

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No.02

A.1 Aim:

Implementation of all dimension tables and fact tables related to the case study mentioned in the first experiment.

A.2 Prerequisite:

Refer to the DBMS manual for SQL Commands and ER diagram.

A.3 Outcome:

After successful completion of this experiment students will be able to design a data warehouse with dimension modelling.

A.4 Theory:

Dimensions are categories by which summarized data can be viewed. E.g. a profit summary in a fact table can be viewed by a Time dimension (profit by month, quarter, year), Region dimension (profit by country, state, city), Product dimension (profit for product1, product2).

A fact table is a table that contains summarized numerical and historical data (facts) and a multipart index composed of foreign keys from the primary keys of related dimension tables.

In data warehousing, a dimension is a collection of reference information about a measurable event. These events are known as facts and are stored in a fact table. Dimensions categorize and describe data warehouse facts and measures in ways that support meaningful answers to business questions. They form the very core of dimensional modelling.

Dimension tables are referenced by fact tables using keys. When creating a dimension table in a data warehouse, a system-generated key is used to uniquely identify a row in the dimension. This key is also known as a surrogate key. The surrogate key is used as the primary key in the dimension table. The surrogate key is placed in the fact table and a foreign key is defined between the two tables. When the data is joined, it does so just as any other join within the database.

Algorithm:

CREATION OF OLTP TABLES

1. Customer table

```
SQL>CREATE TABLE customer
2 ( customer_id  VARCHAR2(10) PRIMARY KEY,
3 name          VARCHAR2(40) NOT NULL,
4 addrVARCHAR2(10) NOT NULL,
5 dob          DATE,
6 in_range     NUMBER,
7 h_owner      NUMBER,
8 c_ownerNUMBER );
```

The CUSTOMER table is populated using the following sample DML statement

```
SQL>INSERT INTO CUSTOMER VALUES ('R41', 'Devarseebanerjee', 'Chennai',
'16-dec-1947', 1, 0, 1);
```

2. Item table

```
SQL>CREATE TABLE item
2 ( item_id      VARCHAR2(20) PRIMARY KEY,
3 name          VARCHAR2(20) NOT NULL,
4 brand         VARCHAR2(20) NOT NULL,
5 dept          NUMBER,
6 c_price       NUMBER,
7 s_price       NUMBER,
8 stock         NUMBER);
```

The ITEM table is populated using the sample DML statement

```
SQL>INSERT INTO ITEM VALUES ( 'R4CB84', 'talc', 'ponds', 5, 28, 34, 21);
```

3. Trans Table

```
SQL>CREATE TABLE trans
  2 ( transid    VARCHAR2(20) PRIMARY KEY,
  3 custid      VARCHAR2(20) REFERENCES
  4            customer(customer_id),
  5 datet       DATE,
  6 amt         NUMBER,
  7 branchid    NUMBER(2) );
```

The TRANS table is populated using the following sample DML statement

```
SQL>INSERT INTO TRANS VALUES ('R4T81', 'R4200', '9-Jan-2003', 47684, 4);
```

4. Item_sold Table

```
SQL> CREATE TABLE item_sold
  2 ( transid    VARCHAR2(20) REFERENCES
  trans(transid),
  3 itemid       VARCHAR2(20) REFERENCES
  item(item_id),
  4 qty         NUMBER);
```

The ITEM_SOLD table is populated using the following DML statement

```
SQL>INSERT INTO ITEM_SOLD VALUES ('R4T996', 'R4SP16', 3);
```

5. Branch Table

```
SQL>CREATE TABLE branch
  2 ( branchid   NUMBER(2) PRIMARY KEY ,
  3 street      VARCHAR2(54),
  4 city        VARCHAR2(54),
  5 state       VARCHAR2(54) );
```

The BRANCH table is populated using the following DML statement

```
SQL>INSERT INTO BRANCH VALUES ( 'Prasad', 'Banglore', 'karnataka' );
```

IMPLEMENTATION OF ENTERPRISE DATAMART

CREATION OF DIMENSION TABLES

1. Customer Dimension Table

```
SQL> CREATE TABLE customer
      2 ( customer_id VARCHAR2 PRIMARY KEY,
      3 name          VARCHAR2(40) NOT NULL,
      4 addr           VARCHAR2(10) NOT NULL,
      5 dob            DATE,
      6 in_range       NUMBER,
      7 h_owner        NUMBER,
      8 c_owner        NUMBER);
```

