MONASH University
Information Technology

FIT 5195 Major Assignment

Student name: Yue Zhang Student ID: 30976316
Student name: Lei Hu Student ID: 30841933

MONASH University
Information Technology

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
30841933	Hu	Lei
30976316	Zhang	Yue

***** Please include the names of all other group members.

Unit name and code	FIT5195 Business intelligence and data warehousing	
Title of assignment	FIT5195 Major Assignment	
Lecturer/tutor	Soon Lay-Ki / Arif Hidayat & Shuyi Sun	
Tutorial day and time	Tue 3:00pm – 5:00pm	**Campus Clayton**
Is this an authorised group assignment? ☒ **Yes** ☐ **No**		
Has any part of this assignment been previously submitted as part of another unit/course? ☐ **Yes** ☒ **No**		
Due Date Week 9, Wednesday 5-May-2021, 11:55pm		**Date submitted 5-May-2021**

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date) ..... Signature of lecturer/tutor .....

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  - i. provide to another member of faculty and any external marker; and/or
  - ii. submit it to a text matching software; and/or
  - iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .....Yue Zhang.....Lei Hu..... Date……05/05/2021……

* delete (iii) if not applicable

Signature ___Yue Zhang _____Date:_____*05/05/2021*_____ Signature _____Date:_____

Signature ___Lei_Hu_____Date:_____*05/05/2021*_____ Signature _____Date:_____

Signature _____Date:_____ Signature _____Date:_____

Privacy Statement

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

MONASH University
Information Technology

Contribution Declaration Form
(to be completed by all team members)

Please fill in the form with the contribution from each student towards the assignment.

1 NAME AND CONTRIBUTION DETAILS

Student ID	Student Name	Contribution Percentage
30841933	Lei Hu	50%
30976316	Yue Zhang	50%

List of parts that each student did:
1. Lei Hu: Data clean 50%, Star schema 50%, Implementation 50%, Report Writing 50%
2. Yue Zhang: Data clean 50%, Star schema 50%, Implementation 50%, Report Writing 50%

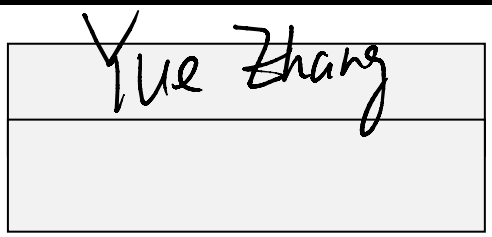
2 DECLARATION

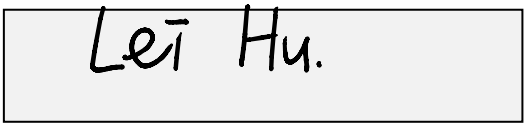
We declare that:

- The information we have supplied in or with this form is complete and correct.
- We understand that the information we have provided in this form will be used for individual assessment of the assignment.

3 SIGNATURE

Signatures





Day Month Year

Date 05 / 05 / 2021

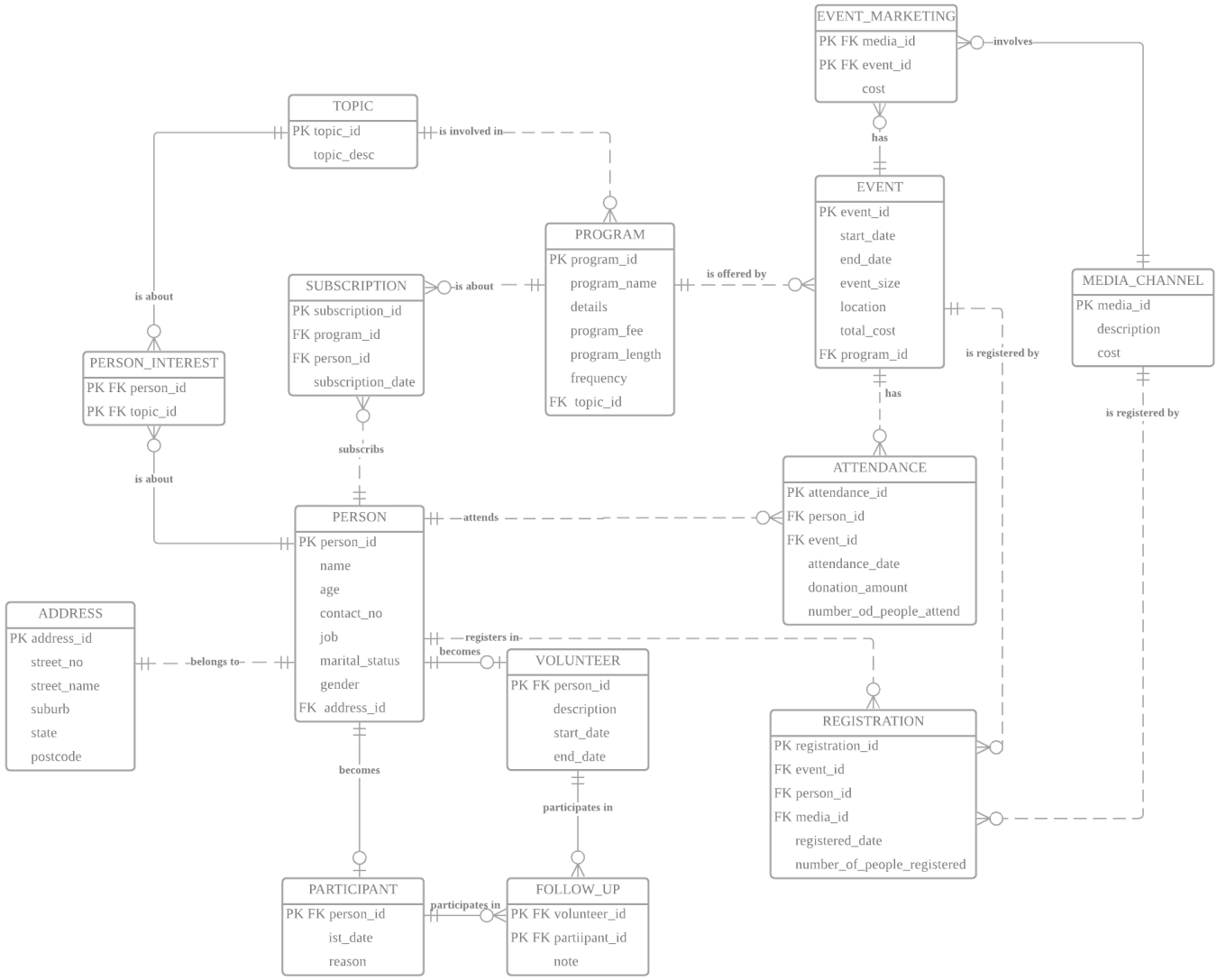
ORACLE accounts :
Yue Zhang :
Username : S30976316 Password : student
Lei Hu :
Username : S30841933 Password : student

MONASH University
Information Technology

Table of Contents

Preparation stage.

a) The E/R diagram of the operational database



b) Data cleaning process

Before cleaning, we copy all tables from the Monexplore and save them in our own account to explore and clean, the code for copy is as below:

```
CREATE TABLE TOPIC AS SELECT * FROM MONEXPLORE.TOPIC;

CREATE TABLE PROGRAM AS SELECT * FROM MONEXPLORE.PROGRAM;

CREATE TABLE EVENT AS SELECT * FROM MONEXPLORE.EVENT;

CREATE TABLE MEDIA_CHANNEL AS SELECT * FROM MONEXPLORE.MEDIA_CHANNEL;

CREATE TABLE PERSON AS SELECT * FROM MONEXPLORE.PERSON;

CREATE TABLE ADDRESS AS SELECT * FROM MONEXPLORE.ADDRESS;

CREATE TABLE PERSON_INTEREST AS SELECT * FROM MONEXPLORE.PERSON_INTEREST;

CREATE TABLE SUBSCRIPTION AS SELECT * FROM MONEXPLORE.SUBSCRIPTION;
```

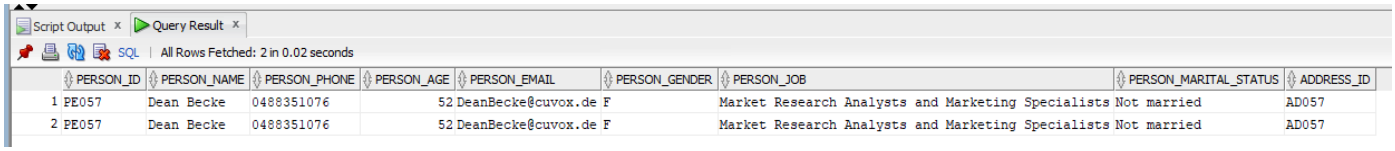
```
CREATE TABLE ATTENDANCE AS SELECT * FROM MONEXPLORE.ATTENDANCE;

CREATE TABLE REGISTRATION AS SELECT * FROM MONEXPLORE.REGISTRATION;

CREATE TABLE EVENT_MARKETING AS SELECT * FROM MONEXPLORE.EVENT_MARKETING;

CREATE TABLE VOLUNTEER AS SELECT * FROM MONEXPLORE.VOLUNTEER;

CREATE TABLE PARTICIPANT AS SELECT * FROM MONEXPLORE.PARTICIPANT;

CREATE TABLE FOLLOW_UP AS SELECT * FROM MONEXPLORE.FOLLOW_UP;
```

- Duplication:

Error 1: duplicate results in Person table: person id ‘PE057’, ‘PE078’, ‘PE021’ has duplicate records.

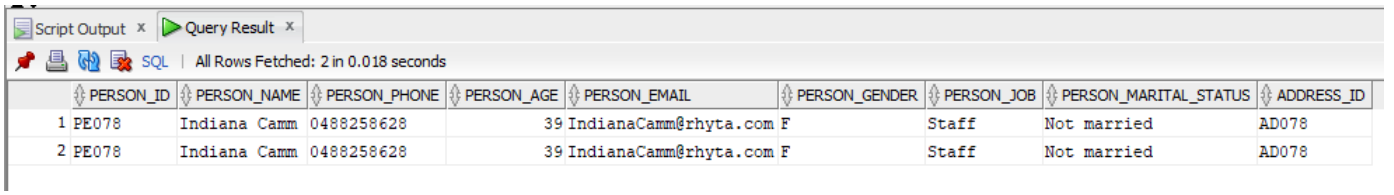
- Code to identify it:

```
SELECT PERSON_ID, COUNT(*)

FROM PERSON

GROUP BY PERSON_ID

HAVING COUNT(*) > 1; -- ID PE057, PE078, PE021

SELECT * FROM PERSON

WHERE PERSON_ID = 'PE057'; -- DUPILICATE
```



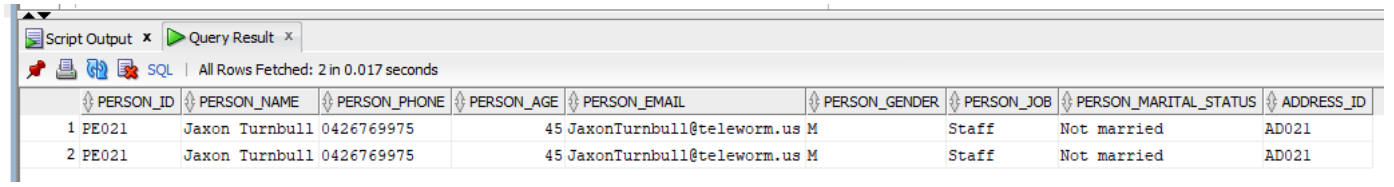
	PERSON_ID	PERSON_NAME	PERSON_PHONE	PERSON_AGE	PERSON_EMAIL	PERSON_GENDER	PERSON_JOB	PERSON_MARITAL_STATUS	ADDRESS_ID
1	PE057	Dean Becke	0488351076	52	DeanBecke@cuvox.de	F	Market Research Analysts and Marketing Specialists	Not married	AD057
2	PE057	Dean Becke	0488351076	52	DeanBecke@cuvox.de	F	Market Research Analysts and Marketing Specialists	Not married	AD057

```
SELECT * FROM PERSON

WHERE PERSON_ID = 'PE078'; -- DUPILICATE
```



	PERSON_ID	PERSON_NAME	PERSON_PHONE	PERSON_AGE	PERSON_EMAIL	PERSON_GENDER	PERSON_JOB	PERSON_MARITAL_STATUS	ADDRESS_ID
1	PE078	Indiana Camm	0488258628	39	IndianaCamm@rhyta.com	F	Staff	Not married	AD078
2	PE078	Indiana Camm	0488258628	39	IndianaCamm@rhyta.com	F	Staff	Not married	AD078

```
SELECT * FROM PERSON

WHERE PERSON_ID = 'PE021'; -- DUPILICATE
```

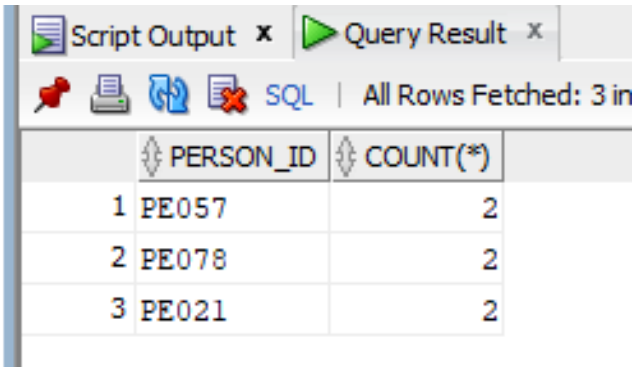


	PERSON_ID	PERSON_NAME	PERSON_PHONE	PERSON_AGE	PERSON_EMAIL	PERSON_GENDER	PERSON_JOB	PERSON_MARITAL_STATUS	ADDRESS_ID
1	PE021	Jaxon Turnbull	0426769975	45	JaxonTurnbull@teleworm.us	M	Staff	Not married	AD021
2	PE021	Jaxon Turnbull	0426769975	45	JaxonTurnbull@teleworm.us	M	Staff	Not married	AD021

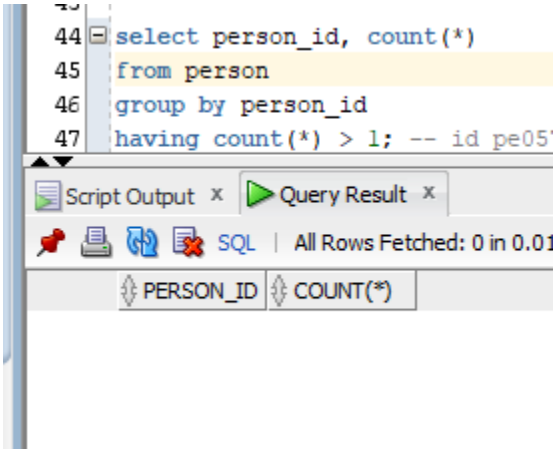
- Code to clean it:

```
--CLEAN

DROP TABLE PERSON;

CREATE TABLE PERSON AS SELECT DISTINCT * FROM MONEXPLORE.PERSON;
```

Before deleting: each person id has 2 same records.



