Introduction:

This report is based on conducting a Data mart implementation method on students' overall criteria of progression. The capacity to assess data, plan crucial facts, and take action in response to changes in the business environment is essential for a firm to advance. This is Only if the structure is appropriate for the decision-making process and precise information is accessible in a certain data format can ability be fulfilled. (Chaudhuri and Dayal , 1997) The ETL (EXTRACT, TRANSFORM, AND LOAD) process is done in this report and also shows OLAP (ONLINE ANALYTICS PROCESSING). This report also conducted social ethics. I'm going to discuss how schools, colleges, and universities analyse student academic performance based on modules, years, and marks.

1 Data Integration (ETL) and Maintenance

1.1 Original datasets

There are few datasets:

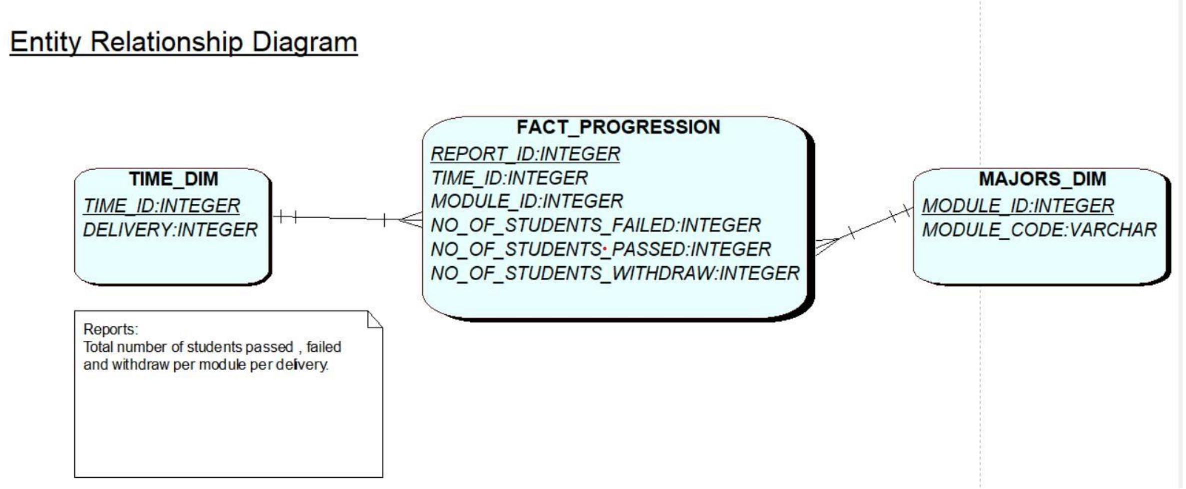
TIME, MODULE, and FACT_1.

There are few tables:

(FACT_ASSESSMENTS, TIME_DIM, MODULE_DIM);

I am utilizing these tables for the Start schema and further process to ETL to clean data and load data into a star schema.

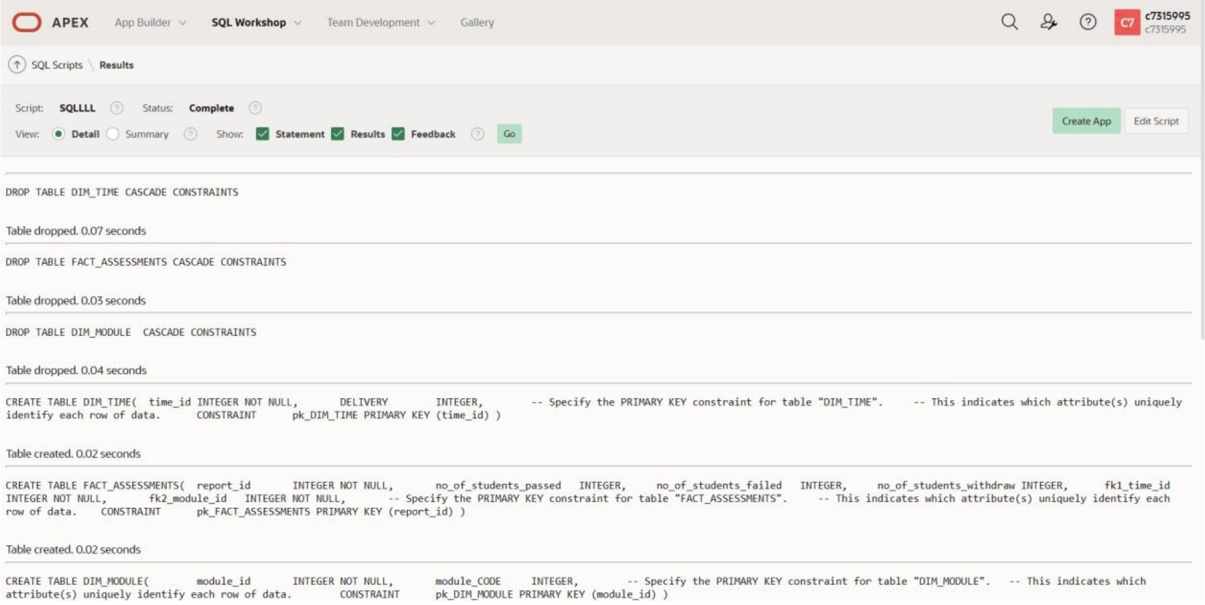
1.2 Star schema:


Entity Relationship Diagram
TIME_DIM
TIME_ID:INTEGER
DELIVERY:INTEGER
FACT_PROGRESSION
REPORT_ID:INTEGER
TIME_ID:INTEGER
MODULE_ID:INTEGER
NO_OF_STUDENTS_FAILED:INTEGER
NO_OF_STUDENTS_PASSED:INTEGER
NO_OF_STUDENTS_WITHDRAW:INTEGER
MAJORS_DIM
MODULE_ID:INTEGER
MODULE_CODE:VARCHAR
Reports:
Total number of students passed , failed and withdraw per module per deivery.

Star schema evolute the overall progress of the student's academic performance. This contains tables and columns. Referenced table is FACT_PROGRESSION where all primary keys are stored. Fact tables store transactional or measured data and DIMENSION tables store attributes about the data.

The support for attribute hierarchies in star schemas is implicit. The dimensional hierarchy is explicitly expressed in snowflake schemas by normalising the dimension tables, which is an improvement over star schemas.(Chaudhuri and Dayal,1997).

Star schema query generate:



Script: SQLLLL Status: Complete

View: ● Detail ○ Summary Show: ☑ Statement ☑ Results ☑ Feedback Go

```
DROP TABLE DIM_TIME CASCADE CONSTRAINTS
```

Table dropped. 0.07 seconds

```
DROP TABLE FACT_ASSESSMENTS CASCADE CONSTRAINTS
```

Table dropped. 0.03 seconds

```
DROP TABLE DIM_MODULE  CASCADE CONSTRAINTS
```

Table dropped. 0.04 seconds

```
CREATE TABLE DIM_TIME(  time_id INTEGER NOT NULL,       DELIVERY       INTEGER,      -- Specify the PRIMARY KEY constraint for table "DIM_TIME".    -- This indicates which attribute(s) uniquely
identify each row of data.      CONSTRAINT      pk_DIM_TIME PRIMARY KEY (time_id) )
```

Table created. 0.02 seconds

```
CREATE TABLE FACT_ASSESSMENTS(  report_id       INTEGER NOT NULL,       no_of_students_passed   INTEGER,     no_of_students_failed   INTEGER,     no_of_students_withdraw INTEGER,      fk1_time_id
INTEGER NOT NULL,       fk2_module_id   INTEGER NOT NULL,       -- Specify the PRIMARY KEY constraint for table "FACT_ASSESSMENTS".    -- This indicates which attribute(s) uniquely identify each
row of data.    CONSTRAINT      pk_FACT_ASSESSMENTS PRIMARY KEY (report_id) )
```

Table created. 0.02 seconds

```
CREATE TABLE DIM_MODULE(        module_id       INTEGER NOT NULL,       module_CODE     INTEGER,       -- Specify the PRIMARY KEY constraint for table "DIM_MODULE".   -- This indicates which
attribute(s) uniquely identify each row of data.        CONSTRAINT      pk_DIM_MODULE PRIMARY KEY (module_id) )
```

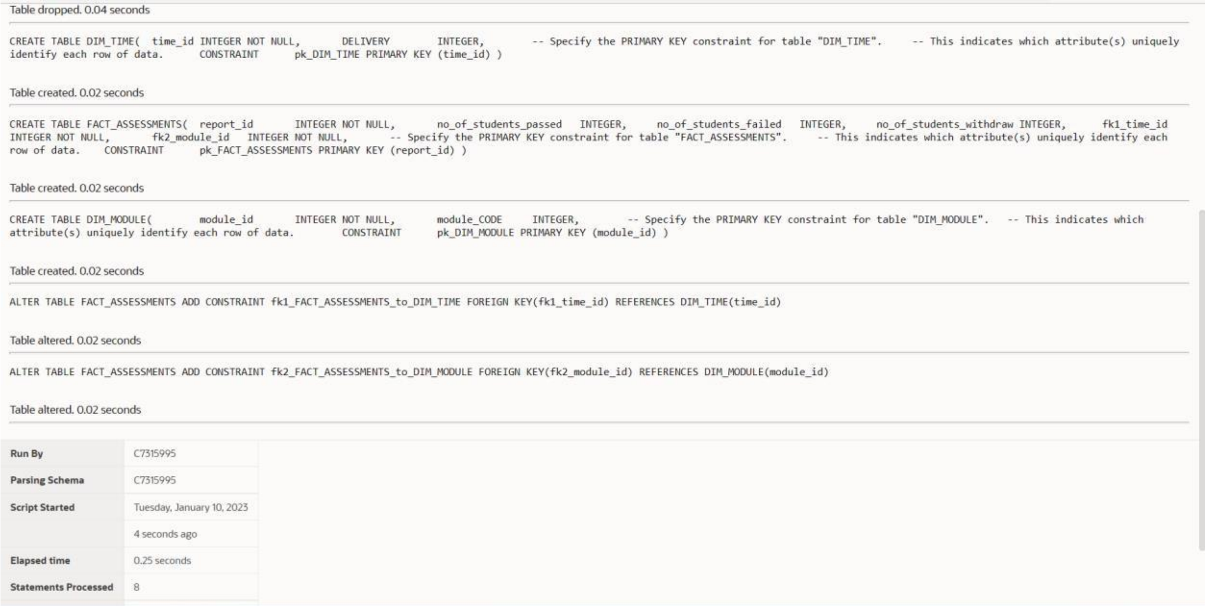


Table dropped. 0.04 seconds

```
CREATE TABLE DIM_TIME(  time_id INTEGER NOT NULL,       DELIVERY       INTEGER,      -- Specify the PRIMARY KEY constraint for table "DIM_TIME".    -- This indicates which attribute(s) uniquely
identify each row of data.      CONSTRAINT      pk_DIM_TIME PRIMARY KEY (time_id) )
```

Table created. 0.02 seconds

```
CREATE TABLE FACT_ASSESSMENTS(  report_id       INTEGER NOT NULL,       no_of_students_passed   INTEGER,     no_of_students_failed   INTEGER,     no_of_students_withdraw INTEGER,      fk1_time_id
INTEGER NOT NULL,       fk2_module_id   INTEGER NOT NULL,       -- Specify the PRIMARY KEY constraint for table "FACT_ASSESSMENTS".    -- This indicates which attribute(s) uniquely identify each
row of data.    CONSTRAINT      pk_FACT_ASSESSMENTS PRIMARY KEY (report_id) )
```