2. Item dimension table

```
SQL> CREATE TABLE item
      2 ( item_id      VARCHAR2(20) PRIMARY KEY,
      3 name            VARCHAR2(20) NOT NULL,
      4 brand           VARCHAR2(20) NOT NULL,
      5 dept            NUMBER,
      6 c_price          NUMBER,
      7 s_price          NUMBER,
      8 stock            NUMBER);
```

3. Branch dimension table

```
SQL> CREATE TABLE branch
      2 (branchidNUMBER(2) PRIMARY KEY,
      3 street    VARCHAR2(54),
      4 city      VARCHAR2(54),
      5 state     VARCHAR2(54));
```

4. Time dimension table

```
SQL> CREATE TABLE time
      2 ( timeid   date PRIMARY KEY,
      3 day         VARCHAR2(3),
      4 month       VARCHAR2(4),
      5 year        VARCHAR2(3));
```

CREATION OF FACT TABLE

1. Sales_fact table

```
SQL> CREATE TABLE sales_fact
2 (custid    VARCHAR2 (20) REFERENCES customer(customer_id),
3  itemid    VARCHAR2 (20) REFERENCES item(item_id),
4  branchid  NUMBER          REFERENCES branch(branchid) ,
5  timeid    DATE,           REFERENCES time(timeid)
6  qty       NUMBER,
7  profit    NUMBER);
```

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: Comps TE B	Batch: B3
Date of Experiment: 02/02/2021	Date of Submission: 02/02/2021
Grade:	

B.1 Software Code written by a student:

(Paste your problem statement related to your case study completed during the 2 hours of practice in the lab here)

Dimension Table

CUSTOMER

CREATE TABLE CUSTOMER

(CUSTOMER_ID VARCHAR2(10) PRIMARY KEY,
 NAME VARCHAR2(40) NOT NULL,
 GENDER VARCHAR(10));

INSERT INTO CUSTOMER VALUES (1, 'Archit Konde', 'MALE');

INSERT INTO CUSTOMER VALUES (2, 'Saakshi Deokar', 'FEMALE');

INSERT INTO CUSTOMER VALUES (3, 'Hasan Rizvi', 'MALE');

INSERT INTO CUSTOMER VALUES (4, 'Mega Modha', 'FEMALE');

INSERT INTO CUSTOMER VALUES (5, 'Mayuresh Phansikar', 'MALE');

INSERT INTO CUSTOMER VALUES (6, 'Injila Hussain', 'FEMALE');

Live SQL

Feedback

Help

ameythakur@ternaengg.ac.in

SQL Worksheet

ClearFind

Actions

Save

Run

1

CREATE TABLE CUSTOMER

2

(CUSTOMER_ID VARCHAR2(10)

3

NAME VARCHAR2(40)

4

GENDER VARCHAR2(10)

5

);

6

7

INSERT INTO CUSTOMER VALUES (1, 'Archit Konde', 'MALE');

8

INSERT INTO CUSTOMER VALUES (2, 'Saakshi Deokar', 'FEMALE');

9

INSERT INTO CUSTOMER VALUES (3, 'Hasan Rizvi', 'MALE');

10

INSERT INTO CUSTOMER VALUES (4, 'Mega Modha', 'FEMALE');

11

INSERT INTO CUSTOMER VALUES (5, 'Mayuresh Phansikar', 'MALE');

12

INSERT INTO CUSTOMER VALUES (6, 'Injila Hussain', 'FEMALE');

Table created.

1 row(s) inserted.

1 row(s) inserted.



1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

SELECT * FROM CUSTOMER

  **Live SQL** Feedback Help ameythakur@ternaengg.ac.in

SQL Worksheet Clear Find Actions Save Run

1 `SELECT * FROM CUSTOMER`

CUSTOMER_ID	NAME	GENDER
1	Archit Konde	MALE
2	Saakshi Deokar	FEMALE
3	Hasan Rizvi	MALE
4	Mega Modha	FEMALE
5	Mayuresh Phansikar	MALE
6	Injila Hussain	FEMALE

[Download CSV](#)
6 rows selected.