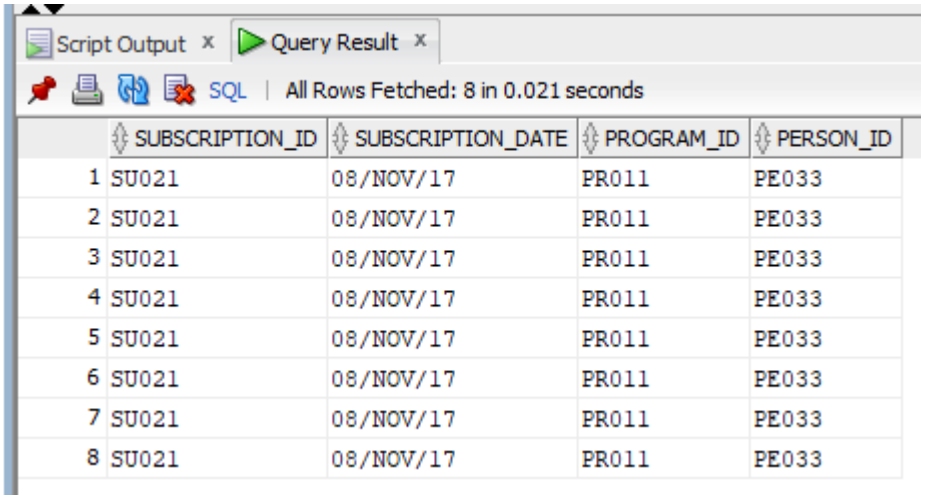
After cleaning: No person id with records more than 1.



Error 2: duplicate results in Subscription table: subscription id ‘SU021’, ‘SU243’ has duplicate records.

- Code to identify it:

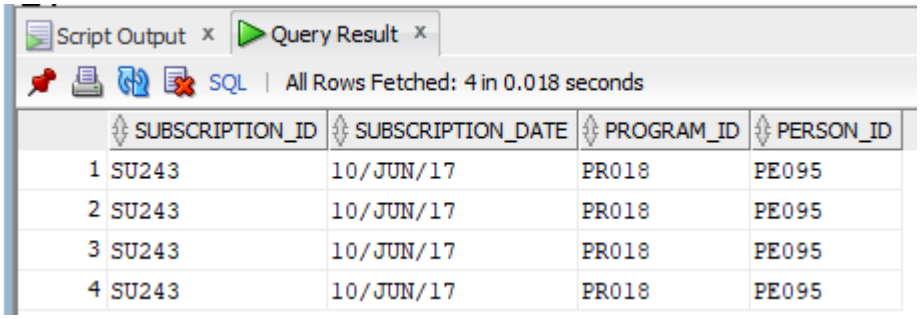
```
SELECT SUBSCRIPTION_ID, COUNT(*)

FROM SUBSCRIPTION

GROUP BY SUBSCRIPTION_ID

HAVING COUNT(*) > 1; --ID SU021, SU243

--CHECK FOR EACH RECORD

SELECT * FROM SUBSCRIPTION

WHERE SUBSCRIPTION_ID = 'SU021'; -- DUPILICATE
```



Script Output | Query Result | SQL | All Rows Fetched: 8 in 0.021 seconds

	SUBSCRIPTION_ID	SUBSCRIPTION_DATE	PROGRAM_ID	PERSON_ID
1	SU021	08/NOV/17	PR011	PE033
2	SU021	08/NOV/17	PR011	PE033
3	SU021	08/NOV/17	PR011	PE033
4	SU021	08/NOV/17	PR011	PE033
5	SU021	08/NOV/17	PR011	PE033
6	SU021	08/NOV/17	PR011	PE033
7	SU021	08/NOV/17	PR011	PE033
8	SU021	08/NOV/17	PR011	PE033

```
SELECT * FROM SUBSCRIPTION

WHERE SUBSCRIPTION_ID = 'SU243'; -- DUPILICATE
```



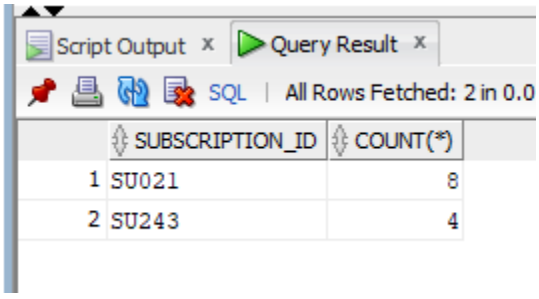
Script Output | Query Result | SQL | All Rows Fetched: 4 in 0.018 seconds

	SUBSCRIPTION_ID	SUBSCRIPTION_DATE	PROGRAM_ID	PERSON_ID
1	SU243	10/JUN/17	PR018	PE095
2	SU243	10/JUN/17	PR018	PE095
3	SU243	10/JUN/17	PR018	PE095
4	SU243	10/JUN/17	PR018	PE095

- Code to clean it:

```
--CLEAN

DROP TABLE SUBSCRIPTION;

CREATE TABLE SUBSCRIPTION AS SELECT DISTINCT * FROM MONEXPLORE.SUBSCRIPTION;
```

Before deleting: each subscription has more than 1 same records.



Script Output | Query Result | SQL | All Rows Fetched: 2 in 0.0

	SUBSCRIPTION_ID	COUNT(*)
1	SU021	8
2	SU243	4

After cleaning: No subscription id with records more than 1.

```
68
69  select subscription_id, count(*)
70  from subscription
71  group by subscription_id
72  having count(*) > 1; --id su021, su2
```

Script Output | Query Result | SQL | All Rows Fetched: 0 in 0.019 secc

SUBSCRIP...	COUNT(*)

- Null values:

Error 3: PK null error, one record that Pk(media_id) is null in media table.

- Code to identify it:

```
SELECT * FROM MEDIA_CHANNEL WHERE MEDIA_ID IS NULL;
```

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0.017 seconds

	MEDIA_ID	MEDIA_DESCRIPTION	MEDIA_COST
1	(null)	Unknown	0

- Code to clean it:

```
--CLEAN

DELETE FROM MEDIA_CHANNEL WHERE MEDIA_ID IS NULL;
```

Before deleting: One null record.

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0.017 seconds

	MEDIA_ID	MEDIA_DESCRIPTION	MEDIA_COST
1	(null)	Unknown	0

After cleaning: No null pk record.

```
1 row deleted.
```

```
102  select * from event where event_id is null;
103  select * from media_channel where media_id is null;
```

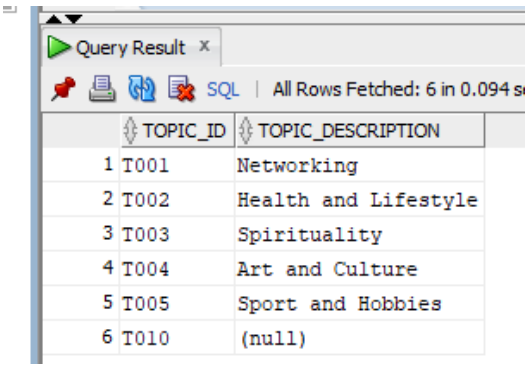
Script Output | Query Result | SQL | All Rows Fetched: 0 in 0.015 seconds

MEDIA_ID	MEDIA_D...	MEDIA_C...

Error 4: other attributes null error, in topic table, there topic T010 has null topic_description, since this record only has two column: id and description, so if description is missing, it will influence the business analysis when we create data warehouse, because we need description to help us know what is topic T010 and help further decision making.

- Code to identify it:

```
SELECT * FROM TOPIC; -- ONE NULL IN DESCRIPTION
```

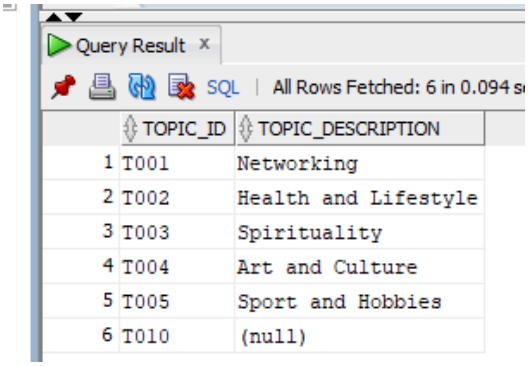


	TOPIC_ID	TOPIC_DESCRIPTION
1	T001	Networking
2	T002	Health and Lifestyle
3	T003	Spirituality
4	T004	Art and Culture
5	T005	Sport and Hobbies
6	T010	(null)

- Code to clean it:

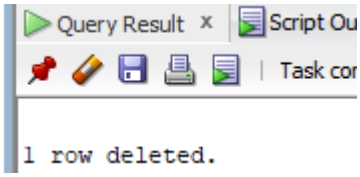
```
DELETE FROM TOPIC WHERE TOPIC_DESCRIPTION IS NULL;
```

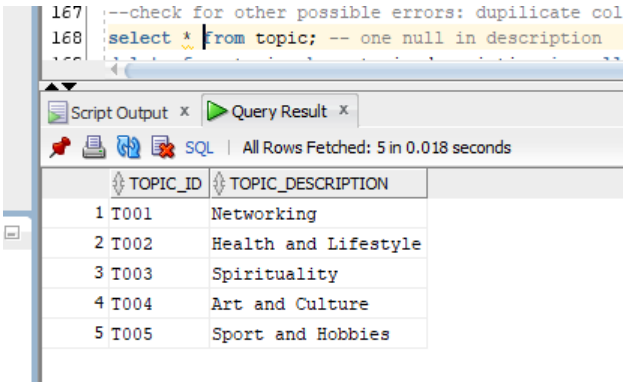
Before deleting: One null record.



	TOPIC_ID	TOPIC_DESCRIPTION
1	T001	Networking
2	T002	Health and Lifestyle
3	T003	Spirituality
4	T004	Art and Culture
5	T005	Sport and Hobbies
6	T010	(null)

After cleaning: No null record.





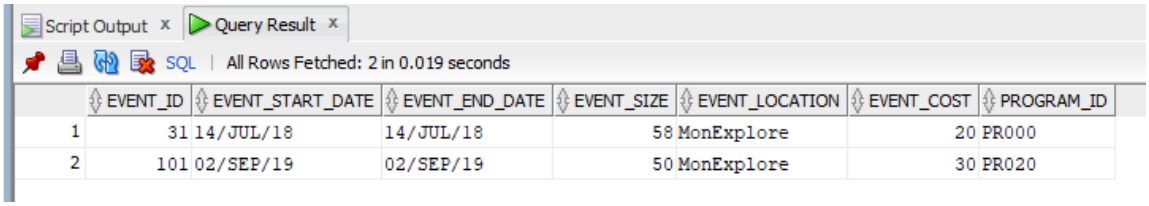
	TOPIC_ID	TOPIC_DESCRIPTION
1	T001	Networking
2	T002	Health and Lifestyle
3	T003	Spirituality
4	T004	Art and Culture
5	T005	Sport and Hobbies

- Relationship:

Error 5: In event table, there are two program_id (PR000, PR020) that are not in program table. However, one event must belongs to one program, so we can not just set fk to be null, so we need to delete those invalid fk records.

- Code to identify it:

```
SELECT * FROM EVENT

WHERE PROGRAM_ID NOT IN (SELECT PROGRAM_ID FROM PROGRAM);--PROGRAM ID PR000, PR020
```

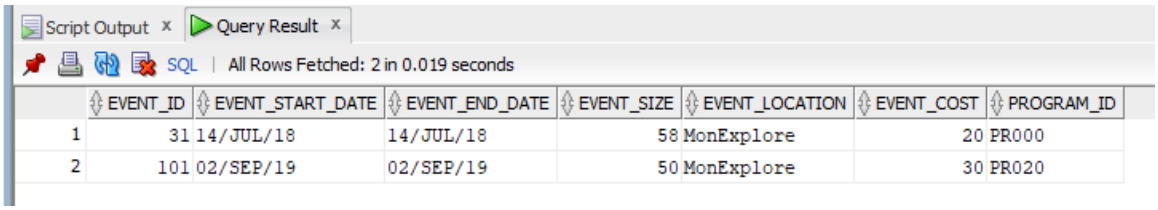


	EVENT_ID	EVENT_START_DATE	EVENT_END_DATE	EVENT_SIZE	EVENT_LOCATION	EVENT_COST	PROGRAM_ID
1	31	14/JUL/18	14/JUL/18	58	MonExplore	20	PR000
2	101	02/SEP/19	02/SEP/19	50	MonExplore	30	PR020

- Code to clean it:

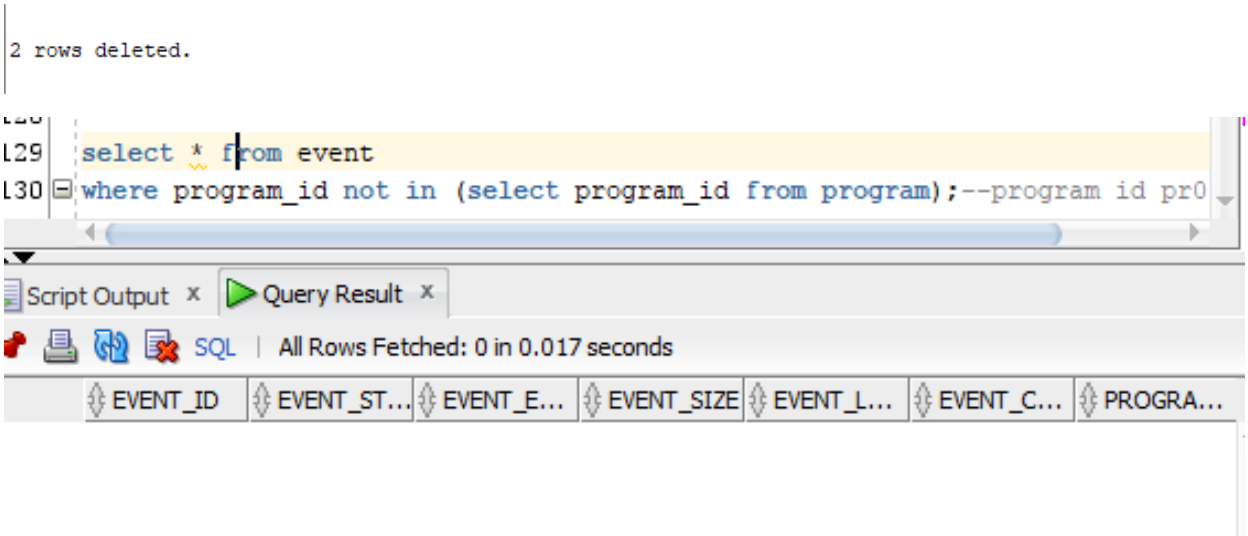
```
DELETE FROM EVENT WHERE PROGRAM_ID IN ('PR000', 'PR020');
```

Before deleting: PR000, PR020 is not as a program_id in program table but it is in event table.



	EVENT_ID	EVENT_START_DATE	EVENT_END_DATE	EVENT_SIZE	EVENT_LOCATION	EVENT_COST	PROGRAM_ID
1	31	14/JUL/18	14/JUL/18	58	MonExplore	20	PR000
2	101	02/SEP/19	02/SEP/19	50	MonExplore	30	PR020

After cleaning: No invalid foreign key.



Error 6: In attendance table, there are two event_id (31, 101) that are not in event table, it is because that we delete in error 5 because of the fk error in event, so 31 and 101 are not in event table now. However, one attendance must belongs to one event, so we can not just set fk to be null. Also, we do not know the event 31 and 101 information, so we can not manually add event 31 and 101 to event table and we need to delete those invalid fk records.

