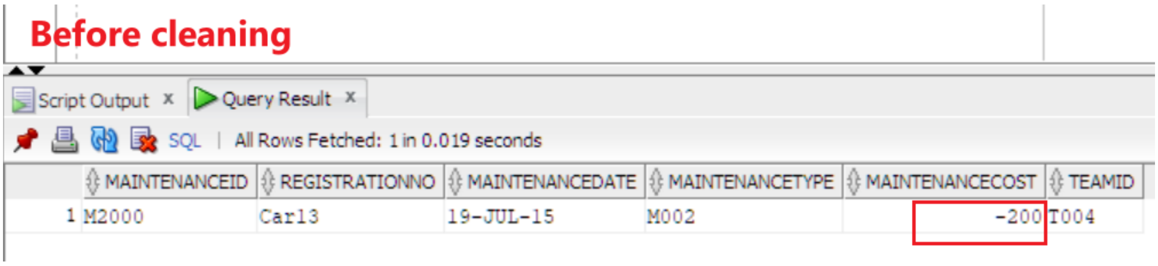
All Rows Fetched: 0 in 0.083 seconds

ACCIDEN...	ACCIDEN...	CAR_DAM...	ERRORC...

```
DROP TABLE Clean_Accidentinfo CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_Accidentinfo as
SELECT *
FROM MonCity.ACCIDENTINFO
WHERE ACCIDENTID IS NOT NULL;

SELECT *
FROM Clean_Accidentinfo
WHERE ACCIDENTID IS NULL;
```

▼ Problem 3: One entry found with negative amount of maintenance cost where `maintenanceid = 'M2000'`

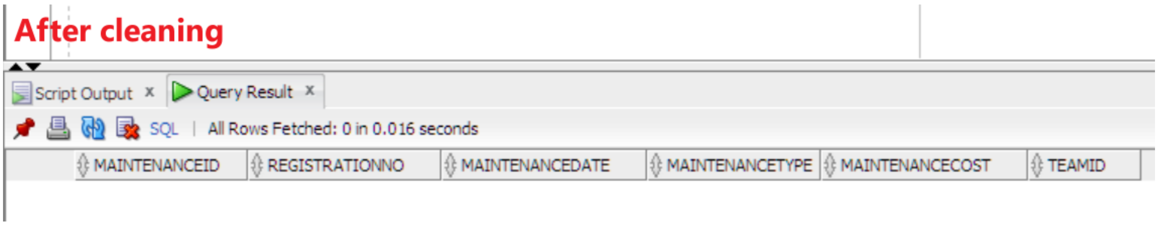


Before cleaning

All Rows Fetched: 1 in 0.019 seconds

	MAINTENANCEID	REGISTRATIONNO	MAINTENANCEDATE	MAINTENANCETYPE	MAINTENANCECOST	TEAMID
1	M2000	Car13	19-JUL-15	M002	-200	T004

```
SELECT *
FROM moncity.maintenance
WHERE maintenancecost < 0;
```



After cleaning

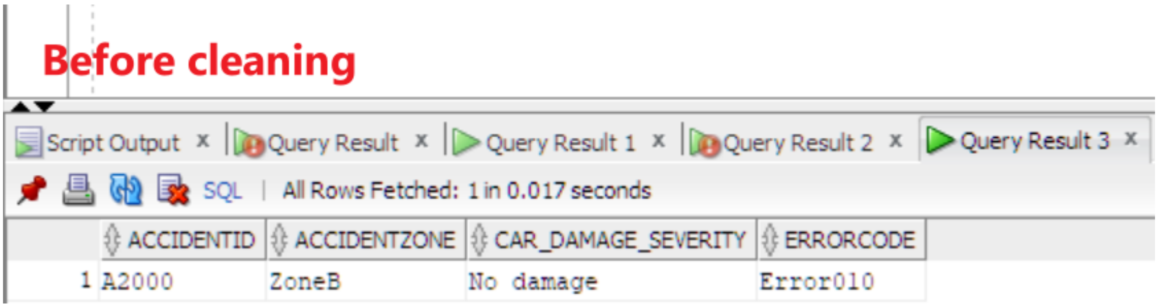
All Rows Fetched: 0 in 0.016 seconds

	MAINTENANCEID	REGISTRATIONNO	MAINTENANCEDATE	MAINTENANCETYPE	MAINTENANCECOST	TEAMID

```
DROP TABLE Clean_maintenance CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_maintenance as
SELECT *
FROM MonCity.MAINTENANCE
WHERE MAINTENANCECOST > 0;

SELECT *
FROM Clean_maintenance
WHERE maintenancecost < 0;
```

▼ Problem 4: One entry found `MONCITY.ACCIDENTINFO` in where `accidentid = 'A2000'` does not exist in `MONCITY.error` AND `error code = ‘Error010’` which is inconsistent