PRODUCT

```
CREATE TABLE PRODUCT
( PRODUCT_ID    VARCHAR2(10)    PRIMARY KEY,
  PRODUCT_NAME  VARCHAR2(40)    NOT NULL,
  PRODUCT_COST  VARCHAR(10) );
```

```
INSERT INTO PRODUCT VALUES ( 1, 'Cadbury Dairy Milk Silk', '140');
INSERT INTO PRODUCT VALUES ( 2, 'Nestle Kit Kat White', '380');
INSERT INTO PRODUCT VALUES ( 3, 'Twix Miniature', '449');
INSERT INTO PRODUCT VALUES ( 4, 'Ferrero Rocher Truffles', '849');
INSERT INTO PRODUCT VALUES ( 5, 'Toblerone Dark Bar', '1299');
INSERT INTO PRODUCT VALUES ( 6, 'Galaxy Smooth Milk', '300');
```







The screenshot shows the Live SQL web application interface. At the top, there is a navigation bar with a menu icon, the 'Live SQL' logo, and links for 'Feedback', 'Help', and a user profile 'ameythakur@ternaengg.ac.in'. Below the navigation bar is a toolbar with 'Clear', 'Find', 'Actions', 'Save', and a 'Run' button. The main area is titled 'SQL Worksheet' and contains the following SQL code:






```
1 CREATE TABLE PRODUCT
2 ( PRODUCT_ID    VARCHAR2(10)    PRIMARY KEY,
3   PRODUCT_NAME  VARCHAR2(40)    NOT NULL,
4   PRODUCT_COST  VARCHAR(10) );
5
6
7 INSERT INTO PRODUCT VALUES ( 1, 'Cadbury Dairy Milk Silk', '140');
8 INSERT INTO PRODUCT VALUES ( 2, 'Nestle Kit Kat White', '380');
9 INSERT INTO PRODUCT VALUES ( 3, 'Twix Miniature', '449');
10 INSERT INTO PRODUCT VALUES ( 4, 'Ferrero Rocher Truffles', '849');
11 INSERT INTO PRODUCT VALUES ( 5, 'Toblerone Dark Bar', '1299');
12 INSERT INTO PRODUCT VALUES ( 6, 'Galaxy Smooth Milk', '300');
```

Below the code editor, the execution results are displayed:

- Table created.
- 1 row(s) inserted.
- 1 row(s) inserted.
- 1 row(s) inserted.
- 1 row(s) inserted.
- 1 row(s) inserted.
- 1 row(s) inserted.