- Code to identify it:

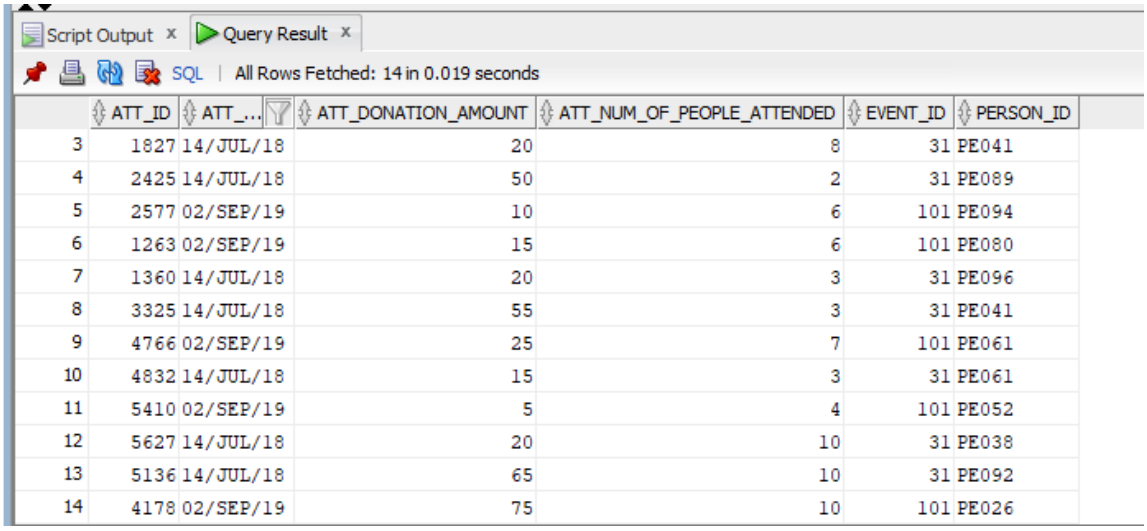
```
SELECT * FROM ATTENDANCE

WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON)

OR EVENT_ID NOT IN (SELECT EVENT_ID FROM EVENT);--EVENT ID 101,31
```

- Code to clean it:

```
DELETE FROM ATTENDANCE
```

```
WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON)

OR EVENT_ID NOT IN (SELECT EVENT_ID FROM EVENT);
```

Before deleting: 31, 101 is not as a event_id in event table but it is in attendance table.

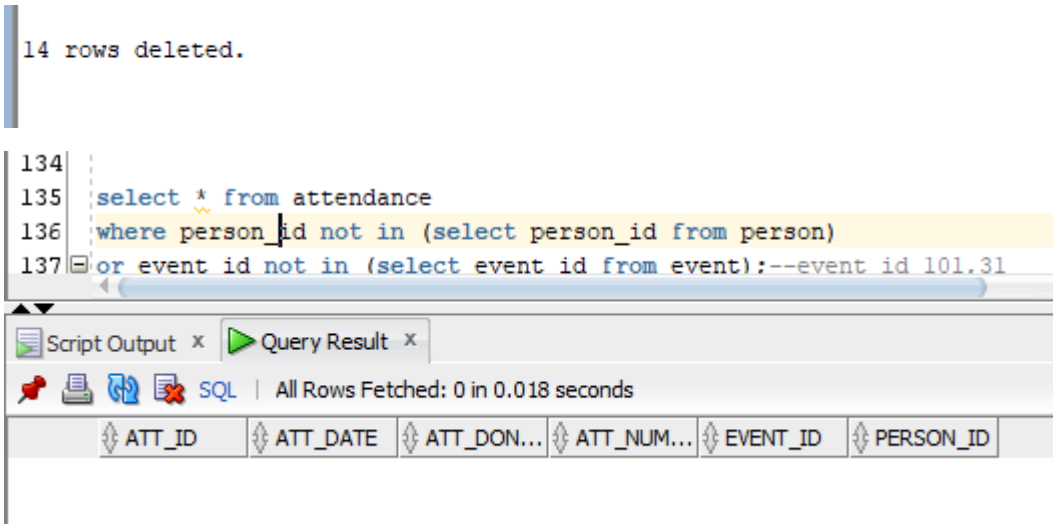


Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.019 seconds

	ATT_ID	ATT_...	ATT_DONATION_AMOUNT	ATT_NUM_OF_PEOPLE_ATTENDED	EVENT_ID	PERSON_ID
3	1827	14/JUL/18	20	8	31	PE041
4	2425	14/JUL/18	50	2	31	PE089
5	2577	02/SEP/19	10	6	101	PE094
6	1263	02/SEP/19	15	6	101	PE080
7	1360	14/JUL/18	20	3	31	PE096
8	3325	14/JUL/18	55	3	31	PE041
9	4766	02/SEP/19	25	7	101	PE061
10	4832	14/JUL/18	15	3	31	PE061
11	5410	02/SEP/19	5	4	101	PE052
12	5627	14/JUL/18	20	10	31	PE038
13	5136	14/JUL/18	65	10	31	PE092
14	4178	02/SEP/19	75	10	101	PE026

After cleaning: No invalid foreign key.



```
14 rows deleted.
```

```
134
135  select * from attendance
136  where person_id not in (select person_id from person)
137  or event_id not in (select event_id from event);--event id 101,31
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.018 seconds

ATT_ID	ATT_DATE	ATT_DON...	ATT_NUM...	EVENT_ID	PERSON_ID

Error 7: In registration table, there are two event_id (31, 101) that are not in event table, it is because that we delete in error 5 because of the fk error in event, so 31 and 101 are not in event table now.. However, one registration must belongs to one event, so we can not just set fk to be null. Also, we do not know the event 31 and 101 information, so we can not manually add event 31 and 101 to event table and we need to delete those invalid fk records.

- Code to identify it:

```
SELECT * FROM REGISTRATION

WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON)

OR EVENT_ID NOT IN (SELECT EVENT_ID FROM EVENT)

OR MEDIA_ID NOT IN (SELECT MEDIA_ID FROM MEDIA_CHANNEL);--EVENT ID 101,31
```

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.023 seconds

	REG_ID	REG_NUM_OF_PEOPLE_REGISTERED	REG_DATE	EVENT_ID	PERSON_ID	MEDIA_ID
1	650	4	07/JUL/18	31	PE089	MC002
2	691	3	26/AUG/19	101	PE094	MC001
3	893	4	07/JUL/18	31	PE041	MC004
4	228	2	26/AUG/19	101	PE095	MC003
5	311	4	26/AUG/19	101	PE080	MC004
6	340	3	07/JUL/18	31	PE096	MC003
7	455	2	26/AUG/19	101	PE061	MC005
8	469	3	07/JUL/18	31	PE041	MC001
9	1087	2	26/AUG/19	101	PE026	MC003
10	1241	3	26/AUG/19	101	PE061	MC001
11	1274	1	07/JUL/18	31	PE061	MC001
12	1356	4	07/JUL/18	31	PE092	MC004
13	1430	3	26/AUG/19	101	PE052	MC001
14	1494	4	07/JUL/18	31	PE038	MC004

- Code to clean it:

```
DELETE FROM REGISTRATION
WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON)
OR EVENT_ID NOT IN (SELECT EVENT_ID FROM EVENT)
OR MEDIA_ID NOT IN (SELECT MEDIA_ID FROM MEDIA_CHANNEL);
```

Before deleting: 31, 101 is not as a event_id in event table but it is in attendance table.

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.023 seconds

	REG_ID	REG_NUM_OF_PEOPLE_REGISTERED	REG_DATE	EVENT_ID	PERSON_ID	MEDIA_ID
1	650	4	07/JUL/18	31	PE089	MC002
2	691	3	26/AUG/19	101	PE094	MC001
3	893	4	07/JUL/18	31	PE041	MC004
4	228	2	26/AUG/19	101	PE095	MC003
5	311	4	26/AUG/19	101	PE080	MC004
6	340	3	07/JUL/18	31	PE096	MC003
7	455	2	26/AUG/19	101	PE061	MC005
8	469	3	07/JUL/18	31	PE041	MC001
9	1087	2	26/AUG/19	101	PE026	MC003
10	1241	3	26/AUG/19	101	PE061	MC001
11	1274	1	07/JUL/18	31	PE061	MC001
12	1356	4	07/JUL/18	31	PE092	MC004
13	1430	3	26/AUG/19	101	PE052	MC001
14	1494	4	07/JUL/18	31	PE038	MC004

After cleaning: No invalid foreign key.

```
14 rows deleted.
```

```
select * from registration
where person_id not in (select person_id from person)
or event_id not in (select event_id from event)
or media_id not in (select media_id from media_channel);--event ic
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.018 seconds

REG_ID	REG_NUM...	REG_DATE	EVENT_ID	PERSON_ID	MEDIA_ID

Error 8: In volunteer table, there are two person_id (PE000, PE110) that are not in person table. However, one volunteer must belongs to one person, so we can not just set fk to be null, so we need to delete those invalid fk records.

- Code to identify it:

```
SELECT * FROM VOLUNTEER  --- *** ERROR ***

WHERE PERSON_ID NOT IN(SELECT PERSON_ID FROM PERSON);
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.031 seconds

	PERSON_ID	VOL_DESCRIPTION	VOL_START_DATE	VOL_END_DATE
1	PE000	Part time	25/OCT/19	25/OCT/19
2	PE110	Part time	16/MAY/20	16/MAY/19

- Code to clean it:

```
DELETE FROM VOLUNTEER

WHERE PERSON_ID NOT IN(SELECT PERSON_ID FROM PERSON);
```

Before deleting: PE000, PE110 is not as a person_id in person table but it is in volunteer table.

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.031 seconds

	PERSON_ID	VOL_DESCRIPTION	VOL_START_DATE	VOL_END_DATE
1	PE000	Part time	25/OCT/19	25/OCT/19
2	PE110	Part time	16/MAY/20	16/MAY/19

After cleaning: No invalid foreign key.

```
2 rows deleted.
```

```
273 select * from volunteer  --- *** ERROR ***
274 where person_id not in(select person_id from person);
275 --clean
```

Script Output | Query Result | SQL | All Rows Fetched: 0 in 0.016 seconds

PERSON_ID	VOL_DES...	VOL_STA...	VOL_END...

Error 10: In person_interest table, the topic_id (T010) that are not in topic table, which is because that we delete topic T010 in the previous stage and regard this topic as an error, and one person_interest must belongs to one topic, so we can not just set fk to be null, so we need to delete those invalid fk records.

- Code to identify it:

```
SELECT * FROM PERSON_INTEREST
WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON)
OR TOPIC_ID NOT IN (SELECT TOPIC_ID FROM TOPIC);--T010
```

Script Output | Query Result | SQL | All Rows Fetched:

	TOPIC_ID	PERSON_ID
1	T010	PE035
2	T010	PE051

- Code to clean it:

```
DELETE FROM PERSON_INTEREST
WHERE PERSON_ID NOT IN (SELECT PERSON_ID FROM PERSON)
OR TOPIC_ID NOT IN (SELECT TOPIC_ID FROM TOPIC);--T010
```

Before deleting: T010 is not as a topic_id in topic table but it is in person_interest table.

Script Output | Query Result | SQL | All Rows Fetched:

	TOPIC_ID	PERSON_ID
1	T010	PE035
2	T010	PE051

After cleaning: No invalid foreign key.

```
2 rows deleted.
```

```
139
140 select * from person_interest
141 where person_id not in (select person_id from person)
142 or topic_id not in (select topic_id from topic);--T010
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.018 seconds

TOPIC_ID	PERSON_ID

Error 10: In event_marketing table, the event_id (101) that are not in event table, it is because that we delete in error 5 because of the fk error in event, so 101 is not in event table now.. However, one event_marketing must belongs to one event, so we can not just set fk to be null, so we need to delete those invalid fk records.

- Code to identify it:

```
SELECT * FROM EVENT_MARKETING  --- *** ERROR ***

WHERE EVENT_ID NOT IN(SELECT EVENT_ID FROM EVENT);

SELECT EVENT_ID FROM EVENT;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.01

	MEDIA_ID	EVENT_ID	EM_COST
1	MC004	101	100

- Code to clean it:

```
DELETE FROM EVENT_MARKETING

WHERE EVENT_ID NOT IN(SELECT EVENT_ID FROM EVENT);
```

Before deleting: 101 is not as a event_id in event table but it is in event_marketing table.

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.01

	MEDIA_ID	EVENT_ID	EM_COST
1	MC004	101	100

After cleaning: No invalid foreign key.

```
1 row deleted.
```

```
304  --- check the event_id from the Event_Marketing table
305  select * from event_marketing  --- *** ERROR ***
306  where event_id not in(select event_id from event);
307  select event_id from event;
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.019 seconds

MEDIA_ID	EVENT_ID	EM_COST

- Incorrect values:

Error 11: In attendance table, there is one record that has att_date in 18/OCT/2021, which is in the future, but attendance only records the person and other related information that has already attended the event, so it cannot happen in the future, so it is an error. Also, we need to analyse this table in time dimension, so it influence our data warehouse, so we need to delete because we do not know which is the correct date.

- Code to identify it:

```
SELECT * FROM ATTENDANCE

WHERE TO_DATE(ATT_DATE, 'DD-MON-RR') > CURRENT_DATE;  ---**** ERROR **** ONE AT 18/OCT/21
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.025 seconds

	ATT_ID	ATT_DATE	ATT_DONATION_AMOUNT	ATT_NUM_OF_PEOPLE_ATTENDED	EVENT_ID	PERSON_ID
1	802	18/OCT/21	95	1	71	PE043

- Code to clean it:

```
DELETE FROM ATTENDANCE

WHERE TO_DATE(ATT_DATE, 'DD-MON-RR') > CURRENT_DATE;
```

Before deleting: One record has incorrect att_date.

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.025 seconds

	ATT_ID	ATT_DATE	ATT_DONATION_AMOUNT	ATT_NUM_OF_PEOPLE_ATTENDED	EVENT_ID	PERSON_ID
1	802	18/OCT/21	95	1	71	PE043

After cleaning: No incorrect date in this table.

```
1 row deleted.
```

```
210  --- 2.1 check the date from the Attendance
211  select * from attendance
212  where to_date(att_date, 'DD-MON-RR') > current_date;  ---**** ERRO
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.027 seconds

ATT_ID	ATT_DATE	ATT_DON...	ATT_NUM...	EVENT_ID	PERSON_ID

Error 12: In attendance table, there are two records (att_id 639, 1001) that have att_donation_amount that is negative number, but for att_donation_amount, it should not be negative, so it is an error. Also, we need to analyse this table using donation amount as a fact measure, so it influences our data warehouse, so we need to delete because we do not know which is the correct date.

- Code to identify it:

```
SELECT * FROM ATTENDANCE  ---**** ERROR **** ATTID 639:-25, ATTID1001:-5
WHERE ATT_DONATION_AMOUNT < 0;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.018 seconds

	ATT_ID	ATT_DATE	ATT_DONATION_AMOUNT	ATT_NUM_OF_PEOPLE_ATTENDED	EVENT_ID	PERSON_I
1	639	12/NOV/20	-25	4	159	PE006
2	1001	28/MAY/19	-5	9	72	PE031

- Code to clean it:

```
DELETE FROM ATTENDANCE
WHERE ATT_DONATION_AMOUNT < 0;
```

Before deleting: Two records have negative att_donation_amount.

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.018 seconds

	ATT_ID	ATT_DATE	ATT_DONATION_AMOUNT	ATT_NUM_OF_PEOPLE_ATTENDED	EVENT_ID	PERSON_I
1	639	12/NOV/20	-25	4	159	PE006
2	1001	28/MAY/19	-5	9	72	PE031

After cleaning: No incorrect date in this table.