Table created. 0.02 seconds

```
CREATE TABLE DIM_MODULE(        module_id       INTEGER NOT NULL,       module_CODE     INTEGER,       -- Specify the PRIMARY KEY constraint for table "DIM_MODULE".   -- This indicates which
attribute(s) uniquely identify each row of data.        CONSTRAINT      pk_DIM_MODULE PRIMARY KEY (module_id) )
```

Table created. 0.02 seconds

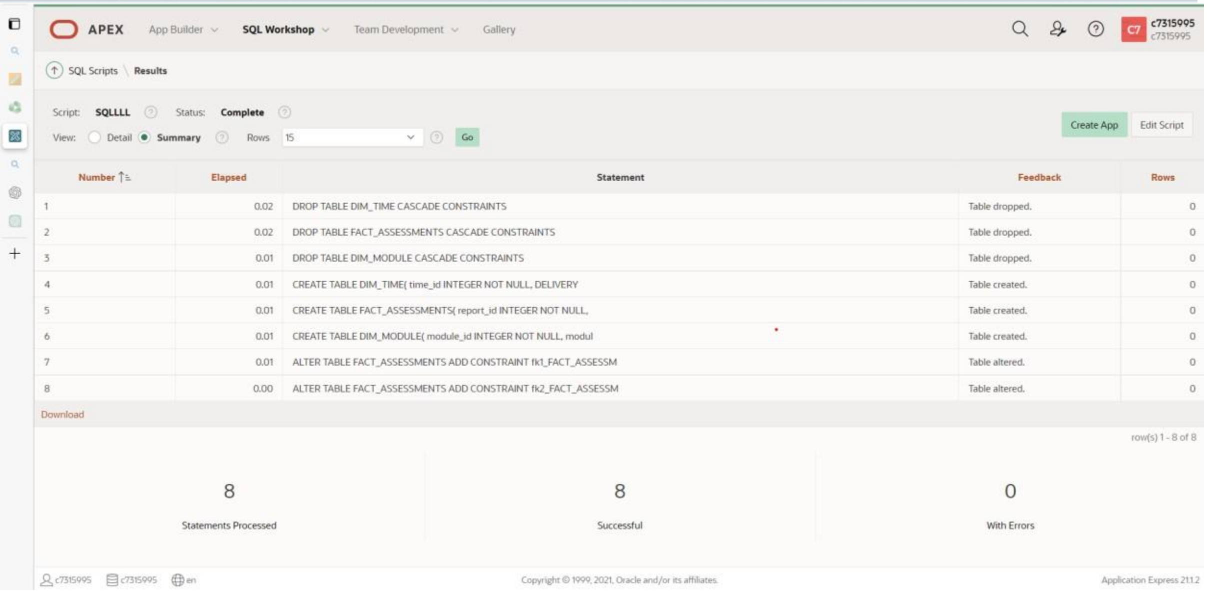
```
ALTER TABLE FACT_ASSESSMENTS ADD CONSTRAINT fk1_FACT_ASSESSMENTS_to_DIM_TIME FOREIGN KEY(fk1_time_id) REFERENCES DIM_TIME(time_id)
```

Table altered. 0.02 seconds

```
ALTER TABLE FACT_ASSESSMENTS ADD CONSTRAINT fk2_FACT_ASSESSMENTS_to_DIM_MODULE FOREIGN KEY(fk2_module_id) REFERENCES DIM_MODULE(module_id)
```

Table altered. 0.02 seconds

Run By	C7315995
Parsing Schema	C7315995
Script Started	Tuesday, January 10, 2023
	4 seconds ago
Elapsed time	0.25 seconds
Statements Processed	8



Script: SQLLLL Status: Complete

View: ○ Detail ● Summary Rows 15 Go

Number	Elapsed	Statement	Feedback	Rows
1	0.02	DROP TABLE DIM_TIME CASCADE CONSTRAINTS	Table dropped.	0
2	0.02	DROP TABLE FACT_ASSESSMENTS CASCADE CONSTRAINTS	Table dropped.	0
3	0.01	DROP TABLE DIM_MODULE CASCADE CONSTRAINTS	Table dropped.	0
4	0.01	CREATE TABLE DIM_TIME(time_id INTEGER NOT NULL, DELIVERY	Table created.	0
5	0.01	CREATE TABLE FACT_ASSESSMENTS(report_id INTEGER NOT NULL,	Table created.	0
6	0.01	CREATE TABLE DIM_MODULE(module_id INTEGER NOT NULL, modul	Table created.	0
7	0.01	ALTER TABLE FACT_ASSESSMENTS ADD CONSTRAINT fk1_FACT_ASSESSM	Table altered.	0
8	0.00	ALTER TABLE FACT_ASSESSMENTS ADD CONSTRAINT fk2_FACT_ASSESSM	Table altered.	0

Download

row(s) 1 - 8 of 8

8 Statements Processed

8 Successful

0 With Errors

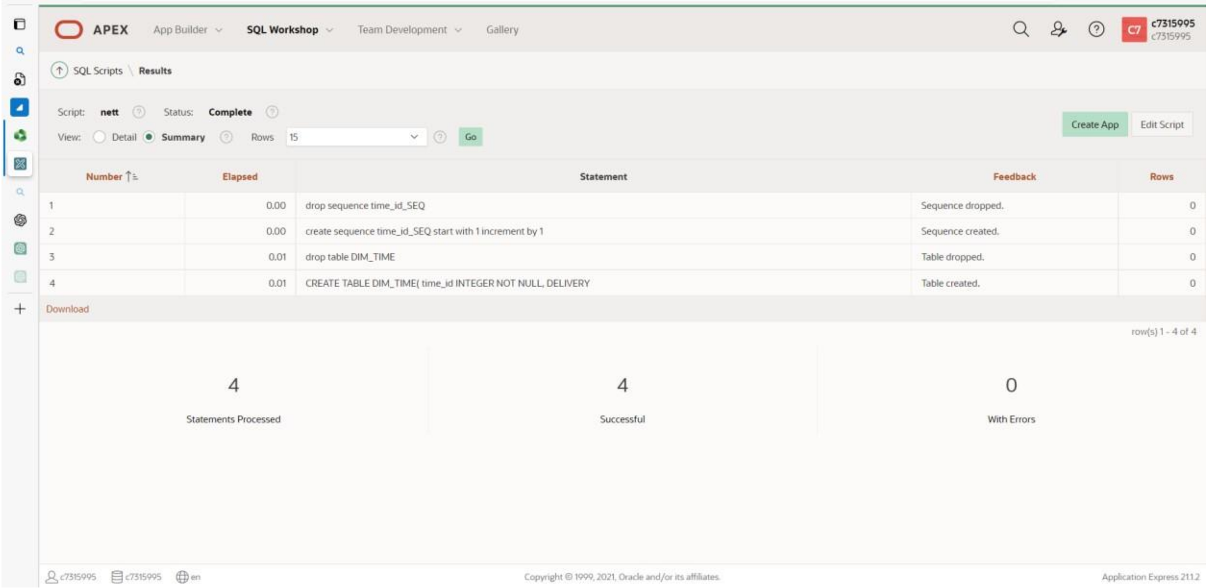
c7315995 c7315995 en

Copyright © 1999, 2021, Oracle and/or its affiliates.

Application Express 21.1.2

Create the sequence for DIM_TIME:

c



Script: nett Status: Complete

View: ○ Detail ● Summary Rows 15 Go

Number	Elapsed	Statement	Feedback	Rows
1	0.00	drop sequence time_id_SEQ	Sequence dropped.	0
2	0.00	create sequence time_id_SEQ start with 1 increment by 1	Sequence created.	0
3	0.01	drop table DIM_TIME	Table dropped.	0
4	0.01	CREATE TABLE DIM_TIME(time_id INTEGER NOT NULL, DELIVERY	Table created.	0

Download

row(s) 1 - 4 of 4

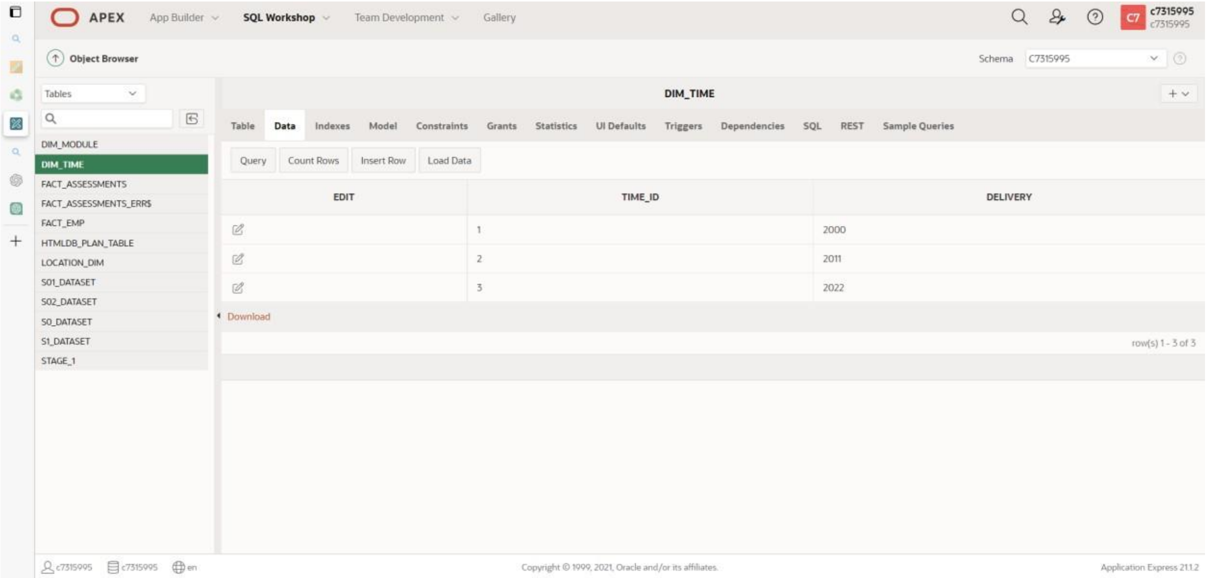
4 Statements Processed

4 Successful

0 With Errors

c7315995
c7315995
en
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.2

Load data into DIM_TIME



Schema C7315995

DIM_MODULE
DIM_TIME
FACT_ASSESSMENTS
FACT_ASSESSMENTS_ERR$
FACT_EMP
HTMLDB_PLAN_TABLE
LOCATION_DIM
S01_DATASET
S02_DATASET
S0_DATASET
S1_DATASET
STAGE_1

DIM_TIME

EDIT	TIME_ID	DELIVERY
	1	2000
	2	2011
	3	2022

Download

row(s) 1 - 3 of 3

c7315995
c7315995
en
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.2

1.3 EXTRACT TRANSFORM AND LOAD :

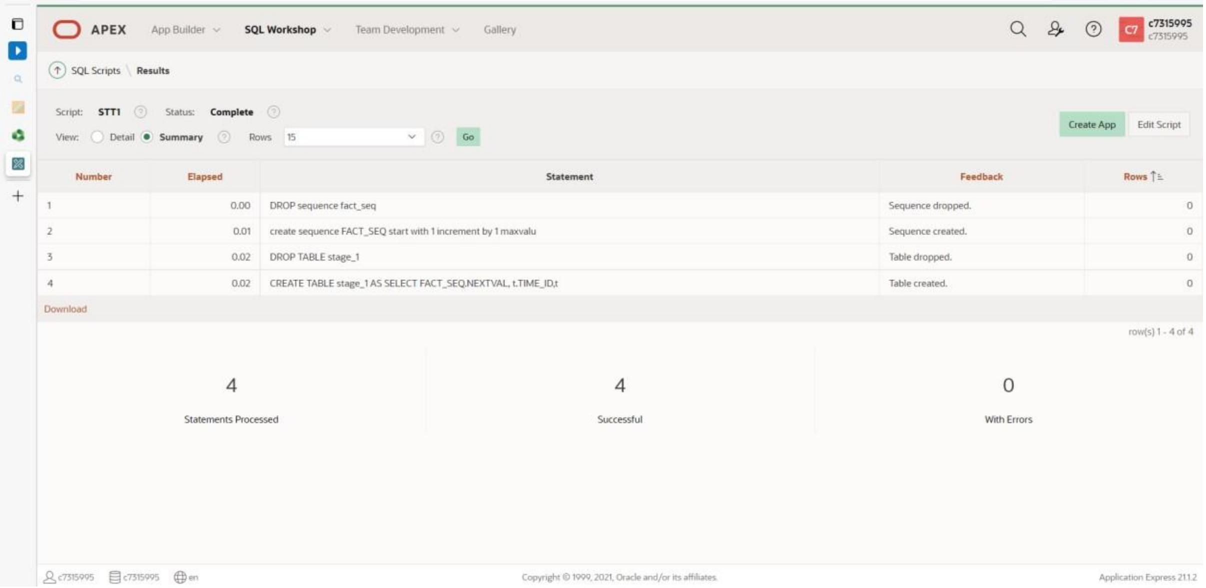
1.3.1 Extract data into a stage table

Designers subconsciously get ready for OLAP queries with quick responses at the ETL step. Measuring, loading dimensions, and idea hierarchies are three crucial components of ETL that help to speed up OLAP. The staging table that was previously developed is where the first two stages—extract and transform—are put into practice. Selecting the necessary data and determining whether they satisfy the desired goal are both steps in the extraction process. (Singh et al.,2016).