SELECT * FROM PRODUCT

  **Live SQL**  Feedback  Help  ameythakur@ternaengg.ac.in 

SQL Worksheet  Clear  Find **Actions**   Save **Run** 

1 `SELECT * FROM PRODUCT`

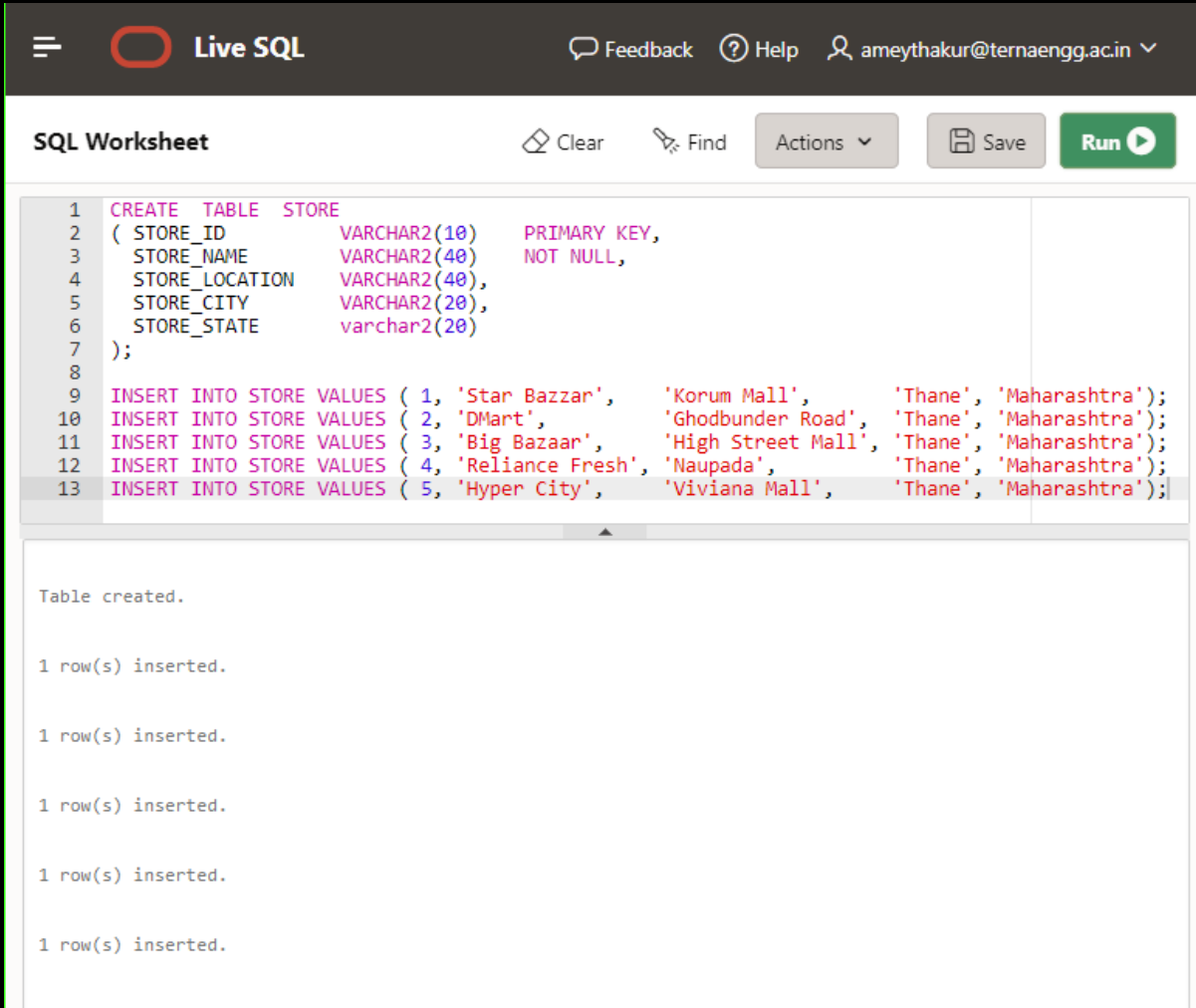
PRODUCT_ID	PRODUCT_NAME	PRODUCT_COST
1	Cadbury Dairy Milk Silk	140
2	Nestle Kit Kat White	380
3	Twix Miniature	449
4	Ferrero Rocher Truffles	849
5	Toblerone Dark Bar	1299
6	Galaxy Smooth Milk	300

[Download CSV](#)
6 rows selected.

STORE

```
CREATE TABLE STORE
( STORE_ID VARCHAR2(10) PRIMARY KEY,
  STORE_NAME VARCHAR2(40) NOT NULL,
  STORE_LOCATION VARCHAR(40),
  STORE_CITY VARCHAR2(20),
  STORE_STATE varchar2(20) );
```

```
INSERT INTO STORE VALUES ( 1, 'Star Bazaar', 'Korum Mall', 'Thane', 'Maharashtra');
INSERT INTO STORE VALUES ( 2, 'DMart', 'Ghodbunder Road', 'Thane', 'Maharashtra');
INSERT INTO STORE VALUES ( 3, 'Big Bazaar', 'High Street Mall', 'Thane', 'Maharashtra');
INSERT INTO STORE VALUES ( 4, 'Reliance Fresh', 'Naupada', 'Thane', 'Maharashtra');
INSERT INTO STORE VALUES ( 5, 'Hyper City', 'Viviana Mall', 'Thane', 'Maharashtra');
```