```
2 rows deleted.
```

```
218  --- 2.2 check amount from Attendance table
219  select * from attendance  ---**** ERROR **** attid 639:-25, attid1001:-5
220  where att_donation_amount < 0;
221  --- modify ......
222  delete from attendance
223  where att_donation_amount < 0;
224
225  --- 2.3 check number of people attended
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.019 seconds

ATT_ID	ATT_DATE	ATT_DON...	ATT_NUM...	EVENT_ID	PERSON_ID

Error 13: In event table, there are two records (event_id 11, 47) that has event_size is negative number, but for event_size, it should not be negative, so it is an error. Also, we need to analyse this table using event size dimension, so it influences our data warehouse, so we need to delete because we do not know which is the correct date. Also, we should delete the records in other table (event_marketing, attendance, registration) that has event_id 11 and 47 to keep relationship correct.

- Code to identify it:

```
SELECT * FROM EVENT  ---- **** ERROR **** EVENT ID 11:-10, EVENT ID 47: -75
WHERE EVENT_SIZE < 0;
```

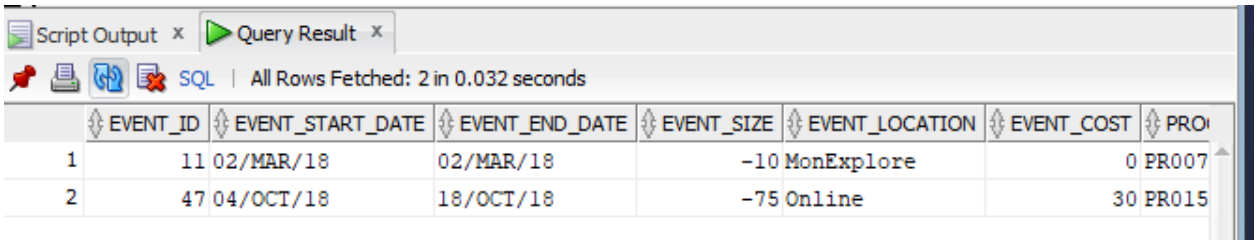
SQL | All Rows Fetched: 2 in 0.032 seconds

	EVENT_ID	EVENT_START_DATE	EVENT_END_DATE	EVENT_SIZE	EVENT_LOCATION	EVENT_COST	PRO
1	11	02/MAR/18	02/MAR/18	-10	MonExplore	0	PR007
2	47	04/OCT/18	18/OCT/18	-75	Online	30	PR015

- Code to clean it:

```
DELETE FROM EVENT_MARKETING WHERE
EVENT_ID IN (SELECT EVENT_ID FROM EVENT
WHERE EVENT_SIZE < 0);
DELETE FROM ATTENDANCE WHERE
EVENT_ID IN (SELECT EVENT_ID FROM EVENT
WHERE EVENT_SIZE < 0);
DELETE FROM REGISTRATION WHERE
EVENT_ID IN (SELECT EVENT_ID FROM EVENT
WHERE EVENT_SIZE < 0);
DELETE FROM  EVENT  ---- **** ERROR **** EVENT ID 11:-10, EVENT ID 47: -75
WHERE EVENT_SIZE < 0;
```

Before deleting: Two records have incorrect event_size.

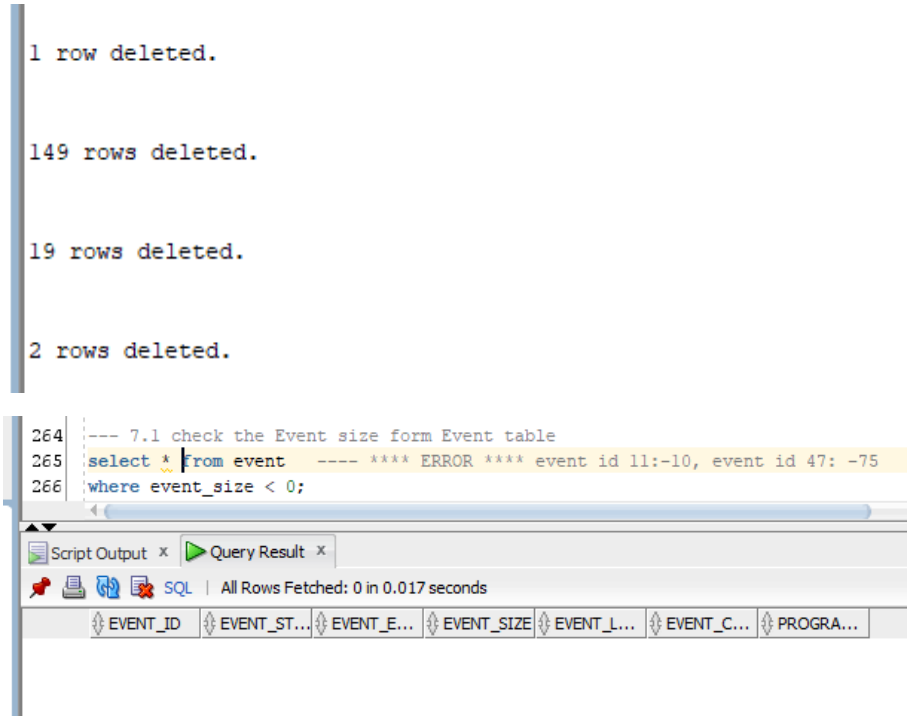


Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.032 seconds

	EVENT_ID	EVENT_START_DATE	EVENT_END_DATE	EVENT_SIZE	EVENT_LOCATION	EVENT_COST	PRO
1	11	02/MAR/18	02/MAR/18	-10	MonExplore	0	PR007
2	47	04/OCT/18	18/OCT/18	-75	Online	30	PR015

After cleaning: No incorrect date in this table and other table.



```
1 row deleted.


149 rows deleted.


19 rows deleted.


2 rows deleted.
```

```
264  --- 7.1 check the Event size form Event table
265  select * from event   ---- **** ERROR **** event id 11:-10, event id 47: -75
266  where event_size < 0;
```

Script Output | Query Result

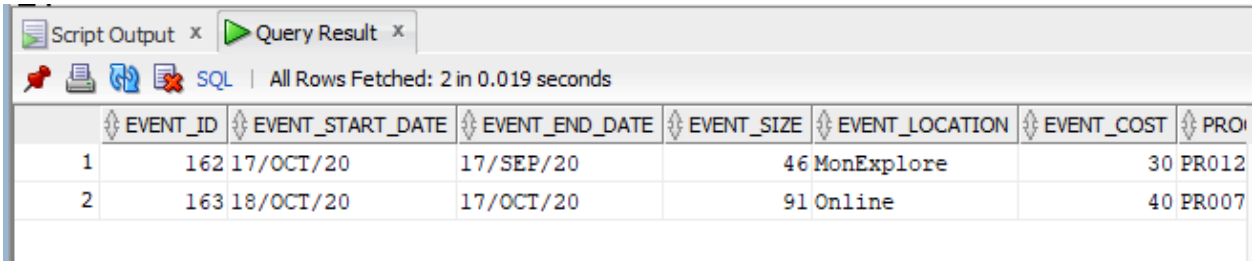
SQL | All Rows Fetched: 0 in 0.017 seconds

EVENT_ID	EVENT_ST...	EVENT_E...	EVENT_SIZE	EVENT_L...	EVENT_C...	PROGRA...

Error 14: In event table, there is one record that has event start date is after the end date, so it is an error. Also, we need to analyse this table in time dimension, so it influences our data warehouse, so we need to delete because we do not know which is the correct date. Also, we should delete the records in other table (event_marketing, attendance, registration) for this incorrect event by event id to keep relationship correct.

- Code to identify it:

```
SELECT * FROM EVENT   --- **** ERROR **** EVENT ID 162, 163

WHERE TO_DATE(EVENT_START_DATE, 'DD-MON-RR') > TO_DATE(EVENT_END_DATE, 'DD-MON-RR');
```



Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.019 seconds

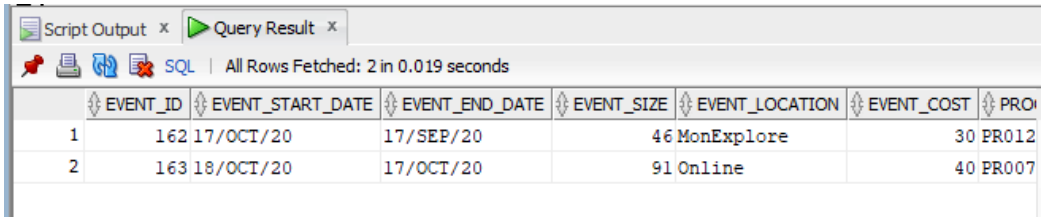
	EVENT_ID	EVENT_START_DATE	EVENT_END_DATE	EVENT_SIZE	EVENT_LOCATION	EVENT_COST	PRO
1	162	17/OCT/20	17/SEP/20	46	MonExplore	30	PR012
2	163	18/OCT/20	17/OCT/20	91	Online	40	PR007

- Code to clean it:

```
DELETE FROM EVENT_MARKETING WHERE
```

```
EVENT_ID IN (SELECT EVENT_ID FROM EVENT

WHERE TO_DATE(EVENT_START_DATE, 'DD-MON-RR') > TO_DATE(EVENT_END_DATE, 'DD-MON-RR'));

DELETE FROM ATTENDANCE WHERE

EVENT_ID IN (SELECT EVENT_ID FROM EVENT

WHERE TO_DATE(EVENT_START_DATE, 'DD-MON-RR') > TO_DATE(EVENT_END_DATE, 'DD-MON-RR'));

DELETE FROM REGISTRATION WHERE

EVENT_ID IN (SELECT EVENT_ID FROM EVENT

WHERE TO_DATE(EVENT_START_DATE, 'DD-MON-RR') > TO_DATE(EVENT_END_DATE, 'DD-MON-RR'));

DELETE FROM  EVENT   ---- **** ERROR **** EVENT ID 11:-10, EVENT ID 47: -75

WHERE TO_DATE(EVENT_START_DATE, 'DD-MON-RR') > TO_DATE(EVENT_END_DATE, 'DD-MON-RR');
```

Before deleting: One record has incorrect att_date.

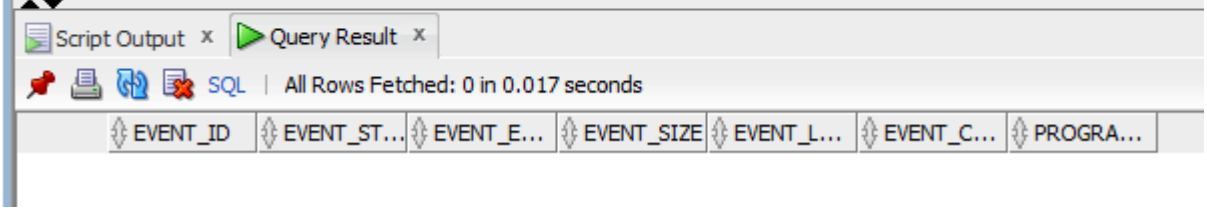


	EVENT_ID	EVENT_START_DATE	EVENT_END_DATE	EVENT_SIZE	EVENT_LOCATION	EVENT_COST	PRO
1	162	17/OCT/20	17/SEP/20	46	MonExplore	30	PR012
2	163	18/OCT/20	17/OCT/20	91	Online	40	PR007

After cleaning: No incorrect date in this table.

```
2 rows deleted.


15 rows deleted.


15 rows deleted.


2 rows deleted.
```

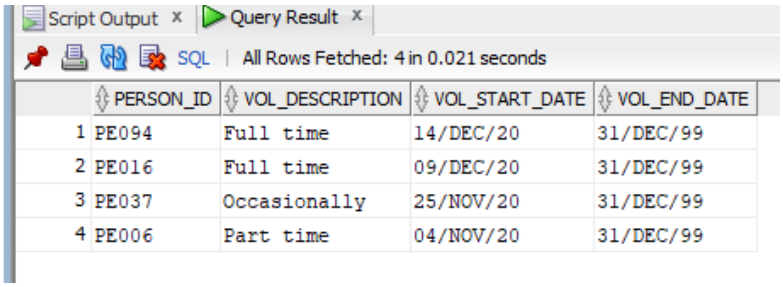
```
284  --- 7.3 check the date from Event table
285  select * from event    --- **** Error **** event id 162, 163
286  where TO_DATE(event_start_date, 'DD-MON-RR') > TO_DATE(event_end_date, 'DD-MON-
287  --clean
288  delete from  event    ---- **** ERROR **** event id 11:-10, event id 47: -75
```



Error 15: In volunteer table, there are four records that have start date is after the end date, so it is an error, so we need to delete because we do not know which is the correct date. Also, we should delete the records in other table (follow_up) for this incorrect volunteer to keep relationship correct.

- Code to identify it:

```
SELECT * FROM VOLUNTEER --- *** ERROR ***

WHERE TO_DATE(VOL_START_DATE,'DD-MON-RR') > TO_CHAR(VOL_END_DATE,'DD-MON-RR');
```


Script Output | Query Result

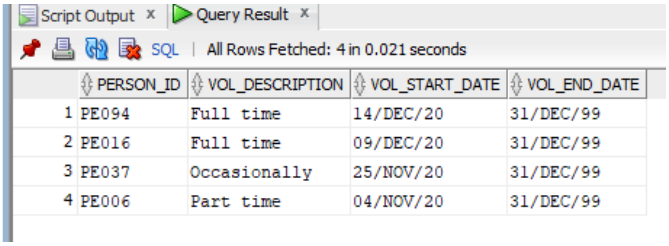
SQL | All Rows Fetched: 4 in 0.021 seconds

	PERSON_ID	VOL_DESCRIPTION	VOL_START_DATE	VOL_END_DATE
1	PE094	Full time	14/DEC/20	31/DEC/99
2	PE016	Full time	09/DEC/20	31/DEC/99
3	PE037	Occasionally	25/NOV/20	31/DEC/99
4	PE006	Part time	04/NOV/20	31/DEC/99

- Code to clean it:

```
DELETE FROM FOLLOW_UP

WHERE PERSON_ID IN (SELECT PERSON_ID FROM VOLUNTEER

WHERE TO_DATE(VOL_START_DATE,'DD-MON-RR') > TO_CHAR(VOL_END_DATE,'DD-MON-RR'));

DELETE FROM VOLUNTEER

WHERE TO_DATE(VOL_START_DATE,'DD-MON-RR') > TO_CHAR(VOL_END_DATE,'DD-MON-RR');
```

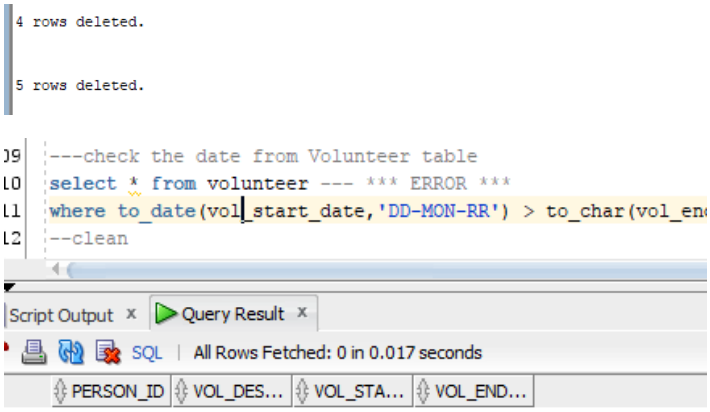
Before deleting: Four records has incorrect start date and end date.


Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.021 seconds

	PERSON_ID	VOL_DESCRIPTION	VOL_START_DATE	VOL_END_DATE
1	PE094	Full time	14/DEC/20	31/DEC/99
2	PE016	Full time	09/DEC/20	31/DEC/99
3	PE037	Occasionally	25/NOV/20	31/DEC/99
4	PE006	Part time	04/NOV/20	31/DEC/99

After cleaning: No incorrect date in this table.


```
4 rows deleted.


5 rows deleted.
```

```
09  ---check the date from Volunteer table
10  select * from volunteer --- *** ERROR ***
11  where to_date(vol_start_date,'DD-MON-RR') > to_char(vol_en
12  --clean
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.017 seconds

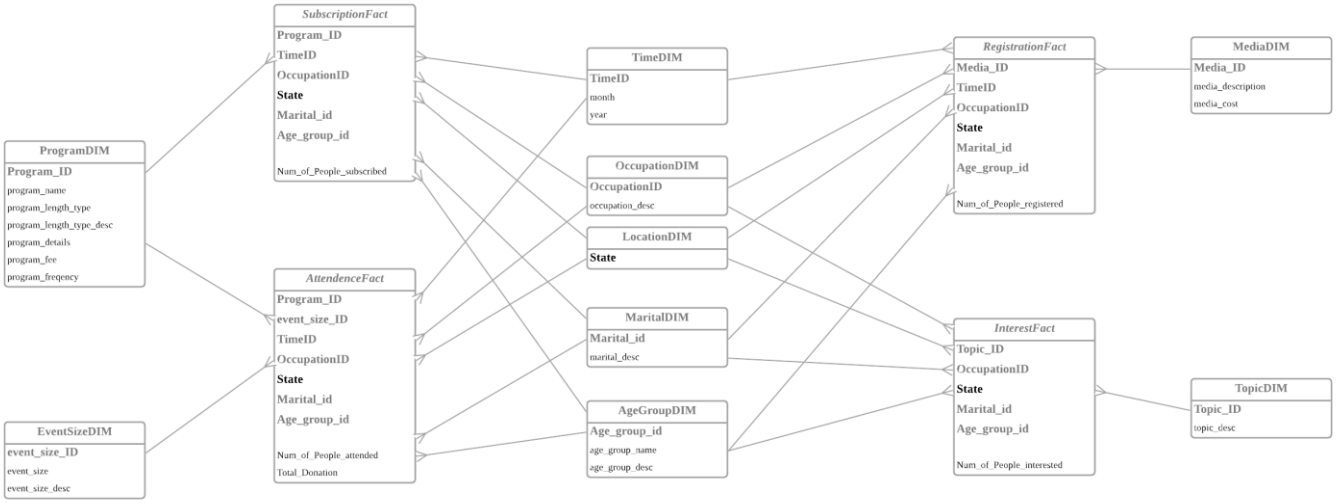
PERSON_ID	VOL_DES...	VOL_STA...	VOL_END...

MONASH University
Information Technology

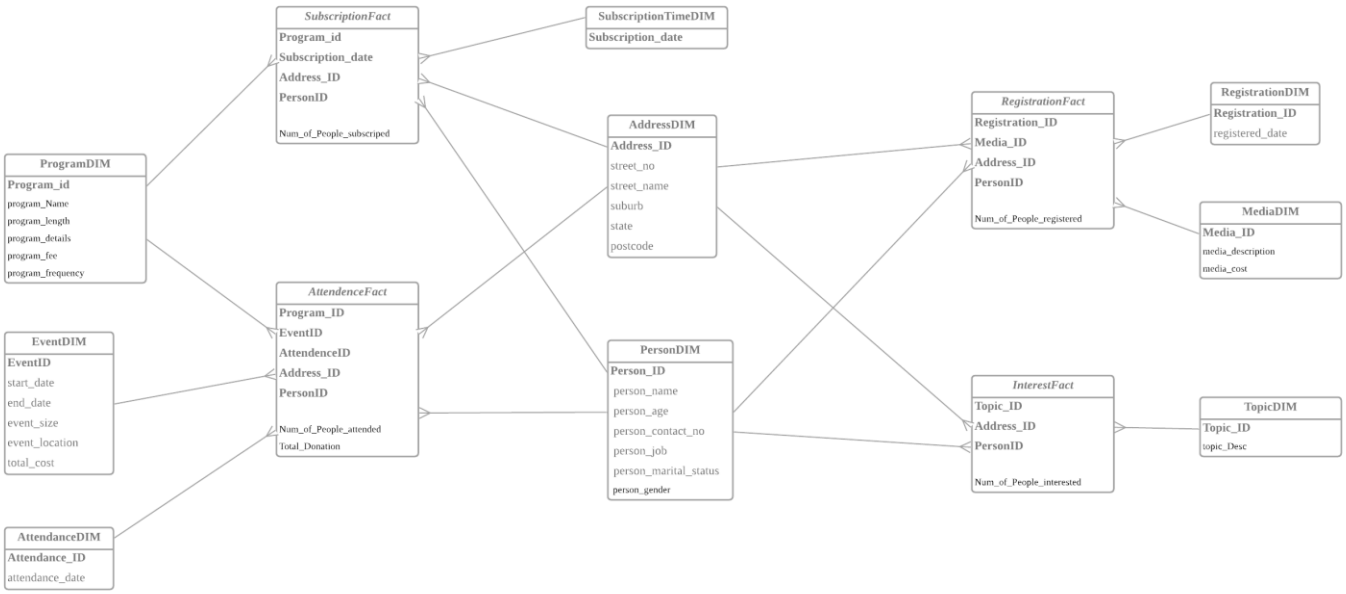
Designing the data warehouse by drawing star/snowflake schema.

c) Two versions of star/snowflake schema diagrams

Level 2:



Level 0:



d) The reasons of the choice of SCD type for temporal dimension

No SCD in the two star schemas.

e) A short explanation of the difference among the two versions of star/snowflake schema.

- Version 1- High Aggregation(Level 2)

In Version 1, we design four fact tables including five fact measures according to their shared dimensions. As you can see in output a) level 2, we put the Fact Measure Number of People Attended and Total Donation into one Fact Table, because they came from the same Table in the operation database. For other fact measures, we put them into separate fact tables, because they have unique and different dimensions compared to others. AS for the Dimension Table, the OccupationDim,

LocationDim, MaritalDim and AgeGroupDim are the common dimension for all fact measures, and they are come from the Person's demographic information in Person table, we separate them to keep a high level of aggregation for fact table. Also, in ProgramDIM, we break the original program length into program_length_type and program_length_type description to help us analyse the program length type (short, medium, long) for the fact, and rest of the dimensions connection can be shown as the figures.

- Version 2- No Aggregation(Level 0) & Differences Between Version 1 and Version 2

In Version 2, we keep the fact tables the same as the Version 1. Considering in Version 2 the information should be in detail, so we create the PersonDim and add the personal information like PersonID and Name. What's more, to simplify the dimension table, we also move the Occupation Dimension, Marital Dimension and AgeGroup Dimension into the PersonDim to lower down the level of aggregation. In addition, we using the Address instead of State to keep low aggregation.

For InterestFact, we use Address, Person and Topic as the dimension, which keeps no aggregation in this fact table.

For RegistrationFact, considering that one person in the same day can register to the same event many times because one event has different sessions, so we add a new dimension called RegistrationDIM to help keep no aggregation.

Similar to AttendanceFact, we add a new dimension called AttendanceDIM to help keep no aggregation, because one person can attend to the same event in the same day many times. In addition, the EventSize dimension is changed to Event dimension to lower down the aggregation.

For SubscriptionFact, we use the original Program to create ProgramDIM to keep the no aggregation, and we change the Time dimension to SubscriptionTime dimension which stores the subscription date instead of month and year in level 2.

It is important to note that we store date separately for subscriptionFact, RegistrationFact and AttendaneFact, this is because that the date information is different in different tables (in level 2, month and year is relatively more common through different fact tables), and it is better to keep them separately to reduce the amount of records that stored in time dimension, and the date for RegistrationFact and AttendaneFact can be stored in RegristraionDIM and AttendanceDIM respectively, and it is also easy to retrieve.

The version 1 is a high aggregation star schema, which can provide some general suggestions for decision makers. For example, we can easily check what is the most famous program according to different age groups.

However Version 2 is No aggregation star schema, which is similar to the operational database records but in different structure. With Version 2, the manager can also access the single records without going back to query in the operational database which can improve the efficiency.

Implement the two versions star/snowflake schema using SQL.

a) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema Version-1

---create the EventSizeDim

drop table eventsizedim;

```
create table eventsizeDim
(event_size_ID varchar2(2) not null,
event_size varchar2(20) not null,
event_size_desc varchar2(50) not null
);

insert into EventSizeDim values('1', 'Short', 'less than 10 people');
insert into EventSizeDim values('2', 'Medium','Between 11 and 30 people');
insert into EventSizeDim values('3', 'Large','More than 30 people');

select * from eventsizedim;

--- create OccupationDim
drop table OccupationDim;
create table OccupationDim
(occupationID varchar2(2) not null,
occupation_desc varchar2(20) not null
);

insert into occupationdim values('1', 'Student');
insert into occupationdim values('2', 'Staff');
insert into occupationdim values('3', 'Conmmunity');

---create locationDim
drop table locationDim;
create table locationdim as
select distinct address_state as state from address;

--- create maritaldim
drop table maritaldim;
create table maritaldim
( marital_id varchar2(2) not null,
```

```
marital_desc varchar2(50) not null
);

insert into maritaldim values ('1', 'Not married');
insert into maritaldim values ('2', 'Divorced');
insert into maritaldim values ('3', 'Married');



--- create agegroupdim
drop table agegroupdim;
create table agegroupdim
( age_group_id varchar2(2) not null,
age_group_name varchar2(50) not null,
age_group_desc varchar2(50) not null
);
insert into agegroupdim values ('1', 'Child', '0-16 years old');
insert into agegroupdim values ('2', 'Young adults', '17-30 years old');
insert into agegroupdim values ('3', 'Middle-aged adults', '31-45 years old');
insert into agegroupdim values ('4', 'Old-aged adults', 'Over 45 years old');

select * from agegroupdim;



---- create mediadim
drop table mediadim;
create table mediadim as
select * from media_channel;

---- create topicdim
drop table topicdim;
create table topicdim as
```

```
select * from topic;

---- create timedim
drop table time_1;
drop table time_2;
drop table time_3;
drop table time_4;
drop table time_5;
drop table timedim;

create table time_1 as
select to_char(subscription_date, 'MON-YYYY') as timeid,
to_char(subscription_date, 'MON') as month,
to_char(subscription_date, 'YYYY') as year
from subscription;

create table time_2 as
select to_char(event_start_date,'MON-YYYY') as timeid,
to_char(event_start_date, 'MON') as month,
to_char(event_start_date, 'YYYY') as year from event;

create table time_3 as
select to_char(event_end_date,'MON-YYYY') as timeid,
to_char(event_end_date, 'MON') as month,
to_char(event_end_date, 'YYYY') as year
from event;

create table time_4 as
select to_char(att_date,'MON-YYYY') as timeid,
to_char(att_date, 'MON') as month,
to_char(att_date, 'YYYY') as year
from attendance;
```

```
create table time_5 as
select to_char(reg_date, 'MON-YYYY') as timeid,
to_char(reg_date, 'MON') as month,
to_char(reg_date, 'YYYY') as year
from registration;


insert into time_1 select * from time_2;
insert into time_1 select * from time_3;
insert into time_1 select * from time_4;
insert into time_1 select * from time_5;

create table timedim as
select distinct * from time_1;


---  create programdim
drop table programdim;
create table programdim as
select program_id, program_name, program_details, program_fee,program_length,
program_frequency
from program;

alter table programdim
add program_length_type varchar2(10)
add program_length_type_desc varchar2(50);

update programdim
set
program_length_type = 'Short', program_length_type_desc = 'Less than 3 sessions'
where program_length in ('1 session','2 sessions');
```

```
update programdim
set
program_length_type = 'Medium', program_length_type_desc = 'Between 3 to 6 sessions'
where program_length in ('3 sessions','4 sessions','5 sessions','6 sessions');

update programdim
set
program_length_type = 'Long', program_length_type_desc = 'More than 6 sessions'
where program_length not in ('1 session','2 sessions','3 sessions','4 sessions','5 sessions','6 sessions');

alter table programdim
drop column program_length;

--- create subscriptionFact
drop table subscriptionFact;
drop table subscriptionFacttemp;

create table subscriptionFacttemp as
select s.subscription_id, pr.program_id, to_char(s.subscription_date,'MON-YYYY') as timeid,
pe.person_job, ad.address_state as state,
pe.person_marital_status, pe.person_age
from program pr, subscription s, person pe, address ad
where pr.program_id = s.program_id and s.person_id = pe.person_id
and pe.address_id = ad.address_id;


alter table subscriptionFacttemp
add occupationid varchar2(2)
add marital_id varchar2(2)
add age_group_id varchar2(2);

update subscriptionFacttemp
set
```

```
occupationid = '1' where person_job = 'Student';

update subscriptionFacttemp
set
occupationid = '2' where person_job = 'Staff';

update subscriptionFacttemp
set
occupationid = '3' where person_job not in('Student','Staff');

update subscriptionFacttemp
set
marital_id = '1' where person_marital_status = 'Not married';

update subscriptionFacttemp
set
marital_id = '2' where person_marital_status = 'Divorced';

update subscriptionFacttemp
set
marital_id = '3' where person_marital_status = 'Married';

update subscriptionFacttemp
set
age_group_id = '1' where person_age between '0' and '16';

update subscriptionFacttemp
set
age_group_id = '2' where person_age between '17' and '30';

update subscriptionFacttemp
set
```

```
age_group_id = '3' where person_age between '31' and '45';

update subscriptionFacttemp
set
age_group_id = '4' where person_age > 45;

select * from subscriptionFacttemp;

create table subscriptionFact as
select program_id, timeid, state ,occupationid, marital_id, age_group_id,
count(subscription_id) as Num_of_People_subscribed
from subscriptionFacttemp
group by program_id, timeid, state ,occupationid, marital_id, age_group_id;


---- create registrationFact
drop table registrationFact;
drop table registrationFacttemp;

create table registrationFacttemp as
select re.reg_id, to_char(re.reg_date, 'MON-YYYY') as timeid, pe.person_job, ad.address_state as
state,
pe.person_marital_status, pe.person_age, re.media_id
from registration re, person pe, address ad, media_channel me
where re.media_id = me.media_id and re.person_id = pe.person_id
and pe.address_id = ad.address_id;

alter table registrationFacttemp
add occupationid varchar2(2)
add marital_id varchar2(2)
add age_group_id varchar2(2);

update registrationFacttemp
```

```
set
occupationid = '1' where person_job = 'Student';

update registrationFacttemp
set
occupationid = '2' where person_job = 'Staff';

update registrationFacttemp
set
occupationid = '3' where person_job not in('Student','Staff');

update registrationFacttemp
set
marital_id = '1' where person_marital_status = 'Not married';

update registrationFacttemp
set
marital_id = '2' where person_marital_status = 'Divorced';

update registrationFacttemp
set
marital_id = '3' where person_marital_status = 'Married';

update registrationFacttemp
set
age_group_id = '1' where person_age between '0' and '16';

update registrationFacttemp
set
age_group_id = '2' where person_age between '17' and '30';

update registrationFacttemp
```

```
set
age_group_id = '3' where person_age between '31' and '45';

update registrationFacttemp
set
age_group_id = '4' where person_age > 45;

create table registrationFACT as
select timeid, state ,occupationid, marital_id, age_group_id, media_id,
count(reg_id) as Num_of_People_registered
from registrationFacttemp
group by timeid, state ,occupationid, marital_id, age_group_id, media_id;