Since it pulls student data from sources, corrects any problems in the data, integrates the data to fit the model of a target DW, and loads the data into a DW, "Extract, Transform, and Load" (ETL) is a crucial building block of a DW.(Zhao and LIU , 2022)

Create the staging area to show the specific result of NO_OF_STUDENTS_PASSED, NO_OF_STUDENTS_FAILED, NO_OF_STUDENTS_WITHDRAW.

First stage area stage_1 : it's showing NO_of_Students_passed



Script: STT1 Status: Complete

View: ○ Detail ● Summary Rows 15 Go

Number	Elapsed	Statement	Feedback	Rows
1	0.00	DROP sequence fact_seq	Sequence dropped.	0
2	0.01	create sequence FACT_SEQ start with 1 increment by 1 maxvalu	Sequence created.	0
3	0.02	DROP TABLE stage_1	Table dropped.	0
4	0.02	CREATE TABLE stage_1 AS SELECT FACT_SEQ.NEXTVAL, t.TIME_ID,t	Table created.	0

row(s) 1 - 4 of 4

4 Statements Processed

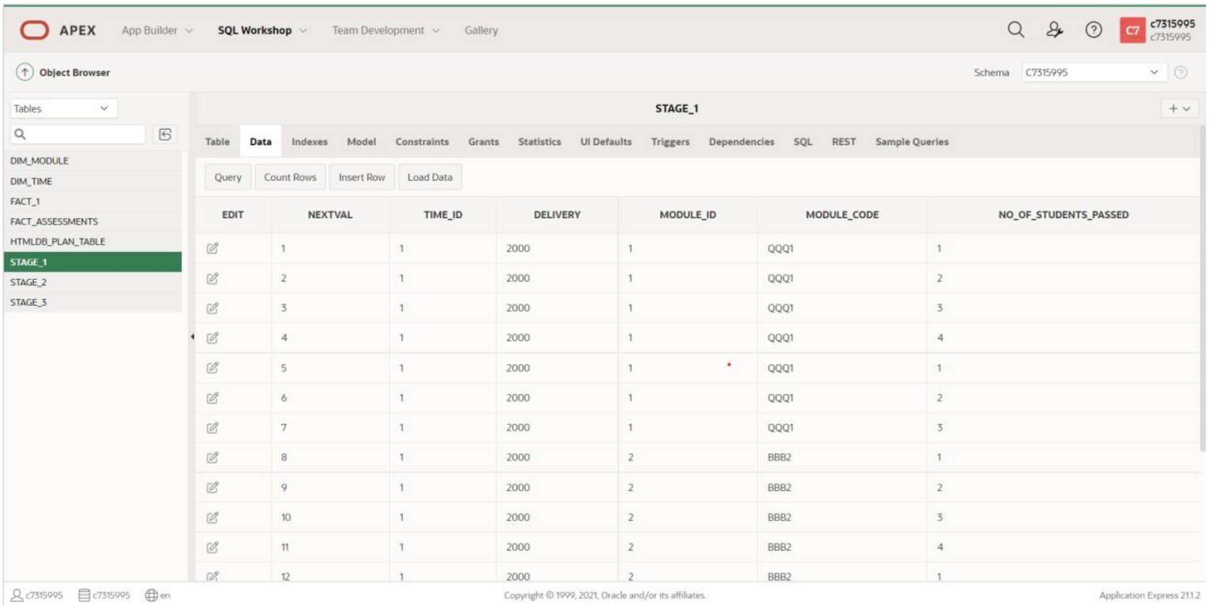
4 Successful

0 With Errors

c7315995 c7315995 en

Copyright © 1999, 2021, Oracle and/or its affiliates.

Application Express 21.1.2



Schema C7315995

STAGE_1

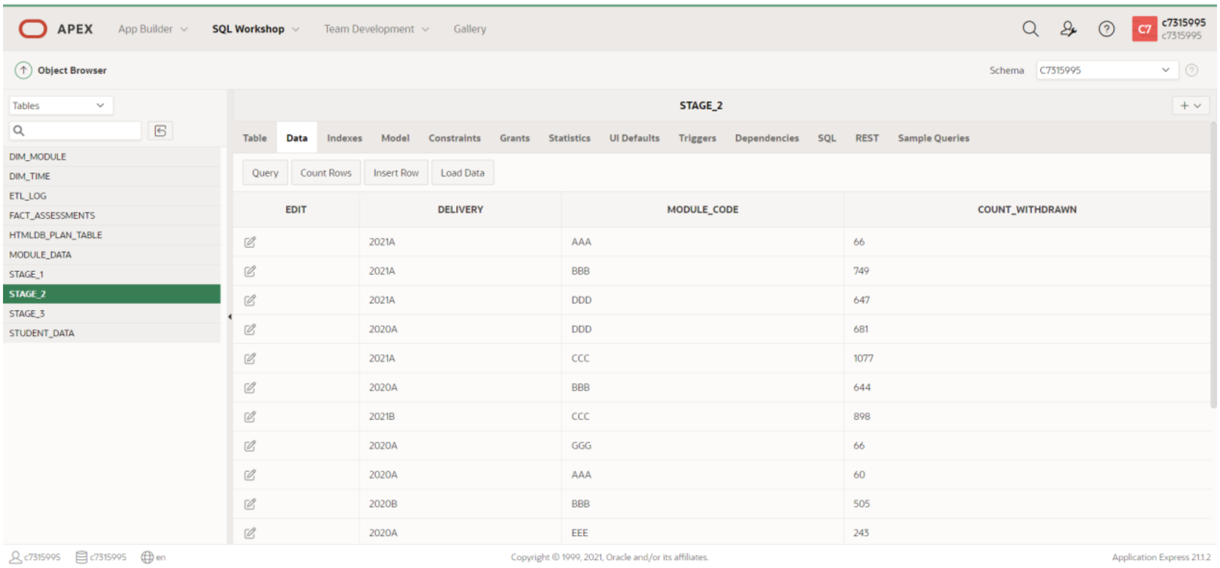
EDIT	NEXTVAL	TIME_ID	DELIVERY	MODULE_ID	MODULE_CODE	NO_OF_STUDENTS_PASSED
✎	1	1	2000	1	QQQ1	1
✎	2	1	2000	1	QQQ1	2
✎	3	1	2000	1	QQQ1	3
✎	4	1	2000	1	QQQ1	4
✎	5	1	2000	1	QQQ1	1
✎	6	1	2000	1	QQQ1	2
✎	7	1	2000	1	QQQ1	3
✎	8	1	2000	2	BBB2	1
✎	9	1	2000	2	BBB2	2
✎	10	1	2000	2	BBB2	3
✎	11	1	2000	2	BBB2	4
✎	12	1	2000	2	BBB2	1

c7315995 c7315995 en

Copyright © 1999, 2021, Oracle and/or its affiliates.

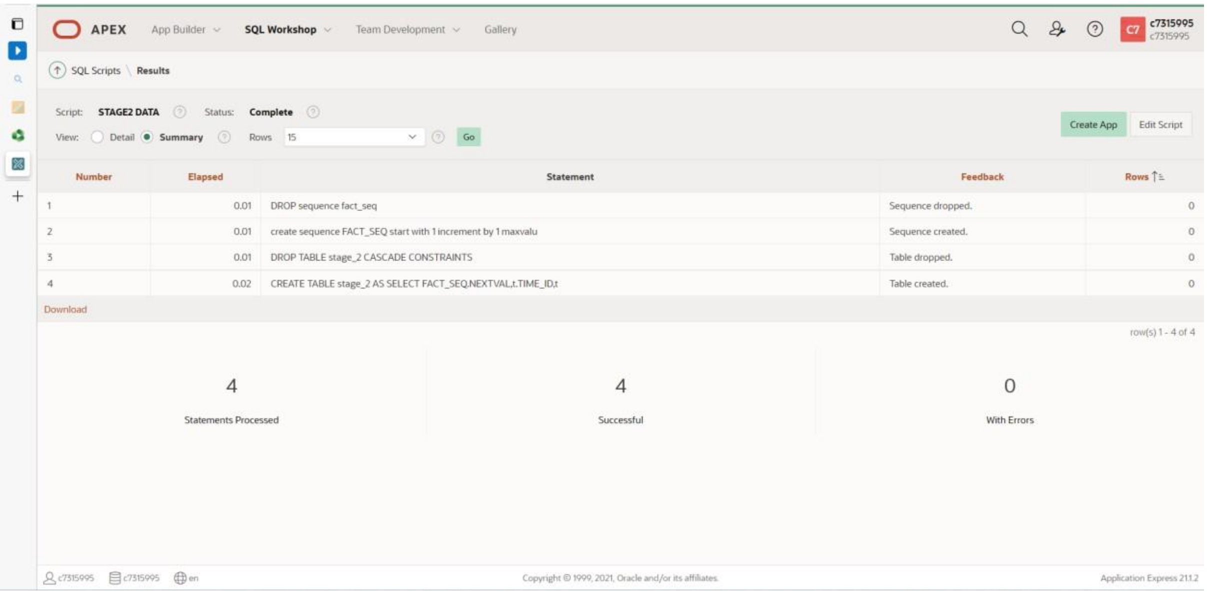
Application Express 21.1.2

Second stage area STAGE_2: NO_OF_STUDents



EDIT	DELIVERY	MODULE_CODE	COUNT_WITHDRAWN
	2021A	AAA	66
	2021A	BBB	749
	2021A	DDD	647
	2020A	DDD	681
	2021A	CCC	1077
	2020A	BBB	644
	2021B	CCC	898
	2020A	GGG	66
	2020A	AAA	60
	2020B	BBB	505
	2020A	EEE	243

c7315995 c7315995 en
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.2



Script: STAGE2 DATA Status: Complete

View: ○ Detail ● Summary Rows 15

Number	Elapsed	Statement	Feedback	Rows
1	0.01	DROP sequence fact_seq	Sequence dropped.	0
2	0.01	create sequence FACT_SEQ start with 1 increment by 1 maxvalu	Sequence created.	0
3	0.01	DROP TABLE stage_2 CASCADE CONSTRAINTS	Table dropped.	0
4	0.02	CREATE TABLE stage_2 AS SELECT FACT_SEQ.NEXTVAL,t.TIME_ID,t	Table created.	0