The screenshot shows a web-based SQL editor interface titled "Live SQL". The top navigation bar includes a menu icon, the "Live SQL" logo, and links for "Feedback", "Help", and a user profile "ameythakur@ternaengg.ac.in". Below the navigation bar is a toolbar with "Clear", "Find", "Actions", "Save", and a "Run" button. The main area displays a "SQL Worksheet" with 13 lines of SQL code. The code consists of a CREATE TABLE statement for a table named "STORE" with columns: STORE_ID (PRIMARY KEY), STORE_NAME (NOT NULL), STORE_LOCATION, STORE_CITY, and STORE_STATE. This is followed by five INSERT INTO statements, each adding a new row of store data. The execution results are shown in a separate pane below the code editor, displaying the following messages: "Table created.", "1 row(s) inserted.", "1 row(s) inserted.", "1 row(s) inserted.", "1 row(s) inserted.", and "1 row(s) inserted."

```
1 CREATE TABLE STORE
2 ( STORE_ID VARCHAR2(10) PRIMARY KEY,
3   STORE_NAME VARCHAR2(40) NOT NULL,
4   STORE_LOCATION VARCHAR2(40),
5   STORE_CITY VARCHAR2(20),
6   STORE_STATE varchar2(20)
7 );
8
9 INSERT INTO STORE VALUES ( 1, 'Star Bazaar', 'Korum Mall', 'Thane', 'Maharashtra');
10 INSERT INTO STORE VALUES ( 2, 'DMart', 'Ghodbunder Road', 'Thane', 'Maharashtra');
11 INSERT INTO STORE VALUES ( 3, 'Big Bazaar', 'High Street Mall', 'Thane', 'Maharashtra');
12 INSERT INTO STORE VALUES ( 4, 'Reliance Fresh', 'Naupada', 'Thane', 'Maharashtra');
13 INSERT INTO STORE VALUES ( 5, 'Hyper City', 'Viviana Mall', 'Thane', 'Maharashtra');
```

Table created.

1 row(s) inserted.



1 row(s) inserted.





1 row(s) inserted.






1 row(s) inserted.

1 row(s) inserted.

SELECT * FROM STORE

  **Live SQL**

 Feedback  Help  ameythakur@ternaengg.ac.in 

SQL Worksheet  Clear  Find **Actions**   Save **Run** 

1 `SELECT * FROM STORE`

STORE_ID	STORE_NAME	STORE_LOCATION	STORE_CITY	STORE_STATE
1	Star Bazaar	Korum Mall	Thane	Maharashtra
2	DMart	Ghodbunder Road	Thane	Maharashtra
3	Big Bazaar	High Street Mall	Thane	Maharashtra
4	Reliance Fresh	Naupada	Thane	Maharashtra
5	Hyper City	Viviana Mall	Thane	Maharashtra

[Download CSV](#)
5 rows selected.

SALESPERSON

```
CREATE TABLE SALESPERSON
( SALESPERSON_ID      VARCHAR2(10)    PRIMARY KEY,
  SALESPERSON_NAME    VARCHAR2(40)    NOT NULL,
  STORE_ID            VARCHAR2(10),
  LOCATION             VARCHAR(20),
  CITY                 VARCHAR2(10),
  STATE                VARCHAR2(20) );
```

```
INSERT INTO SALESPERSON VALUES ( 1 , 'Supreet', 1, 'Korum Mall', 'Thane',
'Maharahstra');
```

```
INSERT INTO SALESPERSON VALUES ( 2 , 'Filly', 1, 'Korum Mall', 'Thane',
'Maharahstra');
```

```
INSERT INTO SALESPERSON VALUES ( 3 , 'Saiganesh', 2, 'Ghodbunder Road',
'Thane', 'Maharahstra');
```

```
INSERT INTO SALESPERSON VALUES ( 4 , 'Saurabh', 3, 'High Street Mall', 'Thane',
'Maharahstra');
```

```
INSERT INTO SALESPERSON VALUES ( 5 , 'Sakshi', 3, 'High Street Mall', 'Thane',
'Maharahstra');
```

```
INSERT INTO SALESPERSON VALUES ( 6 , 'Hemant', 4, 'Naupada', 'Thane',
'Maharahstra');
```

```
INSERT INTO SALESPERSON VALUES ( 7 , 'Rutika', 5, 'Viviana Mall', 'Thane',
'Maharahstra');
```

Live SQL

Feedback

Help

ameythakur@ternaengg.ac.in

SQL Worksheet

ClearFindActionsSaveRun

1CREATE TABLE SALESPERSON

2(SALESPERSON_ID VARCHAR2(10) PRIMARY KEY,

3SALESPERSON_NAME VARCHAR2(40) NOT NULL,

4STORE_ID VARCHAR2(10),

5LOCATION VARCHAR2(20),

6CITY VARCHAR2(10),

7STATE VARCHAR2(20)