---- create intersetFact
drop table interestFact;
drop table interestFacttemp;

create table interestFacttemp as
select person_interest.person_id, pe.person_job, ad.address_state as state,
pe.person_marital_status, pe.person_age, person_interest.topic_id
from person_interest, person pe, address ad, topic
where person_interest.person_id = pe.person_id and person_interest.topic_id = topic.topic_id
and pe.address_id = ad.address_id;

alter table interestFacttemp
add occupationid varchar2(2)
add marital_id varchar2(2)
add age_group_id varchar2(2);

update interestFacttemp
set
```

```
occupationid = '1' where person_job = 'Student';

update interestFacttemp
set
occupationid = '2' where person_job = 'Staff';

update interestFacttemp
set
occupationid = '3' where person_job not in('Student','Staff');

update interestFacttemp
set
marital_id = '1' where person_marital_status = 'Not married';

update interestFacttemp
set
marital_id = '2' where person_marital_status = 'Divorced';

update interestFacttemp
set
marital_id = '3' where person_marital_status = 'Married';

update interestFacttemp
set
age_group_id = '1' where person_age between '0' and '16';

update interestFacttemp
set
age_group_id = '2' where person_age between '17' and '30';

update interestFacttemp
set
```

```
age_group_id = '3' where person_age between '31' and '45';

update interestFacttemp
set
age_group_id = '4' where person_age > 45;

create table interestFACT as
select state ,occupationid, marital_id, age_group_id, topic_id,
count(person_id) as Num_of_People_interested
from interestFacttemp
group by state ,occupationid, marital_id, age_group_id, topic_id;



------ create attendencefact
drop table eventtemp;
create table eventtemp as
select * from event;

alter table eventtemp
add event_size_id varchar2(2) ;

update eventtemp
set event_size_id = '1' where event_size between '0' and '10';

update eventtemp
set event_size_id = '2' where event_size between '10' and '30';

update eventtemp
set event_size_id = '3' where event_size > 30;

drop table attendanceFact;
```

```
drop table attendanceFacttemp;

select * from attendanceFacttemp;


create table attendanceFacttemp as

select att.att_id, to_char(att.att_date, 'MON-YYYY') as timeid, pe.person_job, ad.address_state as
state,

pe.person_marital_status, pe.person_age, pr.program_id, ev.event_size_id, att.att_donation_amount

from program pr, person pe, address ad, eventtemp ev, attendance att

where att.event_id = ev.event_id and ev.program_id = pr.program_id

and att.person_id = pe.person_id

and pe.address_id = ad.address_id;



alter table attendanceFacttemp

add occupationid varchar2(2)

add marital_id varchar2(2)

add age_group_id varchar2(2);


update attendanceFacttemp

set

occupationid = '1' where person_job = 'Student';


update attendanceFacttemp

set

occupationid = '2' where person_job = 'Staff';


update attendanceFacttemp

set

occupationid = '3' where person_job not in('Student','Staff');


update attendanceFacttemp

set

marital_id = '1' where person_marital_status = 'Not married';
```

```
update attendanceFacttemp
set
marital_id = '2' where person_marital_status = 'Divorced';

update attendanceFacttemp
set
marital_id = '3' where person_marital_status = 'Married';

update attendanceFacttemp
set
age_group_id = '1' where person_age between '0' and '16';

update attendanceFacttemp
set
age_group_id = '2' where person_age between '17' and '30';

update attendanceFacttemp
set
age_group_id = '3' where person_age between '31' and '45';

update attendanceFacttemp
set
age_group_id = '4' where person_age > 45;

drop table attendanceFact;
create table attendanceFact as
select program_id, timeid, event_size_id, state ,occupationid, marital_id, age_group_id,
count(att_id) as Num_of_People_attended,
sum(att_donation_amount) as Total_donation
from attendanceFacttemp
group by program_id, timeid, event_size_id, state ,occupationid, marital_id, age_group_id;
```

```
select * from interestfact;
select * from registrationfact;
select * from attendancefact;
select * from subscriptionfact;

select * from programdim;
select * from eventsizedim;
select * from timedim;
select * from occupationdim;
select * from locationdim;
select * from maritaldim;
select * from agegroupdim;
select * from mediadim;
select * from topicdim;
```

b) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema Version-2

```
--using the cleanned table in data_clean process
drop table programdim;
drop table eventdim;
drop table attendancedim;
drop table subscriptiontimedim;
drop table addressdim;
drop table persondim;
drop table registrationdim;
drop table mediadim;
drop table topicdim;
drop table subscriptionfact;
drop table attendancefact;

--level 0
```

```
--ProgramDIM
select * from program;
create table ProgramDIM as select
program_id, program_name, program_details, program_fee, program_length, program_frequency
from program;

--EventDIM
select * from event;
create table EventDIM as select
event_id, event_start_date as start_date, event_end_date as end_date,
event_size, event_location, event_cost as total_cost
from event;

--AttendanceDIM
select * from attendance;
create table AttendanceDIM as select
att_id, att_date from attendance;

--SubscriptionTimeDIM
select * from subscription;
create table SubscriptionTimeDIM as select
distinct subscription_date from subscription;

--AddressDIM
select * from address;
create table AddressDIM as select * from address;

--PersonDIM
select * from person;
create table PersonDIM as select person_id, person_name, person_age,
person_email, person_gender, person_job, person_marital_status
from person;
```

```
--RegistrationDIM
select * from registration;
create table RegistrationDIM as select reg_id, reg_date
from registration;

--MediaDIM
select * from media_channel;
create table MediaDIM as select * from media_channel;

--TopicDIM
select * from topic;
create table TopicDIM as select * from topic;

--SubscriptionFact
create table SubscriptionFact as select
s.subscription_date,
p.program_id,
pe.person_id,
a.address_id,
count(s.subscription_id) as numer_of_people_subscriped
from subscription s, program p, person pe, address a
where s.program_id = p.program_id
and s.person_id = pe.person_id
and pe.address_id = a.address_id
group by s.subscription_date, p.program_id, pe.person_id, a.address_id;

--AttendanceFact
create table AttendanceFact as select
at.att_id, p.program_id, pe.person_id, a.address_id, e.event_id,
sum(at.att_num_of_people_attended) as numer_of_people_attended,
sum(at.att_donation_amount) as total_donation
```

```
from attendance at, program p, person pe, address a, event e
where e.program_id = p.program_id
and at.person_id = pe.person_id
and pe.address_id = a.address_id
and at.event_id = e.event_id
group by at.att_id, p.program_id, pe.person_id, a.address_id, e.event_id;

--RegistrationFact
create table RegistrationFact as select
re.reg_id, pe.person_id, a.address_id, m.media_id,
sum(re.reg_num_of_people_registered) as numer_of_people_registered
from registration re, media_channel m, person pe, address a
where re.person_id = pe.person_id
and pe.address_id = a.address_id
and re.media_id = m.media_id
group by re.reg_id, pe.person_id, a.address_id, m.media_id;

--InterestFact
create table InterestFact as select
pe.person_id, a.address_id, t.topic_id,
nvl(count(*),0) as numer_of_people_registered
from person_interest pi, topic t, person pe, address a
where pi.person_id = pe.person_id
and pe.address_id = a.address_id
and pi.topic_id = t.topic_id
group by pi.topic_id, pe.person_id, a.address_id, t.topic_id;
```

c) Screenshots of the tables that you have created; this includes the contents of each table that you have created. If the table is very big, you can show only the first part of the data.

- Level 2:

```
select * from programdim;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 19 in 0.016 seconds

	PROGRAM_ID	PROGRAM_NAME	PROGRAM_DETAILS	PROGRAM_FEE	PROGRAM_FREQ...	PROGRA...	PROGRAM_LENGTH_TYPE_DESC
1	PR001	Resume and Inte...	Teach how to write a resume an...	0	Twice a year	Short	Less than 3 sessions
2	PR002	PTE Preparation...	Teach the structure of PTE exa...	0	Twice a year	Short	Less than 3 sessions
3	PR003	Career Development	Discuss different skills to pr...	0	Twice a year	Long	More than 6 sessions
4	PR004	The Future CEO ...	Help to find the direction in ...	0	Once a year	Long	More than 6 sessions
5	PR005	Optimize Your B...	Help to improve mental perform...	0	Twice a year	Long	More than 6 sessions
6	PR006	Stress Management	Learn to manage stress through...	0	Twice a year	Long	More than 6 sessions
7	PR007	Plant-Based Coo...	Learn how to cook delicious pl...	0	Monthly	Short	Less than 3 sessions
8	PR008	Hiking	Walk in the nature	0	Twice a month	Short	Less than 3 sessions
9	PR009	Positive Relati...	Learn how to think positively ...	0	Twice a year	Short	Less than 3 sessions
10	PR010	Weight Loss	Learn how to lose weight safel...	0	Twice a year	Long	More than 6 sessions
11	PR011	Depression and ...	Help people with depression an...	0	Twice a year	Long	More than 6 sessions
12	PR012	Pilates	Weekly exercises	8	Weekly	Short	Less than 3 sessions
13	PR013	Health and Spir...	Learn about the connection bet...	0	Twice a year	Long	More than 6 sessions
14	PR014	Life of Excelle...	Learn how to live a life of ex...	0	Twice a year	Long	More than 6 sessions
15	PR015	Isolation Inspi...	Learn to make artworks during ...	0	Monthly	Long	More than 6 sessions

select * from eventsizedim;

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.019 seconds

	EVENT_SIZE_ID	EVENT_SIZE	EVENT_SIZE_DESC
1	1	Short	less than 10 people
2	2	Medium	Between 11 and 30 people
3	3	Large	More than 30 people

select * from timedim;

Script Output | Query Result

SQL | All Rows Fetched: 43 in (

	TIMEID	MONTH	YEAR
4	SEP-2018	SEP	2018
5	OCT-2018	OCT	2018
6	OCT-2019	OCT	2019
7	JAN-2020	JAN	2020
8	AUG-2020	AUG	2020
9	FEB-2018	FEB	2018
10	APR-2018	APR	2018
11	MAR-2019	MAR	2019
12	APR-2019	APR	2019
13	JUN-2019	JUN	2019
14	DEC-2019	DEC	2019
15	FEB-2020	FEB	2020

select * from occupationdim;

Script Output | Query Result | SQL | All Rows Fetched: 3 in 0.016 seconds

	OCCUPATIONID	OCCUPATION_DESC
1	1	Student
2	2	Staff
3	3	Conmmunity

select * from locationdim;

Script Output | Query Result | SQL | All Rows Fetched: 7 in

	STATE
1	QLD
2	SA
3	NSW
4	WA
5	ACT
6	VIC
7	TAS

select * from maritaldim;

779

Script Output | Query Result | SQL | All Rows Fetched: 3 in 0.019 se

	MARITAL_ID	MARITAL_DESC
1	1	Not married
2	2	Divorced
3	3	Married

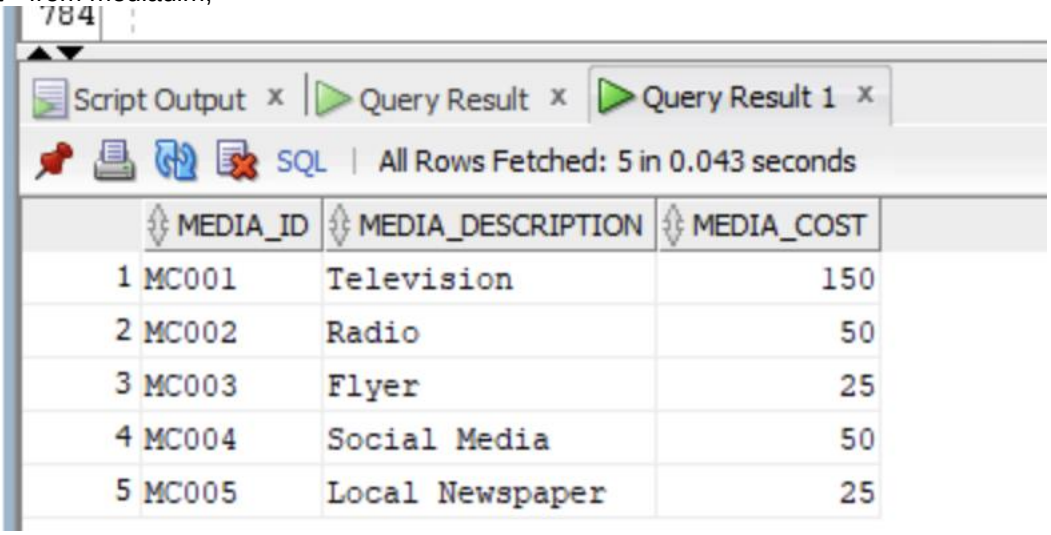
select * from agegroupdim;

398 insert into agegroupdim values ('4', 'Old-aged adults', '

Script Output | Query Result | SQL | All Rows Fetched: 4 in 0.018 seconds

	AGE_GROUP_ID	AGE_GROUP_NAME	AGE_GROUP_DESC
1	1	Child	0-16 years old
2	2	Young adults	17-30 years old
3	3	Middle-aged adults	31-45 years old
4	4	Old-aged adults	Over 45 years old

select * from mediadim;

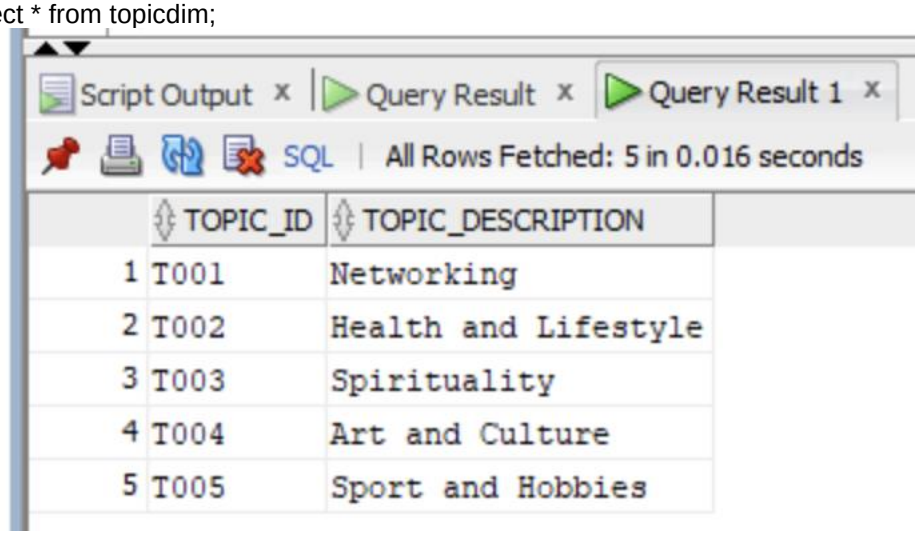


Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.043 seconds

	MEDIA_ID	MEDIA_DESCRIPTION	MEDIA_COST
1	MC001	Television	150
2	MC002	Radio	50
3	MC003	Flyer	25
4	MC004	Social Media	50
5	MC005	Local Newspaper	25

select * from topicdim;

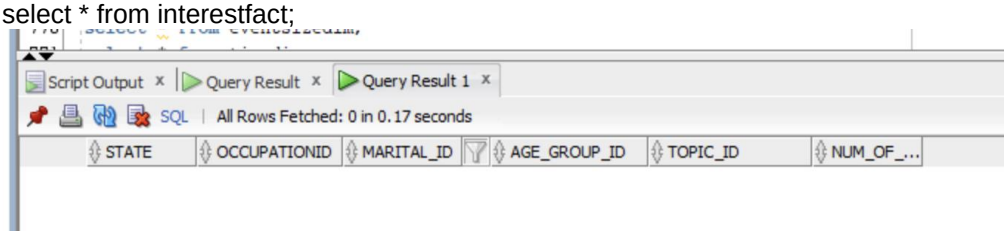


Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.016 seconds

	TOPIC_ID	TOPIC_DESCRIPTION
1	T001	Networking
2	T002	Health and Lifestyle
3	T003	Spirituality
4	T004	Art and Culture
5	T005	Sport and Hobbies

select * from interestfact;