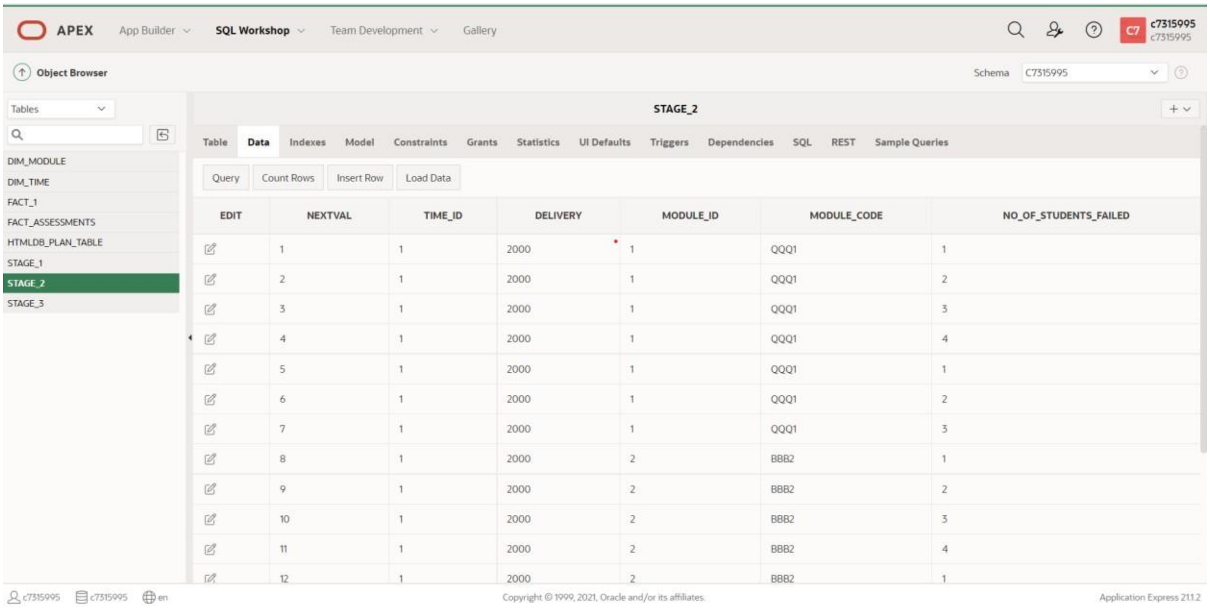
row(s) 1 - 4 of 4

4 Statements Processed

4 Successful

0 With Errors

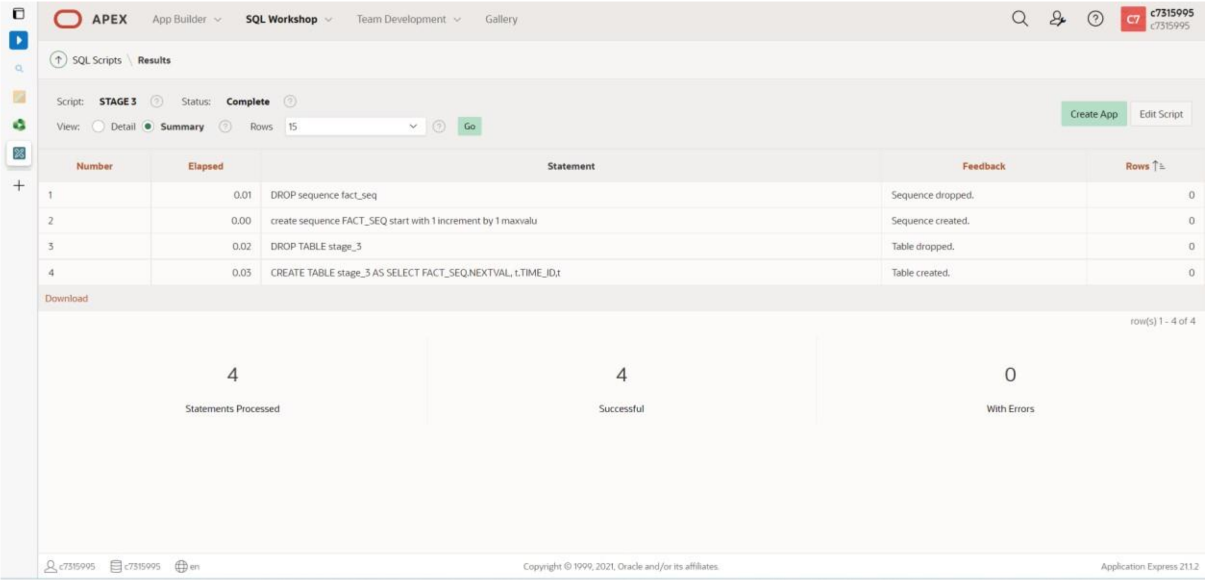
c7315995 c7315995 en
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.2



STAGE_2

EDIT	NEXTVAL	TIME_ID	DELIVERY	MODULE_ID	MODULE_CODE	NO_OF_STUDENTS_FAILED
	1	1	2000	1	QQQ1	1
	2	1	2000	1	QQQ1	2
	3	1	2000	1	QQQ1	3
	4	1	2000	1	QQQ1	4
	5	1	2000	1	QQQ1	1
	6	1	2000	1	QQQ1	2
	7	1	2000	1	QQQ1	3
	8	1	2000	2	BBB2	1
	9	1	2000	2	BBB2	2
	10	1	2000	2	BBB2	3
	11	1	2000	2	BBB2	4
	12	1	2000	2	BBB2	1

Third stage STAGE_3: NO_OF_STUDENTS_WITHDRAW



Script: STAGE 3 Status: Complete

View: ○ Detail ● Summary Rows 15 Go

Number	Elapsed	Statement	Feedback	Rows
1	0.01	DROP sequence fact_seq	Sequence dropped.	0
2	0.00	create sequence FACT_SEQ start with 1 increment by 1 maxvalu	Sequence created.	0
3	0.02	DROP TABLE stage_3	Table dropped.	0
4	0.03	CREATE TABLE stage_3 AS SELECT FACT_SEQ.NEXTVAL, t.TIME_ID,t	Table created.	0

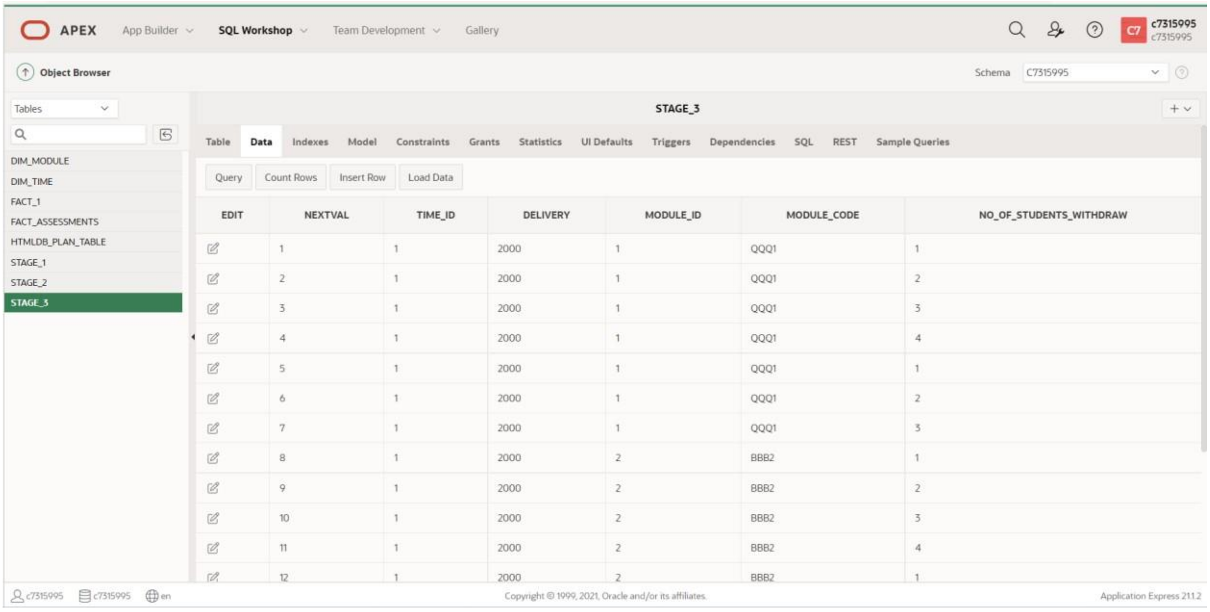
row(s) 1 - 4 of 4

4 Statements Processed

4 Successful

0 With Errors

c7315995 c7315995 en Copyright © 1999, 2021, Oracle and/or its affiliates. Application Express 21.1.2



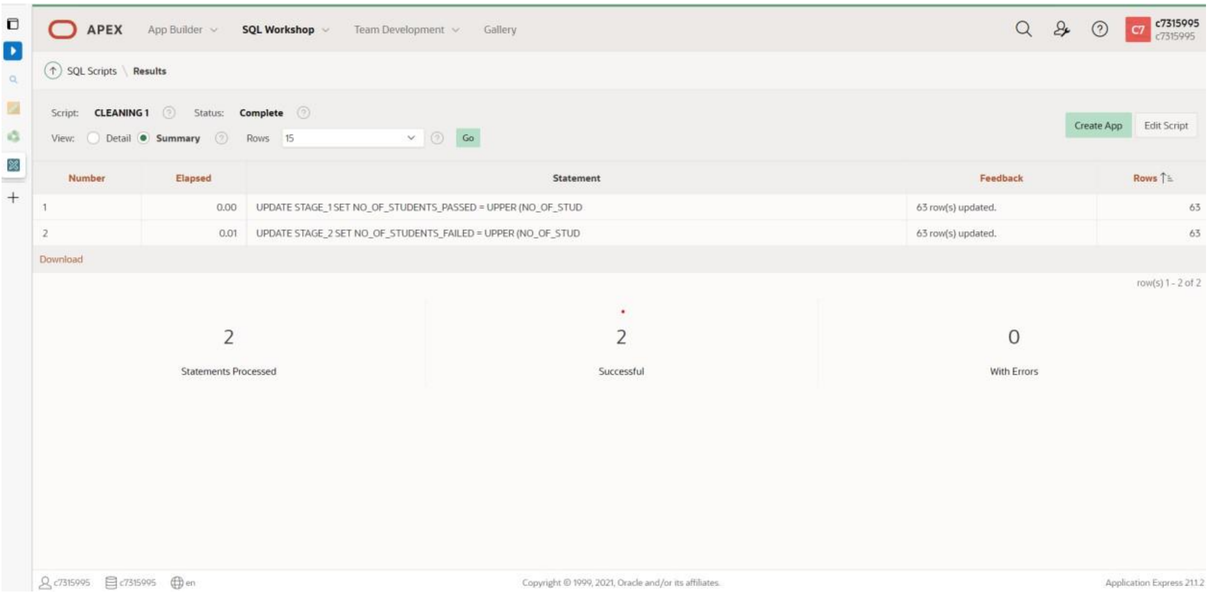
Schema C7315995

STAGE_3

EDIT	NEXTVAL	TIME_ID	DELIVERY	MODULE_ID	MODULE_CODE	NO_OF_STUDENTS_WITHDRAW
	1	1	2000	1	QQQ1	1
	2	1	2000	1	QQQ1	2
	3	1	2000	1	QQQ1	3
	4	1	2000	1	QQQ1	4
	5	1	2000	1	QQQ1	1
	6	1	2000	1	QQQ1	2
	7	1	2000	1	QQQ1	3
	8	1	2000	2	BBB2	1
	9	1	2000	2	BBB2	2
	10	1	2000	2	BBB2	3
	11	1	2000	2	BBB2	4
	12	1	2000	2	BBB2	1

c7315995 c7315995 en Copyright © 1999, 2021, Oracle and/or its affiliates. Application Express 21.1.2