8);

9

10INSERT INTO SALESPERSON VALUES (1 , 'Supreet', 1, 'Korum Mall', 'Thane', 'Maharahstra');

11INSERT INTO SALESPERSON VALUES (2 , 'Filly', 1, 'Korum Mall', 'Thane', 'Maharahstra');

12INSERT INTO SALESPERSON VALUES (3 , 'Saiganesh', 2, 'Ghodbunder Road', 'Thane', 'Maharahstra');

13INSERT INTO SALESPERSON VALUES (4 , 'Saurabh', 3, 'High Street Mall', 'Thane', 'Maharahstra');

14INSERT INTO SALESPERSON VALUES (5 , 'Sakshi', 3, 'High Street Mall', 'Thane', 'Maharahstra');

15INSERT INTO SALESPERSON VALUES (6 , 'Hemant', 4, 'Naupada', 'Thane', 'Maharahstra');

16INSERT INTO SALESPERSON VALUES (7 , 'Rutika', 5, 'Viviana Mall', 'Thane', 'Maharahstra');

Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.



1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

SELECT * FROM SALESPERSON

 **Live SQL**

Feedback ? Help ameythakur@ternaengg.ac.in

SQL Worksheet Clear Find Actions Save Run

1 SELECT * FROM SALESPERSON

SALESPERSON_ID	SALESPERSON_NAME	STORE_ID	LOCATION	CITY	STATE
1	Supreet	1	Korum Mall	Thane	Maharahstra
2	Filly	1	Korum Mall	Thane	Maharahstra
3	Saiganesh	2	Ghodbunder Road	Thane	Maharahstra
4	Saurabh	3	High Street Mall	Thane	Maharahstra
5	Sakshi	3	High Street Mall	Thane	Maharahstra
6	Hemant	4	Naupada	Thane	Maharahstra
7	Rutika	5	Viviana Mall	Thane	Maharahstra

Download CSV
7 rows selected.

Fact Table

```
CREATE TABLE PRODUCTSALSAES
( TRANSACTION_ID      VARCHAR2(20)      PRIMARY KEY,
  INVOICE_NO          VARCHAR2(10),
  TOTAL_AMOUNT        VARCHAR2(10),
  CUSTOMER_ID         VARCHAR2(10) REFERENCES CUSTOMER(CUSTOMER_ID),
  PRODUCT_ID          VARCHAR2(10) REFERENCES PRODUCT(PRODUCT_ID),
  STORE_ID            VARCHAR2(10) REFERENCES STORE(STORE_ID),
  SALESPERSON_ID      VARCHAR2(10) REFERENCES SALESPERSON(SALESPERSON_ID)
);
```

```
INSERT INTO PRODUCTSALSAES VALUES('1', '501', '1678', '4', '2,3,4', '5', '2');
```

The screenshot shows the 'Live SQL' web application interface. At the top, there is a navigation bar with a menu icon, the 'Live SQL' logo, and links for 'Feedback', 'Help', and a user profile 'ameythakur@ternaengg.ac.in'. Below this is a 'SQL Worksheet' section with buttons for 'Clear', 'Find', 'Actions', 'Save', and a green 'Run' button. The main area contains a SQL script with line numbers 1 through 11. The script creates a table named 'PRODUCTSALSAES' with various columns and foreign key references, and then inserts a single row of data. Below the script, a status box displays the results: 'Table created.' and '1 row(s) inserted.'

```
1 CREATE TABLE PRODUCTSALSAES
2 ( TRANSACTION_ID      VARCHAR2(20)      PRIMARY KEY,
3   INVOICE_NO          VARCHAR2(10),
4   TOTAL_AMOUNT        VARCHAR2(10),
5   CUSTOMER_ID         VARCHAR2(10) REFERENCES CUSTOMER(CUSTOMER_ID),
6   PRODUCT_ID          VARCHAR2(10) REFERENCES PRODUCT(PRODUCT_ID),
7   STORE_ID            VARCHAR2(10) REFERENCES STORE(STORE_ID),
8   SALESPERSON_ID      VARCHAR2(10) REFERENCES SALESPERSON(SALESPERSON_ID)
9 );
10
11 INSERT INTO PRODUCTSALSAES VALUES('1', '501', '1678', '4', '2,3,4', '5', '2');
```

Table created.