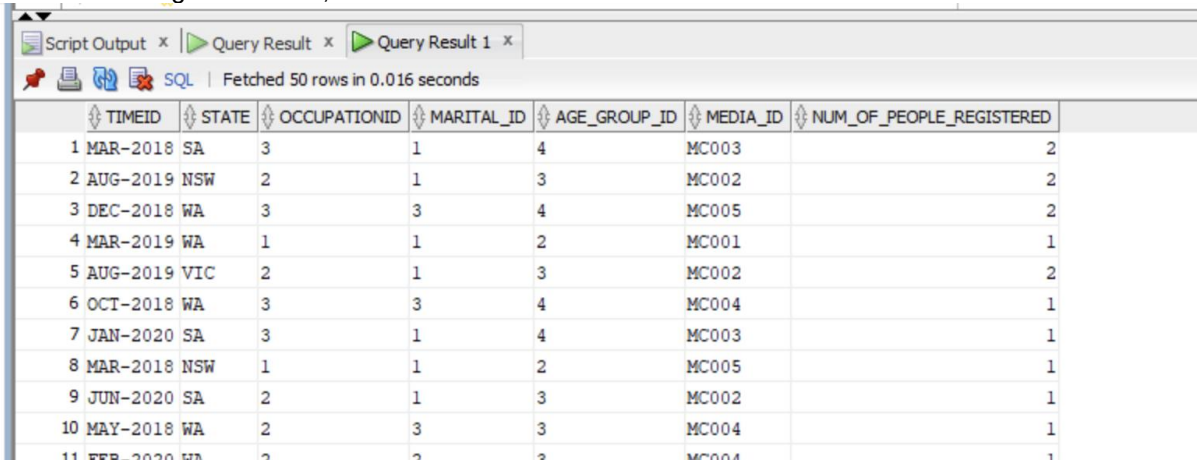


Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 0 in 0.17 seconds

STATE	OCCUPATIONID	MARITAL_ID	AGE_GROUP_ID	TOPIC_ID	NUM_OF_...

select * from registrationfact;



Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.016 seconds

	TIMEID	STATE	OCCUPATIONID	MARITAL_ID	AGE_GROUP_ID	MEDIA_ID	NUM_OF_PEOPLE_REGISTERED
1	MAR-2018	SA	3	1	4	MC003	2
2	AUG-2019	NSW	2	1	3	MC002	2
3	DEC-2018	WA	3	3	4	MC005	2
4	MAR-2019	WA	1	1	2	MC001	1
5	AUG-2019	VIC	2	1	3	MC002	2
6	OCT-2018	WA	3	3	4	MC004	1
7	JAN-2020	SA	3	1	4	MC003	1
8	MAR-2018	NSW	1	1	2	MC005	1
9	JUN-2020	SA	2	1	3	MC002	1
10	MAY-2018	WA	2	3	3	MC004	1

select * from attendancefact;

```
768
769  select * from programdim;
```

Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.016 seconds

	PROGRAM_ID	TIMEID	EVENT_SIZE_ID	STATE	OCCUPATIONID	MARITAL_ID	AGE_GROUP_ID	NUM_OF_PEOPLE_ATTENDED	TOTAL_DONATION
1	PR014	OCT-2020	2	QLD	3	1	4	5	195
2	PR014	DEC-2020	2	QLD	3	1	4	2	130
3	PR012	JAN-2018	3	WA	2	3	3	1	60
4	PR003	JUL-2019	3	QLD	2	1	3	2	135
5	PR012	NOV-2020	3	QLD	1	3	3	1	85
6	PR012	DEC-2018	3	SA	3	1	4	1	65
7	PR001	JAN-2020	3	TAS	1	3	2	1	40
8	PR011	MAR-2018	2	SA	2	1	3	3	150
9	PR015	APR-2019	3	QLD	1	1	3	14	630
10	PR011	MAR-2019	3	VIC	1	3	2	4	280

select * from subscriptionfact;

```
769  select * from programdim;
```

Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.019 seconds

	PROGRAM_ID	TIMEID	STATE	OCCUPATIONID	MARITAL_ID	AGE_GROUP_ID	NUM_OF_PEOPLE_SUB...
1	PR016	AUG-2017	WA	1	1	2	1
2	PR015	DEC-2017	VIC	2	3	3	1
3	PR008	JUL-2017	VIC	1	3	2	1
4	PR003	JUN-2017	ACT	3	2	4	1
5	PR007	JUN-2017	QLD	2	1	3	1
6	PR011	JUN-2017	NSW	1	3	3	1
7	PR001	NOV-2017	NSW	1	1	2	1
8	PR015	NOV-2017	VIC	2	2	3	1
9	PR018	AUG-2017	QLD	3	1	4	1
10	PR008	SEP-2017	TAS	2	2	3	1
11	PR015	DEC-2017	VIC	1	1	2	1

- Level 0:

select * from programdim;

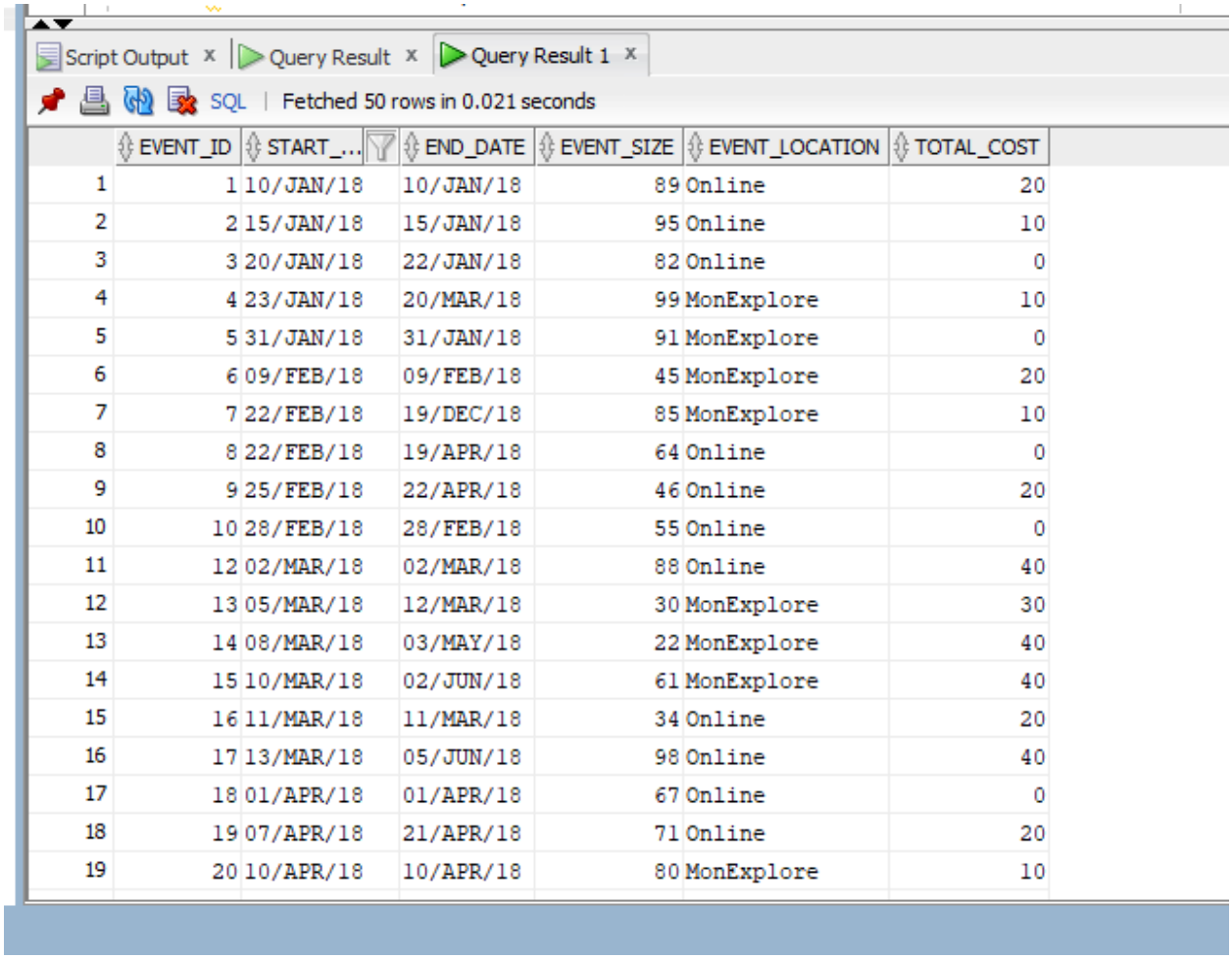
Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 19 in 0.02 seconds

	PROGRAM_ID	PROGRAM_NAME	PROGRAM_DETAILS	PROGRAM_FEE	PROGRAM_LENGTH	PROGRAM_FREQUENCY
1	PR001	Resume and Interview Skills	Teach how to write a resume and prepare for an interview	0	1 session	Twice a year
2	PR002	PTE Preparation Workshop	Teach the structure of PTE exam and provide hints	0	2 sessions	Twice a year
3	PR003	Career Development	Discuss different skills to prepare for job	0	8 sessions	Twice a year
4	PR004	The Future CEO Program	Help to find the direction in life and explore the career path; building friendships and networks	0	10 sessions	Once a year
5	PR005	Optimize Your Brain	Help to improve mental performance and emotional health	0	8 sessions	Twice a year
6	PR006	Stress Management	Learn to manage stress through lifestyle and analysing thoughts	0	8 sessions	Twice a year
7	PR007	Plant-Based Cooking Class	Learn how to cook delicious plant-based dishes	0	1 session	Monthly
8	PR008	Hiking	Walk in the nature	0	1 session	Twice a month
9	PR009	Positive Relationship	Learn how to think positively and maintain positive relationship	0	1 session	Twice a year
10	PR010	Weight Loss	Learn how to lose weight safely and healthly	0	7 sessions	Twice a year
11	PR011	Depression and Anxiety Recovery	Help people with depression and anxiety symptoms	0	8 sessions	Twice a year
12	PR012	Pilates	Weekly exercises	8	1 session	Weekly
13	PR013	Health and Spirituality	Learn about the connection between health and spirituality	0	12 sessions	Twice a year
14	PR014	Life of Excellence Seminar	Learn how to live a life of excellence	0	12 sessions	Twice a year
15	PR015	Isolation Inspiration	Learn to make artworks during isolation	0	14 sessions	Monthly
16	PR016	Art Therapy	Learn how to use art to for depression healing	0	1 session	Twice a year
17	PR017	Dance Chance	Dance group for people of all ages	0	1 session	Monthly
18	PR018	Geo-Coaching	Scavenger hunt play for boxes filled with goodies	10	3 sessions	Twice a year
19	PR019	Golf Coaching Clinic	Learn basic skills in golf	0	1 session	Twice a year

Line 107 Column 20 | Insert

select * from eventdim;

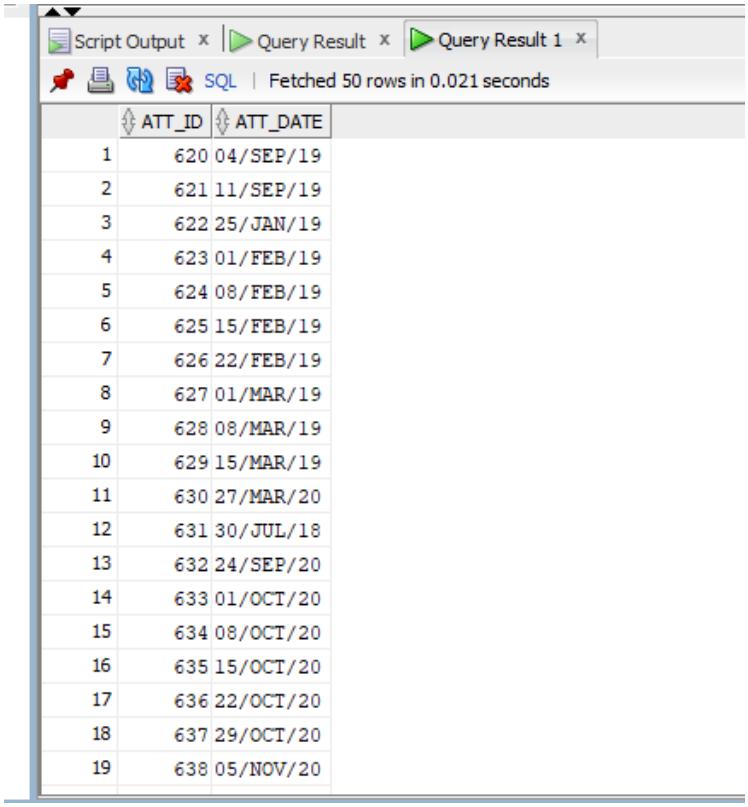


Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.021 seconds

	EVENT_ID	START_...	END_DATE	EVENT_SIZE	EVENT_LOCATION	TOTAL_COST
1	1	10/JAN/18	10/JAN/18	89	Online	20
2	2	15/JAN/18	15/JAN/18	95	Online	10
3	3	20/JAN/18	22/JAN/18	82	Online	0
4	4	23/JAN/18	20/MAR/18	99	MonExplore	10
5	5	31/JAN/18	31/JAN/18	91	MonExplore	0
6	6	09/FEB/18	09/FEB/18	45	MonExplore	20
7	7	22/FEB/18	19/DEC/18	85	MonExplore	10
8	8	22/FEB/18	19/APR/18	64	Online	0
9	9	25/FEB/18	22/APR/18	46	Online	20
10	10	28/FEB/18	28/FEB/18	55	Online	0
11	12	02/MAR/18	02/MAR/18	88	Online	40
12	13	05/MAR/18	12/MAR/18	30	MonExplore	30
13	14	08/MAR/18	03/MAY/18	22	MonExplore	40
14	15	10/MAR/18	02/JUN/18	61	MonExplore	40
15	16	11/MAR/18	11/MAR/18	34	Online	20
16	17	13/MAR/18	05/JUN/18	98	Online	40
17	18	01/APR/18	01/APR/18	67	Online	0
18	19	07/APR/18	21/APR/18	71	Online	20
19	20	10/APR/18	10/APR/18	80	MonExplore	10

select * from attendancedim;



Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.021 seconds

	ATT_ID	ATT_DATE
1	620	04/SEP/19
2	621	11/SEP/19
3	622	25/JAN/19
4	623	01/FEB/19
5	624	08/FEB/19
6	625	15/FEB/19
7	626	22/FEB/19
8	627	01/MAR/19
9	628	08/MAR/19
10	629	15/MAR/19
11	630	27/MAR/20
12	631	30/JUL/18
13	632	24/SEP/20
14	633	01/OCT/20
15	634	08/OCT/20
16	635	15/OCT/20
17	636	22/OCT/20
18	637	29/OCT/20
19	638	05/NOV/20

select * from subscriptiontimedim;

Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.018 seconds

	SUBSCRIPTION_DATE
1	01/JUL/17
2	05/JUL/17
3	14/JUL/17
4	17/SEP/17
5	24/NOV/17
6	08/JUL/17
7	05/OCT/17
8	14/JUN/17
9	02/AUG/17
10	08/DEC/17
11	10/NOV/17
12	06/DEC/17
13	17/JUN/17
14	06/AUG/17
15	07/JUL/17
16	12/OCT/17
17	07/NOV/17
18	04/SEP/17
19	02/JUL/17