1.3.2 Cleaning process of data:



Script: CLEANING 1 Status: Complete

View: ○ Detail ● Summary Rows 15

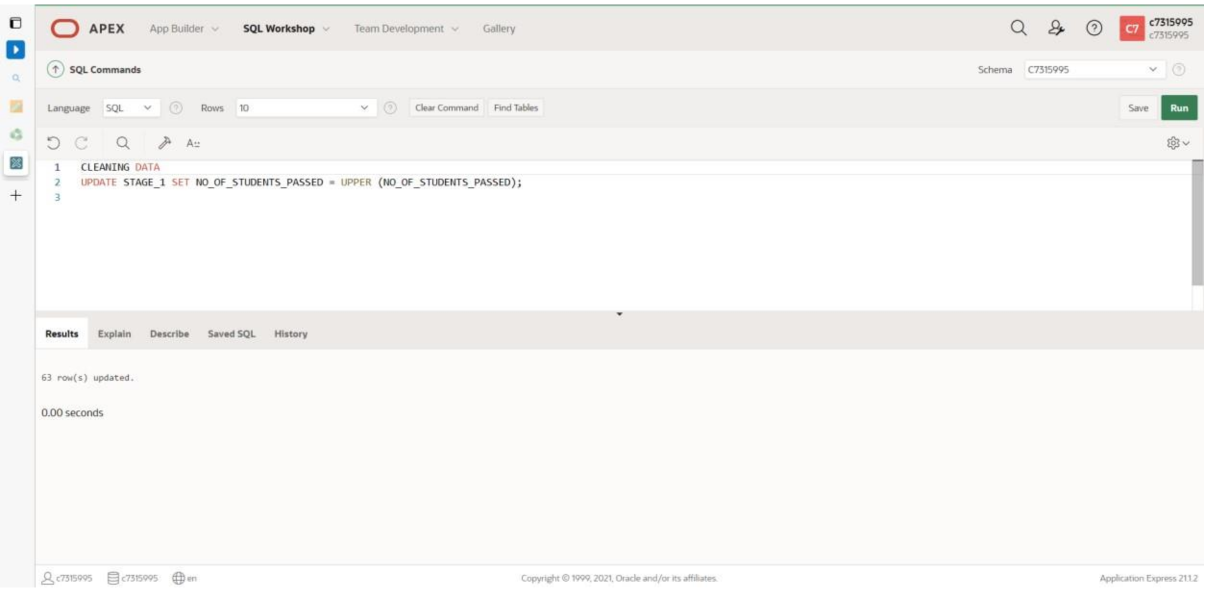
Number	Elapsed	Statement	Feedback	Rows
1	0.00	UPDATE STAGE_1 SET NO_OF_STUDENTS_PASSED = UPPER (NO_OF_STUD	63 row(s) updated.	63
2	0.01	UPDATE STAGE_2 SET NO_OF_STUDENTS_FAILED = UPPER (NO_OF_STUD	63 row(s) updated.	63

row(s) 1 - 2 of 2

Statements Processed	Successful	With Errors
2	2	0

c7315995 c7315995 en Copyright © 1999, 2021, Oracle and/or its affiliates. Application Express 21.1.2

Rows are updated as per the cleaning process.



Schema C7315995

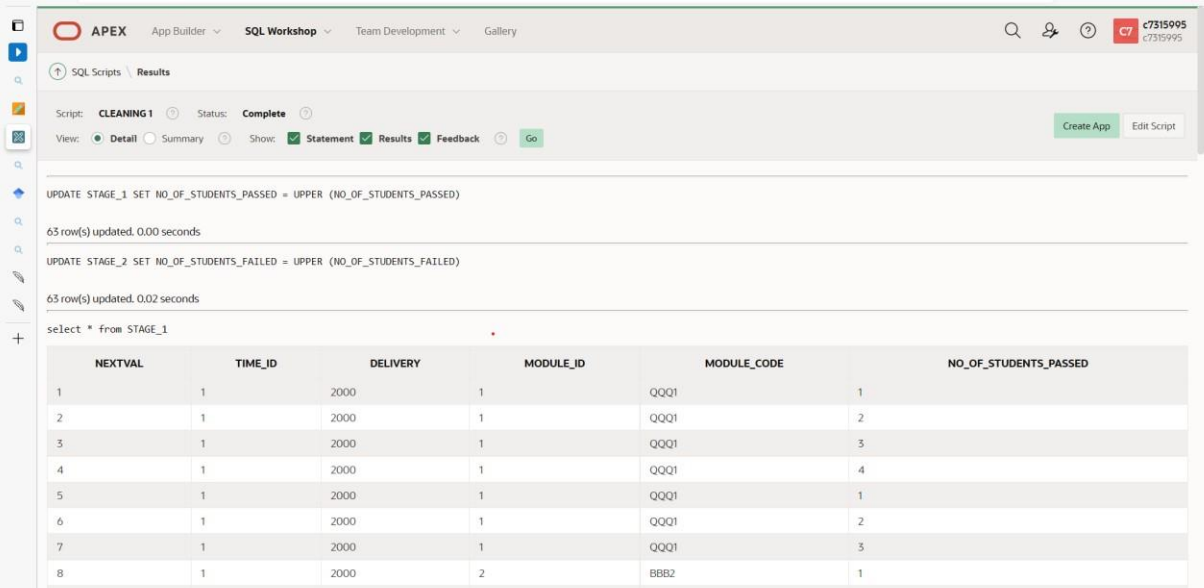
Language SQL Rows 10

```
CLEANING DATA
UPDATE STAGE_1 SET NO_OF_STUDENTS_PASSED = UPPER (NO_OF_STUDENTS_PASSED);
```

```
63 row(s) updated.
```

0.00 seconds

c7315995 c7315995 en Copyright © 1999, 2021, Oracle and/or its affiliates. Application Express 21.1.2



Script: CLEANING 1 Status: Complete

View: ● Detail ○ Summary Show: ☑ Statement ☑ Results ☑ Feedback Go

```
UPDATE STAGE_1 SET NO_OF_STUDENTS_PASSED = UPPER (NO_OF_STUDENTS_PASSED)
```

63 row(s) updated. 0.00 seconds

```
UPDATE STAGE_2 SET NO_OF_STUDENTS_FAILED = UPPER (NO_OF_STUDENTS_FAILED)
```

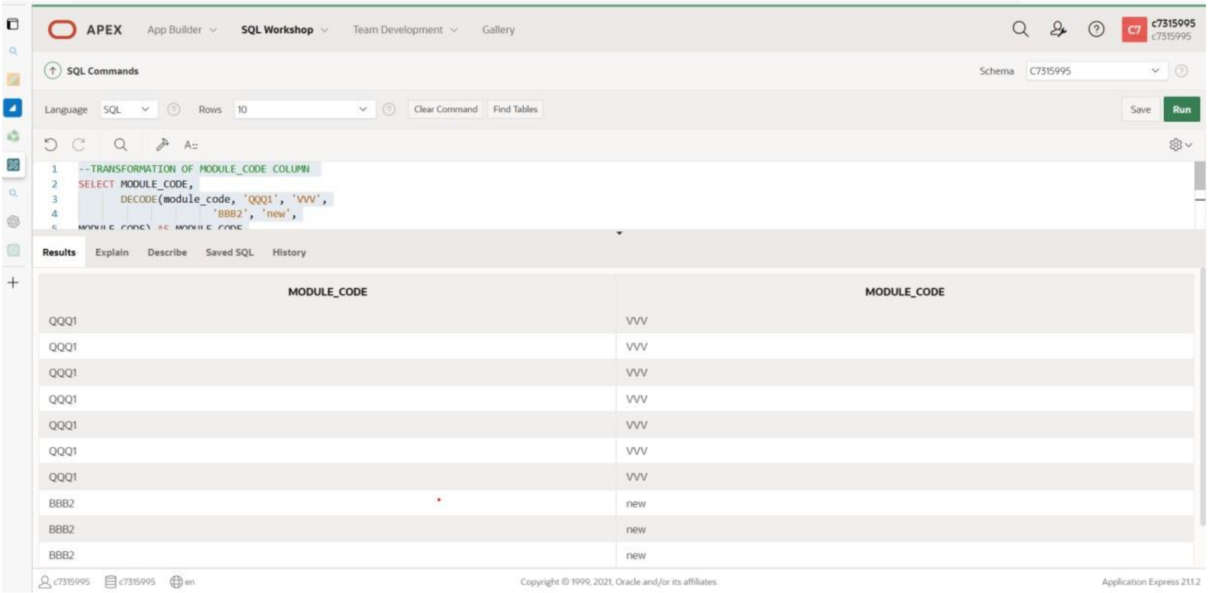
63 row(s) updated. 0.02 seconds

```
select * from STAGE_1
```

NEXTVAL	TIME_ID	DELIVERY	MODULE_ID	MODULE_CODE	NO_OF_STUDENTS_PASSED
1	1	2000	1	QQQ1	1
2	1	2000	1	QQQ1	2
3	1	2000	1	QQQ1	3
4	1	2000	1	QQQ1	4
5	1	2000	1	QQQ1	1
6	1	2000	1	QQQ1	2
7	1	2000	1	QQQ1	3
8	1	2000	2	BBB2	1

1.3.3 Transforming data

I have converted the module code name as it's shown in below table



Schema C7315995

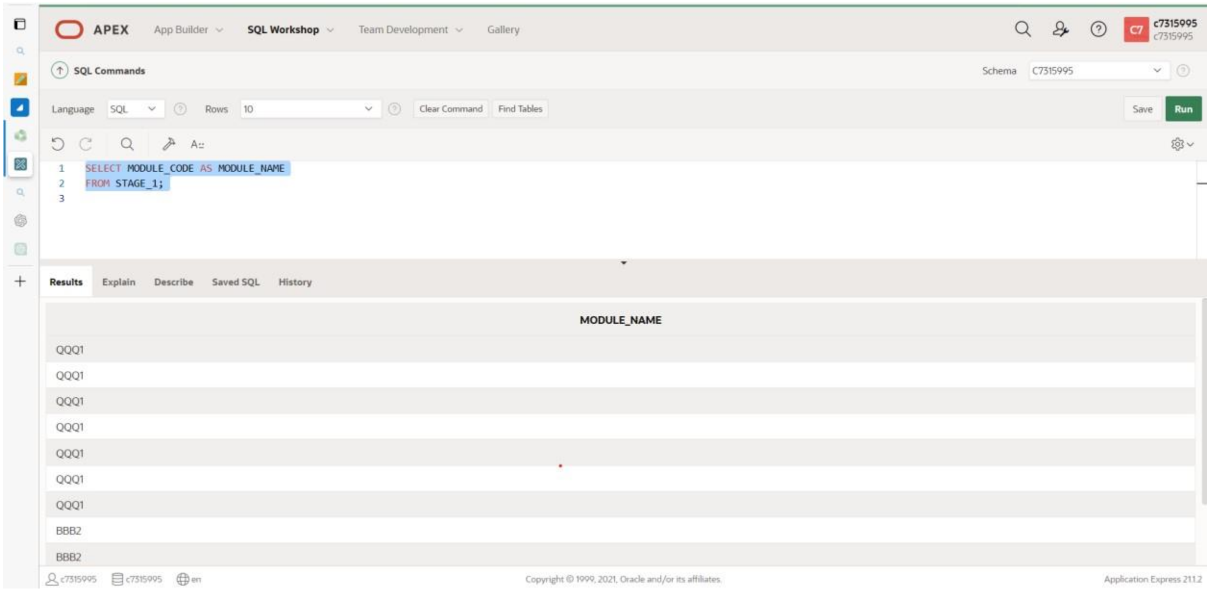
Language SQL Rows 10

```
--TRANSFORMATION OF MODULE_CODE COLUMN
SELECT MODULE_CODE,
       DECODE(module_code, 'QQQ1', 'VVV',
                          'BBB2', 'new',
MODULE_CODE) AS MODULE_CODE
```

MODULE_CODE	MODULE_CODE
QQQ1	VVV
QQQ1	VVV
QQQ1	VVV
QQQ1	VVV
QQQ1	VVV
QQQ1	VVV
QQQ1	VVV
BBB2	new
BBB2	new
BBB2	new

c7315995 c7315995 en Copyright © 1999, 2021, Oracle and/or its affiliates. Application Express 21.1.2

This table shows the transform process of the module column like MODULE_CODE INTO MODULE_NAME.