1 row(s) inserted.

SELECT * FROM PRODUCTSALLES;

The screenshot shows the 'Live SQL' web application. At the top, there's a navigation bar with a menu icon, the 'Live SQL' logo, and links for 'Feedback', 'Help', and a user profile 'ameythakur@ternaengg.ac.in'. Below this is a 'SQL Worksheet' section with buttons for 'Clear', 'Find', 'Actions', 'Save', and a green 'Run' button. The SQL editor contains the query 'SELECT * FROM PRODUCTSALLES;'. Below the editor, the output is displayed as a table with 7 columns: TRANSACTION_ID, INVOICE_NO, TOTAL_AMOUNT, CUSTOMER_ID, PRODUCT_ID, STORE_ID, and SALESPERSON_ID. The first row of data shows values: 1, 501, 1678, 4, 3, 5, and 2. A 'Download CSV' link is visible below the table.

TRANSACTION_ID	INVOICE_NO	TOTAL_AMOUNT	CUSTOMER_ID	PRODUCT_ID	STORE_ID	SALESPERSON_ID
1	501	1678	4	3	5	2

B.2 Input and Output:

(Paste your program input and output in the following format, If there is an error then paste the specific error in the output part. In case of an error with the due permission of the faculty, an extension can be given to submit the error-free code with output in due course of time. Students will be graded accordingly.)

Note: Input and Output are mentioned in section B.1

Input :

SQL commands/script which satisfies Two different outcomes mentioned in Problem statements.

Output:

- 1. Dimensional Tables created after firing the above SQL commands.**
- 2. The output satisfies 2 different outcomes mentioned in Problem statements.**

B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ subpart assigned)

A dimensional model is a data structure technique optimized for Data warehousing tools. The concept of Dimensional Modelling is comprised of "fact" and "dimension" tables. These dimensional and relational models have their unique way of data storage that has specific advantages.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

The concept of a fact table and dimension table was implemented using dimension modelling.

B.5 Question of Curiosity

(To be answered by the student based on the practical performed and learning/observations)

Q1: What are the differences between the Dimension table and the fact table?

Ans:

Parameters	Fact Table	Dimension Table
Definition	Measurements, metrics or facts about a business process.	The companion table to the fact table contains descriptive attributes to be used as query constraining.
Characteristic	Located at the centre of a star or snowflake schema and surrounded by dimensions.	Connected to the fact table and located at the edges of the star or snowflake schema
Design	Defined by their grain or its most atomic level.	Should be wordy, descriptive, complete, and quality assured.
Task	A fact table is a measurable event for which dimension table data is collected and is used for analysis and reporting.	Collection of reference information about a business.
Type of Data	Facts tables could contain information like sales against a set of dimensions like Product and Date.	Every dimension table contains attributes that describe the details of the dimension. E.g., Product dimensions can contain Product ID, Product Category, etc.
Key	The primary Key in the fact table is mapped as foreign keys to Dimensions.	The dimension table has primary key columns that uniquely identifies each dimension.
Storage	Helps to store report labels and filter domain values in dimension tables.	Load detailed atomic data into dimensional structures.
Hierarchy	Does not contain Hierarchy	Contains Hierarchies. For example, Location could contain, country, pin code, state, city, etc.

Q2: Explain Primary Keys, Surrogate Keys & Foreign Keys with an example.

Ans:

Primary key:

A column or group of columns in a table that helps us to uniquely identifies every row in that table is called a primary key. This DBMS can't be a duplicate. The same value can't appear more than once in the table.

Surrogate key:

An artificial key that aims to uniquely identify each record is called a surrogate key. These kinds of keys are unique because they are created when you don't have any natural primary key. They do not lend any meaning to the data in the table. The surrogate key is usually an integer.

Foreign key:

A foreign key is a column that is added to create a relationship with another table. Foreign keys help us to maintain data integrity and also allows navigation between two different instances of an entity. Every relationship in the model needs to be supported by a foreign key.