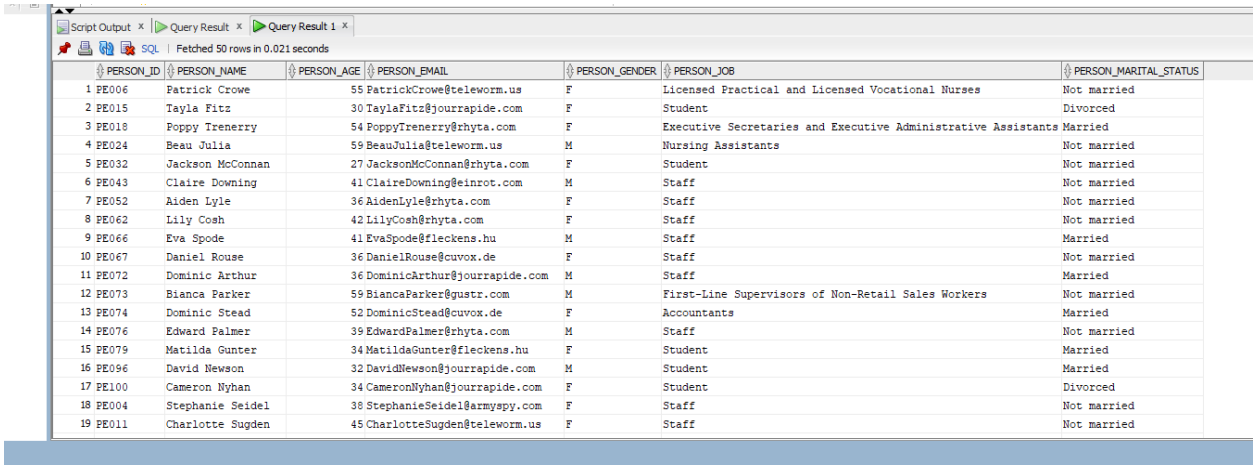
select * from addressdim;

Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.02 seconds

	ADDRESS_ID	ADDDRESS_STREET_NO	ADDRESS_STREET_NAME	ADDRESS_SUBURB	ADDRESS_STATE	ADDRESS_POSTCODE
1	AD001	571	Walker Street	Upper Ferntree Gully	VIC	3156
2	AD002	851	Parsons Street	Murrays Run	NSW	2325
3	AD003	336	Barnett Street	Mount White	NSW	2250
4	AD004	914	Magistrates Walk	Veradilla	QLD	4347
5	AD005	799	Witchwood Close	Pimlico	QLD	4812
6	AD006	435	Rankins Road	Charters Towers City	QLD	4820
7	AD007	962	Macaulay Road	Borung	VIC	3518
8	AD008	743	Rankins Road	Trayning	WA	6488
9	AD009	859	Hotham Place	Fitzroy Crossing	WA	6765
10	AD010	908	Lambeth Street	Glenden	QLD	4743
11	AD011	774	Smith Street	Wheeny Creek	NSW	2758
12	AD012	262	Melrose Street	East Kempsey	NSW	2440
13	AD013	30	Lothian Street	Talisker	WA	6701
14	AD014	876	Epsom Road	Hope Vale	QLD	4895
15	AD015	999	Market Street	Beerwah	QLD	4519
16	AD016	476	Collett Street	Guildford	WA	6055
17	AD017	130	Magistrates Walk	Bourke	NSW	2840
18	AD018	777	Fisken Place	Lort River	WA	6447
19	AD019	562	Barnett Street	Mount Erin	WA	6532

select * from persondim;



Script Output | Query Result | Query Result 1

Fetched 50 rows in 0.021 seconds

	PERSON_ID	PERSON_NAME	PERSON_AGE	PERSON_EMAIL	PERSON_GENDER	PERSON_JOB	PERSON_MARITAL_STATUS
1	PE006	Patrick Crowe	55	PatrickCrowe@teleworm.us	F	Licensed Practical and Licensed Vocational Nurses	Not married
2	PE015	Tayla Fitz	30	TaylaFitz@jourrapide.com	F	Student	Divorced
3	PE018	Poppy Trenerry	54	PoppyTrenerry@rhyta.com	F	Executive Secretaries and Executive Administrative Assistants	Married
4	PE024	Beau Julia	59	BeauJulia@teleworm.us	M	Nursing Assistants	Not married
5	PE032	Jackson McConnan	27	JacksonMcConnan@rhyta.com	F	Student	Not married
6	PE043	Claire Downing	41	ClaireDowning@einrot.com	M	Staff	Not married
7	PE052	Aiden Lyle	36	AidenLyle@rhyta.com	F	Staff	Not married
8	PE062	Lily Cosh	42	LilyCosh@rhyta.com	F	Staff	Not married
9	PE066	Eva Spode	41	EvaSpode@fleckens.hu	M	Staff	Married
10	PE067	Daniel Rouse	36	DanielRouse@cuvox.de	F	Staff	Not married
11	PE072	Dominic Arthur	36	DominicArthur@jourrapide.com	M	Staff	Married
12	PE073	Bianca Parker	59	BiancaParker@gustr.com	M	First-Line Supervisors of Non-Retail Sales Workers	Not married
13	PE074	Dominic Stead	52	DominicStead@cuvox.de	F	Accountants	Married
14	PE076	Edward Palmer	39	EdwardPalmer@rhyta.com	M	Staff	Not married
15	PE079	Matilda Gunter	34	MatildaGunter@fleckens.hu	F	Student	Married
16	PE096	David Newson	32	DavidNewson@jourrapide.com	M	Student	Married
17	PE100	Cameron Nyhan	34	CameronNyhan@jourrapide.com	F	Student	Divorced
18	PE004	Stephanie Seidel	38	StephanieSeidel@armyspy.com	F	Staff	Not married
19	PE011	Charlotte Sugden	45	CharlotteSugden@teleworm.us	F	Staff	Not married

select * from registrationdim;

```
124  select * from Registrationfact order by reg_id;
125  select * from interestfact;
```

Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.033 seconds

	REG_ID	REG_DATE
1	617	06/MAR/18
2	618	16/JAN/18
3	619	25/AUG/19
4	620	20/DEC/18
5	621	23/JUN/18
6	622	24/FEB/20
7	623	20/AUG/18
8	624	21/JUN/20
9	625	11/FEB/20
10	626	26/MAR/19
11	627	08/JAN/18
12	628	29/DEC/18
13	629	11/AUG/20
14	630	01/MAR/19
15	631	29/MAR/19
16	632	12/MAY/20
17	633	17/AUG/20
18	634	17/AUG/19
19	635	16/SEP/19

select * from mediadim;

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.019 seconds

	MEDIA_ID	MEDIA_DESCRIPTION	MEDIA_COST
1	MC001	Television	150
2	MC002	Radio	50
3	MC003	Flyer	25
4	MC004	Social Media	50
5	MC005	Local Newspaper	25

select * from topicdim;

```
125  select * from interestfact;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.02 seconds

	TOPIC_ID	TOPIC_DESCRIPTION
1	T001	Networking
2	T002	Health and Lifestyle
3	T003	Spirituality
4	T004	Art and Culture
5	T005	Sport and Hobbies

select count(*) from subscription;

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0.018 seconds

	COUNT(*)
1	500

select count(*) from subscriptionfact;

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0.016 seconds

	COUNT(*)
1	500

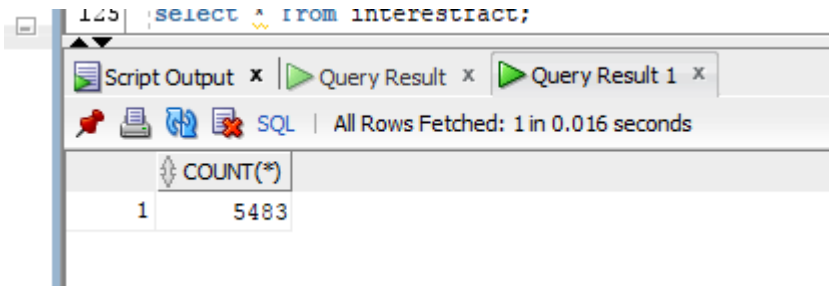
select * from subscriptionfact;

Script Output | Query Result | Query Result 1

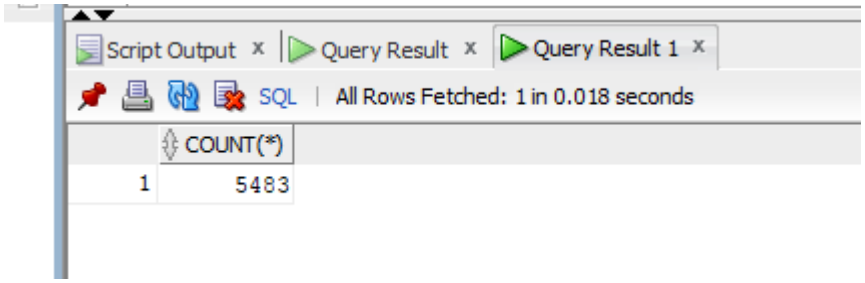
SQL | Fetched 50 rows in 0.02 seconds

	SUBSCRIPTION_DATE	PROGRAM_ID	PERSON_ID	ADDRESS_ID	NUMER_OF_PEOPLE_SUBSCRIPED
1	18/OCT/17	PR013	PE017	AD017	1
2	03/JUN/17	PR002	PE004	AD004	1
3	11/DEC/17	PR016	PE005	AD005	1
4	06/JUN/17	PR010	PE049	AD049	1
5	25/AUG/17	PR014	PE098	AD098	1
6	05/DEC/17	PR008	PE042	AD042	1
7	10/JUL/17	PR004	PE091	AD091	1
8	24/NOV/17	PR019	PE060	AD060	1
9	21/NOV/17	PR001	PE003	AD003	1
10	07/SEP/17	PR008	PE095	AD095	1
11	21/DEC/17	PR008	PE084	AD084	1
12	13/AUG/17	PR011	PE019	AD019	1
13	16/AUG/17	PR016	PE095	AD095	1
14	06/DEC/17	PR019	PE027	AD027	1
15	10/NOV/17	PR005	PE082	AD082	1
16	09/NOV/17	PR018	PE082	AD082	1
17	15/SEP/17	PR015	PE072	AD072	1
18	04/JUN/17	PR005	PE051	AD051	1
19	08/DEC/17	PR018	PE016	AD016	1

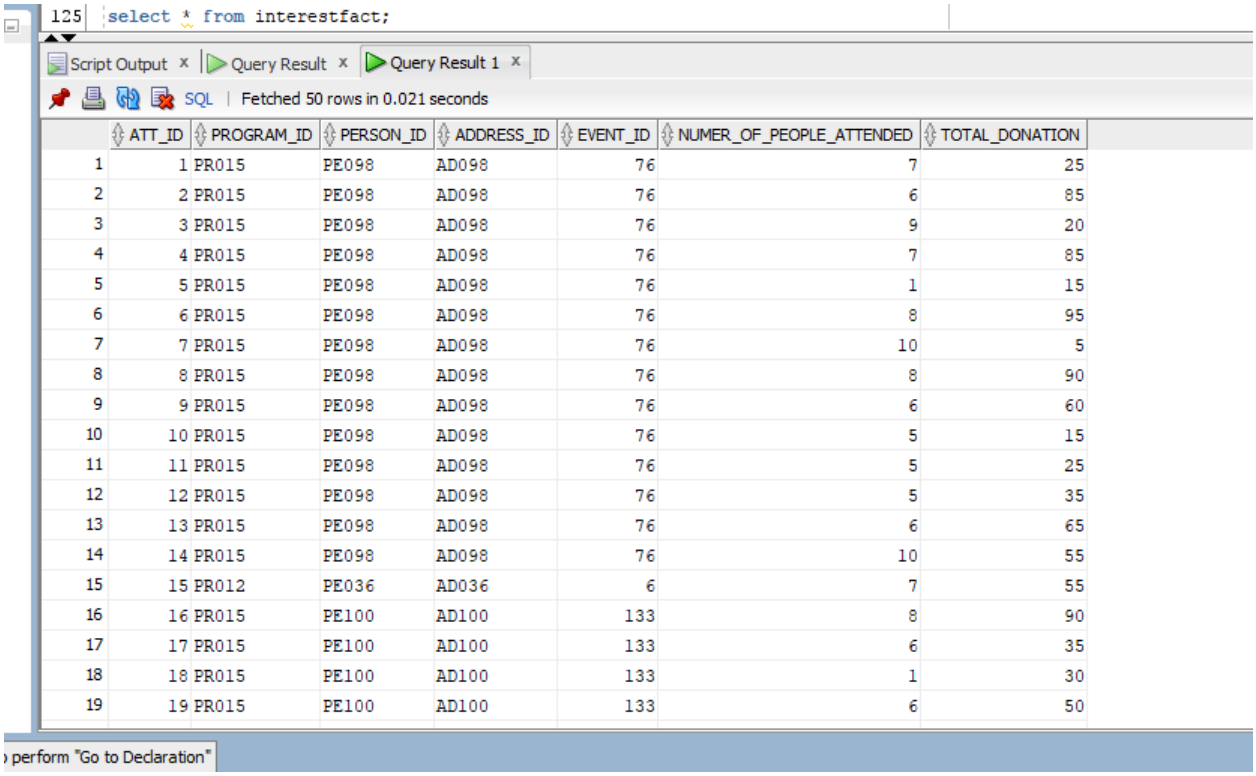
select count(*) from attendance;



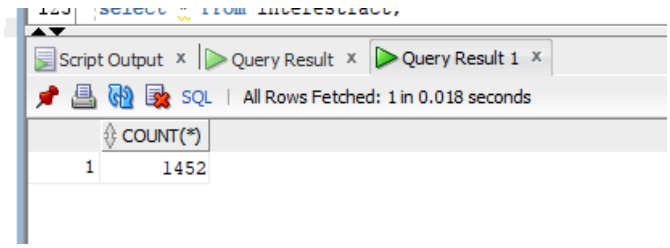
select count(*) from attendancefact;



select * from attendancefact order by att_id;



select count(*) from Registration;



select count(*) from Registrationfact;

```
125  select * from interestfact;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0.015 seconds

	COUNT(*)
1	1452

select * from Registrationfact order by reg_id;

Script Output | Query Result | Query Result 1

SQL | Fetched 50 rows in 0.018 seconds

	REG_ID	PERSON_ID	ADDRESS_ID	MEDIA_ID	NUMER_OF_PEOPLE_REGISTERED
1	1	PE098	AD098	MC001	2
2	2	PE036	AD036	MC005	1
3	3	PE100	AD100	MC001	2
4	4	PE012	AD012	MC003	1
5	5	PE035	AD035	MC001	3
6	6	PE043	AD043	MC003	1
7	7	PE080	AD080	MC003	1
8	8	PE085	AD085	MC004	3
9	9	PE082	AD082	MC004	1
10	10	PE047	AD047	MC002	2
11	11	PE002	AD002	MC001	2
12	13	PE060	AD060	MC001	3
13	14	PE078	AD078	MC002	3
14	15	PE085	AD085	MC003	1
15	16	PE069	AD069	MC004	4
16	17	PE082	AD082	MC002	4
17	18	PE038	AD038	MC003	4
18	19	PE030	AD030	MC005	3
19	20	PE067	AD067	MC003	4

select * from interestfact; --no records in the cleaned person_interest table

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 0 in 0.016 seconds

PERSON_ID	ADDRESS...	TOPIC_ID	NUMER_O...