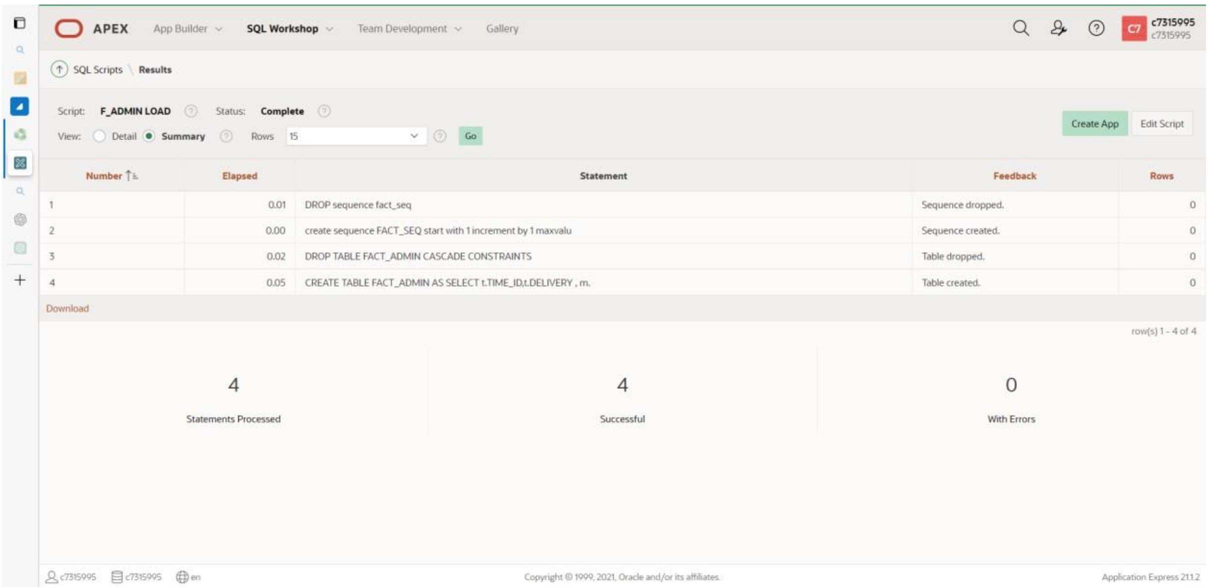


```
SELECT MODULE_CODE AS MODULE_NAME
FROM STAGE_1;
```

MODULE_NAME
QQQ1
QQQ1
QQQ1
QQQ1
QQQ1
QQQ1
QQQ1
BBB2
BBB2

c7315995
c7315995
en
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.2

1.3.4 Loading data into fact table :



Script: F_ADMIN LOAD Status: Complete

View: ○ Detail ● Summary Rows 15

Number	Elapsed	Statement	Feedback	Rows
1	0.01	DROP sequence fact_seq	Sequence dropped.	0
2	0.00	create sequence FACT_SEQ start with 1 increment by 1 maxvalu	Sequence created.	0
3	0.02	DROP TABLE FACT_ADMIN CASCADE CONSTRAINTS	Table dropped.	0
4	0.05	CREATE TABLE FACT_ADMIN AS SELECT t.TIME_ID,t.DELIVERY , m.	Table created.	0

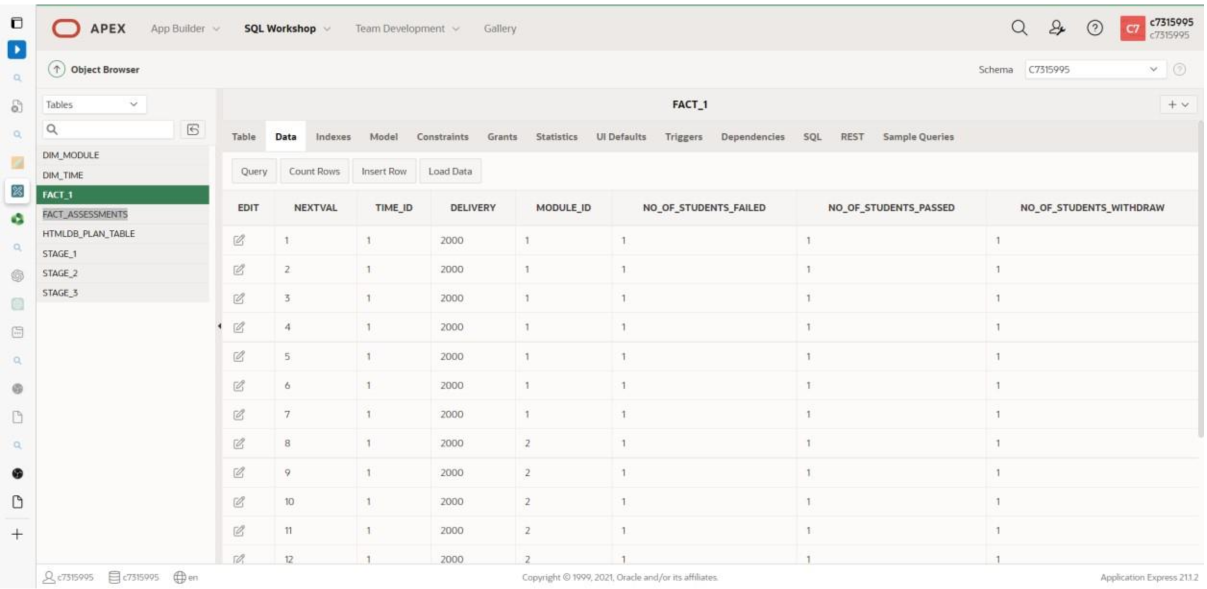
row(s) 1 - 4 of 4

4 Statements Processed

4 Successful

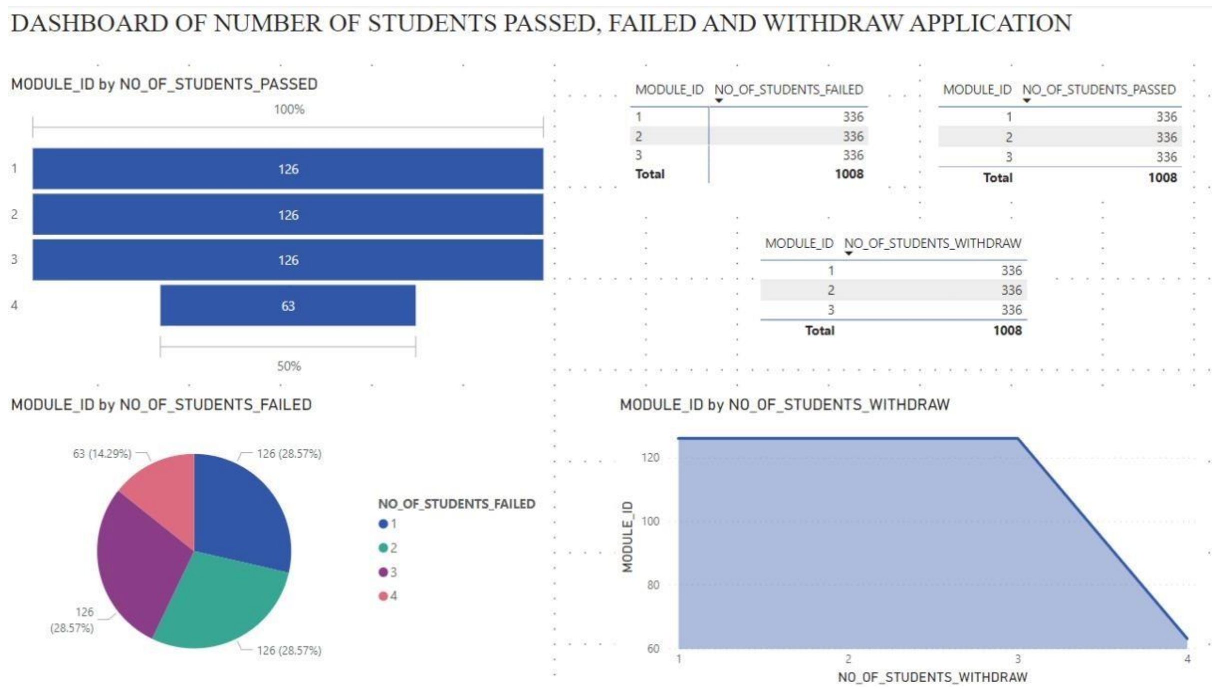
0 With Errors

c7315995
c7315995
en
Copyright © 1999, 2021, Oracle and/or its affiliates.
Application Express 21.1.2



EDIT	NEXTVAL	TIME_ID	DELIVERY	MODULE_ID	NO_OF_STUDENTS_FAILED	NO_OF_STUDENTS_PASSED	NO_OF_STUDENTS_WITHDRAW
✎	1	1	2000	1	1	1	1
✎	2	1	2000	1	1	1	1
✎	3	1	2000	1	1	1	1
✎	4	1	2000	1	1	1	1
✎	5	1	2000	1	1	1	1
✎	6	1	2000	1	1	1	1
✎	7	1	2000	1	1	1	1
✎	8	1	2000	2	1	1	1
✎	9	1	2000	2	1	1	1
✎	10	1	2000	2	1	1	1
✎	11	1	2000	2	1	1	1
✎	12	1	2000	2	1	1	1