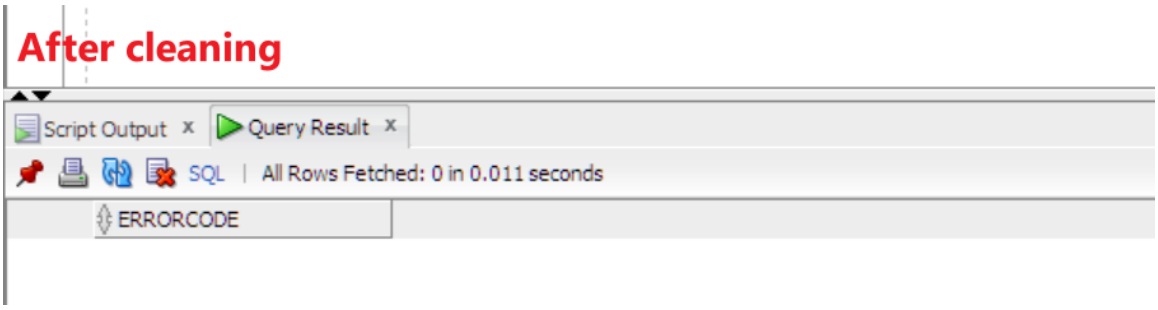
Before cleaning

All Rows Fetched: 1 in 0.017 seconds

	ACCIDENTID	ACCIDENTZONE	CAR_DAMAGE_SEVERITY	ERRORCODE
1	A2000	ZoneB	No damage	Error010

```
SELECT ERRORcode
FROM ACCIDENTINFO
```

```
WHERE ERRORcode  NOT IN ( SELECT ERRORcode FROM MonCity.ERROR );
```



After cleaning

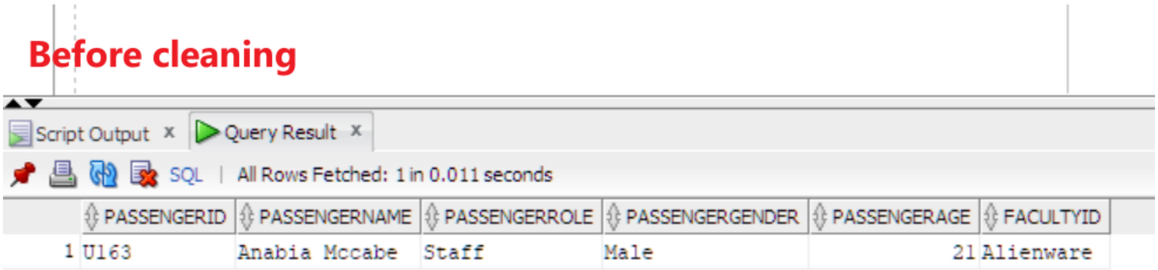
All Rows Fetched: 0 in 0.011 seconds

	ERRORCODE

```
DELETE
FROM CLEAN_ACCIDENTINFO
WHERE ERRORcode NOT IN ( SELECT ERRORcode FROM MonCity.ERROR );

SELECT ERRORcode
FROM CLEAN_ACCIDENTINFO
WHERE ERRORcode  NOT IN ( SELECT ERRORcode FROM MonCity.ERROR );
```

▼ Problem 5: One entry found `MONCITY.passenger` in where `passenger_id = 'U163'` does not exist in `MONCITY.faculty`

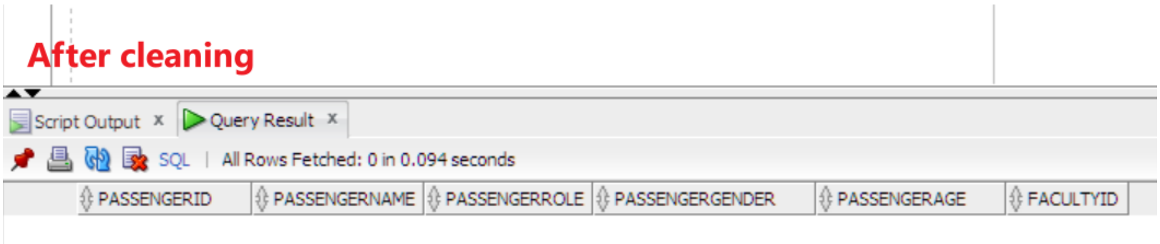


Before cleaning

All Rows Fetched: 1 in 0.011 seconds

	PASSENGERID	PASSENGERNAME	PASSENGERROLE	PASSENGERGENDER	PASSENGERAGE	FACULTYID
1	U163	Anabia Mccabe	Staff	Male	21	Alienware

```
SELECT *
FROM MONCITY.passenger
WHERE FACULTYID NOT IN
(SELECT FACULTYID
FROM moncity.faculty);
```



After cleaning

All Rows Fetched: 0 in 0.094 seconds

	PASSENGERID	PASSENGERNAME	PASSENGERROLE	PASSENGERGENDER	PASSENGERAGE	FACULTYID

```
DROP TABLE Clean_passenger CASCADE CONSTRAINTS PURGE;
CREATE TABLE Clean_passenger as
SELECT DISTINCT *
FROM MonCity.passenger;

DELETE
FROM Clean_passenger
WHERE FACULTYID NOT IN
( SELECT FACULTYID
FROM moncity.faculty );

SELECT *
FROM Clean_passenger WHERE FACULTYID NOT IN
(SELECT FACULTYID
FROM moncity.faculty);
```