2. OLAP (Online Analytical Processing)



DASHBOARD OF NUMBER OF STUDENTS PASSED, FAILED AND WITHDRAW APPLICATION

MODULE_ID	NO_OF_STUDENTS_FAILED
1	336
2	336
3	336
Total	**1008**

MODULE_ID	NO_OF_STUDENTS_PASSED
1	336
2	336
3	336
Total	**1008**

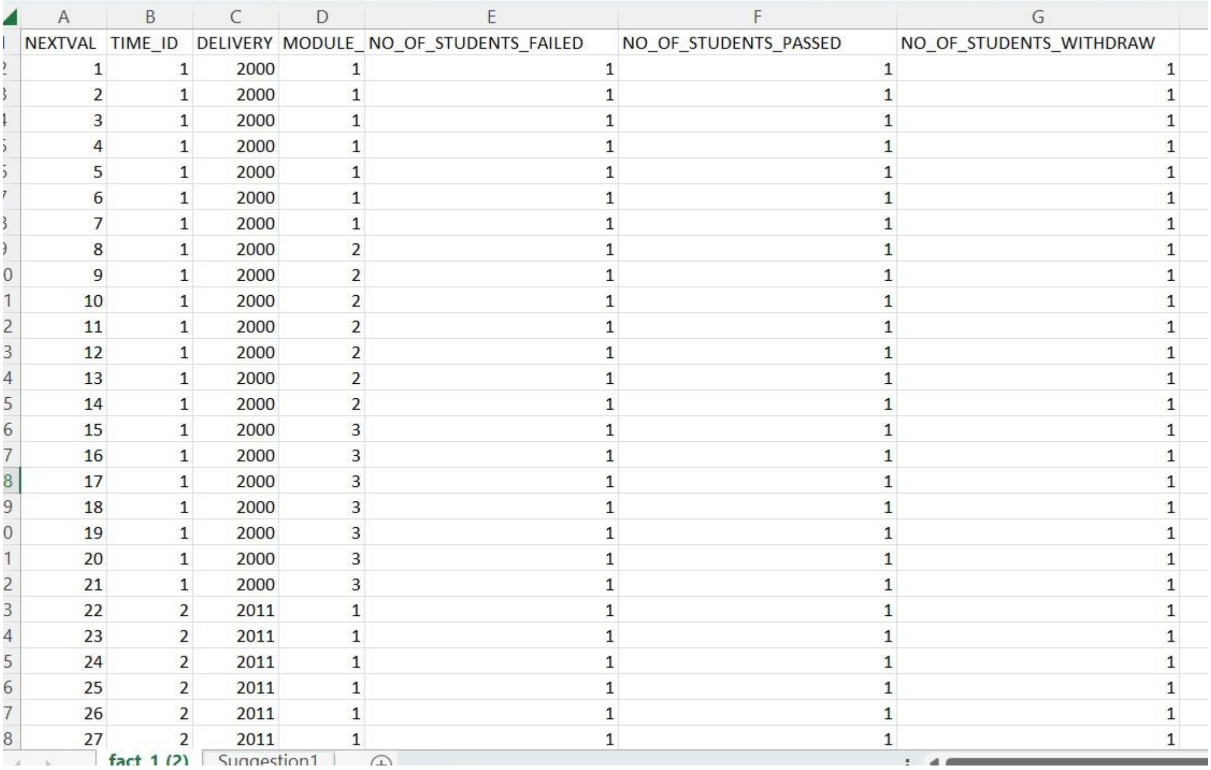
MODULE_ID	NO_OF_STUDENTS_WITHDRAW
1	336
2	336
3	336
Total	**1008**

OLAP database system, in-depth data analysis and data visualization are used in data warehouses.(Yu and Golub).Research on the use of OLAP technologies in management tasks. OLAP and data warehousing systems, with a focus on their updated needs We discuss back-end resources for OLAPtypical multidimensional data models, front-end client tools for querying and data analysis, server extensions for swift query processing, tools for metadata management, and tools for administering the warehouse. Extracting, cleaning, and loading data into a data warehouse. The operational databases generally offer on-line transaction processing (OLTP), whereas the data warehouse enables on-line analytical processing (OLAP), which has quite different functional and performance requirements. (Chaudhuri and Dayal,1997).

The following factors contributed to the development of data warehousing despite the existence of operational databases: • An operational database is designed and tuned from well-known tasks and workloads, such as indexing using primary keys, searching for specific records, and optimising "canned queries." Due to the complexity of data warehouse queries, it may be necessary to apply specialised data organisation, access, and implementation methods based on multidimensional views. These approaches often entail computing huge groupings of data at summary levels. Operational tasks would perform significantly worse if OLAP queries were processed in operational databases. Multiple transactions can be processed concurrently by an operational database. For transactions to be consistent and resilient, concurrency control and recovery techniques like locking and logging are needed. While an OLAP query frequently requires. (Reddy et al., 2010).

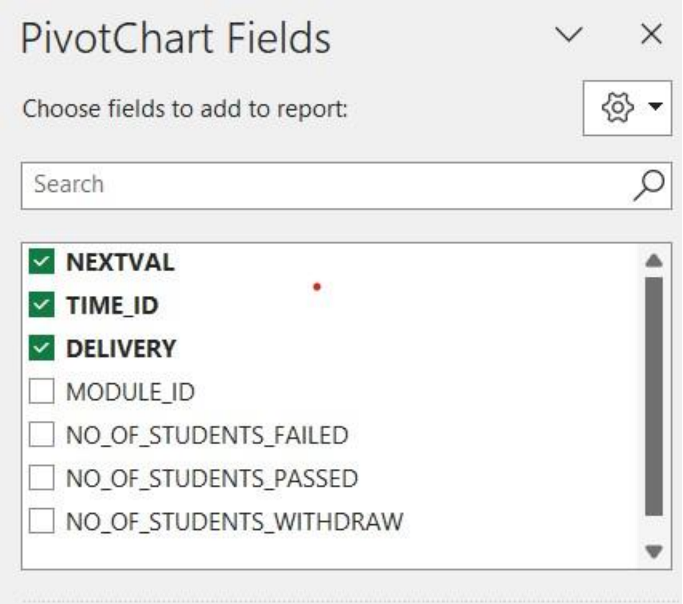
The goal of the decision-support system is to implement data analysis activities so that universities can make wise management judgements.

This below table shows my fact table :



A	B	C	D	E	F	G
NEXTVAL	TIME_ID	DELIVERY	MODULE_	NO_OF_STUDENTS_FAILED	NO_OF_STUDENTS_PASSED	NO_OF_STUDENTS_WITHDRAW
1	1	2000	1	1	1	1
2	1	2000	1	1	1	1
3	1	2000	1	1	1	1
4	1	2000	1	1	1	1
5	1	2000	1	1	1	1
6	1	2000	1	1	1	1
7	1	2000	1	1	1	1
8	1	2000	2	1	1	1
9	1	2000	2	1	1	1
10	1	2000	2	1	1	1
11	1	2000	2	1	1	1
12	1	2000	2	1	1	1
13	1	2000	2	1	1	1
14	1	2000	2	1	1	1
15	1	2000	3	1	1	1
16	1	2000	3	1	1	1
17	1	2000	3	1	1	1
18	1	2000	3	1	1	1
19	1	2000	3	1	1	1
20	1	2000	3	1	1	1
21	1	2000	3	1	1	1
22	2	2011	1	1	1	1
23	2	2011	1	1	1	1
24	2	2011	1	1	1	1
25	2	2011	1	1	1	1
26	2	2011	1	1	1	1
27	2	2011	1	1	1	1

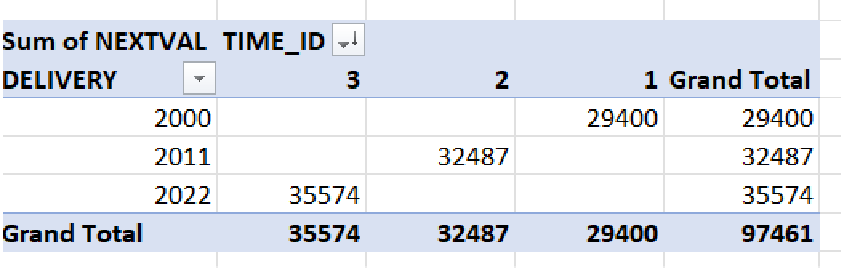
This is pivot table field for creating pivot table.



PivotChart Fields

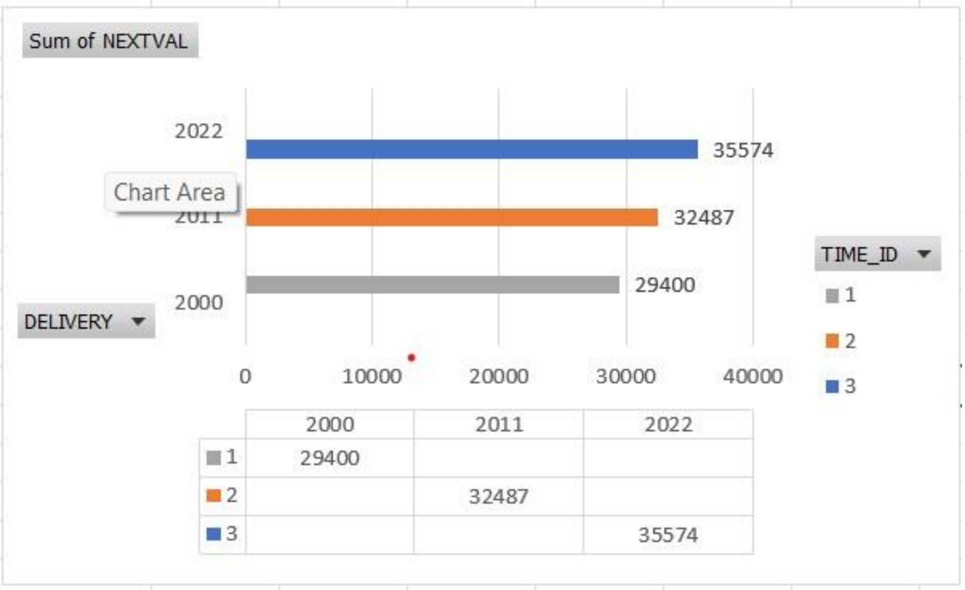
Choose fields to add to report:

☑ NEXTVAL
☑ TIME_ID
☑ DELIVERY
□ MODULE_ID
□ NO_OF_STUDENTS_FAILED
□ NO_OF_STUDENTS_PASSED
□ NO_OF_STUDENTS_WITHDRAW



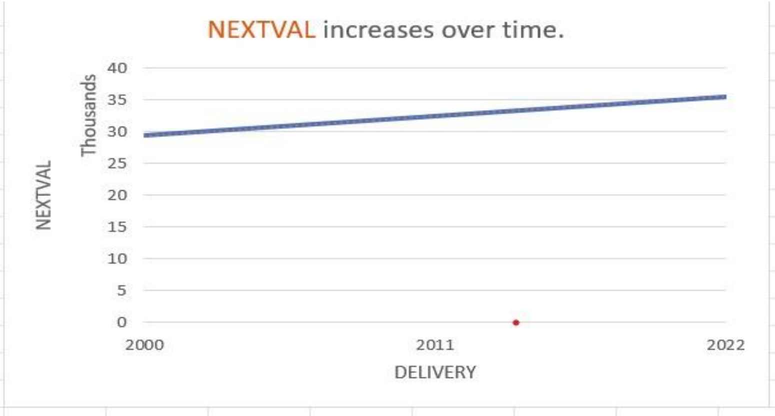
Sum of NEXTVAL	TIME_ID			
DELIVERY	3	2	1	Grand Total
2000			29400	29400
2011		32487		32487
2022	35574			35574
Grand Total	35574	32487	29400	97461

Above Table shows data of delivery and time_id of student progression and also shows overall total of student.

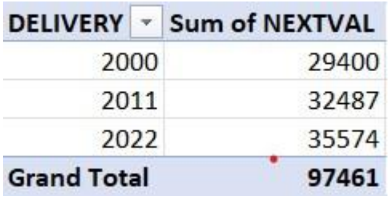


	2000	2011	2022
■ 1	29400		
■ 2		32487	
■ 3			35574

Bar chart address data of each delivery and time_id



Here Nextval is report id which increasing over time.



DELIVERY	Sum of NEXTVAL
2000	29400
2011	32487
2022	35574
Grand Total	**97461**

3. FINDING AND REFLECTION:

Traditional data warehousing solutions employed by corporations encounter several challenges, such as limitations in both the gathering and processing of students' physical data. The fact that parallel storage and processing are required to achieve the growing number of "information efficiency," "scalability," and "elasticity" limitations. Due to the fact that all data marts draw their loading from the same DWH, they all have the same uniform dimensional representation of data. (Zhao and LIU , 2022).

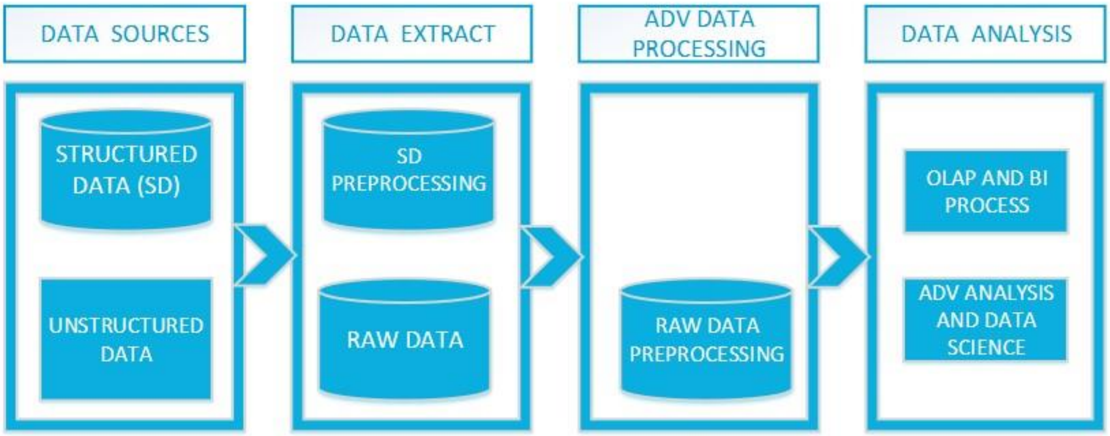


Fig . The architecture of system(Source ; Santoso .,2017)

Bill Inmon's and Ralph Kimball's techniques stand out among the several that scholars have suggested for creating a DWH. Based on the demands of the project, practitioners and researchers can select the one that best fits a certain circumstance. One advantage of Inmon's technique is that because all data marts are loaded from the same central DWH, a very consistent dimensional representation of data is provided across all data marts. Given that it considers the company as a whole, the top-down strategy is adaptable to assist change management [30]. However, because data marts are produced progressively, Kimball's technique is advised when time and money are limited. Kimball's bottom-up methodology's ETL procedure. While the extraction and transformation are the same as Inmon's method, the loading procedure is different since cleansed data is imported into data marts first and then into a central DWH. The DWH design technique developed by Kimball is advised for usage in educational environments, according to a case study conducted in a university [12]. This advice is based on the observation that departments and units typically operate as information silos in universities since they are not interconnected. This paper's goal is to identify the methodology that has been used the most frequently in the collection of publications it has been analysed. (Moscoso et al., 2018).

A stakeholder category's influence in the current study may be veto power on committees overseeing educational quality or high position in society or business using utilitarian or normative methods. In terms of having a say in programme development, a strong stakeholder is not only present in programme committees but also strongly impacts the changes made and has a strong say in programme development. One way to conceptualise the validity of a stakeholder category is as broadly held expectations to be represented in various governance bodies or programme committees. Last but not least, a stakeholder's urgency is demonstrated by his or her attendance at meetings, involvement, and contributions as well as by how they follow up on action outcomes like programme enhancements.(Leisyte and Westerheijen , 2014).

Managers may also model issues with OLAP that would be difficult without it.less adaptable systems with slow and irregular reaction times. Decision-making is made easier when there is more control and quicker access to strategic information. This gives library managers power by enabling them to simulate real-world forecasts and a more effective use of resources.The business as a whole can respond to market needs more swiftly thanks to OLAP. In consequence, market response frequently results in increased revenue and profitability. (Reddy et al., 2010).

References :

1. Chaudhuri, S. and Dayal, U., 1997. An overview of data warehousing and OLAP technology. *ACM Sigmod record*, *26*(1), pp.65-74.

2. Leisyte, L. and Westerheijden, D.F., 2014. Stakeholders and quality assurance in higher education. In *Drivers and barriers to achieving quality in higher education* (pp. 83-97). Brill.

3. Moscoso-Zea, O., Paredes-Gualtor, J. and Luján-Mora, S., 2018. A holistic view of data warehousing in education. *IEEE access*, *6*, pp.64659-64673.

4. Reddy, G.S., Srinivasu, R., Rao, M.P.C. and Rikkula, S.R., 2010. Data Warehousing, Data Mining, OLAP and OLTP Technologies are essential elements to support decision-making process in industries. *International Journal on Computer Science and Engineering*, *2*(9), pp.2865-2873.
5. Singh, R.P. and Singh, K., 2016, September. Design and research of data analysis system for student education improvement (case study: Student progression system in university). In *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)* (pp. 508-512). IEEE.

6. Santoso, L.W., 2017. Data warehouse with big data technology for higher education. *Procedia Computer Science*, *124*, pp.93-99.

7. Zhao, B. and Liu, Y., 2022. Application of Data Warehouse Technology Based on Neural Network in Physical Education Quality Management. *Mathematical Problems in Engineering*, *2022*.

APPENDIX:

```
--------------------------------------------------------------
-- Database creation Script

-- Auto-Generated by QSEE-SuperLite (c) 2001-2004 QSEE-Technologies Ltd.

-- Verbose generation: ON

-- note: spaces within table/column names have been replaced by underscores
(_)

-- Target DB: SQL2

-- Entity Model :Entity Relationship Diagram

-- To drop the tables generated by this script run -
--    'C:\Users\mona\OneDrive\Desktop\SS.SQL1_drop.sql'

--------------------------------------------------------------
DROP TABLE DIM_TIME CASCADE CONSTRAINTS;

DROP TABLE FACT_ASSESSMENTS CASCADE CONSTRAINTS ;

DROP TABLE DIM_MODULE  CASCADE CONSTRAINTS;


--------------------------------------------------------------
-- Table Creation --

-- Each entity on the model is represented by a table that needs to be created
within the Database.
-- Within SQL new tables are created using the CREATE TABLE command.
-- When a table is created its name and its attributes are defined.
-- The values of which are derived from those specified on the model. --
Certain constraints are sometimes also specified, such as identification
of primary keys.

-- Create a Database table to represent the "DIM_TIME" entity.
CREATE TABLE DIM_TIME(     time_id INTEGER NOT NULL,
    DELIVERY    INTEGER,
    -- Specify the PRIMARY KEY constraint for table "DIM_TIME".
    -- This indicates which attribute(s) uniquely identify each row of data.
    CONSTRAINT  pk_DIM_TIME PRIMARY KEY (time_id)
);

-- Create a Database table to represent the "FACT_ASSESSMENTS" entity.
CREATE TABLE FACT_ASSESSMENTS(     report_id   INTEGER NOT NULL,
no_of_students_passed   INTEGER,     no_of_students_failed   INTEGER,
no_of_students_withdraw INTEGER,     fk1_time_id INTEGER NOT NULL,
fk2_module_id   INTEGER NOT NULL,
    -- Specify the PRIMARY KEY constraint for table "FACT_ASSESSMENTS".
    -- This indicates which attribute(s) uniquely identify each row of data.
    CONSTRAINT  pk_FACT_ASSESSMENTS PRIMARY KEY (report_id)
);

-- Create a Database table to represent the "DIM_MODULE" entity.
CREATE TABLE DIM_MODULE(     module_id   INTEGER NOT NULL,
module_CODE INTEGER,
    -- Specify the PRIMARY KEY constraint for table "DIM_MODULE".
    -- This indicates which attribute(s) uniquely identify each row of data.
    CONSTRAINT  pk_DIM_MODULE PRIMARY KEY (module_id)
);


--------------------------------------------------------------
-- Alter Tables to add fk constraints --

-- Now all the tables have been created the ALTER TABLE command is used to
define some additional
-- constraints.  These typically constrain values of foreign keys to be
associated in some way
-- with the primary keys of related tables.  Foreign key constraints can
actually be specified
-- when each table is created, but doing so can lead to dependency problems
within the script
-- i.e. tables may be referenced before they have been created.
This method is therefore safer.

-- Alter table to add new constraints required to implement the "FACT_ASSESSMENTS_DIM_TIME" relationship

-- This constraint ensures that the foreign key of table "FACT_ASSESSMENTS"
-- correctly references the primary key of table "DIM_TIME"
ALTER TABLE FACT_ASSESSMENTS ADD CONSTRAINT fk1_FACT_ASSESSMENTS_to_DIM_TIME FOREIGN KEY(fk1_time_id) REFERENCES DIM_TIME(time_id) ;
-- Alter table to add new constraints required to implement the "FACT_ASSESSMENTS_DIM_MODULE" relationship

-- This constraint ensures that the foreign key of table "FACT_ASSESSMENTS"
-- correctly references the primary key of table "DIM_MODULE"

ALTER TABLE FACT_ASSESSMENTS ADD CONSTRAINT fk2_FACT_ASSESSMENTS_to_DIM_MODULE FOREIGN KEY(fk2_module_id) REFERENCES DIM_MODULE(module_id);

--------------------------------------------------------------
-- End of DDL file auto-generation
--------------------------------------------------------------
```

DIM_TIME LOADING DATA:

```
-- -- Populate DIM_TIME
INSERT INTO DIM_TIME VALUES (7,2000);
INSERT INTO DIM_TIME VALUES (8, 2011);
INSERT INTO DIM_TIME VALUES (9, 2022);


-- check
SELECT * FROM DIM_TIME;
```

STAGE_1 AREA

```
DROP sequence fact_seq;
create sequence FACT_SEQ
start with 1 increment
by 1 maxvalue 10000
minvalue 1;

DROP TABLE stage_1  ;
CREATE TABLE stage_1 AS


SELECT FACT_SEQ.NEXTVAL, t.TIME_ID,t.DELIVERY , m.MODULE_ID,m.MOD-
ULE_CODE , f.NO_OF_STUDENTS_PASSED
FROM DIM_TIME t , DIM_MODULE m, FACT_ASSESSMENTS f
```

STAGE_2 AREA

```
DROP sequence fact_seq;
create sequence FACT_SEQ
start with 1 increment
by 1 maxvalue 10000
minvalue 1; DROP TABLE
stage_2  ;
CREATE TABLE stage_2 AS
SELECT  FACT_SEQ.NEXTVAL,t.TIME_ID,t.DELIVERY , m.MODULE_ID,m.MOD-
ULE_CODE , f.NO_OF_STUDENTS_FAILED
FROM DIM_TIME t , DIM_MODULE m, FACT_ASSESSMENTS f
```

STAGE_3 AREA

```
DROP sequence fact_seq;
create sequence FACT_SEQ
start with 1 increment
by 1 maxvalue 10000
minvalue 1; DROP TABLE
STAGE_3;
CREATE TABLE STAGE_3 AS
SELECT FACT_SEQ.NEXTVAL, t.TIME_ID,t.DELIVERY , m.MODULE_ID,m.MOD-
```

```
ULE_CODE , f.NO_OF_STUDENTS_WITHDRAW
FROM DIM_TIME t , DIM_MODULE m, FACT_ASSESSMENTS f
```

CLEANING PROCESS

```
UPDATE STAGE_1 SET NO_OF_STUDENTS_PASSED = UPPER (NO_OF_STUDENTS_PASSED);
UPDATE STAGE_2 SET NO_OF_STUDENTS_FAILED = UPPER (NO_OF_STUDENTS_FAILED);

select * from STAGE_1;
```

Transformation process:

```
--TRANSFORMATION OF MODULE_CODE COLUMN
SELECT MODULE_CODE,
       DECODE(module_code, 'QQQ1', 'VVV',
                            'BBB2', 'new',
MODULE_CODE) AS MODULE_CODE
FROM STAGE_1;

SELECT MODULE_CODE AS MODULE_NAME
FROM STAGE_1;
--CHECK
SELECT * FROM STAGE_1;
```